

Summary of Thesis
on
Software Reliability Assessment using
Neuro Fuzzy System

By
BonthuKotaiah
Department of Information Technology

Submitted in fulfillment of the requirements of the award of the
Degree of **DOCTOR OF PHILOSOPHY**
in
INFORMATION TECHNOLOGY



Under the supervision of
Prof. R.A. Khan

Co-Supervised by
Dr. Dharendra Pandey
Department of Information Technology
to the

BABASAHEB BHIMRAO AMBEDKAR UNIVERSITY
LUCKNOW

August - 2015

TABLE OF CONTENTS

S.NO.	CONTENT	PAGE No.
1.	Introduction	1
2.	Research Problems	2
3.	Research Objectives	3
4.	Research Contributions	4
5.	Significance of Study	27
6.	Future Work and Suggestions	27
7.	Limitations and delimitations	28
8.	Thesis Outline	28
	References	30
	List of Publications	48

1. Introduction

Dependency on computer aided systems is rapidly increasing day by day and the software systems operating on it. However this quality of service by the system is degraded by some software failures or faults to meet the required level of performance and this make many of the people to strike off these softwares.

1.1 Software Reliability and its Assessment

Software Reliability Engineering (SRE) is the discipline that helps the organizations to improve the quality of their products and processes. The American Institute of Aeronautics and Astronautics (AIAA) defines SRE as "the application of statistical techniques to data collected during system development and operation to specify, predict, estimate, and assess the reliability of software-based systems"[8]. Among the attributes of software quality, reliability is generally accepted as one of the major factor in software quality since it quantifies the failures. Software Reliability is expressed in execution time [102] [103]. There are three main factors on which the Software Reliability can be judged.

- *Fault*: The result that causes from the mistakes in the software is called as a fault (faulty instruction(s) or data).
- *Error*: When invoking a faulty instruction or a data by an appropriate input pattern, the fault produces an error
- *Failure*: If the erroneous data affect the delivered service (in value and/or in the timing of their delivery), failure occurs.

1.2 Neuro Fuzzy Models

The idea of a Neuro Fuzzy system is to find the parameters of a fuzzy system by means of learning methods obtained from Neural Networks. Here, the basic properties of Neuro Fuzzy systems are discussed. The learning techniques that can be used to create fuzzy systems for data; a common way to

apply a learning algorithm to a fuzzy system is to represent it in a special neural-network-like architecture. Then a learning algorithm – such as back propagation – is used to train the system. The Structure of Adaptive Neuro Fuzzy Inference System is shown in Figure 1.

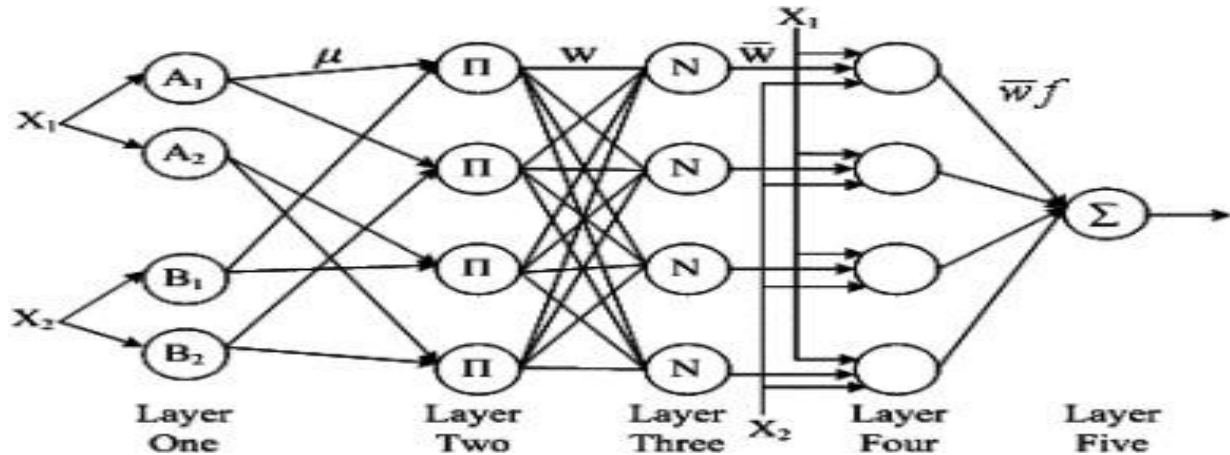


Figure 1: Structure of Adaptive Neuro Fuzzy Inference System.

2. Research Problems

With the rapid development of technology people are tend to depend mainly on the computer aided systems in which the major role is plays by the Operating Softwares(OS) and the application softwares. There is dramatic change in the usage of softwares in the recent years and a wide variety of application softwares are developed to meet the requirements for every area of human life ranging from scientific researchers to kids at kindergarten. In comparison with advancements of hardware components, software achieves less progress caring a larger burden of the total system. The potential of integrating the independently developed software into such hardware system has enabled software designers to develop high complex systems. However, in comparison to the hardware technology, the software technology has not succeeded in keeping measures such as quality. Software reports major source of outages in many systems.

From the two scenarios it is clear that the failures of software system carry huge disaster impacting a lot for human lives. This thesis discuss about the stochastic Software Reliability modeling with Neuro Fuzzy approach that can be utilized at the time of development. Keeping in view of the above discussions, a Neuro Fuzzy based approach is designed in this thesis based on the evaluated metrics Normalized MTBF and Availability to arrive at the best assessment of the Software Reliability for a given dataset of Software observations, the Problem is defined under the title:

“Software Reliability Assessment using Neuro Fuzzy System”

3. Research Objectives

The following objectives are set in the research study:

- To review and examine the literature on Software Reliability in relation with Availability, MTBF and reliability estimation models and the available metrics.
- To identify the factors that affect the Software Reliability at the design phase of SDLC(Software Development Life Cycle)
- 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 develop a model for the assessment of Software Reliability based on Neuro Fuzzy Systems approach and to implement it.
- 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.

4. Research Contributions

To overcome the problems identified in literature survey, the research will use machine learning techniques [158] like Neural Networks, fuzzy-logic approaches for the effective assessment of the Software Reliability after the extensive survey of existing machine learning techniques for the improvement of Software Reliability [74][87][91]. In this research, a combined advantage of theoretical aspects of Artificial Neural Networks and Fuzzy Inference Systems for Software Reliability estimation and modeling is discussed and shown that it is a promising technique in all aspects when compared to all other existing techniques for assessing the Software Reliability [37].

4.1 Proposed Approach for Reliability Assessment

After observed different issues in predicting the Software Reliability [75][90] and also different levels of severity faced by organizations in the case of reliability issues [80], An attempt has been made to implement a model based on the Neuro Fuzzy System.

A generalized block diagram of the proposed approach is shown in the Figure 2. Fuzzy rules employed for the proposed model are:

- *If MTBF (Mean time Between Failure) >0.8 & availability >0.8 then reliability is very high*
- *If $0.7 < \text{MTBF} < 0.8$ & $0.7 < \text{availability} < 0.8$ then reliability is high*
- *If $0.6 < \text{MTBF} < 0.7$ & $0.6 < \text{availability} < 0.7$ then reliability is moderate*
- *If $0.5 < \text{MTBF} < 0.6$ & $0.5 < \text{availability} < 0.6$ then reliability is low*
- *If $0.4 < \text{MTBF} < 0.3$ & $0.4 < \text{availability} < 0.3$ then reliability is very low*

These phases are as follows:

4.1.1 Identification Phase

The objectives of identification phase are:

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

4.1.2 Quantification Phase

The objectives of the quantification phase are:

1. To identify the how the factors are linked with reliability.
2. To evaluate a mathematical analysis for the relationship.

4.1.3 Measurement 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.

4.1.4 Verification and Validation Phase

The objectives of verification and validation phase are

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

4.1.5 Finalization Phase

The objectives of finalization phase are

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

4.1.6 Review & Revisions

There is a review and revision of output of every phase for the refinement of the study. The review and review process is depicted in Figure 3.2.

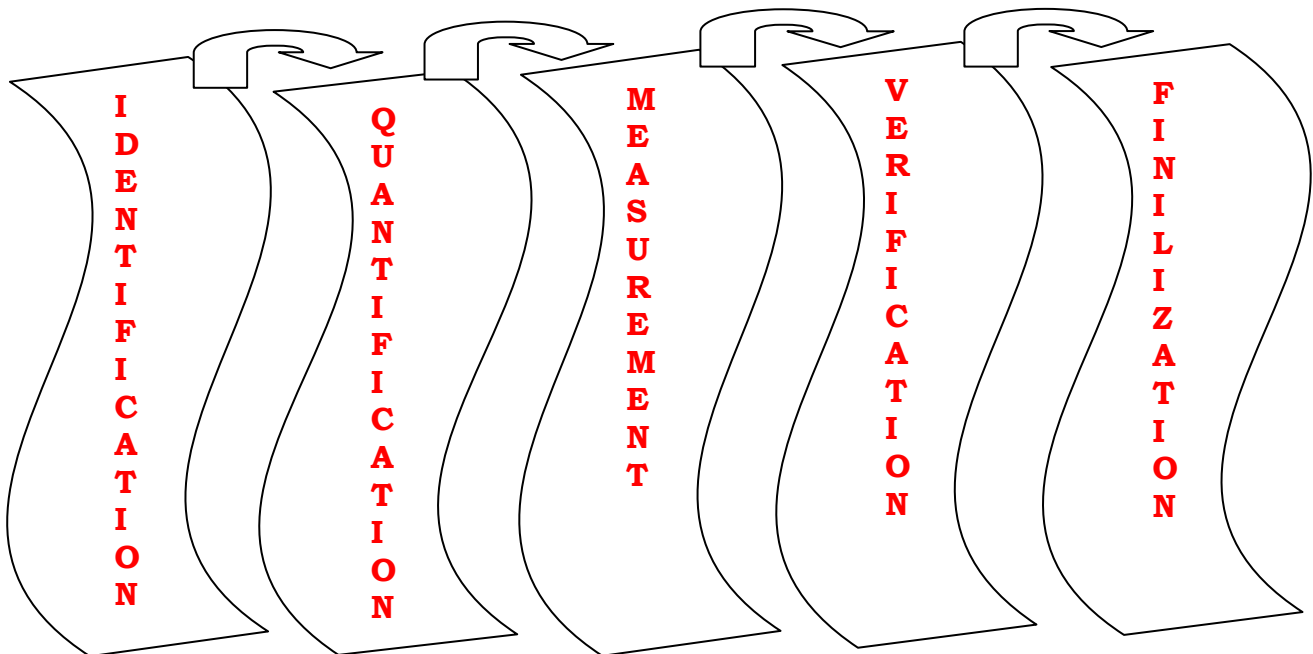


Figure 2: Process Flow of the Proposed Approach

4.2 Implementation of the Proposed Approach

The Figure 3 shows the steps to be followed for implementation of the approach [79]. The values of 2 parameters Normalized MTBF and availability 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).

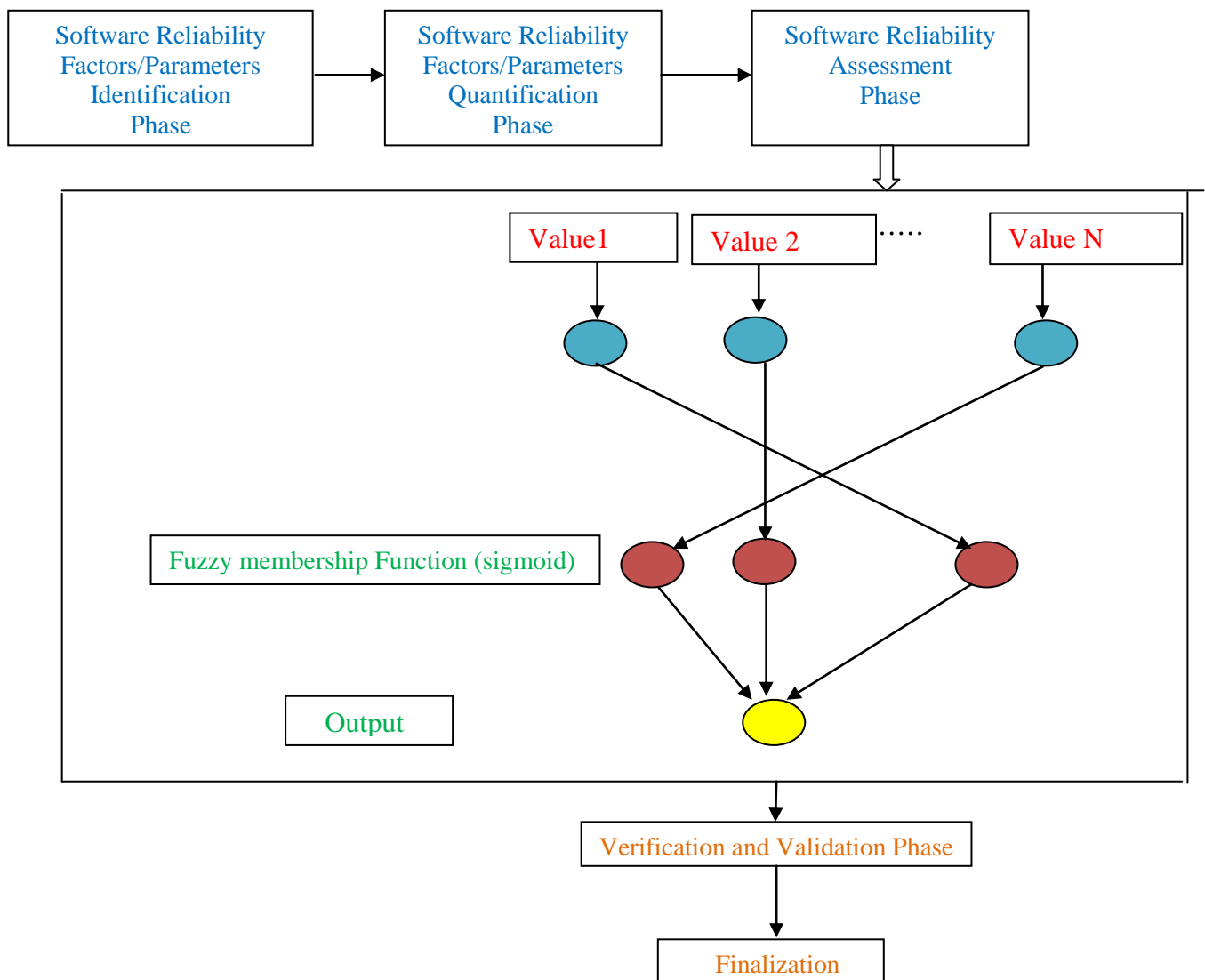


Figure 3: Proposed Neuro Fuzzy based Model of Software Reliability Assessment

4.3 Mathematical Analysis

In this research sigmoid based fuzzy membership function is used for assessment.

