Software Reliability prediction using Ensemble Model

Pravas Ranjan Bal

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Software Reliability prediction using Ensemble Model

Dissertation submitted in partial fulfillment

of the requirements of the degree of

Master of Technology

in

Computer Science and Engineering

by

Pravas Ranjan Bal

(Roll Number: 214CS3484)

based on research carried out

under the supervision of

Prof. Durga Prasad Mohapatra

May, 2016

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Prof. Durga Prasad Mohapatra
Associate Professor

May 22, 2016

Supervisor's Certificate

This is to certify that the work presented in the dissertation entitled *Software Reliability prediction using Ensemble Model* submitted by *Pravas Ranjan Bal*, Roll Number 214CS3484, is a record of original research carried out by him under my supervision and guidance in partial fulfillment of the requirements of the degree of *Master of Technology* in *Computer Science and Engineering*. Neither this dissertation nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

Durga Prasad Mohapatra

Dedication

I would like to dedicate this thesis to my parents, peusa and Buli dei for their love, encourage, patience, and understanding.

Signature

Declaration of Originality

I, *Pravas Ranjan Bal*, Roll Number *214CS3484* hereby declare that this dissertation entitled *Software Reliability prediction using Ensemble Model* presents my original work carried out as a Postgraduate student of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the sections "Reference" or "Bibliography". I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

May 22, 2016
NIT Rourkela

Pravas Ranjan Bal

Acknowledgment

I am grateful to numerous local and global peers who have contributed towards shaping this thesis. At the outset, I would like to express my sincere thanks to Prof. Durga Prasad Mohapatra for his advice during my thesis work. As my supervisor, he has constantly encouraged me to remain focused on achieving my goal. His observations and comments helped me to establish the overall direction of the research and to move forward with investigation in depth. He has been a fatherly figure and source of knowledge through out my research work.

I am really thankful to Deepak Ranjan Nayak, Phd scholar NIT Rourkela who has helped guided me through out my M.Tech research. I am also thankful to Sangaratna Godboley, Phd scholar NIT Rourkela who guided me properly in my research work. My sincere thanks and gratitude to everyone who has provided me new ideas, useful criticism, kind words and their valuable time. I must acknowledge the academic resources that I have received from NIT Rourkela. I would like to thank administrative and technical staff members of the Department who have been kind enough to advise and help in their respective roles.

Last, but not the least, I would like to dedicate this thesis to my parents and peusa and Buli dei for their love, patience, and understanding.

May 22, 2016 *Pravas Ranjan Bal*
NIT Rourkela Roll Number: 214cs3484

Abstract

Software Reliability is the key factor of software quality estimation and prediction during testing period. We have implemented three models such as Radial Basis Function Neural Network (RBFNN) model, Ensemble model based on two types Feed Forward Neural Networks and one Radial Basis Function Neural Network and Radial basis function Neural Network Ensembles (RNNE) model for Software reliability prediction over five benchmark datasets. We have used Bayesian regularization method on all three models to avoid over-fitting problem and generalization of the neural network. We have been used two types of meaningful performance measures such as Relative Error (RE) and Average Errors (AE) for software reliability prediction. The results of all three proposed models have been compared with some traditional models such as Duane model and Artificial neural networks like Feed Forward Neural Network (FFNN) model. The experimental result shows that the nonparametric growth model called Ensemble model (multiple predictors) shows best minimal error than parametric model. Finally, It has been observed that the multiple predictors like Ensemble model always shows the best performance than single predictor like artificial neural network and some other traditional neural network.

Keywords: ***Feed Forward Neural Network***; ***Ensemble Model***; ***Radial Basis Function***; ***Software reliability***; ***Statical Model.***

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Software reliability is the probability of software failure free operation for a specified period of time in a particular specified environment (as defined by ANSI) [1]. Software reliability is an key factor of qualitatively and quantitatively characterizing of software products and estimate the duration of software period. Now a days, the growth of software products increase rapidly so the prediction of software reliability plays a major role in development of software process. We always need every program should be reliable. But failure does not occur if we do not use software. But in case of hardware reliability, some hardware parts occur to failure even if the system is not use. So we need reliability of software in all circumstances.

The main objective of the software reliability prediction is to test performance of the every software under given conditions without any type of corrective measure using known fixed procedures considering its all specifications. The metric of software reliability prediction is Mean Time Between Failure (MTBF) and this measurement depends on different factors of the software such as operational time, calender time and execution time etc.

Software reliability prediction is a special field of software testing. The main application of software reliability prediction not only used in software industry but also used in various fields like solar radiation prediction, military applications, temperature prediction, stock market exchange prediction, whether forecasting of all countries efficiently and accurately.

Most of the software reliability growth models have been proposed in this related work. Mainly, there are two types of growth models for prediction of software reliability such as parametric growth model and non-parametric growth model. The parametric growth models based on the linear software reliability growth models such as some linear equations like Duane model, S-Shaped model, G-O model etc. This parametric growth model especially based on the stochastic nature of the software failure and some development environments. The parametric growth model popularly known as linear Software Reliability Growth Model (SRGM) has been successfully used in practical software reliability environment[2, 3]. However, it has been shown that no single parametric growth model like linear Software

Reliability Growth Model give better performance in all circumstances. So we think about another software reliability prediction models, that is based on the artificial neural network called as nonparametric growth model. The most popular nonparametric growth models [4–6] like Feed Forward Neural Network (FFNN), Jordan Recurrent Neural Network(JRNN) etc. and different types of linear ensemble techniques [7]and nonlinear ensemble techniques. Finally, it is concluded that all nonparametric growth model shows better predictive performance than all parametric growth models in all circumstances [8, 9].

In this work we have designed three models such as RBFNN model, Ensemble model based on ffnn network and RBFN network and RBF Neural Network Ensemble (RNNE) model for software reliability prediction efficiently [7, 10, 11]. Finally, we concluded that the ensemble model called multiple predictor predicts better performance than a single predictor like FFNN network, RBFNN network etc. and other Software Reliability Growth Models (SRGM) like Duane model, Geol Okumoto (G-O) model etc.

1.1 Problem Definition

Reliability of a software is one of the most important factor of estimation of the software of many industries and prediction during the testing time of the software of many industries also. Prediction of software or estimation means that we can always predict some future failure data or performance by using some past failure software reliability datasets of many industries of software.

The important advantage of prediction hardware or software reliability is to predict the software failure data and minimize the error rate of the proposed model and its competent system. The major application of software reliability prediction model is most widely used in military and commercial purpose of many countries such as United States, China, France, and United Kingdom etc. The major benefits of software reliability prediction include better performance, higher reliability, extendibility and efficient solution of application problems.

we plan to develop software reliability model by using soft computing. We also plan to develop the ensemble prediction model for software reliability prediction, weather forecasting, radiation of solar energy prediction and estimation, annual and seasonal temperature prediction of different countries.

1.2 Motivation and Objectives

Some of the very old mathematical growth models such as mathematical models have been used in practical laboratories successfully. But, all mathematical models do not give the good results for all types cases. So we think about another model. The mathematical models, non-mathematical models like some of the neural network have shown to be very effective and alternative techniques. The disadvantage of the parametric model is that parametric model can not perform better in all circumstances. So we have developed a robust predictor Ensemble techniques of RBFN Network (RNNE) for forecasting of reliability of a software.

The main important objective of our proposed research work is to design or develop some of the efficient and efficient software reliability prediction models. To address this important objective of reliability, we identify some of the important following goals :

1. To propose some suitable models for estimating software reliability by using artificial neural networks models such as Feed Forward Neural Network, Recurrent Neural Network, Radial Basis Function Neural Network, Generalized Neural Network etc.
2. To propose some suitable models for predicting software reliability by using artificial neural networks models such as FFNN Network, Recurrent Neural Network, RBFN Network, Generalized Neural Network etc.
3. To implement the proposed model in some industrial dataset, real life datasets like seasonal and annual temperature, Radiation of solar energy, stock price data etc.
4. To evaluate the best performance and effective results of the proposed model.
5. To compare the results of the proposed designed model with other traditional models.

1.3 Organization of the Thesis

Chapter 1 briefly describes the introduction, motivation and objectives of software reliability prediction. Chapter 2 describes basis concepts about software reliability prediction. Chapter 3 describes the details about the related works of software reliability prediction. Chapter 4 describes software reliability prediction based on Radial Basis Function Neural Network model. Chapter 5 describes the software reliability prediction based on ensemble model. Chapter 6 describes software reliability prediction based on Radial basis function Neural Network Ensembles and Chapter 7 concludes the thesis.

Chapter 2

Basic Concepts

Radial basis function is an exponential function used for forecasting of software reliability. Ensemble model is the combination of multiple models. The basic difference between MLP and RBF is, in MLP we use multiple layers as hidden layers and in RBF we use one layer as hidden layer that is called RBF layer.

2.1 Datasets Used

Five types of datasets DS1, DS2, DS3, DS4 AND DS5 [1] have been used for forecasting of reliability of a software in our experimental status.

The dataset DS1 is arranged from real-time application with 21,700 numbers of instructions and 136 number of failures. The dataset DS2 is arranged from dynamic flight application with 10,000 instructions and 118 number of failures. The dataset DS3 is arranged from dynamic flight application with 22,500 instructions and 180 number of failures. The dataset DS4 is arranged from dynamic flight application with 38,500 assembly instructions and 213 number of failures. The dataset DS5 is arranged from single user workstation with 397 failures. All of these datasets having two types of columns such as executional or calender time and number of failures.

2.2 Performance Measurement

Two types of meaningful performance measurement have been used for compare of software reliability prediction. For performance measurement, we have used two types of errors such as Relative Error (RE) and Average Error (AE). They are defined as follows.

$$RE = \frac{\hat{y}_i - y_i}{y_i} * 100 \tag{2.1}$$

$$AE = \frac{1}{n}\sum_{i=1}^{n} RE_i \tag{2.2}$$

where n is called as total number of samples of data, $\hat{y}_i$ is called as predicted value and y_i is called as actual value.

Chapter 3

Literature Survey

This chapter briefly discusses about different types artificial neural network and linear ensemble of artificial neural network for software reliability forecasting and estimation.

Karunanithi et al.[8, 12] first introduce the ffnn network for achieving of software reliability prediction. They had tested this feed forward neural network model on different benchmark datasets of software projects. Finally they observed that this network model behaves as a constant prediction performance than parametric models on all datasets. They also used connectionist model [13, 14] for software reliability prediction. Finally he observed that the usefulness of connectionist model is capable of developing different prediction models of varying complexity.

Renate Sitte [15] has introduced two methods such as Neural Networks and recalibration of its parametric models for measurement of prediction performance. They observed that both methods are predict better than parametric models. They concluded that not only neural networks are much simpler to use but also they give a better prediction performance than other parametric models.

Cai et al.[16] designed a new type of neural network that is called effectiveness of neural network based on different types of reliability datasets. They got several things such as firstly, the neural network approach is the best predictor for all types of datasets secondly, the training results are better predictor than general neural network predictor thirdly, the neural network approach can predict qualitatively what it learned fourthly, due to some essential problem this neural network approach fail to give the quantitatively results.

Ho et al.[17] proposed a Modified Elman Neural Network (MENN) for forecasting of reliability of a software based on different software reliability datasets. They studied different types of feedback weights on the proposed model. They also compared the proposed designed model with simple ffnn network, Jordan neural network and some other mathematical growth models.

