International Journal of Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-4, Issue-7)
Research Article
July 2015

Performance Evaluation of Software Development Effort Estimation Using Neuro-Fuzzy Model

Vidisha Agrawal, Vishal Shrivastava
Department of Computer Science and Engineering,
ARYA College of Engineering and IT, Jaipur, Rajasthan, India

Abstract—

One of the greatest challenges for software developers is forecasting the development effort for a software system for the last decades. The capability to provide a good estimation on software development efforts is necessitated by the project managers. Software effort estimation models divided into two main categories: algorithmic and non-algorithmic. Developers should be able to achieve practices containing effort estimation based on their own programs. New paradigms as Fuzzy Logic may offer an alternative for software effort estimation. In this paper the estimated time for Neuro fuzzy model created for three membership functions is compared with the existing neural network models. The Neuro fuzzy model for Gaussian, triangular and trapezoidal membership function is compared with the neural network models. For this experiment Lopez Martin dataset was used with 41 modules. We compared the three different membership function models (Gaussian MF, Triangular MF and Trapezoidal MF) with the existing neural network models on the basis of five different parameters. Those are Magnitude Relative Error (MRE), Mean Magnitude Relative Error (MMRE), Prediction (Pred), Balanced Relative Error (BRE), Relative standard deviation (RSD) and Root Mean Squared Error (RMSE). Finally it is observed from the comparison that Neuro Fuzzy model using Trapezoidal membership function gives better results than all other models. It is also observed that Trapezoidal MF gives better results for all the five parameters.

Keywords—Software development Effort Estimation, MMRE, Pred, BRE, RSD, RMSE, GMF, Tri MF and Trap MF,

I. INTRODUCTION

Software effort estimation is a necessary feature that guides and supports the planning of software projects. Software effort estimation refers to the predictions of the likely amount of effort, time, and staffing levels required to build a software system. An extremely helpful form of effort prediction is the one made at an early stage during a project, when the costing of the project is proposed for approval. Effort estimation algorithms [1] in general offers estimates of the number of work months required to produce a given amount of code. Age old approaches for software projects effort prediction such as the use of mathematical formulae derived from past data, or the use of expert's judgments, lack in terms of efficiency and robustness in their results. Software effort estimation guides the prediction of the likely amount of effort, time, and staffing levels required to build a software system at an early stage during a project. However, estimates at the preliminary stages of the project are the most difficult to obtain because the primary source to estimate the costing comes from the requirement specification documents [2]. According to Royce [3], a good and effective software cost estimate should fulfil the different types of properties. One is conceptualized and supported by the software project manager and the development team and another is acknowledged by all the stake holders as achievable. The underlying cost model is well-defined on a credible basis. It is based on the careful analysis of the relevant historical project data (similar processes, similar technologies, similar environments, similar people and similar requirements).It is defined in sufficient detail such that its possible key risk areas are clearly understood and probability of success is objectively assessed.

1.1 Artificial Neural Networks: Artificial neural networks (ANN) have been developed as generalizations of mathematical models of biological nervous systems. A first wave of interest in neural networks (also known as connectionist models or parallel distributed processing) emerged after the introduction of simplified neurons by McCulloch and Pitts (1943).The basic processing elements of neural networks are called artificial neurons, or simply neurons or nodes. In a simplified mathematical model of the neuron, the effects of the synapses are represented by connection weights that modulate the effect of the associated input signals, and the nonlinear characteristic exhibited by neurons is represented by a transfer function. The neuron impulse is then computed as the weighted sum of the input signals, transformed by the transfer function. The learning capability of an artificial neuron is achieved by adjusting the weights in accordance to the chosen learning algorithm.