A sigmoid MF is defined as

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

Depending on the sign of the parameter a , the sigmoid membership function is inherently open to the right or to the left, and thus is appropriate for representing concepts such as "very large" or "very negative."

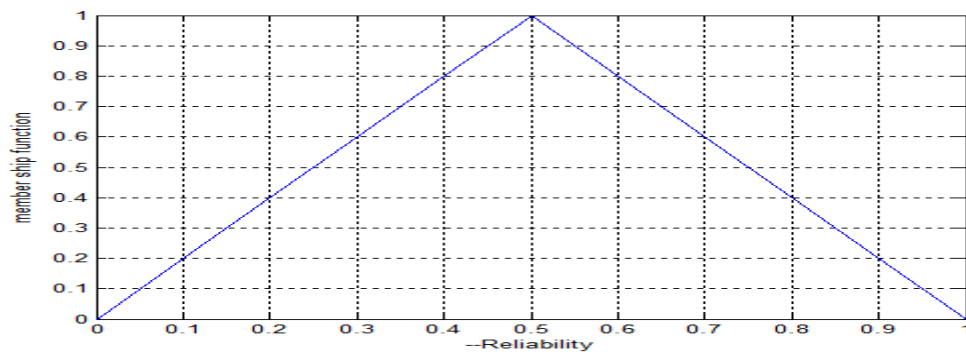


Figure4: The Membership Function of the Proposed Model

The profile of the proposed model is shown in Figure 5.

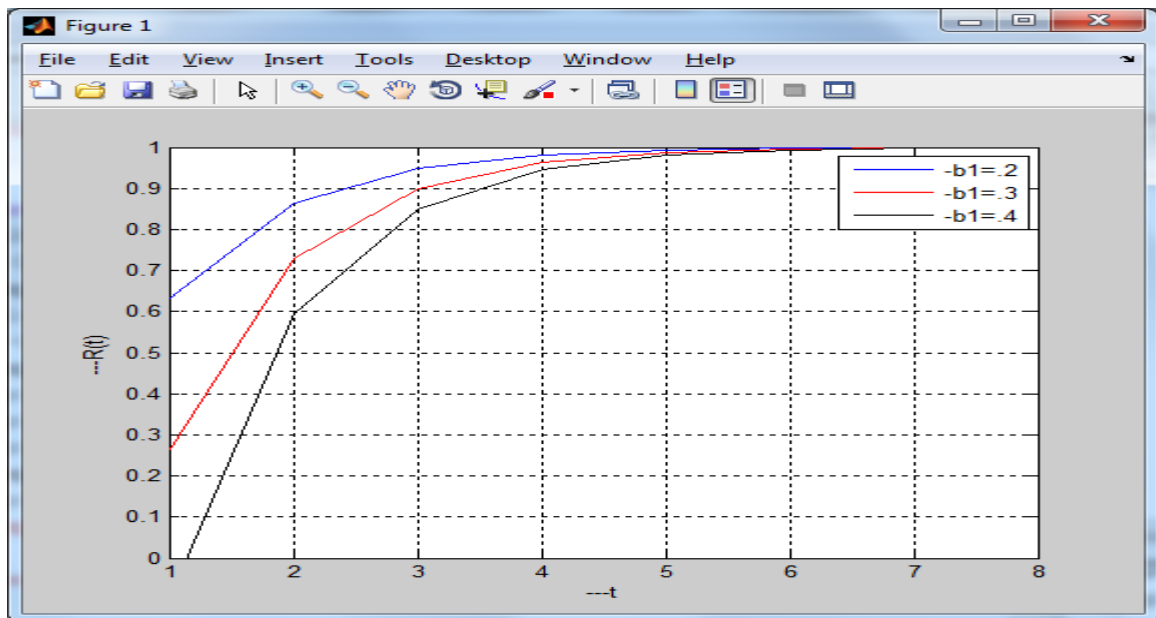


Figure5: Profile of the Proposed Model

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: $C_a(x_i) = C(a) - h \times f(a)$, based on Euler's theorem

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

'h' can be derived by,

$$x_1 = x_0 + n \times h$$

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,

$$f(a) = \text{MTBF} / (1 + \text{MTBF})$$

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

Procedure for 'h' Calculation

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

$$h = 1/17 = 0.058$$

4.4 Empirical Validation

Verification is the process of determining whether the output of one phase of Software Development confirms to that of its previous phase. Validation is the process of determining whether a fully developed system/model/framework confirms to its requirement specification or not. It is also carried out to analyze the validity of model as a whole and phase wise. Validation activities need to be done both theoretically and practically. In the thesis, validation is done to check about the functionality of the proposed model with respect to the given dataset. Validation also helps to find out whether the proposed model offers more advantages than the traditional models. An analysis of Software Reliability assessment based on Neuro Fuzzy Systems to arrive at the conclusion about how to perform validation process [77].

The trained NF system in Figure 6 is then tested for the 17 inputs and the results are compared against test data and FIS output and comparison plot is shown in Figure 7.

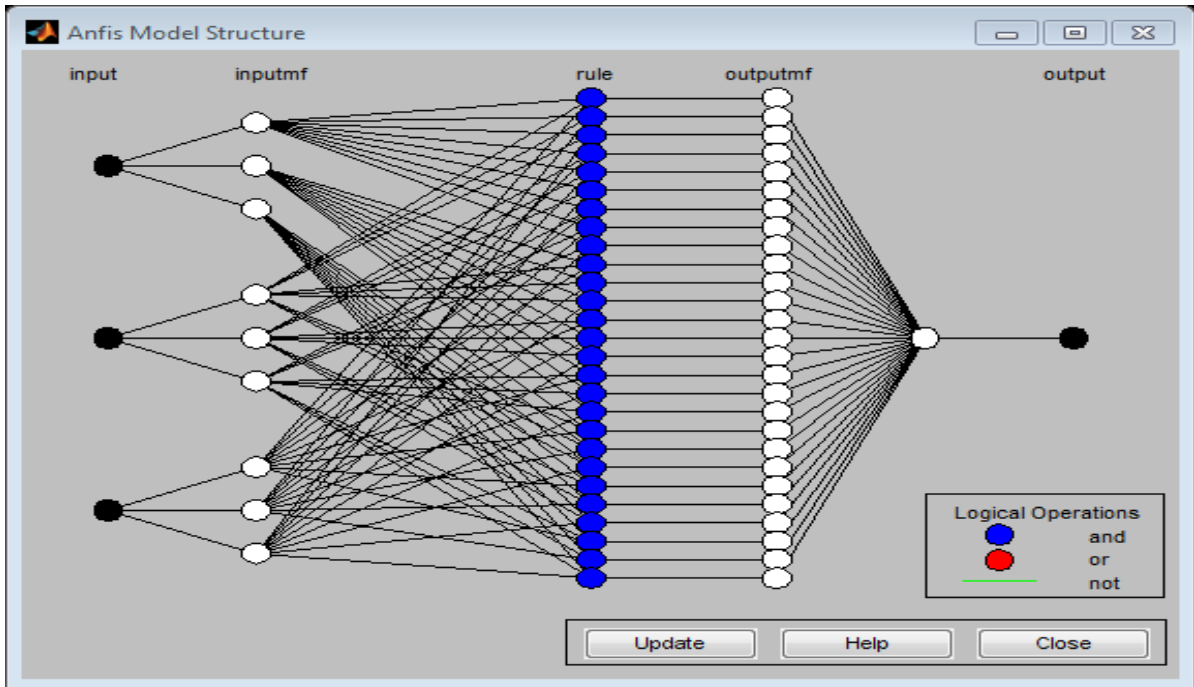


Figure 6: Real time Design of Neuro Fuzzy Structure

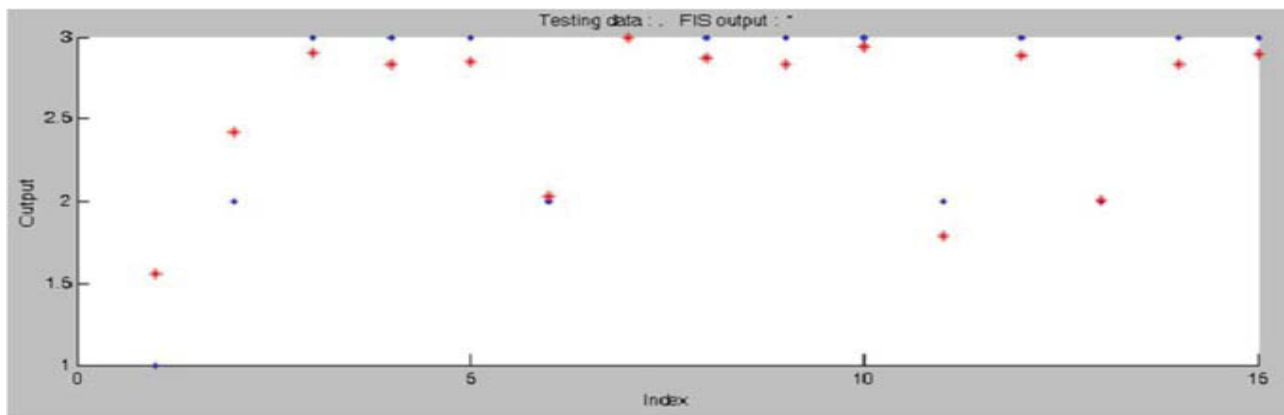


Figure 7: Test Data Vs FIS Output

It shows 0.1571, 0.2140 as NRMSE, RMSE values respectively. The plot of the expected and the output of the NF system for the different inputs are shown in Figure 7.

4.5 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 dataset contains failure observations of 17 programs in Glace EMR Billing Software. Table 1 shows all the 17 projects and the information recorded. The data represents a variety of applications in Glace EMR Billing and is recorded in the 2013. 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 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

4.6 Parameters used for Measurement and Validation

The researchers identified the parameters based on two observations through a revisit of hybrid combination of the parameters used for the assessment of Software Reliability [76].

Software Reliability is measured in terms of Mean Time Between Failures (MTBF). 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. The relationship between MTTF, MTTR, MTBF is shown in Figure 8.

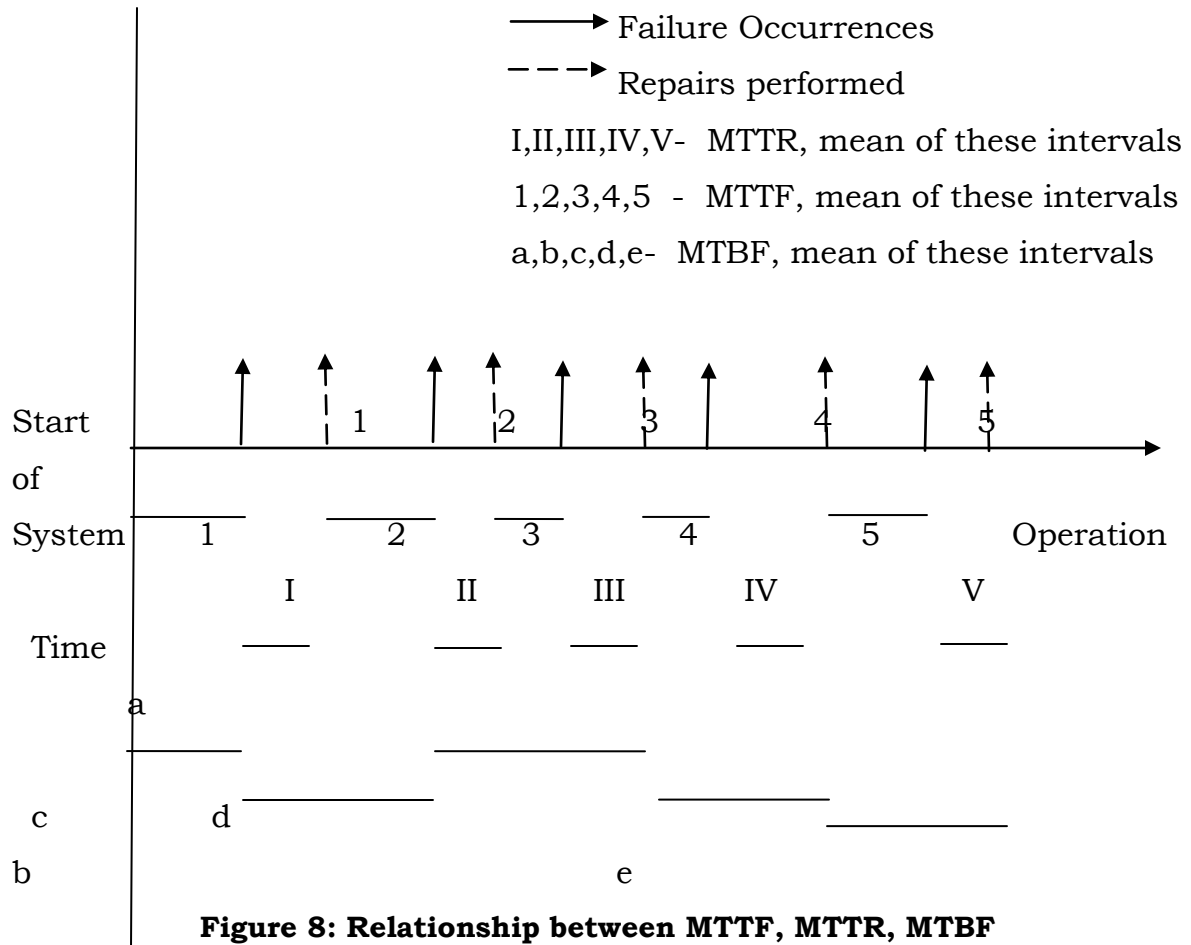


Figure 8: Relationship between MTTF, MTTR, MTBF

Methods to Assess Software Reliability

There are two methods to assess the software reliability:

- Counting failures at periodic intervals of time($\mu(\tau)$)
- Based on Failure Density($\lambda(\tau)$):

For the research, the researcher has used the “Counting failures at periodic intervals of time($\mu(\tau)$)” method. According to this method, the failures are observed at two intervals of time ‘x1’ and ‘x2’. The value of different parameters is calculated as:

$$\text{Software Reliability} = \text{MTBF} / (1 + \text{MTBF}).$$

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

Availability = $MTBF / (MTBF + MTTR)$, is the likelihood that a software system will work at a given time.