Tian and Noore [18, 19] proposed an evolutionarily network for forecasting of reliability of a software based on different types of software reliability datasets. They also applied the genetic algorithm on their proposed model for better prediction results. Finally, they concluded that whenever the new data arrives the model is dynamically reconfigured.

Su and Huang [9] proposed neural network approach such as Dynamically Weighted Combinational Model (DWCM) for reliability prediction and software estimation. Firstly they explain the neural network in mathematical point of view reliability modeling. this model is tested with different types software reliability datasets. This results concluded that the approach prediction model gives better prediction results than other existing model.

Raj Kiran et al.[11] introduced ensemble models for forecasting software reliability. They had designed the ensemble model based on different types of statistical and intelligent techniques. They had designed three linear ensemble models and one nonlinear ensemble models and tested. Finally they observed that nonlinear ensemble model outperformed than all linear ensemble models.

Jun Zheng [7] introduced Neural Network Ensembles of prediction (PNNE) for forecasting of reliability of a software. They compared his proposed model with a single pi sigma neural network and three other parametric models. Finally they observed that the nonparametric models outperformed than parametric models in all datasets.

Chapter 4

Software Reliability Prediction Based On Radial Basis Function Neural Network

Software Reliability is the important thing of quality estimation of the software and forecasting during the testing time of software or hardware. This chapter introduces some of the nonparametric models of the software that uses exponential function of the neural network for forecasting of software reliability. The RBFNN model has been tested over five datasets of software industries . The performance result of the designed model has been compared with some of the very old models and it has been concluded that the proposed designed model shows very efficient performance of the designed competent systems. From experimental result of the proposed and some of the traditional models we observed that the radial basis function neural network gives better performance than some of the very old mathematical software reliability growth model (SRGM).

4.1 Proposed Work

This section introduces the work methodologies of the proposed work such as prediction for software reliability of RBFNN model. Normally, this flowchart follows 4 steps. The overall flowchart diagram of the proposed work is shown in Fig.4.1.

1. Step-1: Collect DataSets
   We have used five types of datasets for forecasting of a reliability software. All these datasets DS1, DS2, DS3, DS4 and DS5 are collected from real-time application, dynamic flight application, dynamic flight application, dynamic flight application and single user workstation respectively.

2. Step-2: Normalize Datasets
   All datasets have been normalized between 0 and 1 by using minmax formula and is given by the following Eq.4.1.

$$Normalized(t_i) = (t_i - min_A)\frac{max - min}{max_A - min_A} + min \tag{4.1}$$

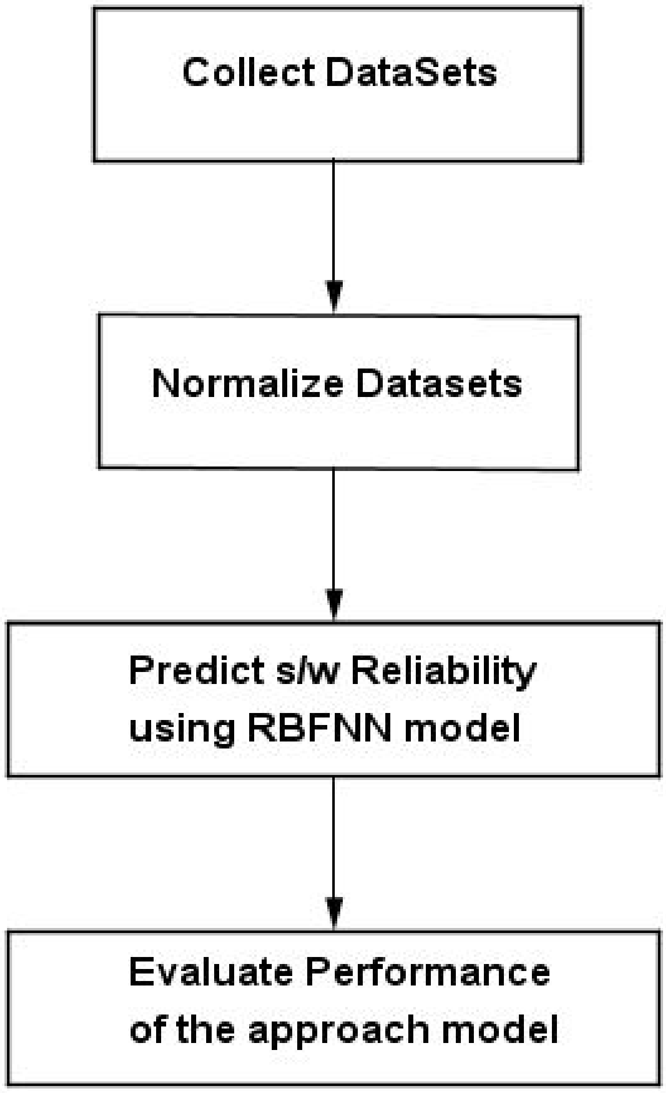


Figure 4.1: Flowchart diagram of RBFNN.

where, max_A is the maximum value of matrix A, min_A is the minimum value of matrix A and t_i is the execution time.

3. Step-3: Predict software reliability using RBFNN model
   The proposed RBFNN model is used for forecasting of software reliability.

4. Step-4: Evaluate Performance of approach model: We have used two types of meaningful Errors such as Relative Error (RE) and Average Error (AE) for performance evaluation of the proposed model and its competent existing model.

4.1.1 Software Reliability data

The reliability of software data has been arranged in two columns such as $\{t_i, N_i\}$, where t_i is the execution time and N_i is the cumulative number of failures. The dataset DS1 has been

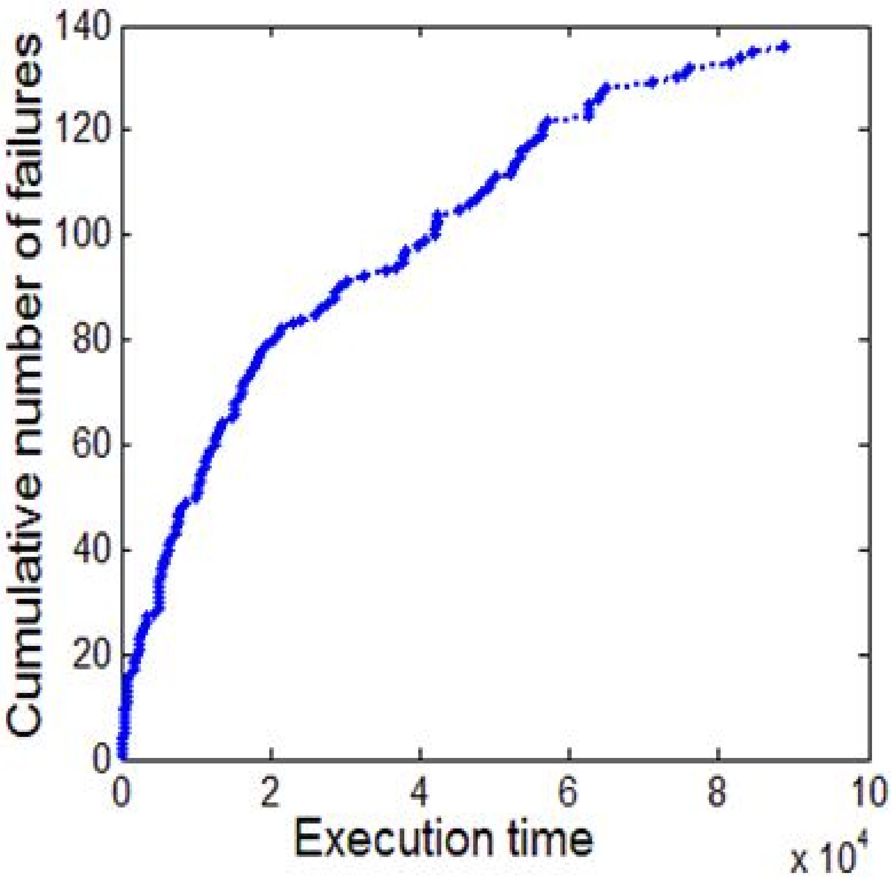


Figure 4.2: Software Reliability data Example for DS1.

plotted in Fig.4.2.

4.1.2 RBFNN model

The proposed model called as RBFNN model is depicted in Fig.4.3. The proposed designed model is of three layers. The hidden layers has been designed by exponential function . This proposed model has been designed by the help of single-input and single-output model of the system. The exponential function has been used in the hidden layer of RBFNN model instead of using sigmoid function. Hidden layer of the RBFNN model consists of k neurons. The execution or calender time of dataset t_i is the input of the designed proposed model and second attribute is used N_i for forecasting output of the proposed model.

4.1.3 Radial Basis Function

The transfer function has been chosen based on RBF function such as exponential function as follows. The function is designed as $f(x) \in R_n \to R$ follows

$$f(x) = \sum_{i=1}^{k} w_i \phi(x - c_i) \tag{4.2}$$

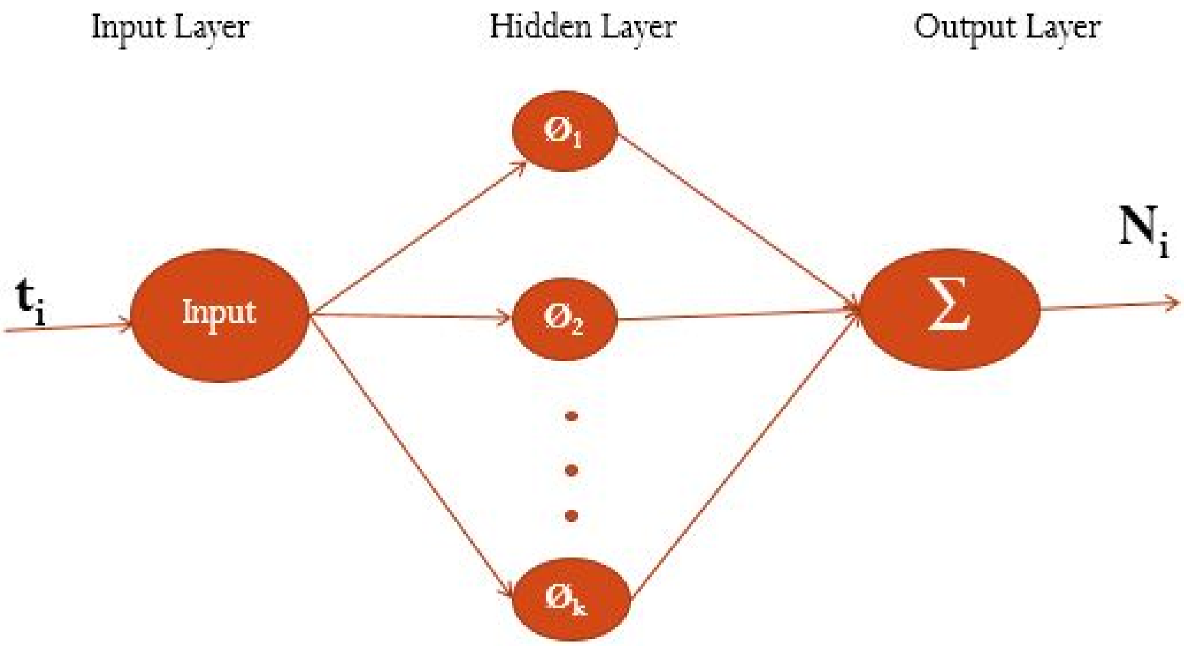


Figure 4.3: RBFNN model.

where, k is called as the number of neurons, x is called as the input numbers, w_i is called as the weight vectors of neuron i and c_i is called as the centroid vector for neuron i. The function $\phi(x - c_i)$ is calculated by the formula given as follows

$$\phi(x - c_i) = \sqrt{\sum_{i=1}^{k}(x - c_i)^2} \tag{4.3}$$

We have been applied the Bayesian regularization method [20] in proposed RBFNN model generalize the the network of the proposed model and is defined by the formula as follows

$$Error = \alpha MSE + (1 - \alpha)MSW \tag{4.4}$$

where, MSE is called as Mean Square Error, MSW is called as Mean Square Weights and α is called as the ratio of performance.