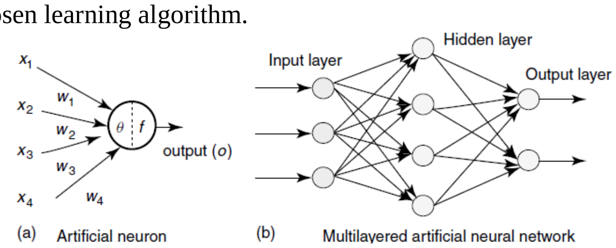

Figure 1: Architecture of an Artificial Neuron and a Multilayered Neural Network

A typical artificial neuron and the modeling of a multilayered neural network are illustrated in Figure 2. Referring to Figure 2, the signal flow from inputs x1. . . xn is considered to be unidirectional, which are indicated by arrows, as is a neuron's output signal flow (O). The neuron output signal O is given by the following relationship:

$$O = f(net) = f(\sum_{j=1}^{n} w_j\ x_j \qquad (1)$$

where wj is the weight vector, and the function f(net) is referred to as an activation (transfer) function. The variable net is defined as a scalar product of the weight and input vectors,

$$net = w^T x = w_1 x_1 + \cdots + w_n x_n \qquad (2)$$

where T is the transpose of a matrix, and, in the simplest case, the output value O is computed as

$$O = f(net) = \begin{cases} 1\ if w^T x \geq \theta \\ 0\ otherwise \end{cases} \qquad (3)$$

where θ is called the threshold level; and this type of node is called a linear threshold unit.
In more than one ways, the human mind is the role of model for soft computing techniques for example, the ability to solve problems expressed in vague terms, or solving problem without making use of explicit solution steps. Arriving at a solution through an evolutionary process is commonplace in nature.
The predominant SC methodologies found in current intelligent systems are:

- Artificial neural networks (ANN)
- Fuzzy systems
- Neurofuzzy systems

The paper is organized as follows. In Section 2, we discuss the related work; Section 3 presents our research methodology of soft computing-based prediction systems. Section 4 presents our results and discussions of our various experiments and Section 5concludes discussions to realize the framework and points out possible directions for future research.

II. RELATED WORK

In this paper, I have made a review on my topic Adaptive Neuro Fuzzy model for software time estimation by reading different kinds of papers and analyzing different techniques which are being used in these papers published by authors which are discussed as follows:
Nassif et al. [7] "Towards an early software estimation using log-linear regression and a multilayer perceptron model" (2013) have proposed a novel log-linear regression model based on the use case point model (UCP) to calculate the software effort based on use case diagrams. A fuzzy logic approach is used to calibrate the productivity factor in the regression model. Moreover, a multilayer perceptron (MLP) neural network model was developed to predict software effort based on the software size and team productivity. The proposed approach outperforms the original UCP model. Furthermore, a comparison between the MLP and log-linear regression models was conducted based on the size of the projects. Results demonstrate that the MLP model can surpass the regression model when small projects are used, but the log-linear regression model gives better results when estimating larger projects.
Shina Dhingra, et al. [8],is an attempt to experiment fourty one dataset for time estimation using three different membership function. The results were analyzed using three different criterionsMMRE, BRE and Pred.It is observed that Neuro Fuzzy model using Trapezoidal membership function gives better results than all other models. It is also observed that Trapezoidal MF gives better results for all the three parameters.
Divya Kashyap, Ashish Tripathi,Prof. A.K.Mishra [9] discussed Neuro-Fuzzy model. Neuro-Fuzzy models are the combination of Artificial Neural Network and Fuzzy Logic. Artificial Neural Network has the ability to learn from previous data. It model complex relationships between both independent variables (cost drivers) and dependent variables (effort). Fuzzy logic simulates the human behavior and reasoning. Fuzzy logic is basically used in situation where decision making is very difficult and conditions are not clearly defined. Facts that may be dismissed are focused in this technique.
Urvashi Rahul Saxena, S.P.Singh [10] in this paper they explore Neuro-fuzzy techniques to design a suitable model to utilize improved estimation of software effort for NASA software projects. Comparative Analysis between Neuro-fuzzy model and the traditional software model(s) such as Halstead, Walston- Felix, Bailey-Basili and Doty models is provided. The evaluation criteria are based upon MMRE (Mean Magnitude of Relative Error) and RMSE (Root mean Square Error). Integration of neural networks, fuzzy logic and algorithmic models into one scheme has resulted in providing robustness to imprecise and uncertain inputs.
Vachik S. Dave, Kamlesh Dutta [11] they have compared Neural Network models and regression model for software Development effort estimation. The comparison reveals that the Neural Network (NN) is better for effort prediction compared to regression analysis model. Further, we have compared two Neural Network models- Feed-Forward Neural Network (FFNN) and Radial Basis Neural Network (RBNN). The evaluation of the models is based on Mean Magnitude Relative Error (MMRE).

