# Software Effort Prediction Using Integrated Approach of Wavelet Transformation and Neural Network

H.S.Hota
*Department of Computer Science*
*Bilaspur University,Bilaspur (C.G.) India*

Ragini Shukla
*Department of Information Technology*
*Dr. C.V.Raman University ,Kota Bilaspur (C.G.) India*

S. K.Singhai
*Department of Electronics*
Government Engineering College, Bilaspur (C.G.) India

**Abstract- Software engineering practitioners have more concerned about accurate estimation of effort of software projects. Accurate estimates are desired but no model has proved to be successful at significantly and effectively software development effort prediction. In software management effort estimation is based on fixed mathematical formulas which are insufficient due to nonlinearity of software effort related data and on the other side Artificial Neural Network (ANN) techniques are very popular due to its capability to mapping non linear input with corresponding output. This paper, explores ANN based Radial Basis Function Neural Network (RBFNN) for prediction of software effort estimation. We propose the use of Stationary Wavelet Transformation (SWT) to de-noising the data and also used ranking based method for feature selection. The effectiveness of hybrid of wavelet transform and RBFNN model was evaluated in terms of the error measures like mean absolute error (MAE), mean absolute percentage error (MAPE), mean squared error (MSE) and root mean squared error (RMSE) for two software effort estimation related data sets: COCOMO81 and COCOMO NASA60.**

**Keywords –Artificial Neural Network (ANN), Radial Basis Function Neural Network (RBFNN), Stationary Wavelet Transformation (SWT).**

## I. INTRODUCTION

In the development of software project, software effort estimation is one of the most important activity. Before the development of the product, estimation of effort is not easy and without effort estimation we cannot progress further. Inaccurate estimation of the cost and time is the main reason which leads to software project development's failure, so the accurate estimate is the potential aspect in estimation of cost, time and effort. In the modern age of technology the complexity of software growing up day by day, because of this reason algorithmic models are not that much affective and strength of their results are not accurate to meet the clients requirements; this is the difficulty faced in software development life cycle. To estimate project delivery time, cost and manpower required to develop software, effort estimation [1] model is very helpful for project managers. In these years learning-based non-algorithmic methods are created to achieve high accuracy estimation for the software and it is also used as alternative of algorithmic techniques. This technique is based on ANN[3], fuzzy logic, decision tree and evolutionary computation that is why it has better performance and accuracy in all aspects.

Authors have used all the above techniques for development of predictive model. Ch. Satyananda Reddy et al.[5] have constructed a cost estimation model based on artificial neural networks. Two different learning algorithms back propagation and Resilient Back Propagation algorithm ( RPROP) were used to train the network to find the best learning algorithm using COCOMO dataset it was observed that the neural network model with RPROP provided significantly better cost estimations than the estimation done using COCOMO model. Jaswinder Kaur et al.[6] explored the use of soft computing techniques to build suitable model structure to utilize improved estimation of software effort for NASA software projects. The result show that ANNs are effective in effort estimation. Ali Idri,

Abdelali Zakrani et al.[7] have designed two RBF networks for software effort estimation. Each one of these networks used a different formula to compute the widths of the Gaussian kernels. These RBFNN models were trained and tested using two software projects datasets. The results show that the use of an adequate formula of width which controls the overlap between Gaussian kernels, according to the number of projects placed in the region covered by centers, improves greatly the estimates generated by RBFNN model. Manpreet Kaur et al.[8] have proposed neuro based system is able to provide good estimation capabilities. It is suggested to use of Neuro based technique to build suitable generalized type of model that can be used for the software effort estimation of all types of the projects. Maryam Molani et al.[12] have compared of the results of the two network models shows that RBF models have much higher accuracy than algorithmic models, and that the reduced model of RBF neural network has the error value less than a simple model of RBF network. Farhad Soleimanian Gharehchopogh et al.[13] have compared ANNs with other algorithmic models it has a high capacity in software cost estimation. They said that ANNs can be a useful tool for software cost estimation and it can be analyzed and estimated its various software projects with large and small dataset. Since data are very much nonlinear, so before giving it to the NN we must smooth the data with the help of smoothing technique like wavelet transformation. Very few literature are available based on this techniques, K. Vinay Kumaret al.[4] used two types of Wavelet Neural network(WNN) with Morlet function and Gaussian function as transfer function and also proposed threshold acceptance training algorithm for wavelet neural network (TAWNN). The results demonstrated that applying WNN to software effort estimation is a feasible approach to improve estimation accuracy and the proposed TAWNN is also comparable to widely used estimation models.