4.1.4 Performance Measures

For efficient measurement of performance of reliability of software, two error functions are used for proposed model. Here, We have trained the proposed model by some part of data. The remaining data of the existing dataset is used for testing. The Relative Error (RE) and Average Error(AE) has been written as follows

$$RE = \frac{\hat{y}_i - y_i}{y_i} * 100 \tag{4.5}$$

where, n is the number of data, $\hat{y}_i$ is the predicted value and y_i is the actual value.

4.2 Experimental Results and Comparison

We have used five types of datasets [1] for software reliability prediction of RBFNN model. All datasets have been normalized between $[0, 1]$ by minmax formula.

In our project works, this proposed designed model has been trained with 50 percent for dataset DS1, 66 percent for dataset DS2, 50 for dataset DS3, 55 percent for dataset DS4 and 65 percent for dataset DS5. we have used the remaining data of each dataset for testing purpose.

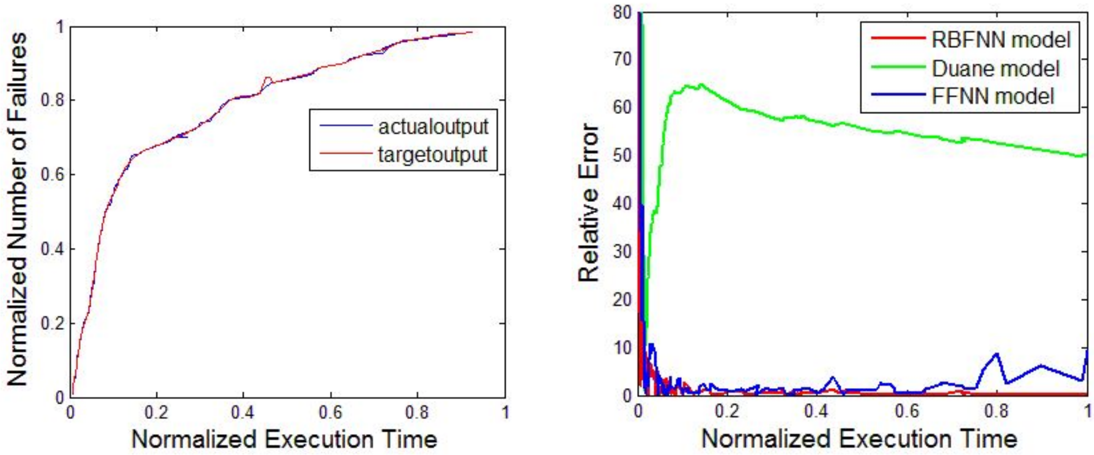
The proposed designed model is compared with FFNN network and mathematical model [21, 22] called as Duane model. The growth Model is given by the following Eq.4.2.

$$\mu(t) = at^b, a > 0, b > 0 \tag{4.6}$$

where a is called as parameter size and b is called as shape of growth curve for Duane model.

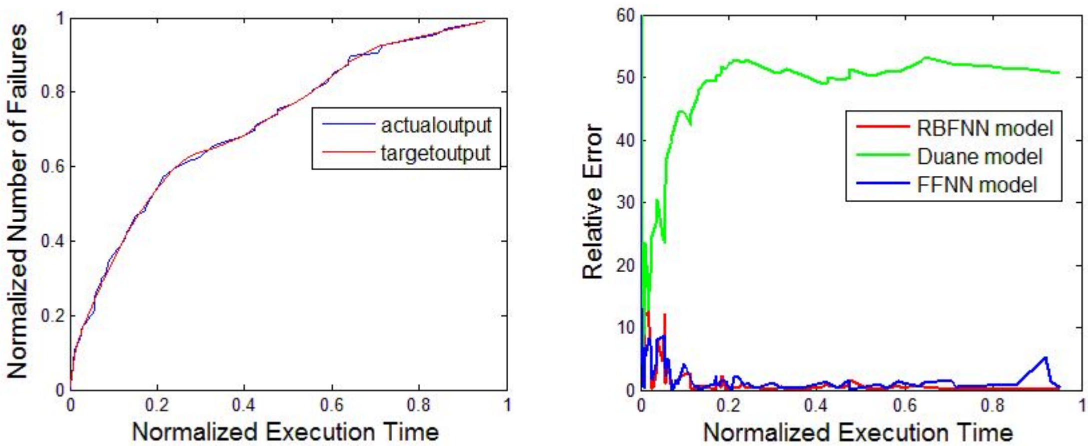
4.2.1 Performance Measures

For RBFNN model, we have selected 14 neurons and for FFNN model we have selected 10 neurons. The results of the prediction of RBFNN model and relative errors of different model on five datasets DS1, DS2, DS3, DS4 and DS5 are depicted in Fig.4.4, Fig. 4.5, Fig. 4.6, Fig.4.7 and Fig. 4.8 respectively. We concluded that the proposed scheme model shows very lower prediction performance than other types of the model.



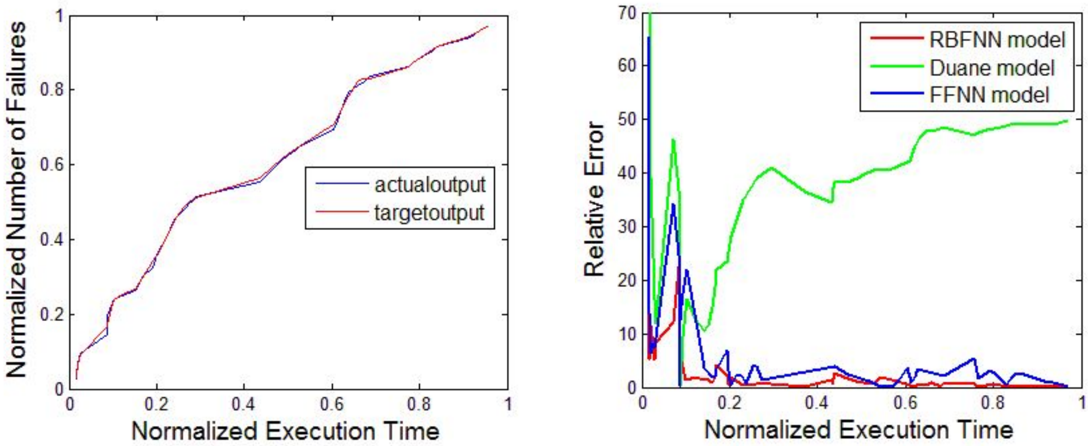
(a) Prediction Performance (b) Relative Errors

Figure 4.4: Prediction Results of RBFNN model and Relative errors of different models of DS1 in (a) and (b)



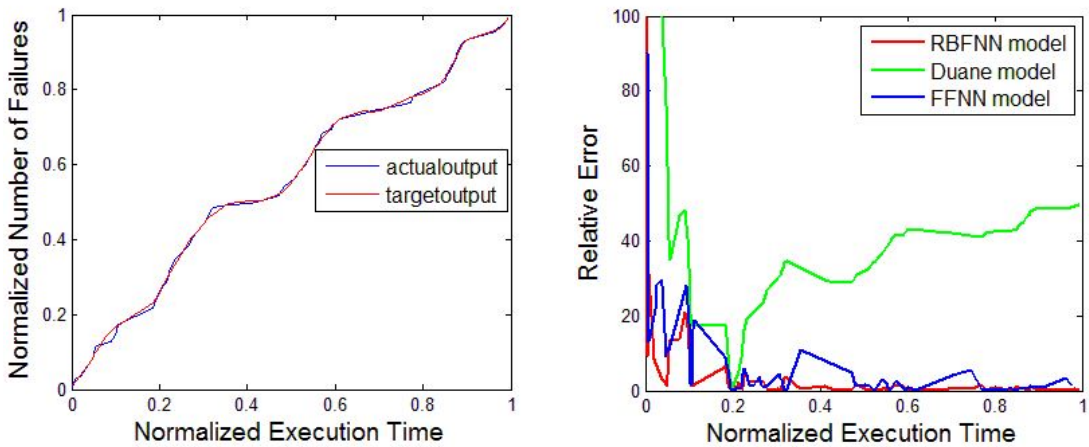
(a) Prediction Performance (b) Relative Errors

Figure 4.5: Prediction Results of RBFNN model and Relative errors of different models of DS2 in (a) and (b)



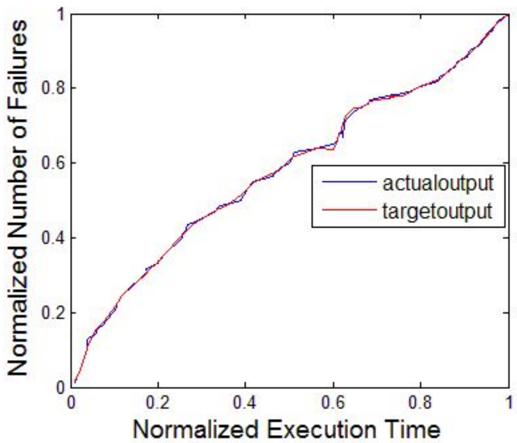
(a) Prediction Performance (b) Relative Errors

Figure 4.6: Prediction Results of RBFNN model and Relative errors of different models of DS3 in (a) and (b)

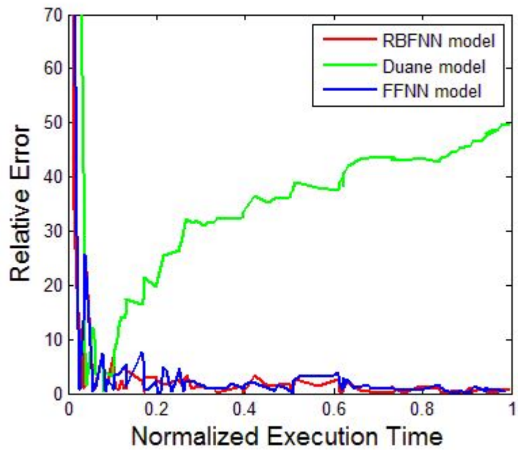


(a) Prediction Performance (b) Relative Errors

Figure 4.7: Prediction Results of RBFNN model and Relative errors of different models of DS4 in (a) and (b)



(a) Prediction Performance



(b) Relative Errors

Figure 4.8: Prediction Results of RBFNN model and Relative errors of different models of DS5 in (a) and (b)

Chapter 5

Software Reliability Prediction based on Ensemble Models

This chapter proposes a nonparametric ensemble technique model for forecasting software reliability of some software industries. The proposed architectural design has been tested over four standard datasets. The proposed model is the best efficient model because it shows very less error rates of its traditional competent model. The experimental results proved that the proposed ensemble model is the efficient model and shows very high quality performance than other models.