III. RESEARCH METHODOLOGY

Neuro-fuzzy was proposed by J. S. R. Jang. Adaptive Neuro fuzzyis a kind of neural network that is based on Takagi–Sugeno fuzzy inference system. Since it integrates both neural networks and fuzzy logic principles, it has potential to capture the benefits of both in a single framework. Its inference system corresponds to a set of fuzzy IF–THEN rules that have learning capability to approximate nonlinear functions.

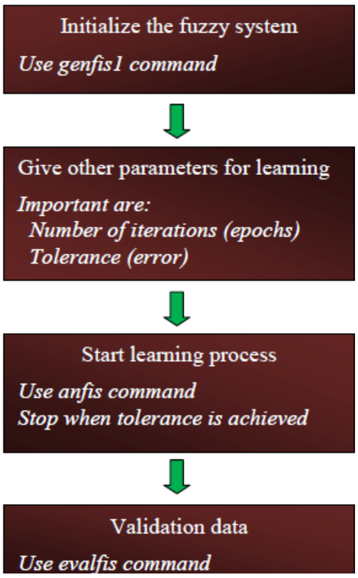

Figure2: Basic Flow Diagram of Computations in ANFIS

The main goal of this paper is to evaluate software development time using an adaptive Neuro fuzzy approach. In this paper an Adaptive Neuro Fuzzy Inference System (ANFIS) tool is used. The network is trained by using learning algorithm i.e. Hybrid Approach (combination of back propagation and least mean square algorithm). This methodology consists of four steps: 1) Loading of Training Data and 2) Generating Fuzzy Inference System 3) Training of ANFIS 4) Development Time Estimation.

START
Step I; Determine the inputs of the model;
Step II; Generate ANFIS model;
Step III; Evaluate the value of Development Time;
Step IV; Evaluate the Value of MRE from result obtained by step III;
Step V; Evaluate the Value of MMRE and PRED from result obtained by step IV;
Step VI; Evaluate the Value of BRE from result obtained by step III;
Step VII; Evaluate the Value of RSD from result obtained by step III, V and VI;
Step VIII; Evaluate the Value of RMSSE from result obtained by VII;
END

3.1 Database used:

For this experiment Lopez Martin dataset was used with 41 modules. Lopez Martin dataset is divided into two sets: Training dataset and testing dataset. Training dataset includes 25 modules and testing dataset includes 11 modules. Training dataset was chosen randomly from 41 modules with which we can achieve the best results. The three different membership functions were trained using the same 25 modules.

3.2 Performance Evaluation of Adaptive Neuro Fuzzy Model:

The development time is estimated for Neuro Fuzzy model using three different membership functions. The estimated time for three different MFs is compared among themselves and the best model is compared with the existing neural network models. The development time is estimated using evalfis command. Three inputs are given for evaluating development time i.e. McCabe complexity (MC), Dhama coupling (DC), Line of Code (LOC). These three inputs are taken from Lopez Martin data set.

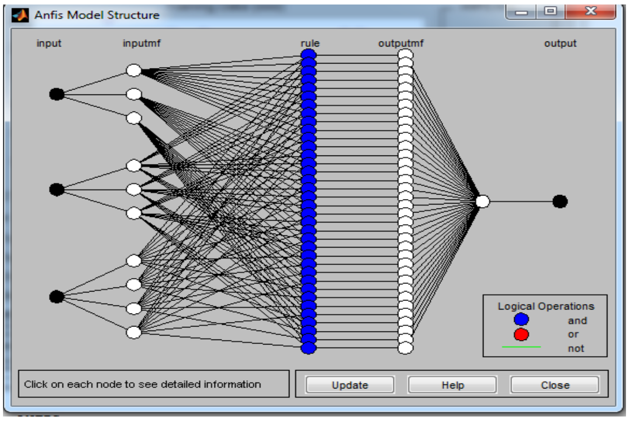

Figure 3: ANFIS Model structure

3.3 Performance Evaluation Metrics:

The following evaluation metrics are adapted to assess and evaluate the performance of the time estimation models.