In this paper we have integrated Wavelet Transformation (WT) and Radial Basis Function Neural Network (RBFNN) to develop a model for software effort estimation. A special form of WT known as Stationary Wavelet Transformation (SWT) was used for data smoothing while RBFNN was used for prediction of software effort using COCOMO81 [2] and COCOMO NASA data sets. Based on various performance measures like MAE, MAPE, MSE and RMSE a model with high performance was selected and feature selection technique (FST) has been applied to reduce irrelevant features.

## II. THE PROPOSED FRAMEWORK

An integrated approach of WT and RBFNN and preprocessing stage is shown in Figure 1. This process can be viewed in different stages: Data preprocessing, Feature selection and Prediction and performance evaluation. Each of these stages are explained below:

### *A. Banchmark Dataset*

There are many benchmark data sets available in the PROMISE (Predictor models in software engineering) [9] repository site related to software effort. These data sets are publicly available and can be downloaded freely. As stated this research work utilizes two data sets namely COCOMO81 and COCOMO NASA60, these data sets are having many features related to software efforts of numeric type. Details of the data sets used for empirical experiment are shown in Table 1. As described in table, COCOMO81 data set contains 17 features out of which 15 are for the effort multipliers, along with LOC and actual development effort. This data set has 63 instances correspond to 63 projects. Another dataset is COCOMO NASA60 dataset obtained from PROMISE [9] data repository site. This dataset is comprised of 60 real software projects. Each project is described by 17 features similar to COCOMO81 data set. In order to feed these data sets to RBFNN, it is divided into two parts: training and testing with 80-20% ratio.

Table 1: Detail of data sets used for software effort estimation

| Data Set | No. of Instance | No. of Feature | Training Data Percentage | Testing Data Percentage |
|---|---|---|---|---|
| **COCOMO81** | **63** | **17** | **80** | **20** |
| **COCOMO NASA60** | **60** | **17** | **80** | **20** |

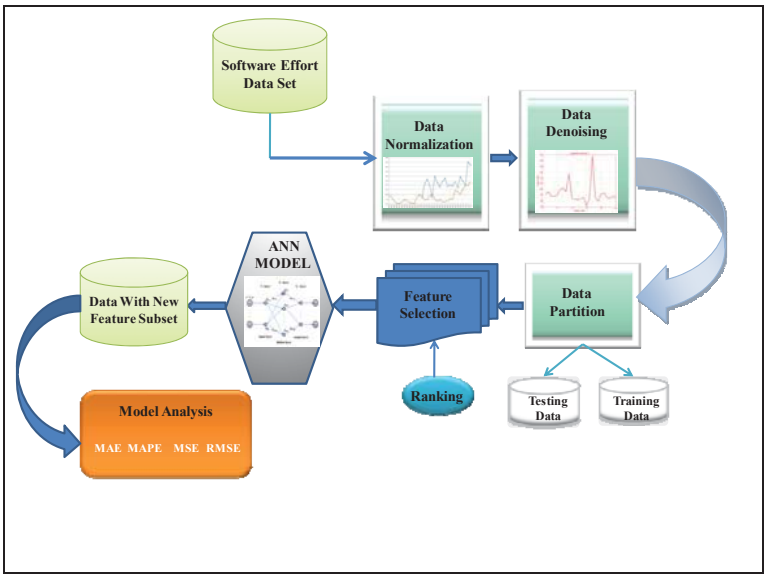


Figure 1. Flow Diagram for Software Effort Estimation

*B. Data Preprocessing*

Many systematic experiments result in a datasets corrupted with noise, either because of the data acquisition process, or because of environmental effects. A first pre-processing step in analyzing such datasets is normalization. Normalization is the process of organizing the columns (attributes) and tables (relations) of a data to minimize data redundancy. It involves decomposing a table into less redundant tables without losing information. In this research work using MATLAB software we have normalized data between 0 and 1 using the following formula:

$$normc = (A-min(A))/(max(A)-min(A)) \qquad (1)$$

Where A is a matrix and normc stands for columns used to normalize, min(A) and max(A) are respectively minimum value of A and maximum value respectively.