4.7 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 has been used as a base for conducting the experiments.

4.7.1 Parameters Estimation

The calculations of normalized Mean Time Between Failures (MTBF) and availability are done mathematically and the results are shown in Table 3 and table 4 respectively. 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 2.

Table 2: Production Time Analysis for the Program Dataset

S.No	Program #	production time	Uptime (x1)	Uptime(x2)	Downtime(x1)	Downtime(x2)	No. of breakdowns(x1)	No. of breakdowns(x2)
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

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

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

$$\text{Availability (For Repairable software systems)} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

A new automated java program for Software Reliability calculation, Approximation based on Euler's mathematical theorem [159] is presented.

Table 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
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

Table 4: Calculation of Availability

S.No.	Program	MTTR	MTBF	Availability
1	GE01	9.12	45.18	0.832
2	GE02	7.336	20.78	0.739
3	GE03	19.158	46.05	0.706
4	GE04	8.25	23.51	0.739
5	GE05	4.25	15.32	0.783
6	GE06	1.447	24.09	0.943
7	GE07	22.05	38.5	0.635
8	GE08	32.82	86.14	0.724
9	GE09	10.791	33.784	0.757
10	GE10	11	17.12	0.608
11	GE11	4.056	4.80	0.543
12	GE12	5.042	8.43	0.609
13	GE13	4.396	5.03	0.533
14	GE14	4.016	8.46	0.678
15	GE15	4.679	11.08	0.703
16	GE16	2.719	14.18	0.839
17	GE17	1.93	9.69	0.833

The ratio of Availability, MTTR and MTBF w.r.t. number of programs in the dataset is represented in Figure 9, Figure 10, Figure11 respectively.

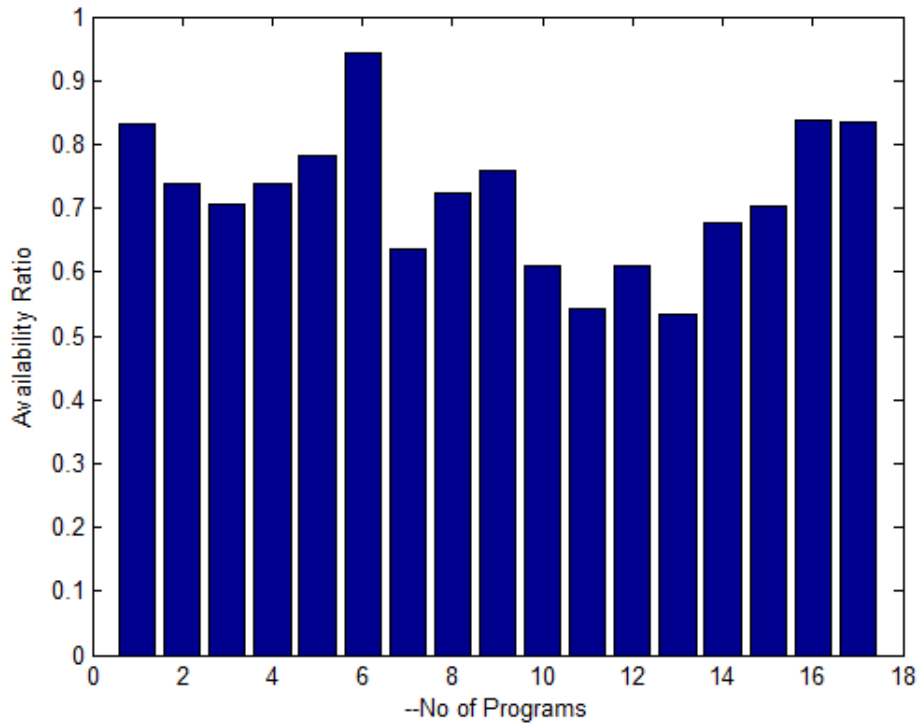


Figure 9: Analysis of Availability Ratio w.r.t. Number of Programs

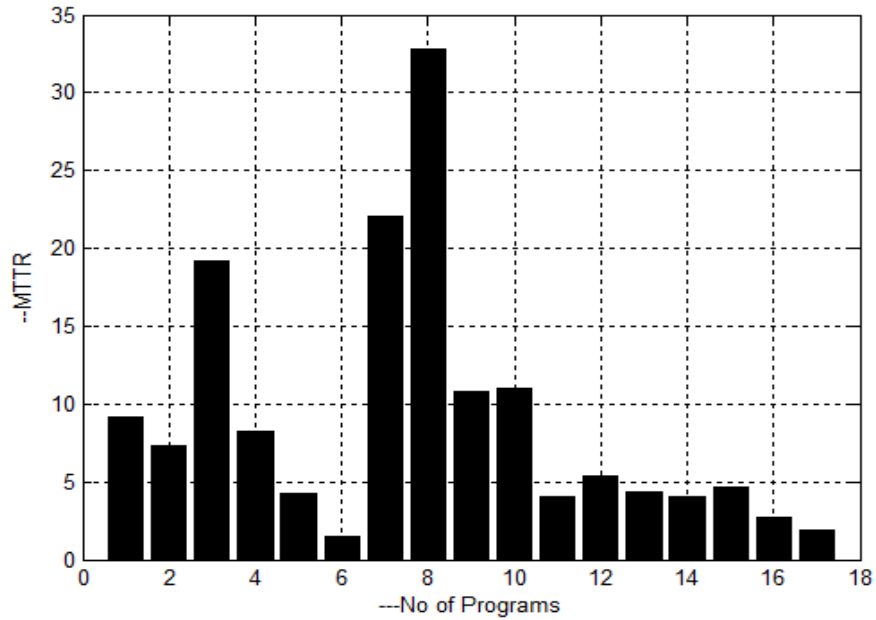


Figure 10: Analysis of MTTR Ratio w.r.t. Number of Programs

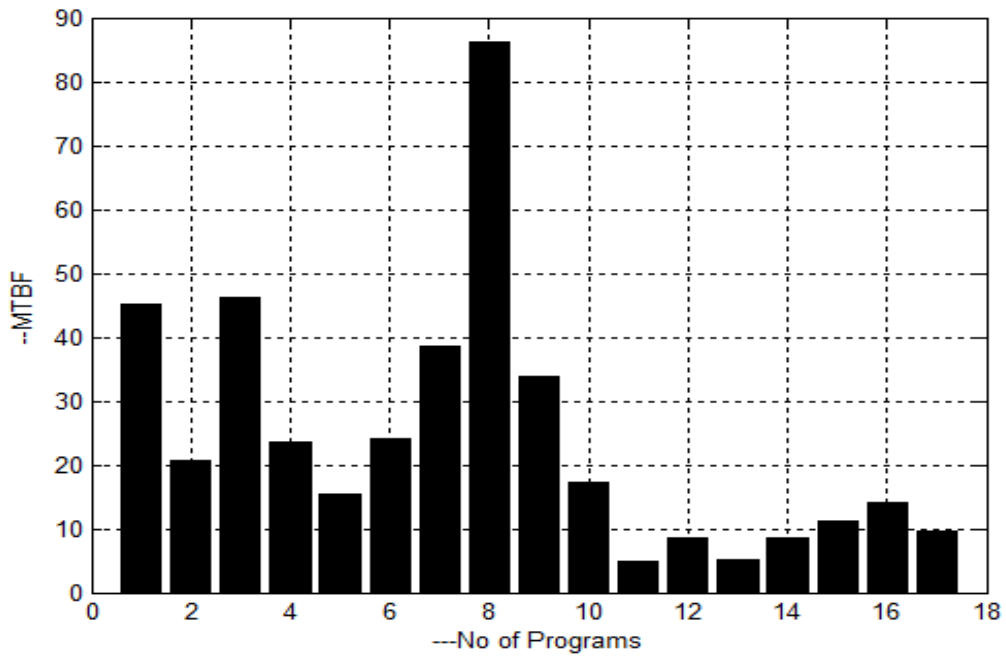


Figure 11: Analysis of MTBF Ratio w.r.t. Number of Programs

4.7.2 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 inference system) and the

proposed Neuro Fuzzy model in MATLAB environment. The Fuzzy Inference Systems based model is presented with the parameters Normalized MTBF and availability in Figure 12. The generated membership function based on employed fuzzy IF-THEN rules ranging from “very low” to “very high” are shown in Figure 13. The dataset will be given as training data to ANFIS systems and the comparison between ANFIS output and FIS output is shown in Figure 14. 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 15. The generated error after learning of inputs through training is based on number of epochs in Figure 16. Finally, the performance level of the proposed approach for conversion of input space to output space is depicted as a surface in Figure 17.