5.1 Proposed Work

The overall Flowchart diagram of the proposed model is depicted in Fig.5.1.

1. Step-1: Collect DataSets
   We have used four types of datasets for software reliability prediction. All these datasets DS1, DS2, DS3 and DS4 are arranged from real-time application, dynamic flight application, dynamic flight application and single user workstation respectively.

2. Step-2: Normalize Datasets
   All datasets have been normalized between 0 and 1 by using minmax formula and is given by the following Eq.5.1.

$$Normalized(t_i) = (t_i - min_A)\frac{max - min}{max_A - min_A} + min \tag{5.1}$$

   where, max_A is the maximum value of matrix A, min_A is the minimum value of matrix A and t_i is the execution time.

3. Step-3: Predict software reliability using Ensemble model
   The proposed Ensemble model based on FFNN Network is used for prediction of software reliability.

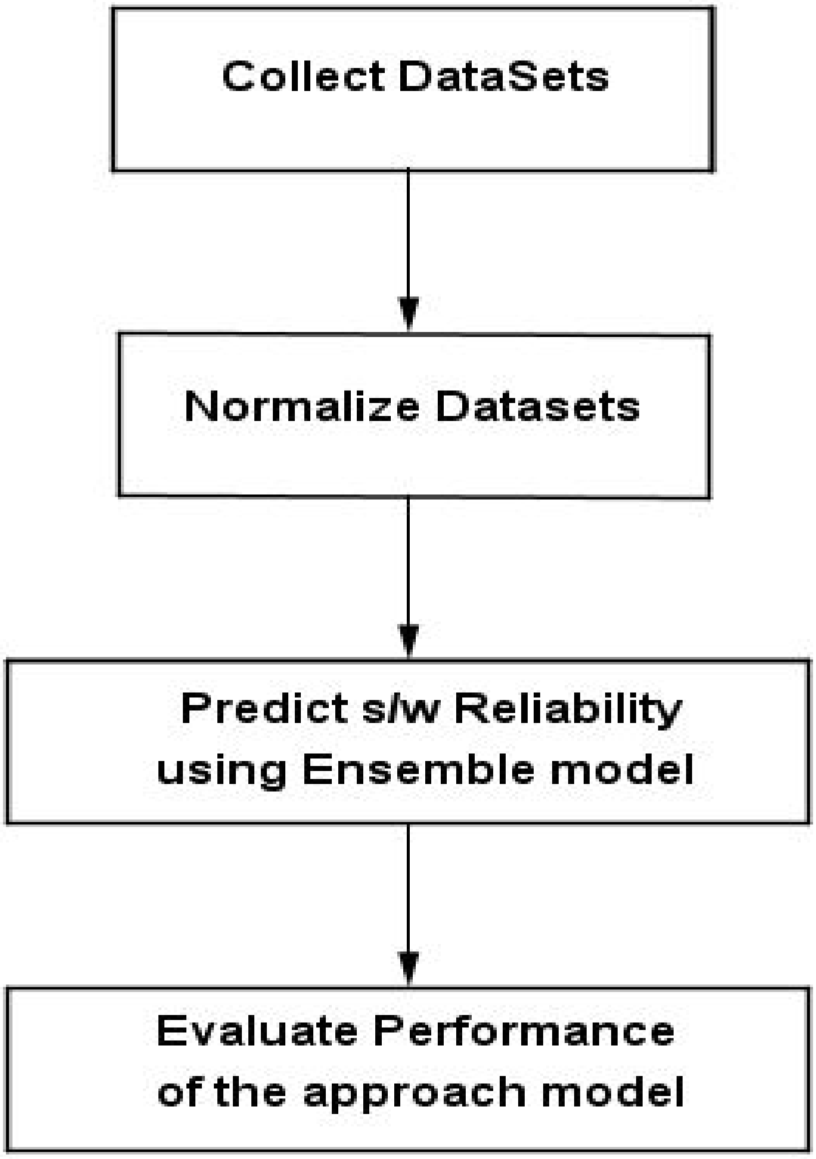


Figure 5.1: Flowchart diagram of Ensemble model .

4. Step-4: Evaluate Performance of the approach model: We have used two types of meaningful Errors such as Relative Error (RE) and Average Error (AE) for performance evaluation of the proposed model and its competent existing model.

5.1.1 Ensemble model

The prediction model based on the Ensemble model is shown in Fig.5.2. Ensemble model is a single input and single output architectural model such as an input layer, a component layer consists of FFNN network, RBF neural network and an output layer is an average combination of output of all component layers. The component layer consists of three components and have used three types of activation functions for three components. The

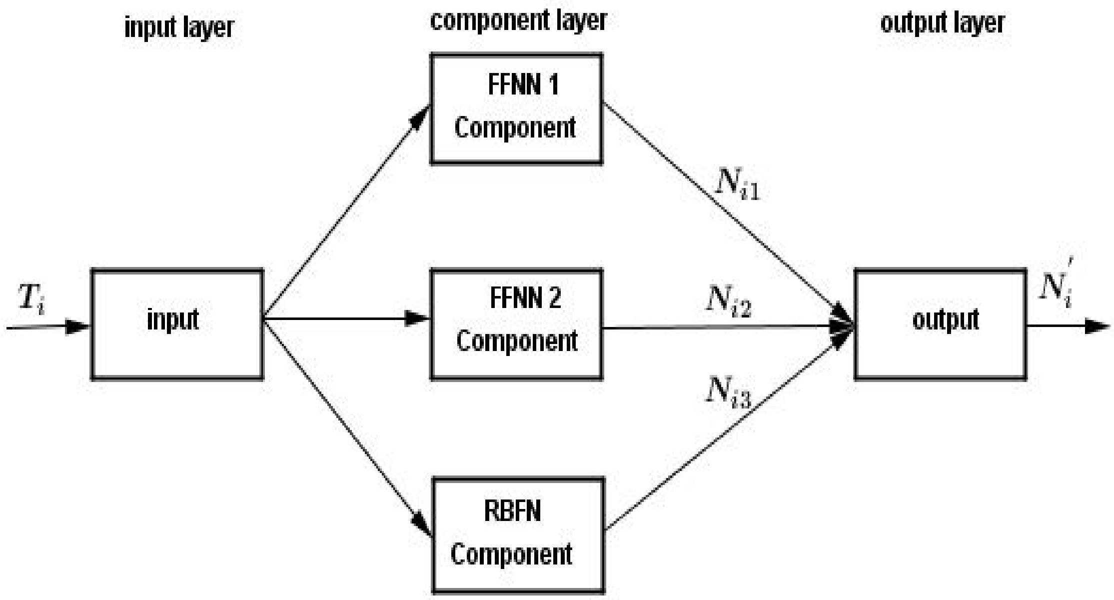


Figure 5.2: Ensemble model .

software reliability data are organized in pair $\{T_i, N_i^{'}\}$, where the calender time T_i is the input and $N_i^{'}$ is the number of failures for forecasting of the proposed model. The output of the Ensemble model is the mean of all three components and is defined as follow.

$$N_i^{'} = \frac{N_{i1} + N_{i2} + N_{i3}}{3} \tag{5.2}$$

5.1.2 FFNN Component

Two types of feed forward neural networks have been used for our Ensemble model such as FFNN1 and FFNN2. The FFNN model is shown in Fig.5.3. The node in the FFNN model is computed as the sum of weighted sum of input data and bias value and the mathematical definition of this process are defined as follows

$$a_i = \sum_{j=1}^{n} w_{ij} x_j + b_i \tag{5.3}$$

$$y_i = f_i(a_i) \tag{5.4}$$

Where a_i is called as the linear combination of input data and b_i is called as the bias value

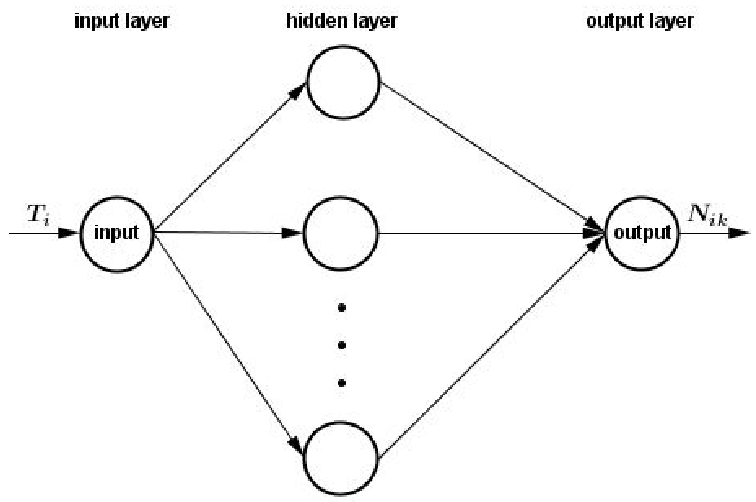


Figure 5.3: FFNN model .

and w_{ij} is called as weight matrices of FFNN model.
For FFNN1 we have used the transfer function such as log sigmoid as activation function and is defined as follows.

$$f(n) = \frac{1}{1 + e^{-n}} \tag{5.5}$$

For FFNN2 we have used the transfer function such as tan sigmoid as activation function and is defined as follows.

$$f(n) = \frac{2}{1 + e^{-2*n}} - 1 \tag{5.6}$$

5.1.3 RBFN Component

Instead of using sigmoid function in FFNN, we have used exponential function as transfer function of FFNN and defined as follows. The input RBF function can be designed as $t \in R_n$ and the output of network can be designed as $f(x) \in R_n \to R$ and defined as follows

$$f(x) = \sum_{i=1}^{k} w_i \phi(x - c_i) \tag{5.7}$$

Where k is called as neurons, x is called as number of inputs and w_i is called as the weight matrices of neuron i and c_i is called as the centroid vector for neuron i.
The function $\phi(x - c_i)$ is given by

$$\phi(x - c_i) = \sqrt{\sum_{i=1}^{k} (x - c_i)^2} \tag{5.8}$$

5.1.4 Generalization Method

To improve the over fitting complication of the Ensemble model, the Bayesian Regularization can be used to modify the error and the error function can be defined as follows

$$Error = \alpha MSE + (1 - \alpha)MSW \tag{5.9}$$

Where MSE is called as the Mean Square Error and MSW is called as the Mean Square Weights.

5.1.5 Performance Measures

We have used Relative Error (RE) for performance measures and compare the proposed model with other existing models and the error function is defined as follows

$$RE = \frac{\hat{y}_i - y_i}{y_i} * 100 \tag{5.10}$$

where, n is called as the total data, $\hat{y}_i$ is called as the predicted value and y_i is called as the actual value.

