Neural Network Model Selection For Software Reliability

Monika Sethi

GGDSD College, Sec-32, Chandigarh, India

Jatender Kumar

SGGS College, Sec-26, Chandigarh, India

ABSTRACT*: The study of this paper focuses on the one of the major problem that exist in the field of software engineering i.e. evaluation of software on the basis of reliability. Software reliability prediction is one of the mind boggling issue in its field. The work proposed in the paper is based upon the fifth generation technology that is Artificial Neural Network (ANN). For the best suitable ANN type selection out of trending models like Feed Forward Neural Network (FFNN) and Cascade Feed Forward Neural Network (CFFNN) by means of performance analysis. The parameters used for evaluating the performance of the ANN models are MSE, RMSE, Correlation and Mean-MSE. Analysis conclude that the CFFNN is providing least mean square error (MSE) compared to FFNN. It is observed that the CFFNN is the best variant of artificial neural network (ANN) which out performs the other variant with respect to the calculation of Mean-MSE. The study is categorized in various catalog as first section highlights the various models that had been used to evaluate the software reliability, Second section describes the motivation or objective behind conducting the research work, Third section is planned for the implementation of the proposed work, Fourth section describes the evaluation and results of the implemented work and Fifth section describes the future work with respect to the proposed work.*

KEYWORDS: Artificial neural network, Software reliability, feed forward neural network (FFNN), Cascade Feed Forward Neural Network (CFFNN).

I. INTRODUCTION

The Forecasting of the lifespan of a system is a fascinated matter among the researchers of different field like Medical science, Software engineering, Information Technology etc [1]. Predicting the software reliability is one the acclaimed work by the precisionist. Software reliability is an indispensable wedge of software quality assurance. A quality acuteness of reliability allude to predicting that a product will be in functionality with respect to given predicament without any collapse [2]. The reliability of the software predominantly relies on the reliability of individual intrinsic and alliance among them [3].Software Reliability Engineering suggests the SRM (Software Reliability Models) as modeling can act as an aid to both developers and testing team as well [4]. Software Reliability model falls into various categories like Parametric reliability Model [5], Analytical reliability model [4], Architecture based reliability model [3], soft computing based reliability model [6] etc.

A. ARCHITECTURE BASED RELIABILITY MODEL

Architectural reliability of software relies on various parameters like size, trustworthiness, complexity of solitary components along with the correlation among them. **FurakhZeshan** [3] perform comparison of various architecture based reliability models such as state model, Composite Model, Path based, Hierarchical Model etc. and derives a conclusion that the hierarchical model can perform better as compare to others.

B. ANALYTICAL BASED RELIABILITY MODEL

Jung-Hua Lo [4], proposed a hybrid analytical model for software reliability prediction. ARIMA (Auto Regression Integrated Moving Average) and SVM (Support Vector Machine) were hybridized into a single model. The grail behind considering these two models were that these models can perfectly predict the culpable both linear and non-linear time series data respectively.

C. PARAMETRIC RELIABILITY MODEL

Anurag Sinha et al. [5] used SRGM (Software Reliability Growth Model) for implementing their work. They performed a comparison between PSRGM (Parametric SRGM) and NPSRGM (Non-Parametric SRGM). Most of the research focused on SRGM because it has an advantage over the PSRGM that it does not necessitate to have assumptions. In this study two parametric SRGM and two non-parametric SRGM along with three real life datasets were considered for evaluation purpose.

D. SOFT COMPUTING BASED RELIABILITY MODELS

As reviewed from previous sort of modeling concept, it can be said that most of the reliability models are manual or follows working criteria as traditional work. Soft Computing Reliability model covers various techniques like ANN (Artificial Neural Network), Fuzzy System (FS), Swarm Intelligence etc. Soft computing also known as Computational Intelligence as it is a scrap of Artificial intelligence Technology [6]. With the drastic thriving in technology the researchers gets attracted towards the modeling techniques of fifth generation.