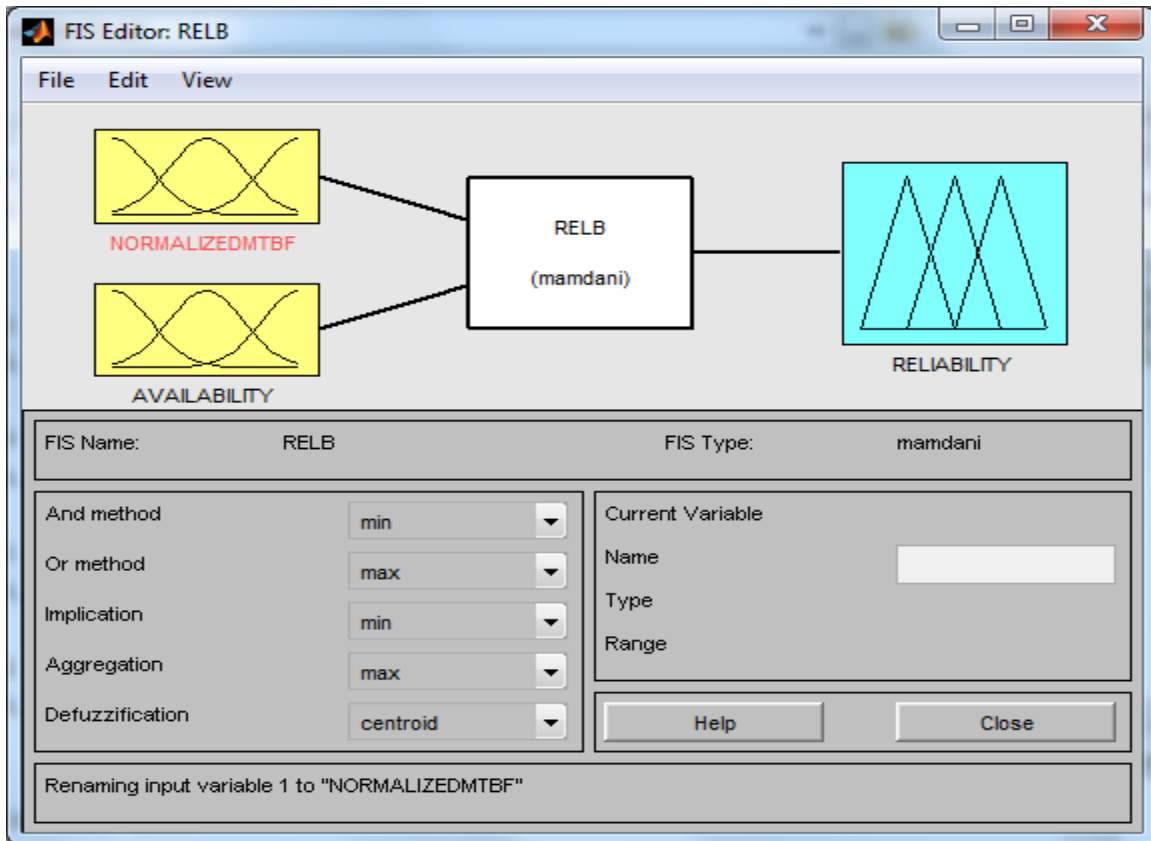


Figure 12: FIS System Model

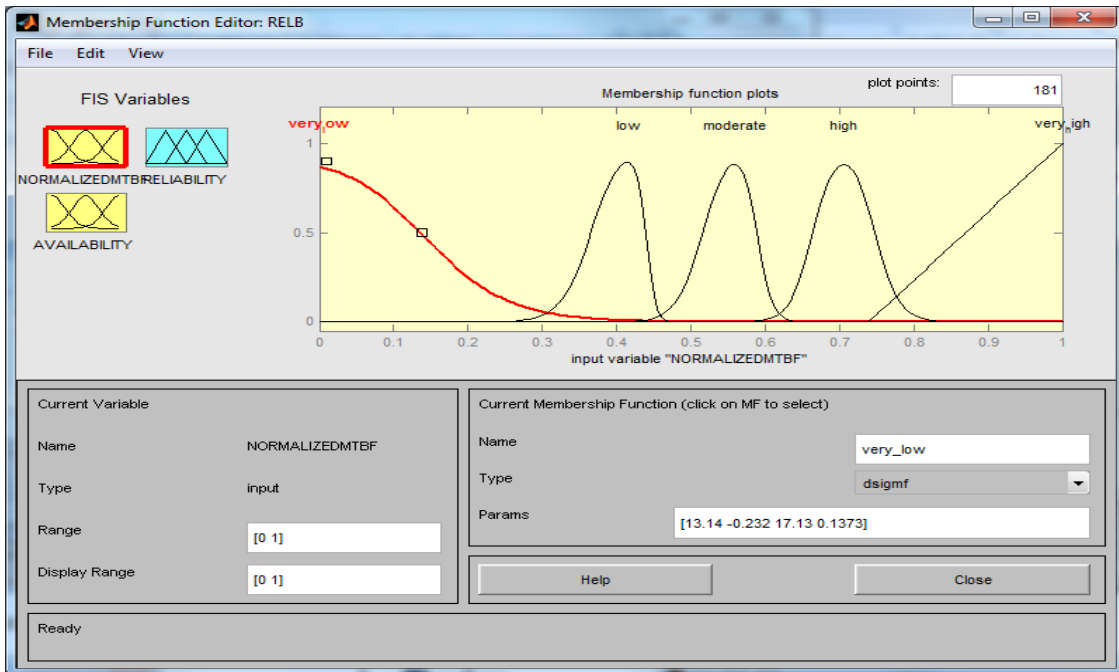


Figure 13: Membership Function for MTBF and Availability

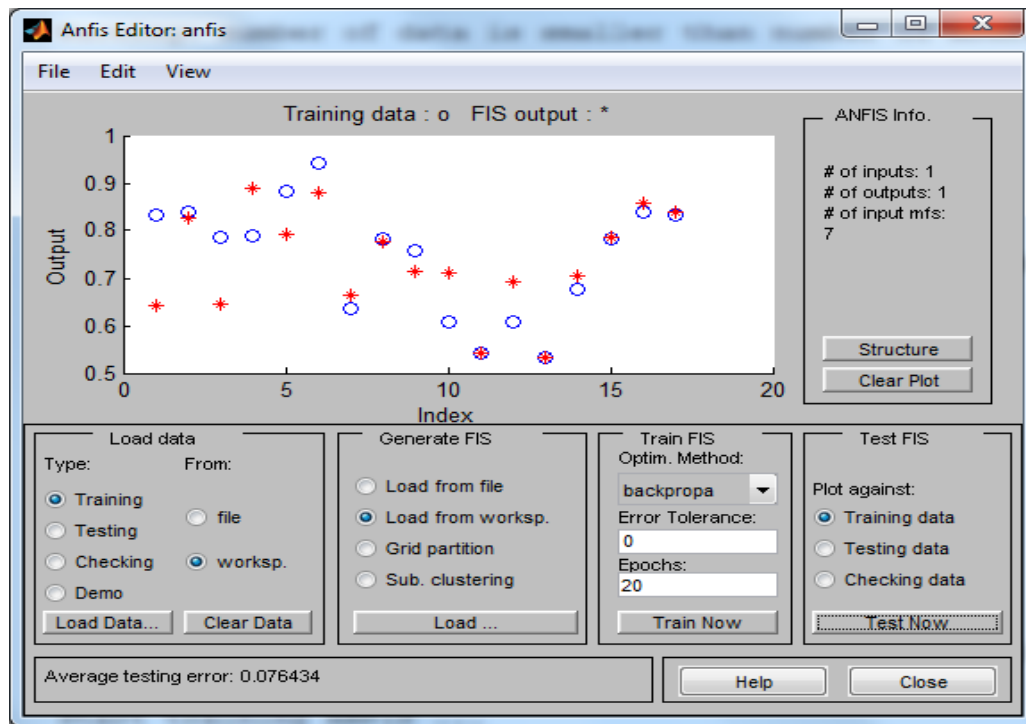


Figure 14: Neuro Fuzzy Inference Model

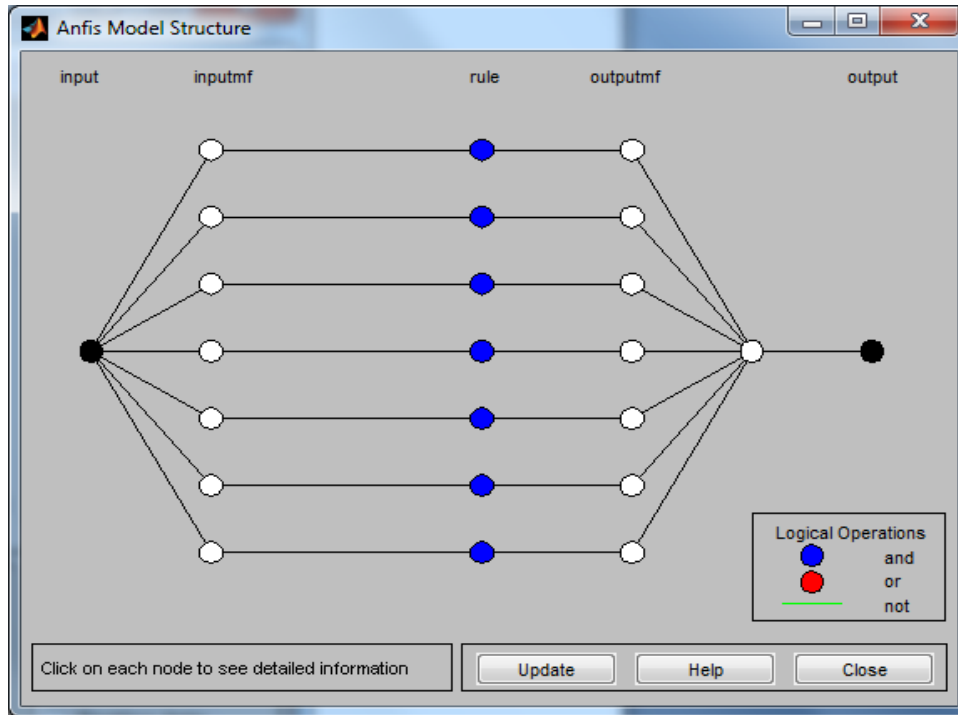


Figure 15: Neuro Fuzzy Structure

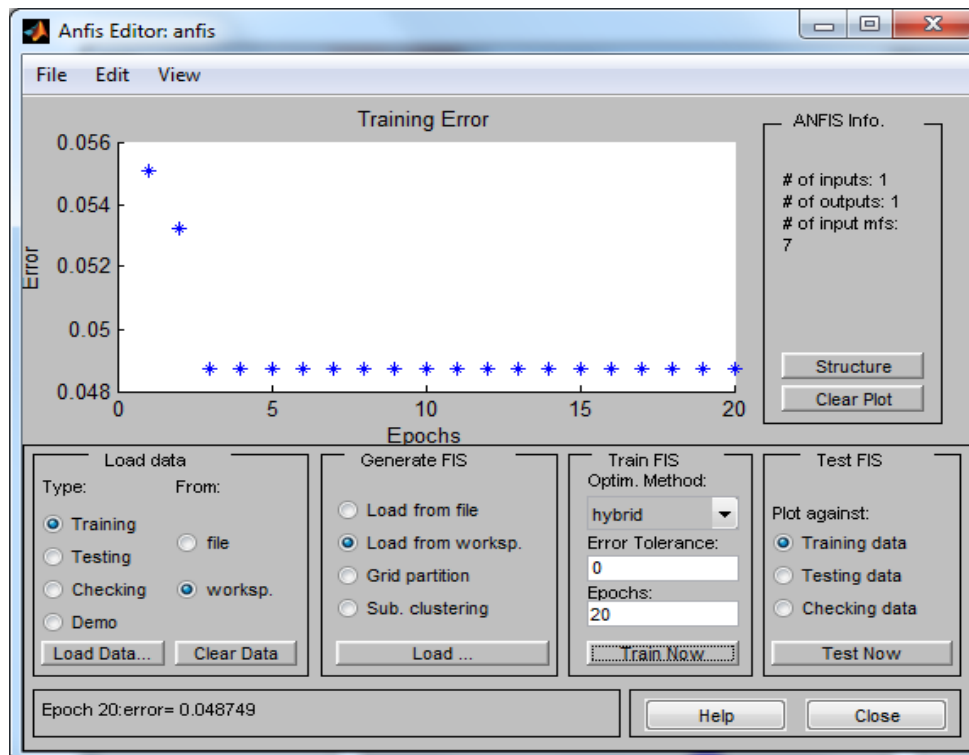


Figure 16: Error Tolerance

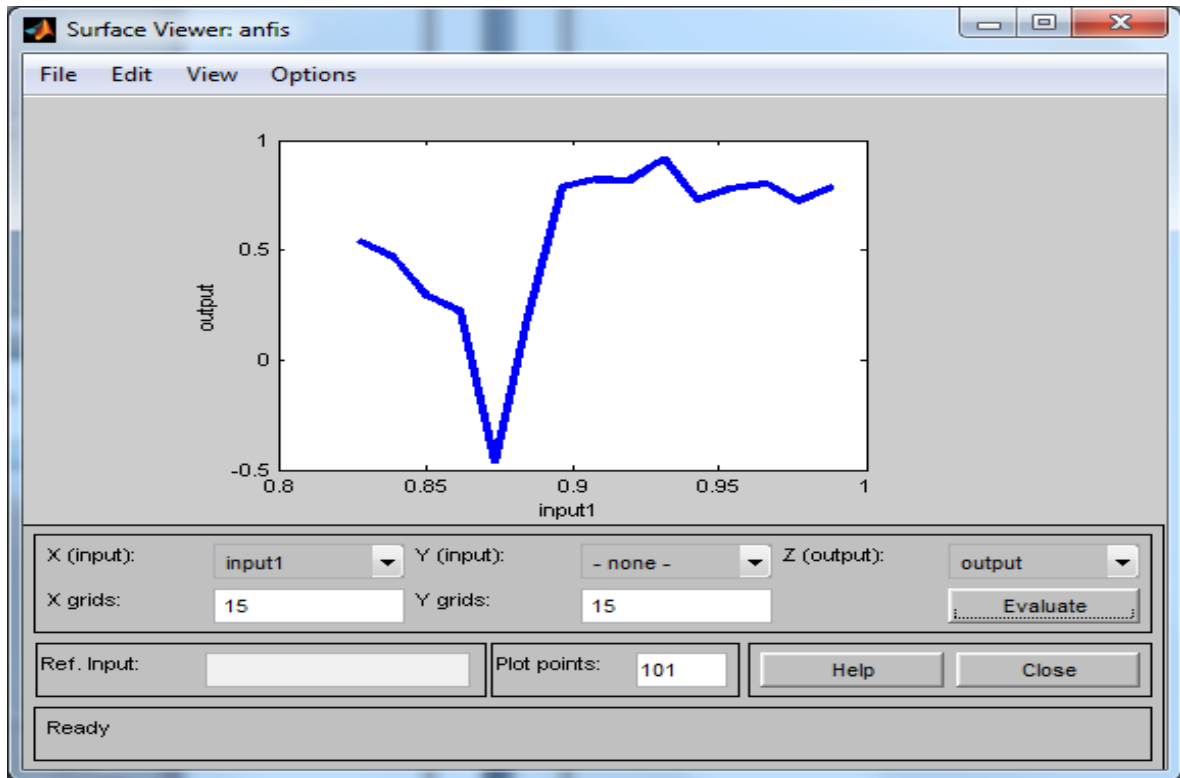


Figure 17: Performance Analysis

4.8 Theoretical Validation

The theoretical validation is performed using mathematical approximation of assessed software reliability using the proposed approach. This mathematical approximation is based on Euler's theorem. A theoretical valuation is done with the formula mentioned in the context. Five iterations are performed for getting the accuracy level of the Approximated Software Reliability and the results are shown in Table 5, Table 6, Table 7, Table 8, Table 9 respectively as follows:

1st Iteration

Table 5: Calculation of Reliability & its Approximation at 1st Iteration

x	y(Measured Value)	f(a)=MTBF/(1+MTBF)	Approximated value= y - h * f(a)
1	45.18	0.97835	45.123
2	20.78	0.95409	20.725
3	46.05	0.97875	45.993
4	23.51	0.9592	23.454
5	15.32	0.93873	15.266
6	24.09	0.96014	24.034
7	38.5	0.97468	38.443
8	86.14	0.98852	86.083
9	33.78	0.97125	33.728
10	17.12	0.94481	17.065
11	4.8	0.82759	4.752
12	8.43	0.89396	8.378
13	5.03	0.83416	4.982
14	8.46	0.89429	8.408
15	11.08	0.91722	11.027
16	14.18	0.93412	14.126
17	9.69	0.90645	9.637

2nd iteration

Table 6: Calculation of Reliability & its Approximation at 2nd Iteration

x	y(Measured Value)	f(a)=MTBF/(1+MTBF)	Approximated value= y - h * f(a)
1	45.12	0.97832	45.07
2	20.73	0.95397	20.67
3	45.99	0.97872	45.94
4	23.45	0.95911	23.4
5	15.27	0.93852	15.21
6	24.03	0.96005	23.98
7	38.44	0.97465	38.39
8	86.08	0.98852	86.03
9	33.73	0.9712	33.67
10	17.07	0.94464	17.01
11	4.752	0.82615	4.704
12	8.378	0.89337	8.326
13	4.982	0.83283	4.934
14	8.408	0.89371	8.356
15	11.03	0.91685	10.97
16	14.13	0.93389	14.07
17	9.637	0.90599	9.584