Further normalized data are used to de-noise using wavelet transformation. Wavelets are a class of functions used to localize a given function in both space and scaling. A wavelet is a waveform of effectively limited duration that has an average value of zero. Wavelet analysis is capable of revealing aspects of data that other signal analysis technique like trends, breakdown points, discontinuities in higher derivatives, and self-similarity. Furthermore, because it affords a different view of data than those presented by traditional techniques, wavelet analysis can often compress or de-noise a signal without appreciable degradation. The important characteristic of wavelets is that they can serve as deterministic or non-deterministic basis for generation and analysis of the most natural signals to provide better time-frequency representation, which is not possible with waves using conventional Fourier analysis [4].

De-noising is estimating the unknown signal of interest from the available noisy data. There are several different approaches to de-noise signals and images. Normalize data are used to de-noise using SWT before giving it to RBFNN model. SWT performs a multilevel 1-D stationary wavelet decomposition using either a specific orthogonal wavelet or specific orthogonal wavelet decomposition filters. Using MATLAB, data can be de-noised through the following formula:

$$x(n) = s(n) + \sigma g(n)\ , \qquad n=1 \text{ to } N \qquad (2)$$

where, s(n) is an N point original signal, x(n) is a noisy signal corrupted by (0,1) additive white Gaussian noise g(n) with a spread of σ as standard deviation.

*C. Feature Selection*

Feature selection is an optimization process which reduces the dimensionality of data by selecting only a subset of measured features to create a model. Selection criteria usually involve the minimization of a specific measure of predictive error for models fit to different subset.

The existing feature selection methods depending on feature selection criterion in two categories one is open-loop methods and another is closed-loop methods [11].In this paper a ranking based feature selection is used to find out a new feature subset , in which each feature is ranked and a feature with lower ranked is reduced one by one.

### *D. Radial Basis Funtion (RBF) Neural Network*

Radial basis function network is an artificial neural network that uses radial basis function as an activation function. The output of the network is a linear combination of radial basis functions of inputs and neuron parameters. RBF network is a popular alternative to the multilayer perceptron network, and has comparatively better performance when there is a large number of training samples. It is similar to the multilayer perceptron network in terms of the number of layers and the architecture. Ali Idari et al.[7] have examined the impact of the radial basis widths on the accuracy of RBF network in the context of software effort estimation.

RBF neural networks have the advantage of not suffering from local minima in the same way as Multi-Layer Perceptions. RBFNN can solve many complex prediction problems with quite satisfying performance. Maryam Molani et al.[12] have also presented a new method which is based on RBFNN that provide greater accuracy than algorithmic models.

Figure 6 shows architecture of a RBFNN[10] with software effort data feature as input and software effort as output. RBFNN is set for default parameters.

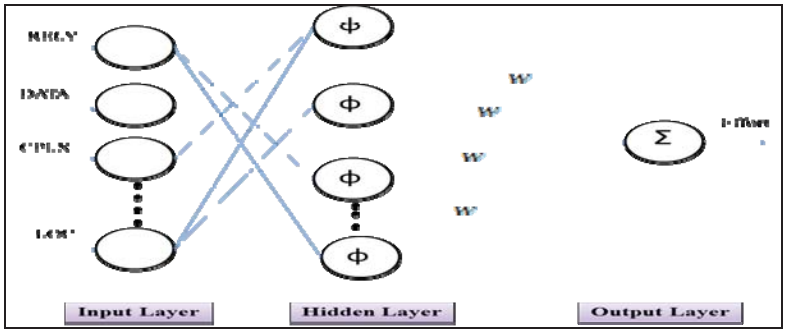


Figure 2: Radial Basis Function Network architecture for Software Effort Estimation

## III. PERFORMANCE MEASURE

Performance of software effort estimation can be evaluated by using some very well-known statistical measures: Mean Absolute Error **(**MAE), Mean Absolute Percentage Error **(**MAPE), Mean Square Error **(**MSE) and Root Mean Square Error (RMSE). Several measures used for performance evaluation of RBFNN model are described below[14]:

MAE: Is an average of the absolute errors which measures of how far the estimates are from actual values. It could be applied to any two pairs of numbers, where one set is "actual" and the other is an estimate prediction.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |Pi - Ai| \qquad (3)$$

MAPE: Used for accuracy estimation based on following formulae.