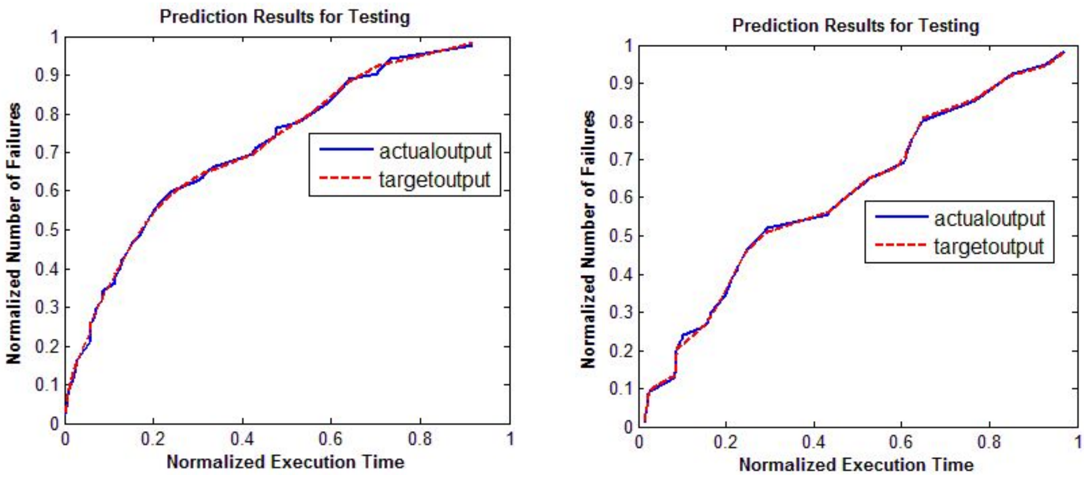
5.2 Experimental Results and Comparison

Four types of datasets DS1, DS2, DS3 and DS4 have been selected [1] for forecasting of reliability in this work. All datasets have been normalized between 0 and 1 by minmax formula.

We have been trained this proposed model with 50 percent for dataset DS1, 66 percent for dataset DS2, 50 percent for dataset DS3 and 55 percent for dataset DS4. For purpose of testing, we have used the remaining datasets . We have used different training ratio of different datasets for better prediction results. The proposed model has been compared with FFNN Network and some of mathematical Growth Models [21, 22] popularly called Duane model. The Duane Model is given by formula as follows

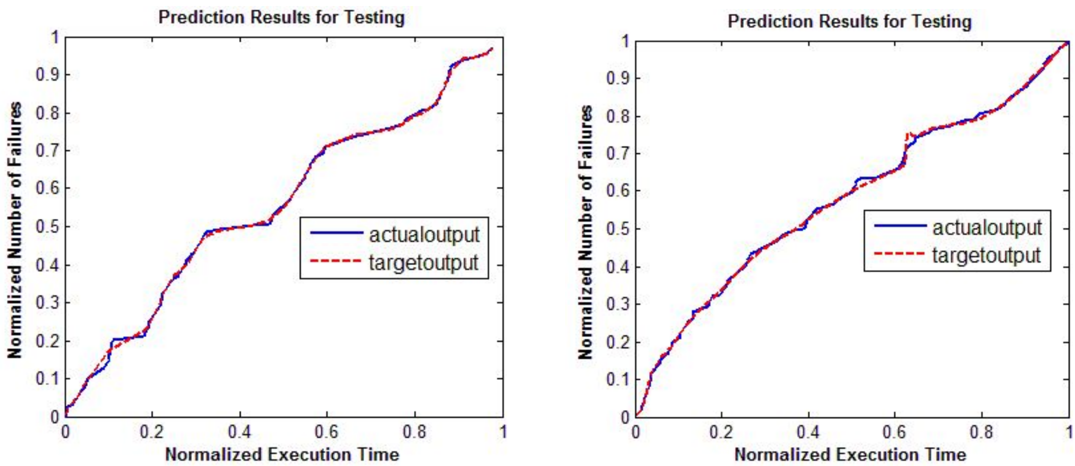
$$\mu(t) = at^b, a > 0, b > 0 \tag{5.11}$$

where a and b is called as the parameters.



(a) DS1 (b) DS2

Figure 5.4: Prediction Results of Ensemble model (a) DS1, (b) DS2

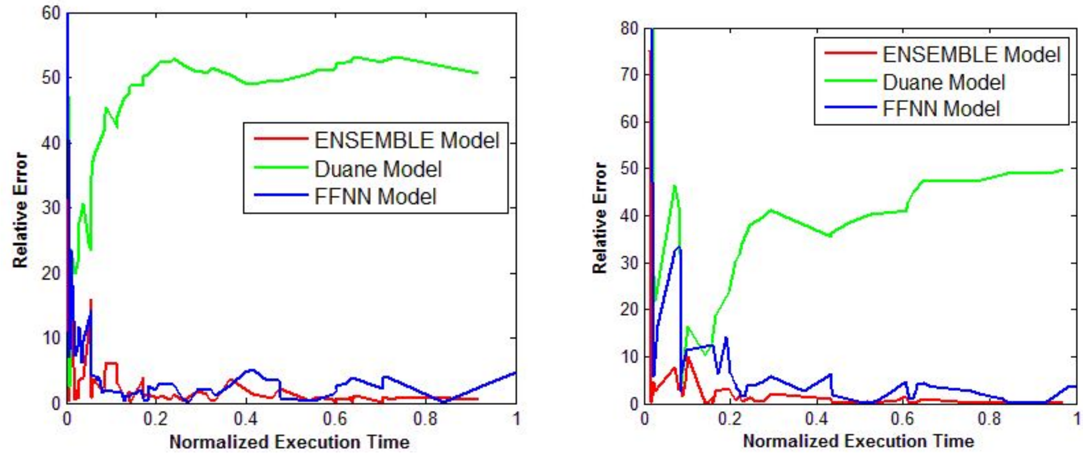


(a) DS1 (b) DS2

Figure 5.5: Prediction Results of Ensemble model (a) DS3, (b) DS4

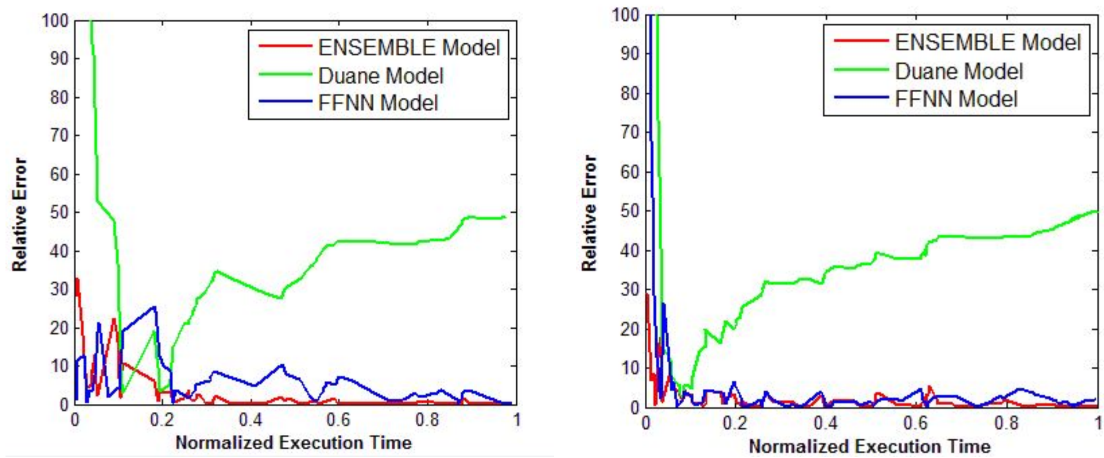
5.2.1 Performance Comparison

For Ensemble model, we have chosen 3 components such as FFNN1, FFNN2, RBFN and each component consists of 5 neurons. For FFNN model, we have taken 5 neurons. The prediction results of Ensemble model on four datasets DS1, DS2, DS3 and DS4 are depicted in Fig.5.4 and Fig.5.5 respectively. For different models, we have depicted the relative error in Fig.5.6 and Fig.5.7. From above comparison results and discussion, it has been concluded that the Ensemble model is best model.



(a) DS1 (b) DS2

Figure 5.6: Relative Errors of different models (a) DS1, (b) DS2



(a) DS1 (b) DS2

Figure 5.7: Relative Errors of different models (a) DS3, (b) DS4

Chapter 6

Neural Network Ensembles of Radial Basis Network for Predicting Software Reliability

This chapter proposes a non-parametric multiple prediction method based on ensemble methods of RBF functional neural network (RNNE) for prediction of s/w reliability efficiently. For most efficient results,we have been used the most efficient learning algorithms called Bayesian Regularization learning algorithm. That's why we have used this type of learning method for improve of the network.The proposed designed model is compared and tested with two other artificial neural networks and another linear parametric software reliability growth model and one ensemble techniques of feed forward neural network model. This ensemble techniques offers a best prediction model to achieve the predictive capability efficiently in all circumstances. The experimental result shows that RNNE model gives better results than the some traditional mathematical models.

6.1 Proposed Work

In this subsection, we discussed about the flowchart of multiple predictor model such as ensemble techniques of rbf network and the flowchart of the RNNE model of software reliability prediction is shown in the Fig.6.1.

1. Step-1: Collect DataSets
   We have used four types of datasets for software reliability prediction. All these datasets DS1, DS2, DS3 and DS4 [1] are arranged from real-time command application, dynamic flight application, dynamic flight application and dynamic flight application respectively.

2. Step-2: Normalize Datasets
   All datasets have been normalized between 0 and 1 by using minmax formula and is

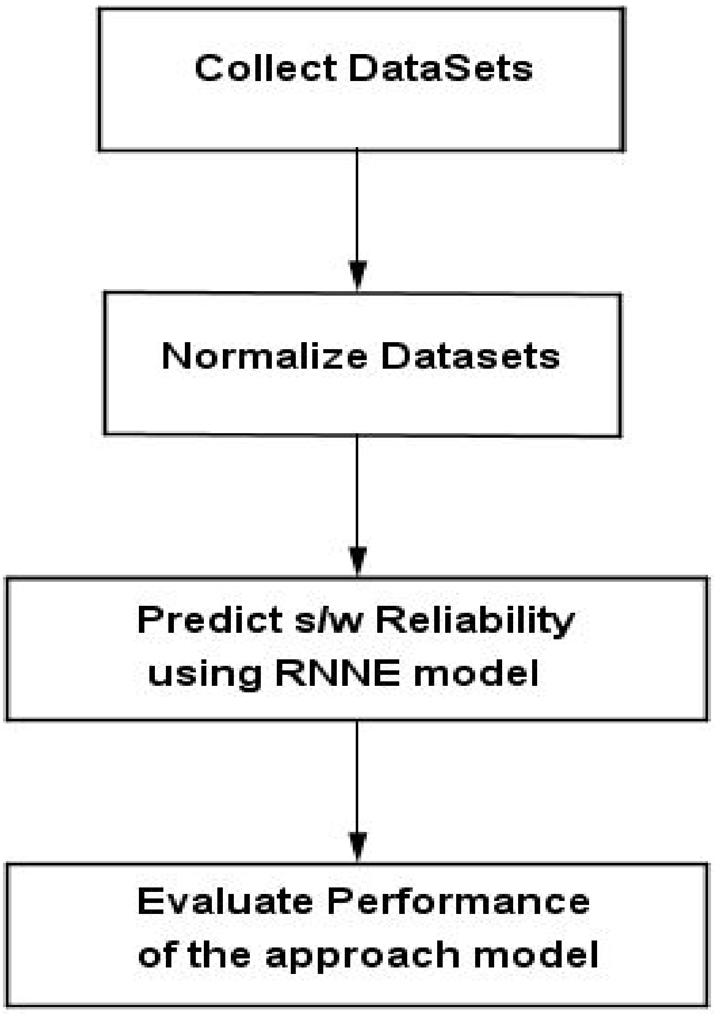


Figure 6.1: Flowchart of RNNE model .

given by the following Eq.6.1.

$$Normalized(t_i) = (t_i - min_A)\frac{max - min}{max_A - min_A} + min \quad (6.1)$$

where, max_A is the maximum value of matrix A, min_A is the minimum value of matrix A and t_i is the execution time.