3rd Iteration

Table 7: Calculation of Reliability & its Approximation at 3rd Iteration

x	y(Measured Value)	f(a)=MTBF/(1+MTBF)	Approximated value= y - h * f(a)
1	45.07	0.97829	45.01
2	20.67	0.95385	20.62
3	45.94	0.97869	45.88
4	23.4	0.95901	23.34
5	15.21	0.93832	15.16
6	23.98	0.95996	23.92
7	38.39	0.97461	38.33
8	86.03	0.98851	85.97
9	33.67	0.97116	33.62
10	17.01	0.94448	16.96
11	4.704	0.82468	4.656
12	8.326	0.89277	8.274
13	4.934	0.83148	4.886
14	8.356	0.89312	8.304
15	10.97	0.91649	10.92
16	14.07	0.93365	14.02
17	9.584	0.90552	9.531

4th Iteration

Table 8: Calculation of Reliability & its Approximation at 4th Iteration

x	y(Measured Value)	f(a)=MTBF/(1+MTBF)	Approximated value= y - h * f(a)
1	45.01	0.97827	44.952
2	20.62	0.95374	20.56
3	45.88	0.97867	45.822
4	23.34	0.95892	23.286
5	15.16	0.93811	15.104
6	23.92	0.95987	23.866
7	38.33	0.97457	38.272
8	85.97	0.9885	85.912
9	33.62	0.97111	33.56
10	16.96	0.94431	16.9
11	4.656	0.8232	4.608
12	8.274	0.89217	8.222
13	4.886	0.83011	4.838
14	8.304	0.89252	8.252
15	10.92	0.91611	10.868
16	14.02	0.93341	13.964
17	9.531	0.90504	9.479

5th Iteration

Table 9: Calculation of Reliability & its Approximation at 5th Iteration

x	y(Measured Value)	f(a)=MTBF/(1+MTBF)	Approximated value= y - h * f(a)
1	44.95	0.97824	44.895
2	20.56	0.95362	20.505
3	45.82	0.97864	45.765
4	23.29	0.95882	23.23
5	15.1	0.9379	15.05
6	23.87	0.95978	23.81
7	38.27	0.97454	38.215
8	85.91	0.98849	85.855
9	33.56	0.97106	33.504
10	16.9	0.94413	16.845
11	4.608	0.82168	4.56
12	8.222	0.89156	8.17
13	4.838	0.82871	4.79
14	8.252	0.89192	8.2
15	10.87	0.91574	10.815
16	13.96	0.93317	13.91
17	9.479	0.90457	9.427

% Reliability= (Average of Approximated vales/ Average of observed Values) x 100

% Reliability after 5th iteration = (23.973/ 24.027)*100=0.9977=99.77%

In 5th iteration, as the study got 99.77% so iteration process will be stopped, since the occurrence of good approximated % of reliability.

4.9 Statistical Validation

A statistical hypothesis is an assertion or conjecture concerning one or more population. To prove that a hypothesis is true, or false, with absolute certainty, we would need absolute knowledge, hypothesis testing concerns on how to use a random sample to judge if it is evidence that supports or not the hypothesis.

4.9.1 Hypothesis Testing

Hypothesis testing or significance testing is a method for testing a claim or hypothesis about a parameter in a population, using data measured in a sample. For obtaining the significance of the approach, the researcher has proposed a null hypothesis as well as an alternate hypothesis. The hypothesis is tested through chi-square test. The objective is to reject the null hypothesis and accept the alternate one. The hypotheses are as follows:

H₀: Reliability estimates obtained through MTBF and availability are not significantly comparable/close to those obtained from mathematical approximated values theoretically.

H_a: Reliability estimates obtained through MTBF and availability are significantly comparable/close to those obtained from mathematical approximated values theoretically.

4.9.2 Chi-Square Test for Approximated Software Reliability

A **chi-squared test**, also referred to as χ^2 test (or **chi-square test**). The chi-square test is always testing what scientists call the **null hypothesis**, which states that there is no significant difference between the expected and observed result [154].

Calculation of Chi-Square variable for Approximated Software Reliability

The Chi- Square test is performed on MTBF values of all 17 programs of dataset. Table 3 shows values of MTBF and MTTR which reflects that for all 17 software systems, all of the metrics are highly correlated with each other, with measured MTBF values and Approximated Software Reliability being the most significantly correlated. In order to further assure, χ^2 test has been used for testing the proposed null hypothesis. The results of chi-sqaure test are presented in Table 10.

Table 10: Calculation of Chi-Square for Approximated Software Reliability (SR)

x	Observed	Expected Frequency(e_i)	f_i-e_i	(f_i-e_i)/ e_i²
1	45.18	1.426104	43.7539	21.51366
2	20.78	1.426104	19.3539	9.516256
3	46.05	1.426104	44.6239	21.94144
4	23.51	1.426104	22.0839	10.85859
5	15.32	1.426104	13.8939	6.83159
6	24.09	1.426104	22.6639	11.14377
7	38.5	1.426104	37.0739	18.22913
8	86.14	1.426104	84.7139	41.65357
9	33.784	1.426104	32.3579	15.91028
10	17.12	1.426104	15.6939	7.716644
11	4.80	1.426104	3.373896	1.658935
12	8.43	1.426104	7.003896	3.443795
13	5.03	1.426104	3.603896	1.772025
14	8.46	1.426104	7.033896	3.458546
15	11.08	1.426104	9.653896	4.746792
16	14.18	1.426104	12.7539	6.271055
17	9.69	1.426104	8.263896	4.063333

From the Table 10 by finding out the average of chi-square variable the value obtained is: $\chi^2 = 11.21938$. The computed value of chi-square 11.21938 is greater than the critical value of chi-square for 1 degree of freedom at 0.05 level of significance, which is 3.84. The test indicates that there is significant relationship between approximated Software Reliability and the measured values of MTBF of the dataset of all 17 systems at 0.05 level of significance. Hence, Null Hypothesis is strongly rejected and alternate hypothesis is accepted.

4.10 Comparison of Proposed Approach with conventional Fuzzy System

The inputs to the Neuro Fuzzy system are Normalized MTBF and availability which is shown in the Figure 13. The researcher has noted down that for the reliability, the outcome from the conventional system is **84.5 %** and from the proposed approach is **95.5 %** using MATLAB software tool. The following program has been run in MATLAB environment and produces the results of validation.

MATLAB Program for Practical Validation

```
clear
% MTBF input
aa= VECTOR OF MTBF VALUES;
af=aa./mean(aa);
% Availability Input
b= VECTOR OF AVAILABILITY VALUES;
% Read the FIS structure named as RELB
F=readfis('RELB.fis');
% Evaluate the input with the given fuzzy structure
ff=evalfis([aa./max(aa)+.7,b+.7],F)
% this section is regarding ANFIS
% train the data for it give MTBF and Availability as inputs
trnData = [af , b];
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([b]',out_fis)'
mean(oo)
```

The results obtained are shown in Figure 18. From the above the performance assessment, the improvement of **11%** is achieved with the proposed approach. The evaluation criteria like Mean Squared Error(MSE) and Average Error(AE) [155] are used to evaluate the proposed model performance against conventional Fuzzy Inference System(FIS) based approach.

MSE= ((Theoretical validation Value- practical Validation Value)/ Total no of readings)²

$$\text{MSE} = ((99.77-95.5)/17)^2 = 0.061038$$

AE = ((Theoretical validation Value- practical Validation Value)/ Total no of readings)=0.247;

$$\text{AE} = (99.77-95.5)/17 = 0.0247$$

Figure 18 shows the results obtained through the experiment done using MATLAB for obtaining the Reliability Assessment differences between conventional FIS and the proposed approach. The performance comparison table is prepared based on MSE and AE and shown in the Table 11.

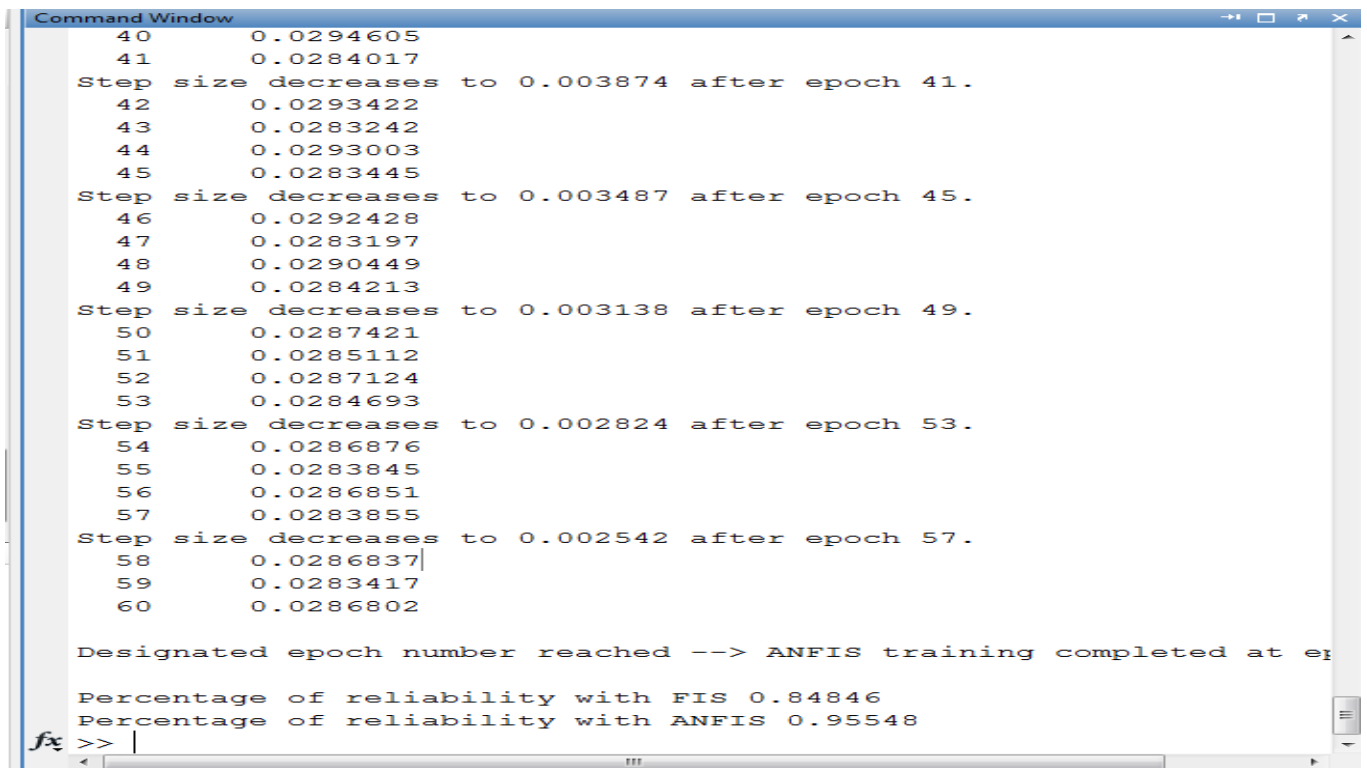


Figure 18: Practical Validation of the Reliability Percentage obtained using MATLAB

Table 11: Performance Comparison between FIS & ANFIS of SR Estimation

Method	MSE	AE
FIS	0.799	0.894
ANFIS	0.061	0.247

5. Significance of Study

This study seems to have the following significance:

- The research will provide an insight to the software developers and testers to agree with high quality software.
- Software Reliability measures suggested may be embedded in an organizations and software engineering process to produce more improved software as far as quality is concerned under techno-economic and reliability constraints.
- It may help to understand the design aspect of the software system clearly, so that development of the system will be made easy by monitoring the reliability.
- It may help to provide best estimation of Software Reliability, so that developers can get clear estimation of total software development cost.
- It facilitates the planning of new activities.
- The proposed model for reliability assessment based on Neuro Fuzzy Systems helps to come out with good availability, MTBF and Reliability estimation models with wider acceptability.
- It also helps future researches in the Neuro Fuzzy based systems to develop new approaches.
- The proposed approach may find a place among the measurement tools for assessing the Software Reliability based on Neuro Fuzzy Systems.

6. Future Work and Suggestions

Research is a continuing activity. As a future researcher plans the following tasks are to be completed:

- 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.

- The researchers may plan to conduct more experiments on real time projects data to derive more clear conclusions about the metrics and models values.
- Accuracy of the Fuzzy Rule Generation for the Fuzzy Inference System can further be improved with decision tree techniques.

7. 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.
- The Neuro Fuzzy system also suffers from stability issues, where the output may not retain the same all the time.
- High values of MTBF and Availability increases the Software Reliability, means there exists a linear relation between the parameters considered.
- The approach can be used to demonstrate how to measure Software Reliability only but doesn't demonstrate how to maximize them for the betterment of Software Reliability.
- The model is validated with only a small dataset(17 programs of GlaceEMR software).

The research concentrated on reliability factors like availability, MTBF and uptime, downtime, number of breakdowns only.

8. Thesis Outline

This thesis on Neuro Fuzzy model based Software Reliability assessment is organized as:

Chapter 2 explains vividly about the literature review on Software Reliability, Software Reliability Engineering process, Neural Networks, Fuzzy Logic, Neuro

Fuzzy models and the earlier approaches that are present on the same context for the assessment of Software Reliability, its flaw and overcoming. It also stress on the reliability model used in the research. It mainly focus on the need of reliability, importance of reliability assessment, with existing literature and the research gaps in the existing literature and what needs to be done.