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{|Pi - Ai|}{Ai} \qquad (4)$$

MSE: Gives high importance to large errors because the errors are squared before they are averaged. Thus MSE is used when large errors are undesirable.

$$MSE = \left[ \frac{1}{n} \sum_{i=1}^{n} (Pi - Ai)^2 \right] \qquad (5)$$

RMSE: Is a frequently used measure of differences between values predicted by a model or estimator and the values actually observed from the thing being modeled or estimated. It is just the square root of the mean square error as shown in equation given below:

$$RMSE=\left[\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Pi-Ai)^2}\right] \quad (6)$$

Where *Pi*=Predicted value for data point *i*, *Ai*=Actual value for data point *i*, *n*=Total number of data points.

## IV. Experiment work and Result

This research work is carried out using the WEKA open source and CLEMENTINE data mining software which provides interactive way to develop and evaluate model.

COCOMO81 and COCOMO NASA60 data were preprocessed using normalization and de-noising using SWT, so that there will be equal influence of each feature during training process.

Integrated approach of SWT and RBFNN were trained and tested with both data sets. A fixed ratio of training and testing samples were chosen randomly, training data are used to construct prediction model while testing data are unseen data and used to check the efficiency of model. In both the cases predicted values are used to calculate MAE, MAPE, MSE and RMSE using equations 3, 4, 5 and 6 respectively and presented in Table 2 and 3 for COCOMO81 and COCOMO NASA60 data set respectively after applying ranking based FST. These tables show that MAPE is reducing while number of features are reducing, for COCOMO81 data set, MAPE in case of 3 features are 2.3064 and 6.2398 at training and testing stages respectively while for COCOMO NASA60 data set these are 21.9704 and 20.6161 at training and testing stages respectively. The same are shown in Figure 3 and 4 for both training and testing data.

Table: 2 Experimental Result of COCOMO81 Data set after applying FST

| No. of Feature | Testing Stage | | | | Training Stage | | | |
|---|---|---|---|---|---|---|---|---|
| | MAE | MAPE | MSE | RMSE | MAE | MAPE | MSE | RMSE |
| 16 | 0.0137 | 25.8132 | 0.0002 | 0.0160 | 0.00487 | 8.4223 | 6.83E-05 | 0.0082 |
| 15 | 0.0145 | 23.9099 | 0.0003 | 0.0187 | 0.0039 | 8.6520 | 2.2E-05 | 0.0046 |
| 14 | 0.0078 | 21.0643 | 0.0001 | 0.0100 | 0.0027 | 6.238 | 9.86E-06 | 0.0031 |
| 13 | 0.0099 | 19.3428 | 0.0001 | 0.0118 | 0.0027 | 6.8124 | 1.22E-05 | 0.0034 |
| 12 | 0.0090 | 17.1506 | 0.0001 | 0.0105 | 0.0024 | 4.9069 | 1.37E-05 | 0.0036 |
| 11 | 0.0101 | 17.9423 | 0.0001 | 0.0134 | 0.0032 | 6.943 | 1.47E-05 | 0.0038 |
| 10 | 0.0084 | 16.6455 | 8.56E-05 | 0.0092 | 0.0031 | 5.815 | 1.79E-05 | 0.0042 |
| 09 | 0.0082 | 14.7405 | 0.0001 | 0.0104 | 0.0026 | 4.409 | 1.08E-05 | 0.0032 |
| 08 | 0.0048 | 13.5146 | 4.46E-05 | 0.0066 | 0.0049 | 8.839 | 4.56E-05 | 0.0067 |
| 07 | 0.0067 | 12.8928 | 7.56E-05 | 0.0086 | 0.0044 | 10.087 | 3.03E-05 | 0.0055 |
| 06 | 0.0036 | 8.6016 | 1.99E-05 | 0.0044 | 0.0045 | 7.00975 | 3.25E-05 | 0.0056 |
| 05 | 0.0038 | 8.0902 | 2.89E-05 | 0.0053 | 0.0023 | 5.1575 | 1.05E-05 | 0.0032 |
| 04 | 0.0034 | 8.0449 | 2.04E-05 | 0.0045 | 0.0017 | 3.62634 | 7.31E-06 | 0.0027 |
| 03 | 0.00286 | 6.2398 | 1.18E-05 | 0.0034 | 0.0012 | 2.3064 | 2.97E-06 | 0.0017 |