3. Step-3: Predict software reliability using RNNE model
   The proposed RNNE model based on ensemble techniques of RBF Network is used for forecasting of software reliability efficiently.

4. Step-4: Evaluate Performance of the approach model
   We have used two types of meaningful Errors such as Relative Error (RE) and Average Error (AE) for evaluation of performance of the proposed model and its competent existing model.

6.1.1 RNNE model

The forecasting model based on RBF Network Ensemble techniques (RNNE) is depicted in Fig. 6.2. RNNE architectural model designed over k artificial neural network components such that each component is called as RBF network component. Each RBF component is a three layer three layer network type such as an i/p layer, a hidden layer and an o/p layer. The hidden layer of the component component consists of exponential functions for give better results and this layer is designed over k neurons. The calender time of software T_i is the input of RNNE model. The number of failures $N_{i1}, N_{i2} \cdots\cdots N_{ik}$ for prediction of $1, 2 \cdots\cdots k$ components and the output of the ensemble model is combining all outputs from each component of RNNE architecture by the help of average rule and the average is defined as follows.

$$N_i' = \frac{1}{k}\sum_{i=1}^{k} N_{ik} \quad (6.2)$$

6.1.2 Artificial Neural Network

The prediction system by using the Ensemble techniques of RBF Network (RNNE) architecture consists of k artificial neural network components and each component like Radial Basis Function Network (RBFN)is depicted in Fig. 6.3. RBFN model is a three layer architecture. The RBF network consists of exponential function in hidden layer instead of other types of transfer function. The input T_i is the execution or calender time of the given

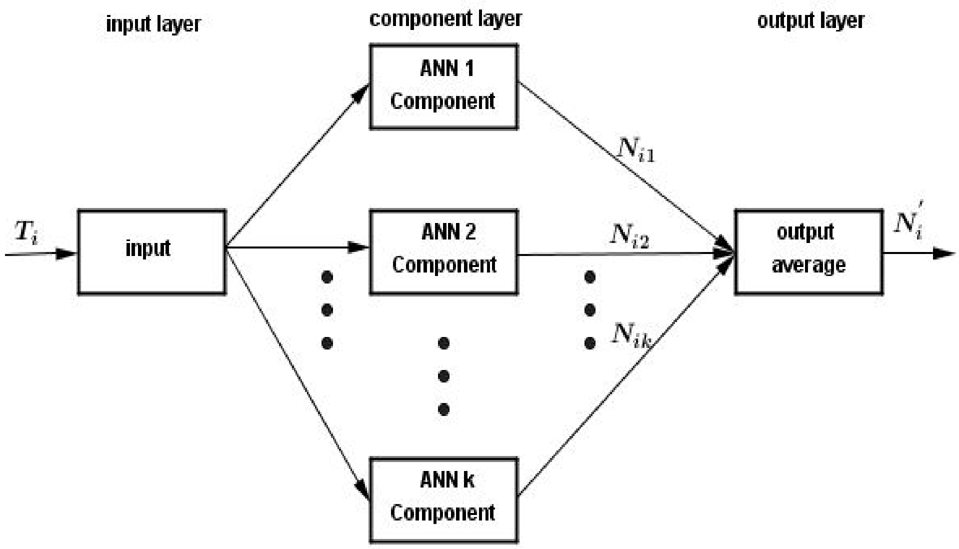


Figure 6.2: RNNE model .

model. The output N_{ik} is software failure for forecasting of k_{th} component of proposed model.

The output function of the RBF network is given by

$$y(x) = \sum_{i=1}^{k} w_i \phi_i(r) \tag{6.3}$$

where, k is called number of neurons, weight vector w_i for neuron i and $\phi_i(r)$ is the nonlinear RBF for neuron i.

We have chosen the nonlinear activation function as Gaussian function of radial basis function neural network as follows.

$$\phi_i(r) = \sum_{i=1}^{k} exp(-\frac{r^2}{2\sigma^2}) \tag{6.4}$$

where, σ is called the standard deviation, r is radius vector of each radial neuron is defined as

$$r = x - c_i \tag{6.5}$$

where, x is called input data and c_i is the centroid of neuron i. Now Eq.4.4 can be written as follows

$$\phi_i(x - c_i) = \sum_{i=1}^{k} exp(-\frac{||x - c_i||^2}{2\sigma^2}) \tag{6.6}$$

The i/p of the RBF network has been written by some real numbers $T_i \in R_n$. The o/p of

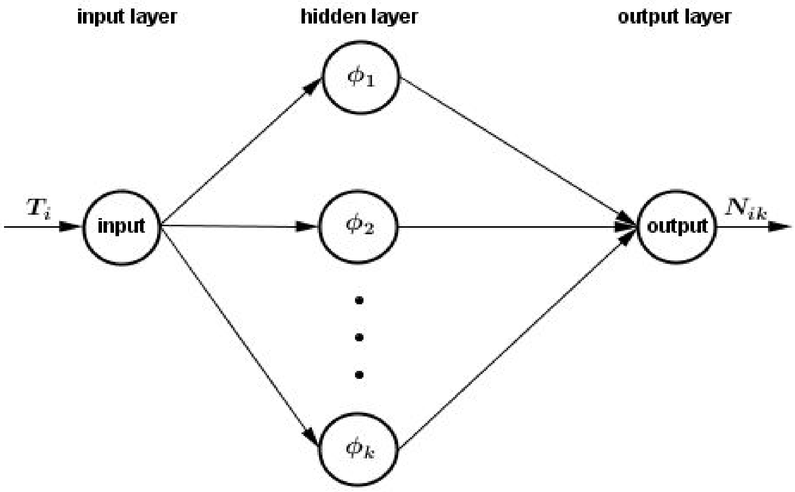


Figure 6.3: Artificial Neural Network Component .

RBF network has been written as $y(x) \in R_n \to R$ is given as follows

$$y(x) = \sum_{i=1}^{k} w_i exp(-\frac{||x - c_i||^2}{2\sigma^2}) \tag{6.7}$$

After fix of the RBFN model, the weight vector matrix of the network will be adjust by a learning algorithm by using training set of data. We have used Levengberg Marquardt (LM) learning algorithm to update the weight vector matrix of the network and achieve the fast convergence [23]. The weight matrices of the network are updated during training process as defined as follows

$$W_{i+1} = W_i - (J^T(W_i)J(W_i) + Z_i I(W_i))^{-1} J^T(W_i)e(W_i) \tag{6.8}$$

Where W_i and $W_(i+1)$ are weight vector matrices of i_{th} and $(i+1)_{th}$ iterations, J and J^T are Jacobian matrix and transpose of the Jacobian matrix of the network with respect to its weight matrices and biases, e is the error vector of the network. [24].

6.1.3 Generalization Method

The prediction result can not give best performance. That's why we have been implement Bayesian regularization learning algorithm [20]. The error function is defined as follows

$$Error = \alpha MSE + (1 - \alpha)MSW \tag{6.9}$$

where, α is called as the ratio of performance.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(|\hat{y}_i - y_i|^2) \tag{6.10}$$

where, m is called as the number of some training samples have been used at the time of training. $\hat{y}_i$ and y_i be the forecasting value and the actual value respectively.

$$MSW = \frac{1}{n}\sum_{i=1}^{n} W_i^2 \tag{6.11}$$

where, n is called as the total number of weights of the network model and w_i is its corresponding network weights.

6.1.4 Performance Measures

We have used two types of error functions such as Relative and Average Errors for performance measures and compare the proposed model with other existing models and both error functions are defined as follows

$$RE = \frac{\hat{y}_i - y_i}{y_i} * 100 \tag{6.12}$$

$$AE = \frac{1}{n}\sum_{i=1}^{n} RE_i \tag{6.13}$$

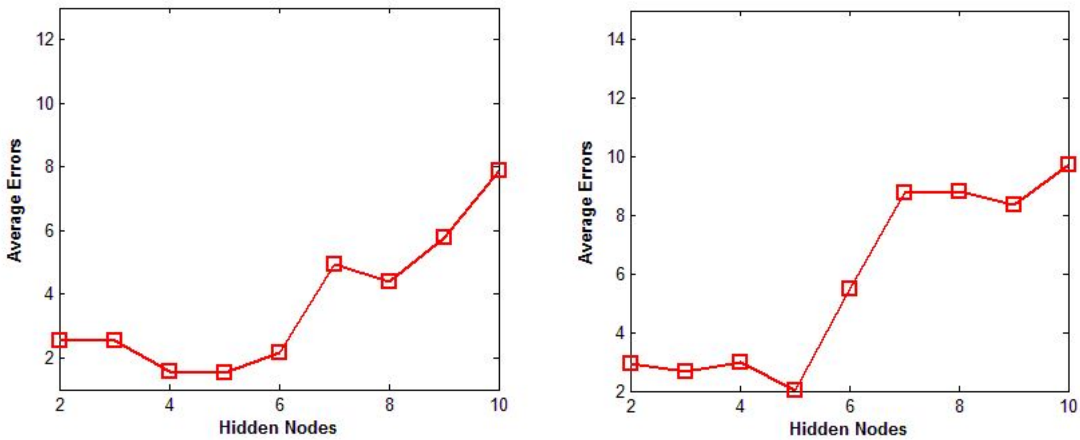
where, n is called data samples, $\hat{y}_i$ is the predicted value and y_i is the actual value.

6.2 Experimental Comparison and Results

Four types of standard datasets DS1, DS2, DS3 and DS4 have been used [1] for software reliability forecasting in this experiments. All datasets have been normalized between 0 and 1 by minmax formula.

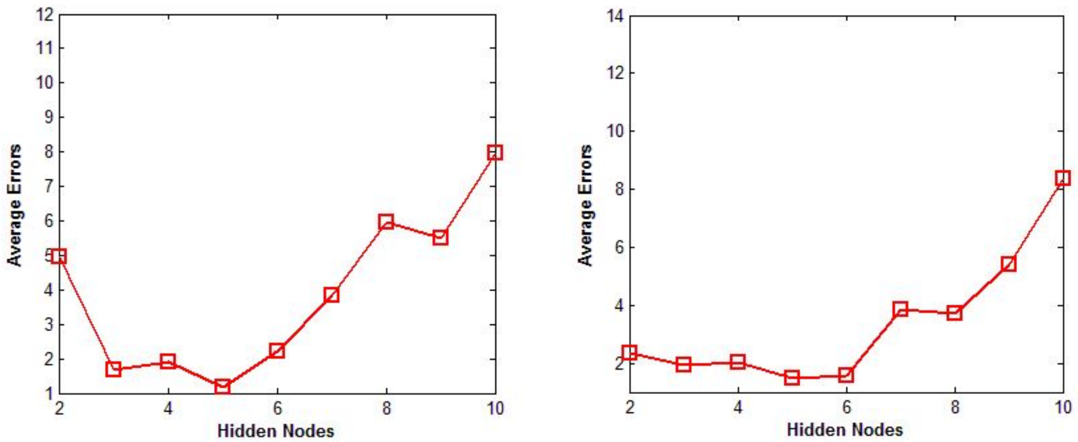
We have been trained the designed model with 50 percent for dataset DS1, 65 percent for dataset DS2, 66 percent for dataset DS3 and 50 percent for dataset DS4. We have been the rest of data for the time of testing.