ANN: It stands for Artificial Neural Network. The working criteria of ANN are homologous to human brain. It abides some internal learning functioning for the purpose of modeling reliability. It is named as neural network since it constitutes of small neurons which are interlinked to each other for establishing the communication among these small elements. These neurons are placed layer by layer in the network which directly makes a consonance. Then the supervised learning mechanism is used for the purpose of training the network. The training procedures continuous until the desired results or responses are produced by the network. ANN model is categorized as follows:

- Feed Forward Learning Neural Network
- Cascade Feed Forward Neural Network

FEED FORWARD NEURAL NETWORK (FFNN): Feed forward neural network is an artificial network, which works apart from working criteria of other neural networks. Since in feed forward neural network the nodes haven't created any cycle or loop like other networks. It is a type of self-patterned algorithm or technique. Feed forward network is assorted in two categories as follows: [7]

- Single layered feed forward NN
- Multi layered feed forward NN

CASCADE FEED FORWARD NEURAL NETWORK (CFNN): Cascade feed forward network work as same as the feed forward network. But the only variation among both of the neural network is that in a cascade neural network there is the weight value connected from each input layer to every subsequent layer.

II. PROPOSED WORK

Over the last decade the concept of Artificial Neural Network was introduced in research area of software reliability prediction. ANN is an artificial intelligent technology which contributes to achieve the maximum software reliability and to decrease the chances of failures. Some of the work leads to the reduction in performance of the system and some other leads to the increment in the performance of the system. Hence the need to develop such a method which can increase the performance of the system in order to check the reliability in an effective and efficient manner has been raised. In this paper we have developed a technique by framing a system for software reliability prediction using most of the prominent methods of ANN and evaluates their performance individually in order to elect the best one. In this paper we have worked on two variants of ANN and are applied to measure the quality of the software in terms of reliability. Two variants Feed forward and Cascade feed forward neural network is applied and evaluated. The selections of the neural variants are done on the basis of earlier reviewed studies.