Table: 3 Experimental Result of COCOMO NASA60 Data set after applying FST

| No. of Feature | Testing Stage | | | | Training Stage | | | |
|---|---|---|---|---|---|---|---|---|
| | MAE | MAPE | MSE | RMSE | MAE | MAPE | MSE | RMSE |
| 16 | 0.039434 | 32.9360 | 0.001928 | 0.04311 | 0.033804 | 22.7849 | 0.001522 | 0.039012 |
| 15 | 0.039014 | 30.9360 | 0.001898 | 0.0435 | 0.032093 | 22.5651 | 0.001738 | 0.04169 |
| 14 | 0.037734 | 28.9360 | 0.001854 | 0.0476 | 0.03283 | 22.7422 | 0.001886 | 0.043426 |
| 13 | 0.036414 | 27.9360 | 0.001828 | 0.042758 | 0.031944 | 22.3557 | 0.00182 | 0.042657 |
| 12 | 0.03437 | 26.2411 | 0.00170 | 0.041244 | 0.033246 | 22.76347 | 0.001988 | 0.044588 |
| 11 | 0.031734 | 23.01957 | 0.001690 | 0.041111 | 0.033047 | 22.35714 | 0.002033 | 0.045091 |
| 10 | 0.032822 | 22.93183 | 0.00194 | 0.044156 | 0.033003 | 22.5885 | 0.002016 | 0.044896 |
| 09 | 0.03158 | 21.8320 | 0.00187 | 0.04330 | 0.033018 | 22.5123 | 0.001982 | 0.044523 |
| 08 | 0.03121 | 21.22121 | 0.001915 | 0.0437696 | 0.032884 | 22.4245 | 0.001968 | 0.044358 |
| 07 | 0.03084 | 21.0843 | 0.00183 | 0.04287 | 0.032941 | 22.3735 | 0.001998 | 0.044697 |
| 06 | 0.03109 | 21.2408 | 0.00187 | 0.04329 | 0.032864 | 22.3465 | 0.001976 | 0.044449 |
| 05 | 0.03095 | 20.9787 | 0.00188 | 0.04339 | 0.032716 | 22.1892 | 0.001972 | 0.044406 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 04 | 0.03061 | 20.8186 | 0.00183 | 0.04288 | 0.032466 | 21.9713 | 0.001952 | 0.044176 |
| 03 | 0.03051 | 20.61613 | 0.00184 | 0.04300 | 0.032406 | 21.9704 | 0.001937 | 0.044016 |

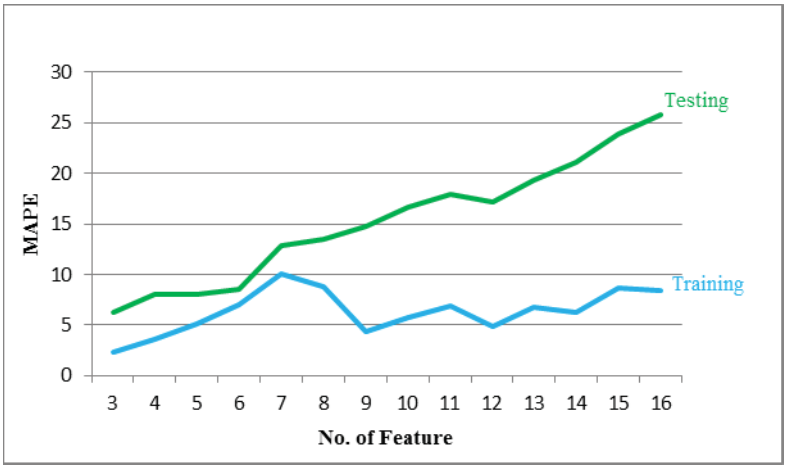


Figure 3: Performance of integration of SWT and RBFNN model according to selection of feature (COCOMO81 Data set)