The proposed designed model has been tested and compared with two other artificial neural network models and one Ensemble techniques of feed forward network model and another mathematical models. [21, 22] called as Duane model: The growth Model is defined as $\mu(t) = at^b, a > 0, b > 0,$. The parameter values of Duane model is calculated by least



(a) DS1 (b) DS2

Figure 6.4: Performance effects on number of hidden nodes of (a)DS1 and (b)DS2



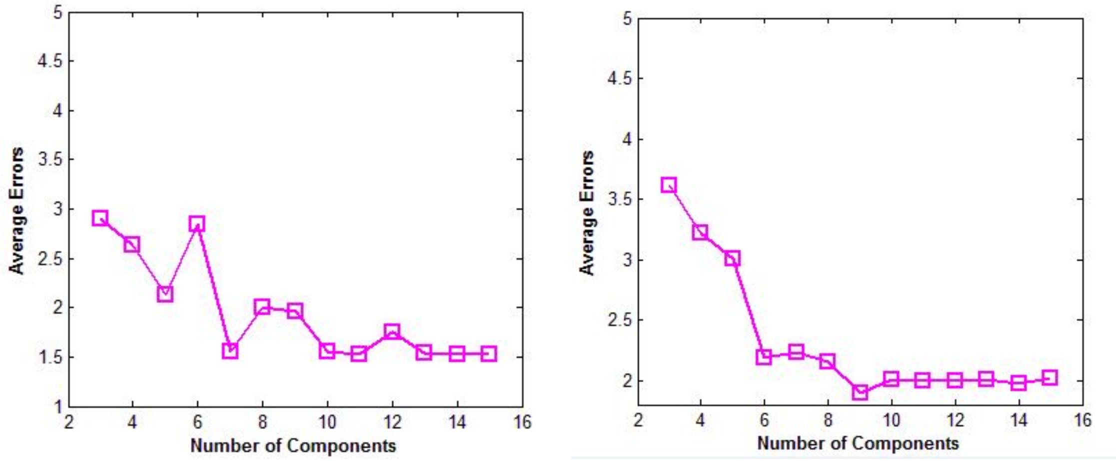
(a) DS3 (b) DS4

Figure 6.5: Performance effects on number of hidden nodes of (a)DS3 and (b)DS4

square method.The hidden layer of FFNN model consists of 5 neurons of our model. We have discussed details about the RBFN model in the above part and also used 5 neurons in the hidden neuron of RBFN model. The Ensemble techniques of Feed Forward Neural network (FFNE) model consists of three layers such as an input layer, a component layer and an output layer. Each component is a feed forward neural network in FFNE model. We have chosen sigmoid function as an activation function in the hidden layer of feed forward neural network and the final output of FFNE model is the average sum of the output of each component.

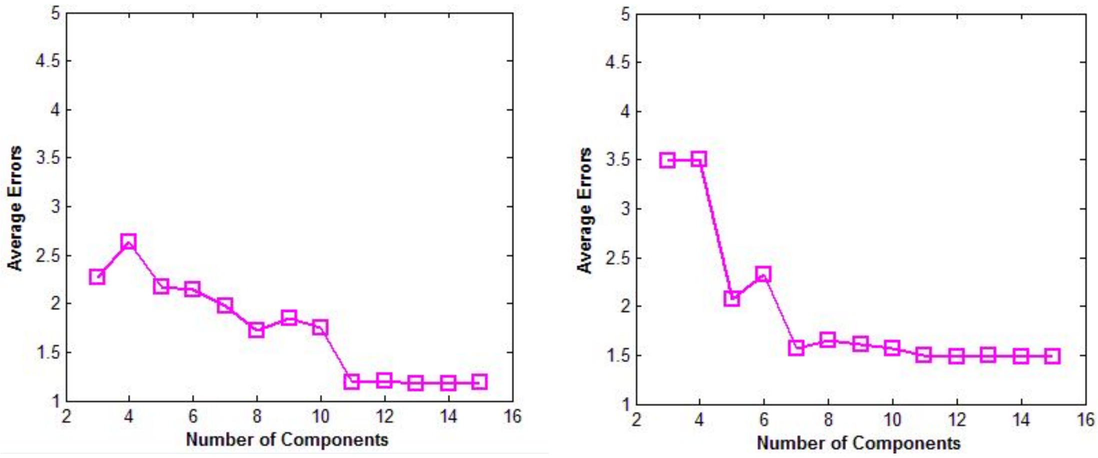
6.2.1 Performance Comparison

We have investigate all combinations of hidden neurons and artificial neural network components of RNNE model and how the performance of RNNE architectural model



(a) DS1 (b) DS2

Figure 6.6: Performance effects on number of artificial neural network components of RNNE model (a) DS1, (b) DS2



(a) DS3 (b) DS4

Figure 6.7: Performance effects on number of artificial neural network components of RNNE model (a) DS3, (b) DS4

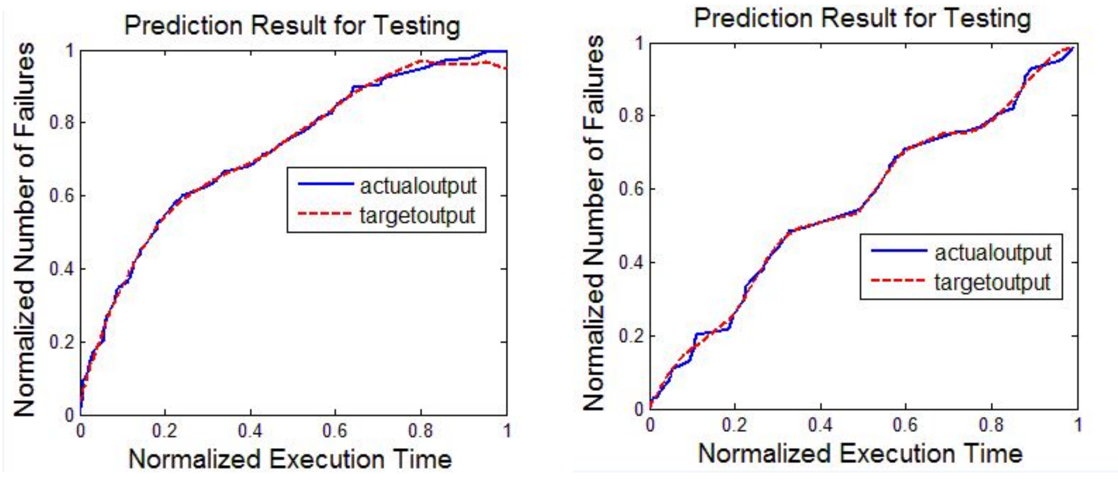
Table 6.1: Comparison AEs of proposed model and other existing models

Dataset	Average Error				
	RNNE	FFNE	FFNN	RBFN	Duane model
DS1	1.2259	1.3912	4.1271	6.2032	41.1720
DS2	1.1842	4.1139	4.0151	4.2155	38.5111
DS3	1.4982	1.6899	1.9366	1.8186	28.2178
DS4	1.4097	2.2511	3.5032	3.2146	35.2281

changes in different number hidden neurons and different number of artificial neural network components. From all combinations we have shown the best performance of hidden neurons and number of components in Fig.6.4, Fig.6.5, Fig.6.6 and Fig.6.7. Fig.6.4 and Fig. 6.5 show the average error plots on the effects of different number of hidden nodes of RNNE model. Fig.6.6 and Fig.6.7 show the average error plots on the effects of different number artificial neural network components of RNNE model.

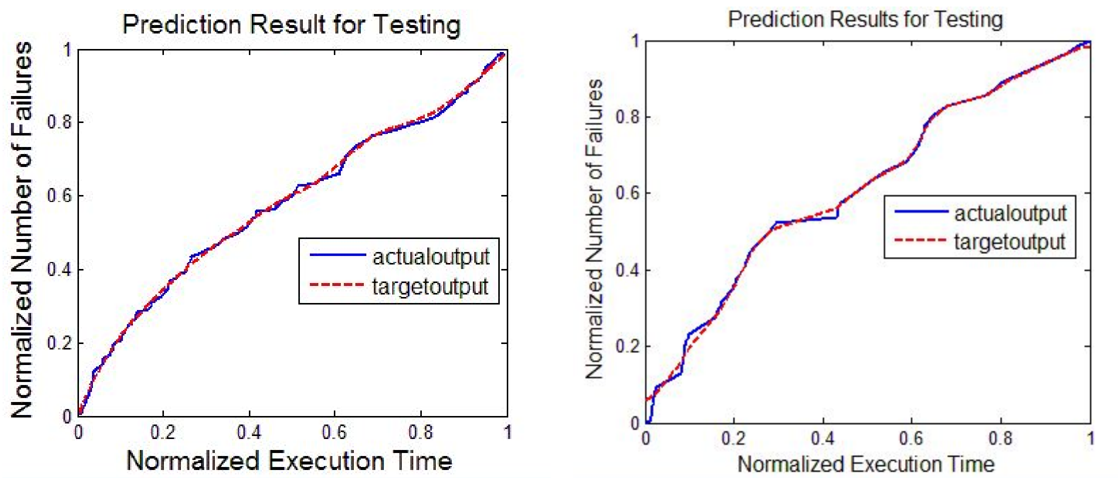
From Fig.6.4 and Fig.6.5 we concluded that larger number of hidden neurons shows worst performance and From Fig.6.6 and Fig.6.7 we concluded that lower number of artificial neural network components shows worst performance. In Fig.6.6 and Fig.6.7 we observe that RNNE model shows the best performance for increasing number of artificial neural network components. If we will chose very large number of components then RNNE model takes more time to predict the software reliability data. For our time complexity and best error performance we have chosen 11 artificial neural network components for RNNE model and each artificial neural network component like radial basis neural network consists of 5 hidden neurons of our architecture model.

The prediction results of RNNE architectural model for four benchmark datasets DS1, DS2, DS3 and DS4 are depicted in Fig.6.7 and Fig.6.8 respectively.The relative errors of different models for dataset DS1, DS2, DS3 and DS4 are shown in Fig.6.10 and Fig.6.11 respectively. The average errors of different models of the datasets DS1, DS2, DS3 and DS4 are shown Table 6.1. From Fig.6.8, Fig.6.9, Fig.6.10, Fig.6.11 and Table 6.1 we observed that RNNE model shows lower prediction errors than two other artificial neural network models such as FFNN model and RBFN model, another classical parametric prediction model and one ensemble techniques of ffnn network. For all datasets parametric prediction model such as Duane model shows worse performance than other artificial network RBF network, ensemble techniques of RBF network and feed forward neural network and also we observed that ensemble techniques gives better prediction performance than simple artificial neural network .