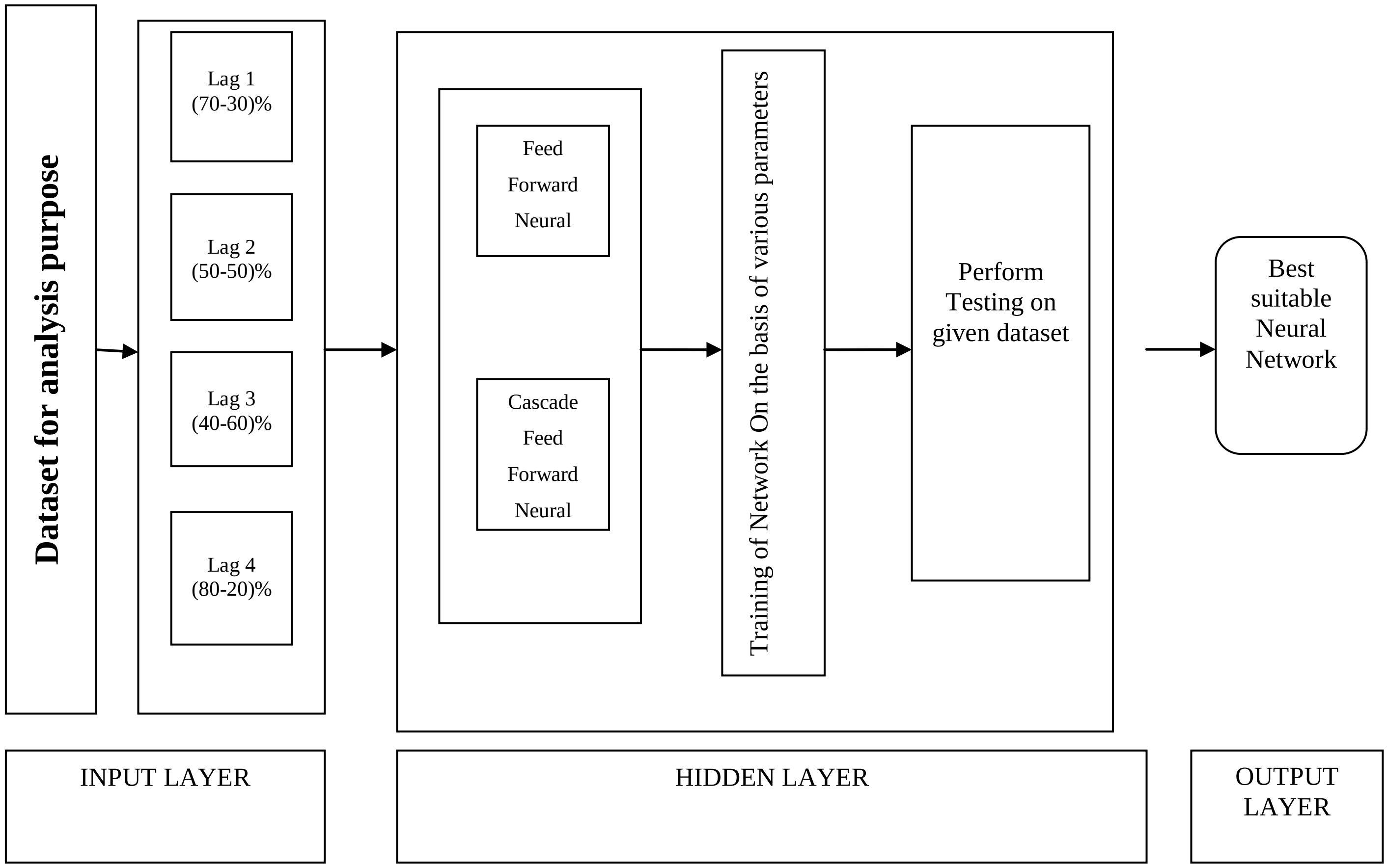

Figure 1. Framework of the Proposed Work

The Figure 1 describes the framework of the proposed work diagrammatically. The working is divided into three layers. These layers are as follows:

- **Input Layer:** Input layer refers to the various lags of the dataset which are used as input. Here the dataset is divided into four lags. Then this lagging data will be an input to the neural network.
- **Hidden Layer: T**his layer constitutes of internal processing of the neural network. Hidden layer consist of variants of neural networks like FFNN and CFFNN; Training of the input dataset; testing or evaluation of trained dataset.
- **Output Layer:** Output layer refers to the obtained output after performing the training and testing

III. IMPLEMENTATION

A. DATASET OBSERVATIONS:

This proposed technique is evaluated by using a dataset which is observed from Musa J.D et al [8]. The database consists of software failure data. This dataset contains 101 (ranges from 0 to 100) observations of as shown in Table 1. Here in data set the parameter named as t represents the time interval at which the particular modification has been done. And Y_t stands for the time to failure of the software with respect to given time interval. This dataset is used for the purpose of training and testing the network. The evaluation or analysis will be done by using the dataset in various proportions. For example the proportion of the dataset has the following lags (%):

- 70-30
- 50-50
- 40-60
- 80-20

Table 1. Dataset (Musa 1979) Software failure [8]