(1) Magnitude of Relative Error (MRE)

$$MRE = \frac{|Actual\ \ Time - Estimated\ \ Time\,|}{Actual\ \ Time} \times 100 \qquad \text{Eq.(4)}$$

(2) Mean Magnitude of Relative Error (MMRE)

$$MMRE = \frac{1}{n}\sum_{i=1}^{i=n} \frac{|\text{Actual Time} - \text{Predicted Time}\,|}{\text{Actual Time}} \qquad \text{Eq.(5)}$$

The MMRE calculates the mean for the sum of the MRE of n projects. Specifically, it is used to evaluate the prediction performance of an estimation model.

(3) Prediction Level (PRED)

$$PRED(l) = \frac{k}{n} \times 100 \qquad \text{Eq.(6)}$$

where l is the maximum MRE of a selected range, n is the total number of projects, and k is number of projectsin a set of n projects whose MRE <= l. PRED calculates the ratio of projects' MREs that falls into the selected range (l) out of the total projects. (e.g. n = 100, k =80, where L= MRE <= 30%: PRED(30%)=80/100=80%).

(4) Balanced Relative Error (BRE)

$$BRE(\%) = \frac{|EstimatedTime\ \ - ActualTime\ \,|}{\min(T,T')} \times 100 \qquad \text{Eq.(7)}$$

Where T= estimated time and T'=actual time

3.4 Flowchart for Proposed Approach: The Proposed approach can be shown through a particular flowchart which shows various steps that in which order our work will be carried out.

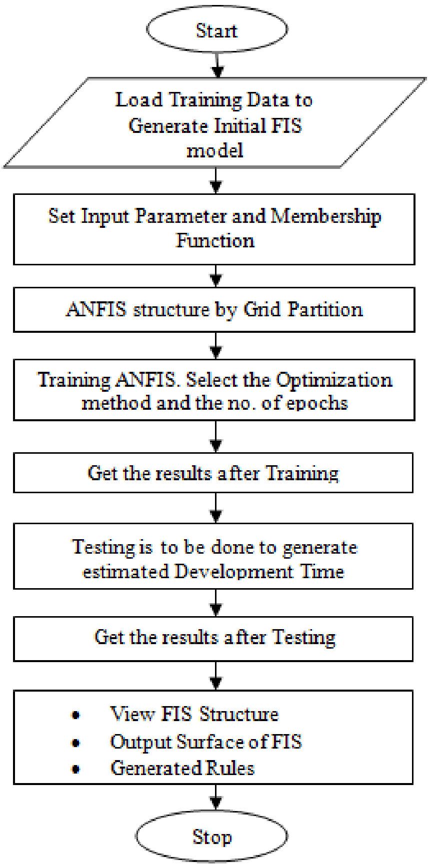

Figure 4: Flow diagram for Proposed ANFIS Model

IV. RESULTS ANALYSIS AND DISCUSSION

In below tables the development time of the three membership functions is compared with each other and the various parameters are calculated for each membership function and the three membership functions are compared with the neural network models. The estimated time for Neuro fuzzy model created for three membership functions is compared with the existing neural network models. The Neuro fuzzy model for Gaussian, triangular and trapezoidal membership function is compared with the neural network models.Below Table 1 summarizes the development time estimates as obtained for Neuro fuzzy model using different membership functions.

Table 1.The DT obtained by different MF using ANFIS model

Project No.	Actual DT	Estimated DT using Gaussian MF for Neuro fuzzy	Estimated DT using Triangular MF using Neuro fuzzy	Estimated DT using Trapezoidal MF using Neuro fuzzy
1	13	1.8101	5.7707	17.6065
2	13	1.8654	0	0
41	18	1.5881	0.0000	0
35	21	0.5063	0.0004	0
39	24	21.0000	21.0000	21.0000
22	18	9.1909	18.5000	18.5000
9	16	20.9734	15.0056	19.5725
35	21	11.4144	9.6954	9.6857
4	15	24.0000	24.0000	24.0000
25	20	24.0000	24.0000	24.0000
10	18	18.0000	18.0000	18.0000

We compared the three different membership function models (Gaussian MF, Triangular MF and Trapezoidal MF) with the existing neural network models on the basis of four different parameters. Those are Magnitude Relative Error (MRE), Mean Magnitude Relative Error (MMRE), Prediction (Pred), Balanced Relative Error (BRE), Relative standard deviation (RSD) and Root Mean Squared Error (RMSE).