Chapter 3 mainly focuses on fuzzy modeling and Neuro Fuzzy modeling, relation between the findings and the approach used in the models. It also provides the mathematical analysis of the proposed approach of Neuro Fuzzy model for Software Reliability Assessment. It presents the model with five phases like Identification Phase, Quantification Phase, Measurement Phase, Verification and Validation Phase and Finalization Phase. The review and revision is also done at every phase of the proposed approach for the effective assessment of Software Reliability.

Chapter 4 shows the implementation of the model under different experimental datasets. Also, analyses the performance and evaluates the outcome of the conventional FIS based system of Software Reliability Assessment and the current approach. The implementation and validation is done using the MATLAB software environment.

Chapter 5 ends up with the conclusions, significance of the research, research findings and suggestions for future work and the limitations of the approach.

References

- [1] Hoyer, R. W. and Hoyer, B. B. Y.(2001). "What is quality?".*Quality Progress*, no. 7, pp. 52-62.
- [2] Crosby P.B. (1979). "Quality is Free the Art of Making Quality Certain".*New York,McGraw-Hill*.
- [3] Deming & Edwards W. (1986). "Out of Crisis".*MIT Press*.
- [4] Feigenbaum A. V.(1983)."Total Quality Control".*McGraw – Hill*.
- [5] IEEE Std 610.12 (1990). *IEEE Standard Glossary of Software Engineering Terminology*, New York.
- [6] ISO/IEC Std 9126-1 (2001)" Software Engineering – Product Quality “, *Part 1: Quality Models, International Organization for Standards2001*.
- [7] McCall J.A, Richards P.K & Walters G.F (1977)" Factors in Software Quality”, *Nat’l Tech Information Services– Vol. 1-2*.
- [8] AIAA/ANSI (1992) "Recommended Practice for Software Reliability, the American Institute of Aeronautics and Astronautics”, *Washington DC, Aerospace Center, R-013, ISBN 1-56347-024-1*.
- [9] Michael R. Lyu(1996),"Handbook of Software Reliability Engineering”, *IEEE*.
- [10] Wasserman, Gary(2002),"Reliability verification, Testing and analysis in engineering design",*Marcel Dekker incorporated, New York, USA*, pp2-10.
- [11] Schneidewind, N. F., (2001), "Modeling the fault correction processes”, *Proceedings of the 12th International Symposium on Software Reliability Engineering*, pp. 185-190.
- [12] Myrtveit, I., Stensrud, E. and Shepperd, M., (2005), "Reliability and validity in comparative studies of software prediction models”, *IEEE Transactions on Software Engineering*, vol. 31, no. 5, pp. 380-391.
- [13] Falcone, G. Hierachy(2010), "Aware Software Metrics in Component Composition Hierarchies”,*PhD thesis, University of Mannheim, 2010*.
- [14] Bass, L., Clements, P., and Kazman, R.(2003), "Software Architecture in Practice”, *Second Edition. Addison-Wesley, Reading, MA, USA*.

- [15] Jain, R.(1991), "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling", *Wiley*.
- [16] Smith, C. U. and Williams, L. G. (2000), Software performance anti patterns, *in Workshop on Software and Performance*, pages 127-136.
- [17] Immonen, A. and Niemela, E. (2008), "Survey of reliability and availability prediction methods from the viewpoint of software architecture", *Software and System Modeling*, 7(1):49{65}.
- [18] AIAA/ANSI (1992) "Recommended Practice for Software Reliability, The American Institute of Aeronautics and Astronautics", *Washington DC, Aerospace Center*, R-013, ISBN 1-56347-024-1.
- [19] Rosenberg Linda, Hammer Ted & Shaw Jack (1998) "Software Matrices and Reliability", *ISSRE*.
- [20] N. E. Fenton and M. Neil(1999), "A critique of software defect prediction models", *IEEE Transactions on Software Engineering*, 25(5):675–689.
- [21] N. E. Fenton and N. Ohlsson(2000), "Quantitative analysis of faults and failures in a complex software system", *IEEE Transactions on Software Engineering*, 26(8):797–814.
- [22] Wood Alan(1996), "Software Reliability Growth Models", *Tandem Technical Report- Vol. 19.1*-part number 130056.
- [23] LyuM.R(1996), "Handbook of Software Reliability Engineering", *IEEE Computer Society Press*.
- [24] Schneidewind N. F(1993), "Software Reliability Model with Optimal Selection of Failure Data" , *IEEE Transactions on Software Engineering - Vol 19(11)*, Pp No. 1095-1104.
- [25] L. Tian and A. Noore (2004), "Software Reliability Prediction Using Recurrent Neural Network with Bayesian Regularization", *International Journal of Neural Systems*, vol. 14, no. 3, pp. 165–174.
- [26] P.Werbos. (1988) "Generalization of Back propagation with Application to Recurrent Gas Market Model," *Neural Network*, vol. 1, pp. 339–356.

- [27] R. Shadmehr and D. Z. DSArgenio. (1990) “A Comparison of a Neural Network Based Estimator and Two Statistical Estimators in a Sparse and Noisy Data Environment,” in *IJCNN, vol. 1, Washington D.C*, pp. 289–292.
- [28] N. Karunanithi, Y. Malaiya, and D. Whitley (1991) “Prediction of Software Reliability Using Neural Networks,” in *Proceedings IEEE International Symposium on Software Reliability Engineering. Austin, TX: IEEE*, pp. 124–130.
- [29] T. M. Khoshgoftaar, A. S. Pandya and H. More (1992) “A Neural Network Approach For Predicting Software Development Faults”,*Research Triangle Park, NC: Proceedings of Third International Symposium on Software Reliability Engineering*, pp. 83–89.
- [30] Y. Singh and P. Kumar(2010) “Prediction of Software Reliability Using Feed Forward Neural Networks,” in *Computational Intelligence and Software Engineering (CiSE), I. Conference, Ed. IEEE*, pp. 1–5.
- [31] M. M. T. Thwin and T. S. Quah, Eds.(2002), “Application of Neural Network for Predicting Software Development Faults using Object-Oriented Design Metrics”, *vol. 5. Proceedings of the 9th International Conference on Neural Information Processing (ICONIP’02)*.
- [32] N. Karunanithi and D. Whitley(1992), “Prediction of Software Reliability Using Feed forward and Recurrent Neural Nets,” in *Neural Networks, 1992. IJCNN, vol.1. Baltimore, MD: IEEE*, pp. 800–805.
- [33] R. Sitte(1999) “Comparison of software-reliability-growth predictions: Neural Networks vs. parametric-recalibration,” *Reliability, IEEE Transactions*, vol. 48, no. 3, pp. 285–291.
- [34] N. RajKiran and V. Ravi (2007), “Software Reliability Prediction using Wavelet Neural Networks,” in *International Conference on Computational Intelligence and Multimedia Applications, vol. 1. Sivakasi, Tamil Nadu: IEEE*, pp. 195 – 199.

- [35] J. H. Lo (2009), “The Implementation of Artificial Neural Networks Applying to Software Reliability Modeling,” *Control and Decision Conference, 2009. CCDC '09, Chinese*, pp. 4349 – 4354.
- [36] L. Zhao, J. pei Zhang, J. Yang, and Y. Chu (2010), “Software Reliability growth model based on fuzzy wavelet neural network” in *2nd International Conference on Future Computer and Communication (ICFCC), vol. 1. Wuhan: IEEE*, pp. 664–668.
- [37] Quah, T. and Thwin, M.(2002),“Application of Neural Networks for Predicting Software Development Faults Using Object-Oriented Metrics”, *Proceedings of Ninth International Conference on Neural Information Processing (ICONIP'02), vol. 5*, pp. 2312-2316.
- [38] R. Sitte. (1999), “Comparison of Software Reliability growth prediction: Neural Networks VS Parametric recalibration,” *IEEE Trans. on Reliability*, vol. 48, no.3, pp. 285-291.
- [39] Aljahdali, S., Sheta, A., and Rine, D. (2002), “Predicting Accumulated Faults in Software Using Radial Basis Function Network”, *Proceedings of the ISCA 17th International Conference on Computers and their Application*, pp. 26-29.
- [40] Kaufmann A and Gupta M. M.(1988), *Fuzzy mathematical models in engineering and management science, North-Holland, Amstredam*.
- [41] Cai K.Y.(1996), System failure and fuzzy methodology: an introductory overview,*Fuzzy Sets and Systems*, 113-133
- [42] Specht, D.F., (1991). A general regression neural network,*IEEE Transactions on Neural Networks*,2 (6), 568-576.
- [43] Kasabov, N.K., (2002). DENFIS: dynamic evolving neural-Fuzzy Inference System and its application for time-series prediction,*IEEE Transactions on fuzzy systems* 10(2).
- [44] Makridakis, S., Anderson, A., Carbone, R., Fildes, R., Hibdon, M., Lewandowski, R., Newton, J., Parzen, E., Winkler, R., (1982), “The

- accuracy of extrapolation (time series) methods: results of a forecasting competition”, *Journal of Forecasting* 1, 111-153.
- [45] Perrone, M.P., Cooper, L.N., (1993), “When networks disagree: ensemble methods for hybrid neural networks”, in *Mammone, R.J. (Ed.), Neural Networks for speech and Image processing*, Chapman Hall, pp. 126-142.
- [46] Yu, L., Wang, S.Y., Lai, K.K., (2005), “A novel non-linear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates”, *Computers and Operations Research* 32 (10), 2523-2541.
- [47] Olmeda, I., Fernandez, E., (1997), “Hybrid classifiers for financial multi-criteria decision making: the case of bankruptcy prediction”, *Computational Economics* 10, 317-335.
- [48] Zhou, Z-H., Wu, J., Tang, W., (2002), “Ensembling Neural Networks: many could be better than all”, *Artificial Intelligence* 137, 239-263.
- [49] J.T. Duane. (1964), "Learning curve (approach to reliability monitoring)", *IEEE Trans. on Aerospace*, AS-2. 563-566.
- [50] Z. Jelinski and P. Moranda (1972), "Software Reliability research in Statistical Computer Performance Evaluation", *W. Freiberger, Ed. New York: Academic*, pp. 465-484.
- [51] M. L. Shooman (1972), "Probabilistic models for Software Reliability prediction," in *Statistical Computer Performance Evaluation*, *W. Freiberger, Ed. New York: Academic*, pp. 485-502.
- [52] J. D. Musa (1971), "A theory of Software Reliability and its application", *IEEE Trans. Software Eng.*, vol. SE-1, pp. 312-327.
- [53] J. D. Musa and K. Okumoto (1983), "A logarithmic Poisson execution time model for Software Reliability measurement," in *Proceedings of 7th International Conference on Software Eng., Orlando, FL*, pp. 230-237.
- [54] A. L. Goel, K. Okumoto (1979), "Time-dependent error detection rate model for Software Reliability and other performance measures", *IEEE Trans. Rel.*, R-28(3), pp. 206-211.

- [55] Yamada, M. Ohba, and S. Osaki (1983), "S-shaped reliability growth modeling for software error detection", *IEEE Trans. Rel.*, vol. R-32, pp. 475-478.
- [56] J. D. Musa, K. Okumoto (1982), "Software Reliability models: concepts, classification, comparisons, and practice", *Proc. Electronic Systems Effectiveness and Life Cycle Costing Conference, Norwich, U. K.*, July 19-31, *NATO ASI Series, Vol. F3*, (Ed: J. W. Skwirzynski) Springer-Verlag, Heidelberg, pp. 395-424.
- [57] Härtler, G. (1989), "The non-homogeneous Poisson process—a model for the reliability of complex repairable systems." *Microelectronics Reliability*, 29.3: 381-386.
- [58] Yamada, Shigeru, Koichi Tokuno, and Shunji Osaki. (1992), "Imperfect debugging models with fault introduction rate for Software Reliability assessment", *International Journal of Systems Science*, 23.12: 2241-2252.
- [59] Yager, Susan E., et al.(1997), "Microcomputer playfulness: stable or dynamic trait?", *ACM SIGMIS Database* 28.2: 43-52.
- [60] Hossain, Syed, and Ram C. Dahiya.(1993), "Estimating the parameters of a non-homogeneous Poisson-process model for Software Reliability", *IEEE Transactions on Reliability*, 42.4: 604-612.
- [61] Kuo, Lynn, and Tae Young Yang.(1996), "Bayesian computation for non-homogeneous Poisson processes in Software Reliability", *Journal of the American Statistical Association*, 91.434: 763-773.
- [62] Singpurwalla, Nozer D., and Simon P. Wilson.(1994)," Software Reliability modeling", *International Statistical Review/Revue Internationale de Statistique* : 289-317.
- [63] Tian, Jeff.(1999), "Measurement and continuous improvement of Software Reliability throughout software life-cycle", *Journal of Systems and Software* 47.2: 189-195.