T	**Y_t**	**t**	**Y_t**	**t**	**Y_t**	**t**	**Y_t**
0	5.7683	**25**	7.5443	**51**	10.3534	**76**	13.3279
1	9.5741	**26**	8.5941	**52**	10.0998	**77**	8.9464
2	9.105	**27**	11.0399	**53**	12.6078	**78**	14.7824
3	7.9655	**28**	10.1196	**54**	7.1546	**79**	14.8969
4	8.6428	**29**	10.1786	**55**	10.0033	**80**	12.1399
5	9.9887	**30**	5.8944	**56**	9.8601	**81**	9.7981
6	10.1962	**31**	9.546	**57**	7.8675	**82**	12.0907
7	11.6399	**32**	9.6197	**58**	10.5757	**83**	13.0977
8	11.6275	**33**	10.3852	**59**	10.9294	**84**	13.368
9	6.4922	**34**	10.6301	**60**	10.6604	**85**	12.7206
10	7.901	**35**	8.3333	**61**	12.4972	**86**	14.192
11	10.2679	**36**	11.315	**62**	11.3745	**87**	11.3704
12	7.6839	**37**	9.4871	**63**	11.9158	**88**	12.2021
13	8.8905	**38**	8.1391	**64**	9.575	**89**	12.2793
14	9.2933	**39**	8.6713	**65**	10.4504	**90**	11.3667
15	8.3499	**40**	6.4615	**66**	10.5866	**91**	11.3923
16	9.0431	**41**	6.4615	**67**	12.7201	**92**	14.4113
17	9.6027	**42**	7.6955	**68**	12.5982	**93**	8.3333
18	9.3736	**43**	4.7005	**69**	12.0859	**94**	8.0709
19	8.5869	**44**	10.0024	**70**	12.2766	**95**	12.2021
20	8.7877	**45**	11.0129	**71**	11.9602	**96**	12.7831
21	8.7794	**46**	10.8621	**72**	12.0246	**97**	13.1585
22	8.0469	**47**	9.4372	**73**	9.2873	**98**	12.753
23	10.8459	**48**	6.6644	**74**	12.495	**99**	10.3533
24	8.7416	**49**	9.2294	**75**	14.5569	**100**	12.4897
		50	8.9671				

B. NEURAL NETWORK CONFIGURATION

This module performs the object creation with respect to the used neural network. Both the neural networks FFNN and CFFNN are created in this step. The parameters defined while configuring the neural network is shown in the Table 2.

Table 2. Configuration of Neural Networks:

S.No.	Parameters	Value
1	Training Function	Trainlm
2	Learning Function	Learngdm
3	Type of Network	linear or non-linear network
4	Number of neurons	30
5	Number of Epochs	1000

The parameters used in Table 2 are as follows:

- Training Function: This parameter refers to the function which is used for train the neural network.
- Learning Function: This parameter defines the function used for testing or learning purpose.
- Type of Network: This parameter refers to the nature of the neural network. Neural network can be of two types one is linear network and other is non-linear network.
- Number of Neurons: As neural network is made up of multiple neurons hence this parameter defines the number neurons used for configuring the neural network.
- Number of Epochs: Epochs refers to the iteration perform by the network. By default 1000 epochs are performed by the neural network.

C. TRAINING DATASET

In this module the inputs is forwarded to the ANN for the purpose of training or learning. In this a proportion of data is set for the training purpose. As the dataset is divided into four ratios (lags), all of the proportions act as an input data for learning procedure one by one.

D. TESTING AND EVALUATION

This module performs evaluation or analysis of the trained data for the purpose of getting the best output. The dataset which is set apart from the training data is used for the purpose of testing or prediction. After testing the evaluation is performed in the form of various parameters as shown in Table 3.

Table 3. Parameter list for evaluation

S. No	Parameter	Formulation
1.	Error	$\sqrt{(x_1 - x_2)^2}$
2.	MSE	$\frac{\sum(x_1 - x_2)^2}{\text{Size of Data}}$
3.	RMSE	$\sqrt{\frac{\sum(x_1 - x_2)^2}{\text{Size of Data}}}$
4.	Correlation	$\mathrm{corr}(X, Y) = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$

IV. RESULTS AND DISCUSSION

LOAD DATASET: The dataset used for training and testing purpose contains the 101 observations related to the software failure as shown in Figure 2.