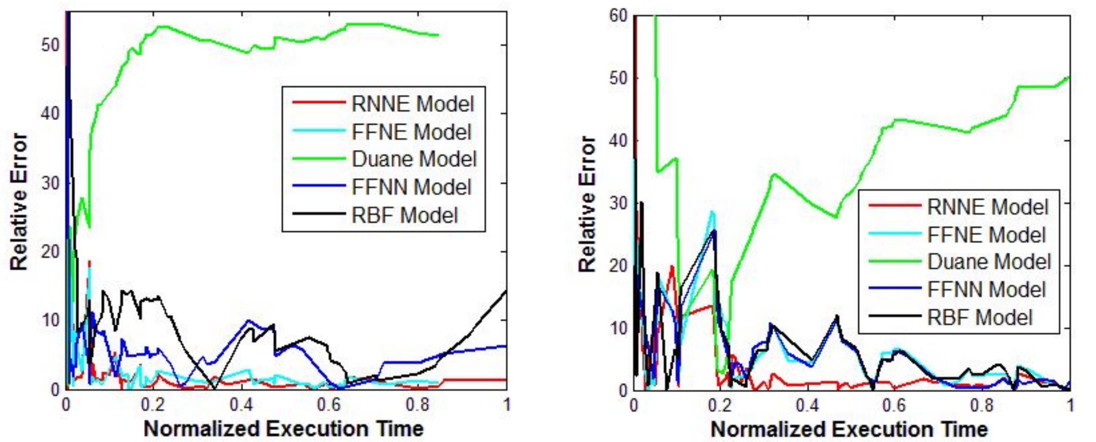
(a) DS3 (b) DS4

Figure 6.8: The prediction results of the proposed model for dataset DS1 and DS2



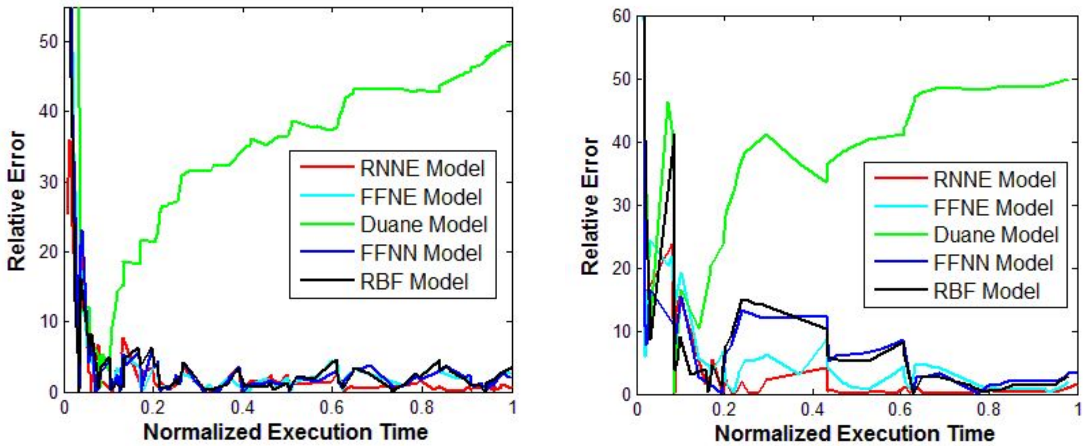
(a) DS3 (b) DS4

Figure 6.9: The prediction results of the proposed model for dataset DS3 and DS4



(a) DS3 (b) DS4

Figure 6.10: Relative Errors of different models for dataset DS1 and DS2



(a) DS3 (b) DS4

Figure 6.11: Relative Errors of different models for dataset DS3 and DS4

Chapter 7

Conclusion

From fourth chapter, we have applied RBFNN for software reliability prediction. This proposed model has been tested on five benchmark datasets. It has been observed that the proposed designed model shows best performance than single artificial neural network like FFNN network and other mathematical Models like Duane model and also we observed that the Duane model shows very worst performance than all other models for all datasets.

From fifth chapter, we have designed the Ensemble model based on the two types of FFNN Network and one RBFN Network. To generalize the network and avoid the over-fitting problem of the proposed model we have implemented the Bayesian regularization method on the proposed model. Finally, it is observed that the Ensemble model called multiple predictor model shows best performance than single predictor like FFNN Network and one statistical method like Duane model.

From sixth chapter, we have implemented the model called RBFN Ensemble (RNNE) model based RBFN Network. we have also used the Bayesian regularization method to avoid the over-fitting problem of the proposed RNNE model and generalize the network. In this chapter, we have implemented two types of multiple predictor models for predicting of software reliability and other artificial neural network models and another statistical model. But it has been observed that the multiple predictor model shows the best performance than all single predictor and Software Reliability Growth Models over all datasets.

From all three chapters, we concluded that nonparametric growth model such as RBFNN model, Ensemble model based on Feed Forward Neural Network and RBFN network and RBFN Network Ensemble (RNNE) model show best performance than Software Reliability Growth Model (SRGM) like Duane model and other Artificial Neural Networks like FFNN etc. in all circumstances. RNNE model is the extension model of all two models such as RBFNN and Ensemble model. Finally, we concluded that the ensemble model is the latest model and also working efficiently for software reliability growth model than other neural networks and Software Reliability Growth Models.

7.1 Summary

Software Reliability and quality is the most and important determinant factor of software estimation and prediction during the testing time of software. Last four or five decades, most of mathematical models have been designed to forecasting the cost of software or hardware reliability and time factors of reliability. software should be reliable, critical for both software users and producers, in our society. Generally, major disruption happens only in failure of softwares, business and can happen services also. So, It is a big challenge for software industries of reliability and be able to enhance the quality of the proposed model in the life cycle. Because it is so very cost-effective to correct many faults of the software early in the process of development. That's why building of software reliability prediction or estimation models have gained considered importance in assessing reliability of software products. So we plan to design a non-parametric prediction model of software reliability prediction based on advanced soft computing techniques, which can predict software reliability efficiently. We will develop also many applications of software reliability prediction such as whether forecasting, greenhouse effects, seasonal and annual temperature deviations etc. The major application of software reliability prediction model is most widely use in military and commercial purpose of many countries such as United States, China, France, United Kingdom etc.

7.2 Scope for Further Research

we will extend the proposed RNNE model and test over more to more software reliability datasets. We plan to implement the proposed RNNE model in other types datasets such as annual and global monthly temperature prediction of United Kingdom, solar radiation datasets and stock price exchange datasets.

References

[1] M. R. Lyu, *Handbook of software reliability engineering*. IEEE computer society press CA, 1996, vol. 222.

[2] Y. K. Malaiya, M. N. Li, J. M. Bieman, and R. Karcich, "Software reliability growth with test coverage," *Reliability, IEEE Transactions on*, vol. 51, no. 4, pp. 420–426, 2002.

[3] H. Pham, L. Nordmann, and X. Zhang, "A general imperfect-software-debugging model with s-shaped fault-detection rate," *Reliability, IEEE Transactions on*, vol. 48, no. 2, pp. 169–175, 1999.

[4] R. Mohanty, V. Ravi, and M. R. Patra, "Hybrid intelligent systems for predicting software reliability," *Applied Soft Computing*, vol. 13, no. 1, pp. 189–200, 2013.

[5] C.-J. Lu, "Sales forecasting of computer products based on variable selection scheme and support vector regression," *Neurocomputing*, vol. 128, pp. 491–499, 2014.

[6] P. Roy, G. Mahapatra, and K. Dey, "Neuro-genetic approach on logistic model based software reliability prediction," *Expert Systems with Applications*, vol. 42, no. 10, pp. 4709–4718, 2015.

[7] J. Zheng, "Predicting software reliability with neural network ensembles," *Expert systems with applications*, vol. 36, no. 2, pp. 2116–2122, 2009.

[8] N. Karunanithi, D. Whitley, and Y. K. Malaiya, "Prediction of software reliability using connectionist models," *Software Engineering, IEEE Transactions on*, vol. 18, no. 7, pp. 563–574, 1992.

[9] Y. S. Su and C. Y. Huang, "Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models," *Journal of Systems and Software*, vol. 80, no. 4, pp. 606–615, 2007.

[10] M. J. L. Orr, "Introduction to radial basis function networks," 1996.

[11] N. R. Kiran and V. Ravi, "Software reliability prediction by soft computing techniques," *Journal of Systems and Software*, vol. 81, no. 4, pp. 576–583, 2008.

[12] N. Karunanithi, Y. K. Malaiya, and D. Whitley, "Prediction of software reliability using neural networks," 1991, pp. 124–130.

[13] N. Karunanithi and D. Whitley, "Prediction of software reliability using feedforward and recurrent neural nets," vol. 1, 1992, pp. 800–805.

[14] N. Karunanithi, D. Whitley, and Y. K. Malaiya, "Using neural networks in reliability prediction," *Software, IEEE*, vol. 9, no. 4, pp. 53–59, 1992.

[15] R. Sitte, "Comparison of software-reliability-growth predictions: neural networks vs parametric-recalibration," *Reliability, IEEE Transactions on*, vol. 48, no. 3, pp. 285–291, 1999.

[16] K. Y. Cai, L. Cai, W. D. Wang, Z. Y. Yu, and D. Zhang, "On the neural network approach in software reliability modeling," *Journal of Systems and Software*, vol. 58, no. 1, pp. 47–62, 2001.

[17] S. L. Ho, M. Xie, and T. N. Goh, "A study of the connectionist models for software reliability prediction," *Computers & Mathematics with Applications*, vol. 46, no. 7, pp. 1037–1045, 2003.

[18] L. Tian and A. Noore, "Evolutionary neural network modeling for software cumulative failure time prediction," *Reliability Engineering & system safety*, vol. 87, no. 1, pp. 45–51, 2005.

[19] ——, "On-line prediction of software reliability using an evolutionary connectionist model," *Journal of Systems and Software*, vol. 77, no. 2, pp. 173–180, 2005.

[20] F. D. Foresee and M. T. Hagan, "Gauss-newton approximation to bayesian learning," vol. 3, 1997, pp. 1930–1935.

[21] L. Shanmugam and L. Florence, "A comparison of parameter best estimation method for software reliability models," *International Journal of Software Engineering & Applications*, vol. 3, no. 5, pp. 91–102, 2012.

[22] A. S. Smith and S. S. Oren, "Reliability growth of repairable systems," *Naval Research Logistics Quarterly*, vol. 27, no. 4, pp. 539–547, 1980.

[23] S. Haykin and R. Lippmann, "Neural networks, a comprehensive foundation," *International Journal of Neural Systems*, vol. 5, no. 4, pp. 363–364, 1994.

[24] B. M. Wilamowski, S. Iplikci, O. Kaynak, and M. Ö. Efe, "An algorithm for fast convergence in training neural networks," in *Proceedings of the international joint conference on neural networks*, vol. 2, 2001, pp. 1778–1782.

Dissemination

Conferences

1. Bal, P. R. and Mohapatra, D. P., ”Software Reliability Prediction Based on Radial Basis Function Neural Network”, Proceedings of international conference on computational intelligence. (ICCI 2015), springer, 2015 (Accepted).

2. Bal, P. R., Jena, J., Mohapatra, D. P., ”Software Reliability Prediction based on ensemble models”, Proceedings of international conference on intelligent communication, control and devices. (ICICCD 2016), springer, 2016 (Accepted).