- [64] Inoue, Shinji, and Shigeru Yamada.(2007), "Discrete program-size dependent Software Reliability assessment: Modeling, estimation, and goodness-of-fit comparisons", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 90.12: 2891-2902.
- [65] R. Kruse, J. Gebhardt and F. Klawonn (1994a).Foundations of Fuzzy Systems.*Wiley, Chichester.*
- [66] R. Kruse, J. Gebhardt and R. Palm, eds. (1994b), Fuzzy Systems in Computer Science. *Vieweg, Braunschweig.*
- [67] W. Pedrycz and H. C. Card (1992). "Linguistic Interpretation of Self-Organizing Maps",*in Proc. IEEE Int. Conf. on Fuzzy Systems 1992*, pages 371–378, San Diego, CA.
- [68] D. Nauck, U. Nauck and R. Kruse (1996),“Generating Classification Rules with the Neuro Fuzzy System NEFCLASS. *inProc. Biennial Conference of the North American Fuzzy Information Processing Society NAFIPS’96*, pages 466–470, Berkeley, CA.
- [69] Glossary Working Party of the International Software Testing Qualification Board (ISTQB), van Veenendaal, E. (ed.) (2006), *Standard glossary of terms used in Software Testing, Version 1.2.*
- [70] Standards Coordinating Committee of the IEEE Computer Society (1991), *IEEE standard computer dictionary 610.*
- [71] Storey, N (1996), Safety-Critical Computer Systems, *Prentice Hall.*
- [72] Ammann, P, Offutt, J. (2008), Introduction to Software Testing, *Cambridge University Press.*
- [73] BonthuKotaiyah, Raees Ahmed Khan, MuralidharVejendla (2013),"Model For Software Errors Prediction using Machine Learning to Improve the Software Reliability", *International Journal of Soft Computing and Software Engineering [JSCSE]*, Vol. 3, No. 3, pp. 314-319, eISSN: 2251-7545,Doi: 10.7321/jscse.v3.n3.47
- [74] BonthuKotaiyah and R.A. Khan (2012), “A Survey on Software Reliability Assessment by using Different Machine Learning Techniques”,

- International Journal of Scientific & Engineering Research(IJSER)*, Volume 3, Issue 6, P. No.1-7.
- [75] BonthuKotaiah, 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.
- [76] BonthuKotaiah, R. A. Khan (2013), “Software Reliability Growth Models- A Revisit”, *IJAC(International Journal Of Advanced Computing (Print Journal)*, ISSN. 0975-7686, P.No.32-36, Online Link: http://Journals.Griet.ac.in/Images/V5i1jan2013_7.Jpg
- [77] BonthuKotaiah, R.A. Khan (2015),“An Analysis of Software Reliability Assessment with Neuro Fuzzy based Expert Systems ”, *Elsevier B.V. ., Procedia Computer Science*, ISSN No.1877-0509, Proceedings of International conference on Soft Computing and Software Engineering(SCSE), University of California, USA (Paper ID-40819).
- [78] BonthuKotaiah, 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.
- [79] BonthuKotaiah, 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.
- [80]BonthuKotaiah, R.A. Khan (2013), “Assessing the level of Software Reliability using Neuro Fuzzy System”, *International Conference on Information & Engineering Sciences (ICIES -2013)*(Print Journal).
- [81] BonthuKotaiah, R.A. Khan(2013), “GMR(Group Maturity Rating) with Fuzzy Logic Based Software Defects Rating for the Assessment of Software Development Life Cycle(SDLC) Performance”,*Advances in Engineering and Technology Convergence(AETC)*, Bangkok, Thailand, ISBN No: 978-93-82208-89-1, P.No.25-29.

- [82] BonthuKotaiah, R.A. Khan (2013), “MLP Neural Networks Based Approach for the Assessment of Software performance to Improve the Software Reliability”, *The International Conference on Mathematics, Statistics and Computer Engineering (ICMSCE)*, ISSN No.2568-4345, (Print Journal).
- [83] BonthuKotaiah, R.A. Khan (2013), “The Preliminary 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.
- [84] BonthuKotaiah, 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.
- [85] BonthuKotaiah, 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*.
- [86] BonthuKotaiah, R.A. Khan,Muralidhar V.(2013), “ Model for Software Errors Prediction Using Machine Learning to Improve The Software Reliability”, *The Proceedings of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13]*, San Francisco State University, California, U.S.A., Doi: 10.7321/jscse.v3.n4.47.
- [87] BonthuKotaiah, B. Sunil Kumar, R.A. Khan(2012), “A Survey on Software Reliability Assessment by using different Machine Learning Techniques, *International Conference on Information Technology, Electronics and Communications(ICITEC-2012)*, Hyderabad, India, Pp No.11-17.
- [88] BonthuKotaiah, 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, Souvenir, BBAU, Lucknow, Conference Proceedings.

- [89] BonthuKotaiah , 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.
- [90] BonthuKotaiah , 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, BabasahebBhimraoAmbedkar University(BBAU), Lucknow, Conference Proceedings.
- [91] BonthuKotaiah, R.A. Khan(2012), “ A Survey on Software Reliability Assessment for Machine Learning Techniques”, *National Conference on Recent Trends in Information Technology*, PVP Siddartha Institute of Technology, Vijayawada, Conference Proceedings, Pp No.95-102.
- [92] BonthuKotaiah , R.A. Khan(2015), “Software Reliability Measurement with Neural Network based Soft Computing Systems”, “*Third International Conference on Emerging Research in Computing, Information, Communication And Applications*” (ERCICA-2015), Springer, Conference Proceedings.
- [93] "Khan, M Taimoor; Mustafa. K; Ahson, SI(2006), "Software Quality,Concepts and Practices”, *Alpha Science Intl ltd, England*.
- [94] "Mustafa, K; Khan RA.(2005), “Quality Metric Development Framework (QMDF)”,*Journal of Computer Science*,1,3,437-444.
- [95] "Yadav A, Khan, RA (2009) ",Complexity: A Reliability Factor, *IEEE International Advance Computing Conference*,2375-2378.
- [96] "Yadav A, Khan, RA (2009) ",Critical review on Software Reliability models,*International Journal of recent trends in Engineering*, 2-3.
- [97] "Nayak, Sandeep Kumar; Khan, Raees Ahmad; Beg, MdRizwan(2012)",Reliable Requirement Specification: Defect Analysis Perspective,*Global Trends in Information Systems and Software*

- Applications*,740-751,Springer.
- [98] "Yadav A, Khan RA (2011), "Class cohesion complexity metric(C3 M),"2nd International Conference on Computer and Communication Technology (ICCCCT), 2011",363-366,IEEE.
- [99] "Nazir, Mohd, Khan, Raees A, Mustafa, K. (2010)", "Testability Estimation Framework,"*Internaional Journal of Computer Applications*.
- [100] Cai, K., Bai, C., Zhong, X., (2003),"Introduction to reliability models of component based software system",*J.Xi'anJiaotong Univ.* 37 (6), 560–564.
- [101] Cheung, R.C., (1980),"A user oriented Software Reliability model",*IEEE Trans. Softw. Eng.* 6 (2), 118–125.
- [102] Dimov, Aleksandar, Sasikumar, Punnekkat, (2010),"Fuzzy reliability model for component-based software systems", *36th EUROMICRO Conference on Software Engineering and Advanced Applications*, pp. 39–46.
- [103] Dong, W., Huang, N., Ming, Y., (2008), Reliability analysis of component-based software based on relationships of components, *IEEE Conference on Web Services*, pp. 814–815.
- [104] Fiondella, Lance, Rajasekaran, Sanguthever, Gokhale, Swapana, (2013), Efficient Software Reliability analysis with correlated component failures,*IEEE Trans. Reliab.* 62 (1), 244–255.
- [105] Gokhle, S.S., (2007),"Architecture based Software Reliability analysis: overview and limitations",*IEEE Trans. Dependable Secure Comput.* 4 (1), 32–40.
- [106] Gokhle, S.S., Dong, W.E., Trivedi, K.S., Horgan, J.R., (1998),"An analytical approach to architecture based Software Reliability prediction", *Proc. of the Third International Computer Performance and Dependability Symposium*, Durham, NC, pp. 13–22.

- [107] Goswami V., Acharya, Y.B., (2009),“Method for reliability estimation of COTS components based software systems”, *International Symposium on Software Reliability Engineering*.
- [108] 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.
- [109] 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.
- [110] Huang, N., Wang, D., Jia, X., (2008), FAST ABSTRACT: an algebra-based reliability prediction approach for composite web services, *19th International Symposium on Software Reliability Engineering*, pp. 285–286.
- [111] Jang, J.R., (1992), ANFIS: adaptive-network-based Fuzzy Inference System, *IEEE Trans. Systems, Man, and Cybernetics*.
- [112] Koziolk H., Becker, S., (2005), Transforming operational profiles of software components for quality of service predictions, *Proc. 10th WCOP’05*, pp. 1–8.
- [113] Krishnamurthy, S., Mathur, A.P., (1997), on the estimation of reliability of a software system using reliabilities of its components, *Proceeding of the Eighth International Symposium on Software Reliability Engineering, Albuquerque, NM*, pp. 146–155.
- [114] Littlewood, B., (1979), Software Reliability model for modular program structure. *IEEE Trans. Reliab.* 28 (3), 241–246.
- [115] Lo, J., (2010), Early Software Reliability prediction based on support vector machines with genetic algorithms, *Fifth IEEE Conference on Industrial Electronics and Applications*, pp. 2221–2226.

- [116] Palviainen, Markov, Evesti, Antti, Ovaska, Eila, (2011),The reliability estimation, prediction and measuring of component – based software. *J. Syst. Softw.*, 1054–1070.
- [117] KirtiTyagi, Arun Sharma, (2014), An adaptive Neuro Fuzzy model for estimating the reliability of component-based software systems, *Applied Computing and Informatics*, 10, 38–51.
- [118] Lee, T., Jung, S. O., In, H. P., & Lee, H. J. (2007),“Cyber Threat Trend Analysis Model Using HMM”,in *Information Assurance and Security, Third International Symposium on* (pp.177-182). *IEEE*.
- [119] Greitzer, Frank L., Andrew P. Moore, Dawn M. Cappelli, Dee H. Andrews, Lynn A. Carroll and Thomas D. Hull (2008), "Combating the insider cyberthreat",*Security & Privacy, IEEE* 6, no. 1:61-64.
- [120] Ingalsbe, Jeffrey A., Louis Kunimatsu, Tim Baeten and Nancy R. Mead, (2008) "Threat modeling: diving into the deep end",*Software, IEEE* 25, no. 1:28-34.
- [121] Ongsakorn, P., Turney, K., Thornton, M., Nair, S., Szygenda, S. &Manikas, T. (2010). Cyber threat trees for large system threat cataloging and analysis. *In Systems Conference, 4th Annual IEEE* (pp. 610-615).*IEEE*.
- [122] Gegick, Michael and Laurie Williams(2005), "Matching attack patterns to security vulnerabilities in software intensive system designs",in *ACM SIGSOFT Software Engineering Notes*, vol. 30, no. 4, pp. 1-7.*ACM*.
- [123] Gandotra, V., Singhal, A., &Bedi, P. (2009),“Threat mitigation, monitoring and management plan-A new approach in risk management”, in *Advances in Recent Technologies in Communication and Computing*.
- [124] Tang, X. & Shen, B. (2009),“Extending Model Driven Architecture with Software Security Assessment”,in*Third IEEE International Conference on Secure Software Integration and Reliability Improvement(SSIRI)* (pp. 436-441).

- [125] Dhillon, Danny(2011), "Developer-Driven Threat Modeling: Lessons Learned in the Trenches",*Security & Privacy, IEEE* 9, no. 4: 41-47
- [126] Okubo, T., Kaiya, H. & Yoshioka, N. (2011), "Effective security impact analysis with patterns for software enhancement", in *Sixth International Conference on Availability, Reliability and Security (ARES)*(pp. 527-534). IEEE.
- [127] Stango, A., Prasad, N. R. &Kyriazanos, D. M. (2009), "A threat analysis methodology for security evaluation and enhancement planning", in *Third International Conference on Emerging Security Information, Systems and Technologies. SECURWARE'09.* (pp. 262-267). IEEE.
- [128] Frantz, T. L., &Carley, K. M. (2009), "Information assurances and threat identification in networked organizations", in *IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA 2009)*. (pp. 1-5). IEEE.
- [129] Debar, H., Becker, M., &Siboni, D. (1992), "A neural network component for an intrusion detection system", in *IEEE Computer Society Symposium on Research in Security and Privacy, Proceedings* (pp. 240-250). IEEE.
- [130] Mukkamala, S., Janoski, G., & Sung, A. (2002), "Intrusion detection using Neural Networks and support vector machines", in *Proceedings of the International Joint Conference on Neural Networks (IJCNN'02)*,(Vol. 2, pp. 1702-1707), IEEE.
- [131] Gong, R. H., Zulkernine, M. &Abolmaesumi, P. (2005), "A software implementation of a genetic algorithm based approach to network intrusion detection" in *Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, First ACIS International Workshop on Self-Assembling Wireless Networks. SNP/SAWN 2005.* (pp. 246-253). IEEE.
- [132] Haslum, K., Abraham, A. &Knapskog, S. (2008), "Hinfra: Hierarchical neuro-fuzzy learning for online risk assessment", in *Second Asia*

- International Conference on Modeling & Simulation, 2008*, AICMS 08. (pp. 631-636). *IEEE*.
- [133] Ngai, E. W. T., & Wat, F. K. T. (2005), "Fuzzy decision support system for risk analysis in E-commerce development", *Decision support systems*, 40(2), 235-255.
- [134] Gandotra, V., Singhal, A. & Bedi, P. (2010), "A step towards secure software system using Fuzzy Logic", in *2nd International Conference on Computer Engineering and Technology (ICCET)* (Vol. 1, pp. V1-427), *IEEE*.
- [135] BLi, S. (2000), "The development of a hybrid intelligent system for developing marketing strategy", *Decision Support Systems*, 27(4), 395-409.
- [136] 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.
- [137] 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.
- [138] Haslum, K., Abraham, A. & Knapskog, S. (2008), "Hinfra: Hierarchical neuro-fuzzy learning for online risk assessment", in *Second Asia International Conference on Modeling & Simulation-AICMS 08* (pp. 631-636). *IEEE*
- [139] Boehm, B. W. (1991), Software risk management: principles and practices, *Software*, *IEEE*, 8(1), 32-41.
- [140] Hu, Y., Zhang, X., Sun, X., Zhang, J., Du, J. & Zhao, J. (2010), "A unified intelligent model for software project risk analysis and planning", in *International Conference on Information Management, Innovation Management and Industrial Engineering (ICIII)*, (Vol. 4, pp. 110-113). *IEEE*.
- [141] Bragina, T. & Tabunshchik, 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.*
- [142] Pooja Rani & Dalwinder Singh Salaria (2013), “Neuro-Fuzzy based Software Risk Estimation Tool”, in *Global Journal of Computer Science and Technology Software & Data Engineering, Vol. 13 Issue 6, Global Journals Inc. (USA), ISSN: 0975-4172, 0975-4350(p.No.1-7)*
- [143] Aggarwal, K.K., Singh, Y., Kaur A, Malhotra R. (2009), “Empirical analysis for investigating the effect of object-oriented metrics on fault proneness: A replicated case study”. *Software Process: Improvement and Practice* 2009; (14): pp. 39-62, DOI=<http://onlinelibrary.wiley.com/doi/10.1002/spip.389>.
- [144] Han, J., Kamber, M. (2001), *Data Mining: Concepts and Method*. Harchort India Private Limited.
- [145] Ho, S, Xie, M, Goh, T.N. (2003), “A study of the connectionist models for Software Reliability prediction”, *Computers and Mathematics with Applications 2003*; (46): pp. 1037-1045.
- [146] Hosmer, D., Lemeshow, S. (1989), *Applied Logistic regression*, John Wiley and Sons.
- [147] Jun, Z. (2007), Predicting Software Reliability with neural network ensembles. *Expert Systems with Applications*; 36(2): pp. 216-222. DOI=<http://10.1016/j.eswa.2007.12.029>.
- [148] Kai, Y.C., Lin, C., Wei, D.W., Zhou, Y.Y., and David, Z. (2001), on the neural network approach in Software Reliability modeling, *Journal of Systems and Software 2001*; (58): pp. 47-62.
- [149] Karunanithi, N., Whitley, D., and Malaiya, Y.K. (1992), “Prediction of Software Reliability using connectionist models”, *IEEE Transactions on Software Engineering*; 18(7): pp. 563-574.
- [150] Lyu, M.R. (1999), “*Handbook of Software Reliability Engineering*”, McGraw Hill, India; 131-151.