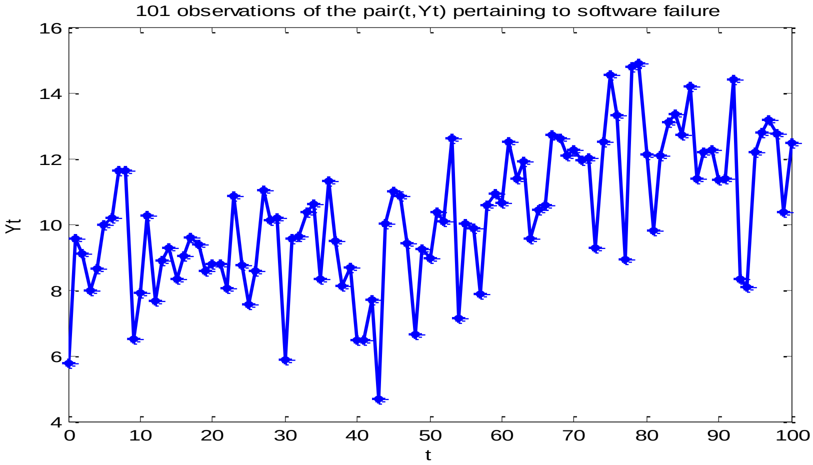

Figure 2. Loading Test Dataset

Training and Testing the NN: The results obtained after applying both of the neural networks on four data set are depicted in the form of various parameters such as MSE, RMSE and Correlation in order to elect the best of the neural network. Table 4. shows the values of resultant parameters MSE which is observed after applying the Feed Forward Neural Network and Cascade Feed Forward Neural Network (CFFNN) on four tested data sets.

Table 4. MSE of FFNN and CFFNN for all Data Lags

S. No.	Data Lags	FFNN	CFFNN
1.	70-30	108.1756	110.1983
2.	50-50	107.8102	94.8880
3.	40-60	673.4878	598.8788
4.	80-20	975.8828	1012.9

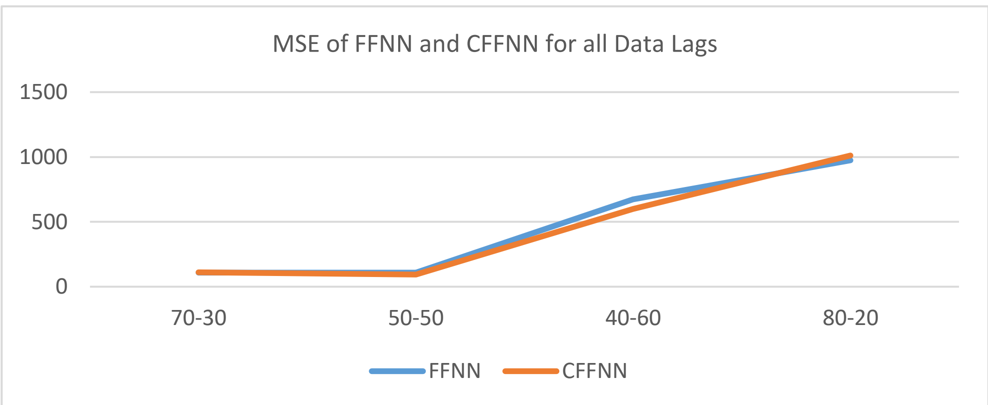

Figure 3. Graph of Mean Square Error for FFNN and CFFNN on the basis of various data lags.

Figure 3 depicts the pictorial representation of the values of MSE on the basis of four data lags(70-30, 50-50, 40-60 and 80-20) with respective to neural variants. As seen from the graph the curve of CFFNN is quite lower as compared to the curve of FFNN. Hence it is analyzed that the CFFNN has a lower Mean Square Error in contrast of FFNN.Table 5 depicts the value of RMSE with respect to Dataset in case of FFNN and CFFNN. The graphical representation of these values is presented in the Figure 4.