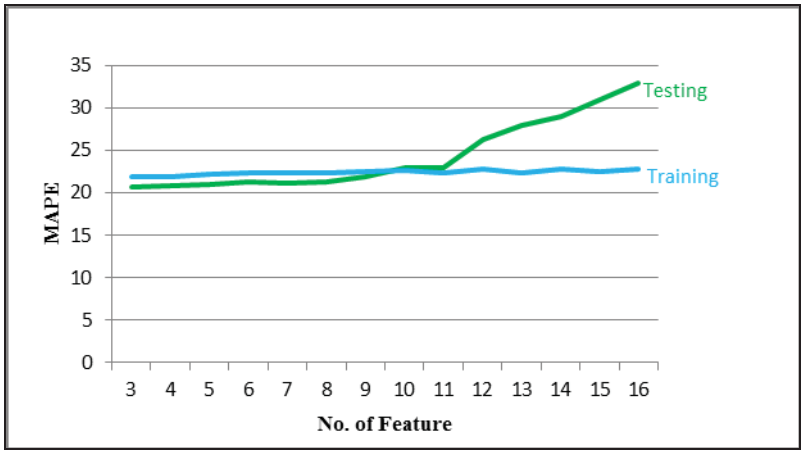


Figure 4: Performance of integration of SWT and RBFNN model according to selection of feature (COCOMO NASA60 Data set)

## V.CONCLUSION

Software effort estimation is complex due to non linear nature of data and important due to manage manpower and timely delivery of software product. This research work is to improve software effort estimation by integrating SWT and RBFNN where SWT is used to remove noise from software effort data set while RBFNN is used for actual software effort prediction. Feature selection technique is also used to reduce features from data sets. Experimental results through WEKA shown that integrated approach of SWT and RBFNN is performing better with least MAPE with 3 features only.

## REFERENCES

[1] Jalote, P. "An Integrated Approach to Software Engineering", *Narosa Publishing House, 2005.*

[2] Pressman, R. S. "Software Engineering", *McGraw Hill Higher Education, 2005.*

[3] Zurada, J. M. "Introduction to Artificial Neural Systems", *Jaico Publishing House, Mumbai 2006.*

[4] Kumar, K.V., V. Ravi , Mahil Carr, and N. Raj Kiran, "Software development cost estimation using wavelet neural networks" *Science Direct The Journal of Systems and Software 81,2008 pp.1853–1867.*

[5] Reddy, Ch.S. and KVSVN Raju, "An Optimal Neural Network Model for Software Effort Estimation", *International Journal of Software Engineering, IJSE Vol.3 No.1 January 2010,pp.63-78.*

[6] Kaur, J. , Satwinder Singh, Dr. Karanjeet Singh Kahlon and Pourush Bassi "NeuralNetwork-A Novel Technique for Software Effort Estimation" *International Journal of Computer Theory and Engineering, Vol.2 No.1 February 2010,pp.17-19.*

[7] Idri1, A., Abdelali Zakrani and Azeddine Zahi, "Design of Radial Basis Function Neural Networks for Software Effort Estimation" *IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 4, No 3, July 2010,pp.11-17.*

[8] Kaur, M. and Mohit Verma "Computing MMRE and RMSSE of Software Efforts using Neural Network Based Approaches and Comparing with Traditional Models of Estimation", *International Journal of Computer Science and technology Vol. 3, Issue 1, Jan. - March 2012, pp.316-318.*

[9] Menzies,T., Caglayan,B., Kocaguneli, E., Krall,J., Peters, F. and Turhan , B. ,"The PROMISE Repository of empirical software engineering data" http://promisedata.goolecode.com, *West Virginia University, Department of Computer Science, 2012.*

[10] Jang, J. S. R., Chuen-Tsai Sun and Eiji Mizutani, "Neuro-Fuzzy and Soft Computing"*PHI Learning Private Limited,2012.*

[11] Hota, H.S. , Akhilesh Kumar Shrivas and S.K.Singhai, " Tuned Artificial Neural Network Model for E-mail Data Classification with Feature Selection" *International Journal of Computer Applications (0975 – 8887) Volume 67– No.25, April 2013.*

[12] Molani,M., Ali Ghaffari and Ahmad Jafarian, " A New Approach to Software Project Cost Estimation using a Hybrid Model of Radial Basis Function Neural Network and Genetic Algorithm", *Indian Journal of Science and Technology, Vol 7(6), 838–843, June 2014.*

[13] Gharehchopogh, F.S. and Awat Maroufi, **"**Approach of software cost estimation with hybrid of imperialist competitive and artificial neural network algorithms", *Journal of Scientific Research and Development 50-57, 2014.*

[14] Hota,H. S. , Ragini Shukla and S.K. Singhai, "Predicting Software Development Effort Using Tuned Artificial Neural Network", *Springer India 2015, Computational Intelligence in Data Mining - Volume 3,Smart Innovation, Systems and Technologies 33, pp.195-21.*