- [151] MATLAB TOOLBOX (2014), <http://www.mathworks.com> “MatLab Toolbox for ANN, FIS, ANFIS”.
- [152] KirtiTyagi, 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,
- [153] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986a), “Learning internal representations by error propagation”, in D. E. Rumelhart, & J. L. McClelland (Eds.), “Parallel distributed processing: Explorations in the microstructure of cognition (pp. 318–362)”, *Cambridge, MA: MIT Press*, 318–362.
- [154] Yates, F (1934). "Contingency table involving small numbers and the χ^2 test". *Supplement to the Journal of the Royal Statistical Society* 1(2): 217–235.
- [155] Vinu V Das, Janahanlal Stephen, YogeshChaba (2011), Computer Networks and Information Technologies: Second International Conference on Advances in Communication, Network, and Computing, CNC 2011, Bangalore, India, March 10-11, Proceedings, Springer, 15-Mar-2011.
- [156] BonthuKotaiah, 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.
- [157] BonthuKotaiah, D.S.R. Krishna, T.Arundathi (2014), “Assessment of Software Reliability via Mathematical Modelling”, *Proceedings of the Second International Conference on “Emerging Research in Computing, Information, Communication and Applications”(ERCICA-2014)*, ISBN: 9789351072638, Elsevier, Pp. No.864-867.
- [158] BonthuKotaiah, T.Arundathi, M.D. Fasihuddin, P.M.Jahan(2014), “Machine Learning Techniques for the effective Assessment of Software Reliability”, *Proceedings of the Second International Conference on*

“Emerging Research in Computing, Information, Communication and Applications”(ERCICA-2014), ISBN: 9789351072638, Elsevier, Pp. No.868-872.

- [159] BonthuKotaiah, R.A. Khan (2015), “ Automation of Assessment and Approximation of Software Reliability”, *International Journal of Research (IJR)* e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, Pp. No. 425-468.

List of Publications

INTERNATIONAL JOURNAL

2012

BonthuKotaiah and R.A. Khan (2012), "A Survey on Software Reliability Assessment by using Different Machine Learning Techniques", *International Journal of Scientific & Engineering Research(IJSER)*, Volume 3, Issue 6, P. No.1-7.

Dr.O. Nagaraju, B.Kotaiah , R.A. Khan, M.Ramireddy, N.S.KalyanChakravarthy," Implementing and Compiling Clustering using Mac Queens Alias K-Means Apriori Algorithm", *International Journal of Database Management Systems(IJDMS)* Vol.4, No.2, April 2012, Pp. No.69-82.

2013

BonthuKotaiah, Raees Ahmed Khan, MuralidharVejendla (2013),"Model For Software Errors Prediction using Machine Learning to Improve the Software Reliability", *International Journal of Soft Computing and Software Engineering [JSCSE]*, Vol. 3, No. 3, pp. 314-319, eISSN: **2251-7545**, Doi: 10.7321/jscse.v3.n3.47

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

BonthuKotaiah, R. A. Khan (2013), "Software Reliability Growth Models- A Revisit", *IJAC(International Journal Of Advanced Computing (Print Journal)*, ISSN. 0975-7686, P.No.32-36, Online Link: http://Journals.Griet.ac.in/Images/V5i1jan2013_7.Jpg

BonthuKotaiah, 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.

2014

B Kotaiah, T.Arundathi, P.M. Jahan, “Comparative Analysis of Machine Learning Techniques for Software Reliability Prediction”, *Mathematical Sciences International Research Journal*: Vol. 3 Spl. Issue(2014), ISSN 2349-1353, PP No.977-992.

2015

BonthuKotaiah, R.A. Khan (2015), “An Analysis of Software Reliability Assessment with Neuro Fuzzy based Expert Systems ”, *Elsevier B.V. ., Procedia Computer Science*, ISSN No.1877-0509, Proceedings of International conference on Soft Computing and Software Engineering(SCSE), University of California, USA (Paper ID-40819).

BonthuKotaiah, 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.

BonthuKotaiah, R.A. Khan (2015), “Automation of Assessment and Approximation of Software Reliability”, *International Journal of Research (IJR)* e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, Pp. No. 425-468.

INTERNATIONAL CONFERENCE

2012

BonthuKotaiah, R.A. Khan, The International Conference on Mathematics, Statistics and Computer Engineering (ICMSCE), September 13-14,2012(Print Journal).

BonthuKotaiah, B. Sunil Kumar, R.A. Khan, “A Survey on Software Reliability Assessment by using different Machine Learning Techniques, International Conference on Information Technology, Electronics and Communications(ICITEC-2012), Hyderabad, India, July 14-15, 2012. PP No.11-17.

2013

BonthuKotaiah, R.A. Khan (2013), “Assessing the level of Software Reliability using Neuro Fuzzy System”, *International Conference on Information & Engineering Sciences (ICIES -2013)*(Print Journal), Vijayawada.

BonthuKotaiah, R.A. Khan(2013), “GMR(Group Maturity Rating) with Fuzzy Logic Based Software Defects Rating for the Assessment of Software Development Life Cycle(SDLC) Performance”,*Advances in Engineering and Technology Convergence(AETC)*, Bangkok, Thailand, ISBN No: 978-93-82208-89-1, P.No.25-29.

BonthuKotaiah, R.A. Khan (2013), “MLP Neural Networks Based Approach for the Assessment of Software performance to Improve the Software Reliability”, *The International Conference on Mathematics, Statistics and Computer Engineering (ICMSCE)*, ISSN No.2568-4345, (Print Journal).

BonthuKotaiah, R.A. Khan, “The Preliminary Prediction of Software Reliability Prediction using Neural Network Based Systems”, International Conference on Mathematical Modelling And Numerical Simulation(ICMMANS)-BBAU. Lucknow, July 1-3,2013, Conference Proceedings.

BonthuKotaiah, 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*.

BonthuKotaiah, R.A. Khan, Muralidhar V., “ Model for Software Errors Prediction using Machine Learning to Improve the Software Reliability”, The Proceedings of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13], San Francisco State University, CA, U.S.A., March 2013 DOI: 10.7321/jscse.v3.n3.47.

BonthuKotaiah, R.A. Khan, “The preliminary prediction of software reliability prediction using neural network based systems”, International Conference on

Mathematical Modeling and Numerical Simulation(ICMMANS), BBAU, Lucknow, July 1-3,2013,

2014

BonthuKotaiah, T. Arundhathi, M. D. Fasihuddin and PathanMehra Jahan, “Machine Learning Techniques for the Effective Assessment of Software Reliability” , Proceedings of the Second International Conference on “Emerging Research in Computing, Information, Communication and Applications(ERCICA-2014), Elsevier, ISBN: 9789351072638, PP No. 868-872.

BonthuKotaiah, T. Arundhathi, M. D. Fasihuddin and PathanMehra Jahan, “Assessment of Software Reliability via Mathematical Modelling” , Proceedings of the Second International Conference on “Emerging Research in Computing, Information, Communication and Applications(ERCICA-2014), Elsevier, ISBN: 9789351072638, PP No.864-867.

BonthuKotaiah, 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, Souvenir, BBAU, Lucknow, Conference Proceedings.

2015

BonthuKotaiah, R.A. Khan (2015),“An Analysis of Software Reliability Assessment with Neuro Fuzzy based Expert Systems ”, *Elsevier B.V. , Procedia Computer Science*, ISSN No.1877-0509, Proceedings of International conference on Soft Computing and Software Engineering(SCSE), University of California, USA (Paper ID-40819).

NATIONAL CONFERENCE

BonthuKotaiah , R.A. Khan, “ A Survey on Software Reliability Assessment for Machine Learning Techniques”, National Conference on Recent Trends in

Information Technology(NCRTIT), 9th Match, 2012, PVP Siddartha Institute of Technology, Vijayawada.

BonthuKotaiah , R.A. Khan, “Software Reliability Assessment by using Neural Networks with Fuzzy Logic Based Systems”, Advances in Engineering and Technology Conference(AET), Delhi, NCR, August,2013, Elsevier.

BonthuKotaiah , R.A. Khan, “Issue in the Assessment of Software Reliability with Mathematical Modelling Based Safety-Critical System”, [National Conference on Information Security Challenges \(NCISC- 2014\)](#), 28thMarch 2014, Organized by Department of Information Technology, BabasahebBhimraoAmbedkar University(BBAU), Lucknow.

BonthuKotaiah , R.A. Khan(2015), “Software Reliability Measurement with Neural Network based Soft Computing Systems”, *“Third International Conference on Emerging Research in Computing, Information, Communication And Applications” (ERCICA-2015), Springer, Conference Proceedings.*

CONFERENCES ATTENDED/PAPERS PRESENTED

Attended and Presented a paper in the International Conference on Mathematics, Statistics And Computer Engineering (ICMSCE), September 13-14,2012(Print Journal).

Attended and Presented a paper on “Assessing the Level of Software Reliability using Neuro Fuzzy System”, International Conference on Information & Engineering Sciences - 2013 (ICIES -2013) (Print Journal), 21st – 23rd Feb. 2013,Vijayawada.

Attended and Presented a paper on “GMR(Group Maturity Rating) with Fuzzy Logic based Software Defects Rating for the Assessment of Software Development Life Cycle(SDLC) Performance”, Advances In Engineering and Technology Convergence(AETC), 28th April, 2013, Bangkok, Thailand, ISBN No: 978-93-82208-89-1, Pp.No.25-29.

Attended and Presented a paper on “MLP Neural Networks based Approach for the Assessment of Software Performance to improve the Software Reliability”, The International Conference on Mathematics, Statistics And Computer Engineering (ICMSCE), ISSN No.2568-4345, August 9-10,2013, Print Journal.

Attended and Presented a paper on “The Preliminary Prediction of Software Reliability Prediction using Neural Network Based Systems”, International Conference on Mathematical Modelling and Numerical Simulation(ICMMANS)-BBAU, Lucknow, July 1-3,2013, Conference Proceedings.

BonthuKotaiah, 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.*

BonthuKotaiah, R.A. Khan (2015),“An Analysis of Software Reliability Assessment with Neuro Fuzzy based Expert Systems ”, *Elsevier B.V. ., Procedia Computer Science*, ISSN No.1877-0509, Proceedings of International conference on Soft Computing and Software Engineering(SCSE), University of California, USA (Paper ID-40819).

Attended and Presented a paper on “ Software Reliability Assessment by using Neural Networks with Fuzzy Logic based systems”, International Conference on Advances in Computer Science(AET-ACS), Elsevier, December,2013.online link: http://searchdl.org/public/book_series/elsevierst/6/540.pdf.

Attended and Presented a paper on “ Model for Software Errors Prediction Using Machine Learning to Improve The Software Reliability”, The Proceedings of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13], San Francisco State University, CA, U.S.A., March 2013 Doi: 10.7321/jscse.v3.n3.47.

Attended and Presented a paper on “The preliminary prediction of software reliability prediction using neural network based systems”, International

Conference on Mathematical Modeling and Numerical Simulation(ICMMANS), BBAU, Lucknow, July 1-3,2013,

Attended and Presented a paper on “Machine Learning Techniques for the Effective Assessment of Software Reliability” , Proceedings of the Second International Conference on “Emerging Research in Computing, Information, Communication and Applications(ERCICA-2014), Elsevier, ISBN: 9789351072638, PP No. 868-872.

Attended and Presented a paper on “Assessment of Software Reliability via Mathematical Modelling” , Proceedings of the Second International Conference on “Emerging Research in Computing, Information, Communication and Applications(ERCICA-2014), Elsevier, ISBN: 9789351072638, PP No.864-867.

Attended and Presented a paper on “A Survey on Software Reliability Assessment by using different Machine Learning Techniques, International Conference on Information Technology, Electronics and Communications(ICITEC-2012), Hyderabad, India, July 14-15, 2012. PP No.11-17.

Attended and Presented a paper on “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, Souvenir 10-11 July,2014, BBAU, Lucknow.

BonthuKotaiah, R.A. Khan (2013), “The Preliminary 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.*