Table 5. RMSE of FFNN and CFFNN for all Data Lags

S. No.	Data Lags	FFNN	CFFNN
1.	70-30	10.4008	10.4975
2.	50-50	10.3832	9.7410
3.	40-60	25.9516	24.4720
4.	80-20	31.2391	31.8259

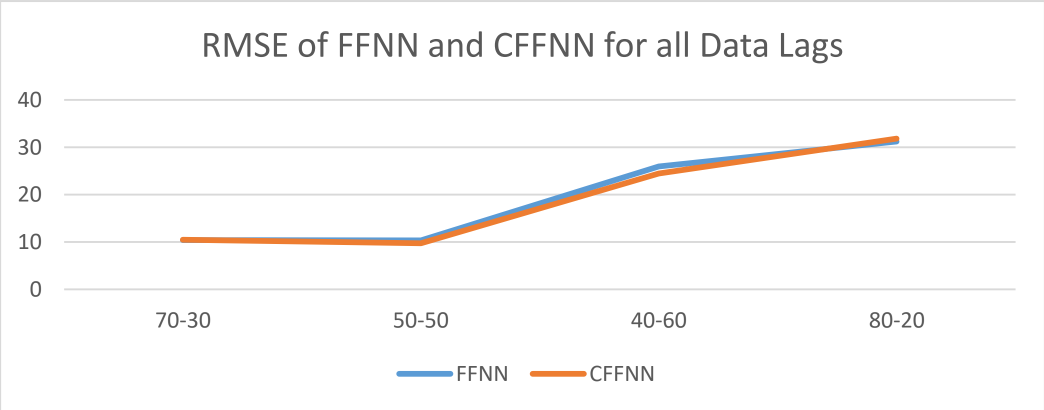

Figure 4. Graph of Root Mean Square Error.

The Figure 4 traces the RMSE of all neural variants in case of variation in tested data lags. It defines the clear difference among anticipated values and the values which are actually observed after implementation with respect to both neural variants and data lags also. In case of CFFNN, RMSE is less as compared to other two cases. Similarly Table 6 unfold the resultant parameter (Correlation) on the basis of various proportions of datasets in case FFNN and CFFNN.

Table 6.Correlation in case of neural variants and datalags.

S. No.	Data Lags	FFNN	CFFNN
1.	70-30	0.0142	0.1575
2.	50-50	0.1319	0.3187
3.	40-60	0.5332	0.8062
4.	80-20	0.5298	0.6944

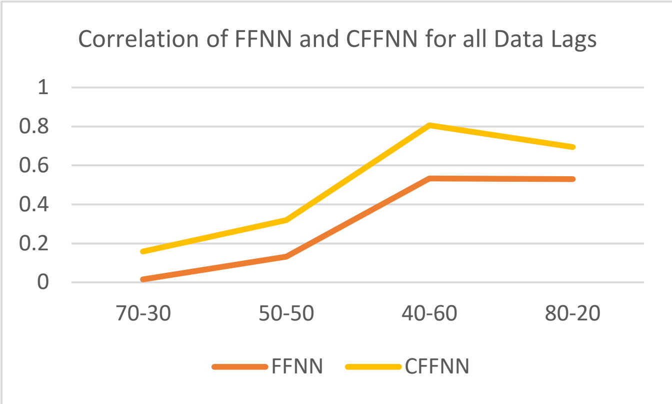

Figure 5. Correlation for FFNN and CFFNN in case of various data lags.

Evaluation of results: As above section displays the results obtained after implementing the proposed work on various proportions of data individually. Now on the basis of observed results the best ANN variant will be decided by measuring the mean MSE of both of the neural variants. The following formulation symbolizes for evaluating the mean results:

$$Mean(MSE) = \frac{\sum(Data\ lag1 + Data\ lag2 + Data\ lag3 + Data\ lag4)}{Total\ number\ of\ data\ lags} \qquad (1)$$

The equation(1) defined above is implemented for calculating the mean results obtained from Feed Forward Neural Network. Here the MSE value that is spotted after implementing the proposed work with respect to four data lags is used for calculating the mean resultant value. Then the mean value of MSE that is calculated is considered in order to elect the best neural variant. After weighing the mean resultant values of both of the neural variants, Table 7 shows the exact mean values of MSE which is gathered on evaluating the individual test data set with respect to both of the neural variants.

Table 7. Mean MSE of FFNN and CFFNN

S.No.	ANN Variants	Resultant Mean MSE
1.	FFNN	466.3421
2.	CFFNN	454.2313

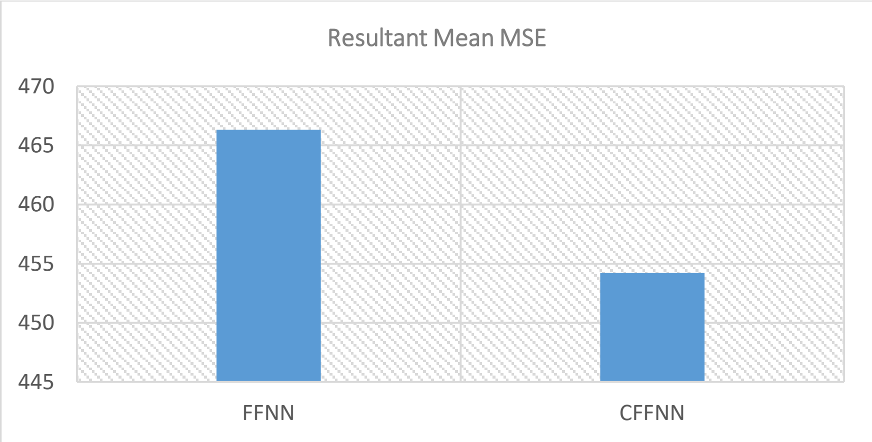

Figure 6. MMSE of FFNN and CFFNN

On the basis of MSE as shown in Figure 3. The final Mean MSE Figure 6 is calculated in order to analyze and evaluate the results. Two considerations are observed as the CFFNN has lower value of MSE, but if the parameter consideration shifts towards the correlation instead of MSE then the FFNN poses higher correlation. But the MSE have more weightage as compare to correlation because the main motive behind conducting the research is to minimize the error while evaluating the software reliability. Hence, after having a view over the values descripted in Table 7. It is observed that in case of CFFNN i.e. Cascade Feed Forward Neural Network the value of the mean error is low as compare to the other neural variant. Hence CFFNN is proven as the better variant of neural network in order to evaluate the reliability of a software in an effective and qualitative manner.

V. CONCLUSION

This study is conducted in order to evaluate the software reliability by applying heterogeneous aberrant of Neural Networks such as feed forward and cascade feed forward. The proposed work is organized in order to elite the best suitable neural network for analyzing the software reliability in a sensible way. On the basis of observed results of implementation, CFFNN is chosen as the best variant with least value of MSE which fulfill the objective behind governing the research of reliability prediction. Further the selected neural network, i.e. CFFNN will be optimized by using the swarm intelligent algorithm for getting the most relevant results. The future work will be done in order to minimize the current observed MSE for the Software reliability prediction system.

REFERENCES

[1] JarosáawSmoczek, “The Survey of Soft Computing echniques for reliability prediction”, Journal of KONES, Vol 19, Issue 3, Pp 1-8, 2012
[2] H.S. Vijaya Kumar, “On Applcation of Soft Computing Assisted Software Reliability”, IJIRAE, Vol 1, Issue 4,, Pp 32-35, 2014
[3] FurkhZeshan, “Software Architecture Reliability Models: An Overview”, IEEE, Pp 119-123, 2011
[4] Jung Hua Lo, “Data driven Model for Software Reliability Prediction”, IEEE, 2013
[5] AnkurChoudhary, “Software Reliability Prediction Modelling : A Comparison of Parametric and Non-Parametric Modeling”, IEEE, Pp 649-653, 2016
[6] Marcia F.P. Salgado, “Soft Computing Approaches in Reliability modeling and analysis of Repairable System”, IEEE, Pp 1-5,2010
[7] Nitin Kumari, “Comparison of ANNs, Fuzzy Logics and Neuro Fuzzy Integrated Approach for diagnosis of coronary heart disease”, IJCSMC, Vol 2, Issue 6, Pp 216-224, 2013
[8] Musa, J.D. et al, “Software reliability data”, IEEE Computer Society –Repository, Pp 239-249, 1979