4.1 Comparison between Gaussian MF, Triangular MF, Trapezoidal MF, FFBP NN, Cascaded FFBP NN and Layer Recurrent NN based on various parameters:

Table 2 represents the comparison between different models on the basis of various parameters and Figure 6 represents the graph of comparison between different models. Above Table1 represents the development time obtained by different models and below table 2 depicts the comparison between different models.

Table2. Comparison of different models using Parameters

Models	RSD (%)	RMSE (%)	MMRE (%)	Pred (25%)	BRE (%)
FFBP NN	5.6	17.4475	32.21	55	.6316
Cascaded FFBP NN	10.9	123.457	38.99	45	.7287
Layer Recurrent NN	7.5	25.0224	39.17	45	.8196
ANFIS Model using GMF	4.2	28.0456	53.626	45.4545	0.584397
ANFIS Model usingTri MF	6.2	25.0556	38.8558	45.4545	3.16369
ANFIS Model using Trap MF	2.7	22.0237	19.9797	72.7273	0.25146

Below figure (5) depicts the results obtained by different modelsGaussian MF, Traiangular MF, Trapezoidal MF graphically.

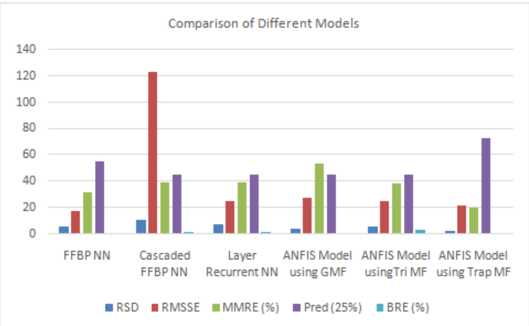

Figure 5: Comparison of different models

Above Figure 5 shows the comparison of 6 different models like FFBP Neural network, Cascaded neural network, Recurrent NN, Gaussian MF for ANFIS model. Triangular MF for ANFIS model and Trapezoidal MF for ANFIS model. The above graph shows that the result for Trapezoidal MF for ANFIS model is best among all other models.

A model which gives lower BRE is better than that which gives higher BRE. A model which gives higher Pred (n) is better than that which gives lower Pred (n). A model which gives lower MMRE is better than that which gives higher MMRE. Hence from Table 2 we can observe that ANFIS model using Trapezoidal Membership Function (Trap MF) is better. Figure 5 shows the comparison of different models based on MMRE, Pred (0.25), BRE and we can see that the Pred (0.25) for Trapezoidal MF is highest among all the models, so it can be concluded that Neuro Fuzzy model for Trapezoidal MF is the better model for Time estimation among all other models.

V. CONCLUSIONS

In this work, Adaptive Neuro Fuzzy Inference model is proposed and three membership functions i.e. Gaussian MF, Triangular MF and Trapezoidal MF are used to predict the future values. The development time is calculated for the three membership functions using the Lopez martin data set. The development time is calculated using evalfis command. The Neuro fuzzy model for three membership functions is compared with various neural network models on the basis of five parameters like Magnitude Relative error, Mean Magnitude Relative error, Balanced Relative error, Prediction, Root Mean Squared Error and Relative Standard Deviation.

It is observed from the comparison that Neuro Fuzzy model using Trapezoidal membership function gives better results than all other models. It is also observed that Trapezoidal MF gives better results for all the five parameters. The model with less BRE is better than the model with more BRE. The model with more prediction is better model than the model with less prediction.

REFERENCES

[1] S.Basha and P.Dhavachelvan, "Analysis of Empirical Software Effort Estimation Models" in International Journal of Computer Science and Information Security (IJCSIS), Vol. 7, 2010.
[2] B.Hughes and M.Cotterell, "Software Project Management", Tata McGraw-Hill, 2006.
[3] T.Gruschke, "Empirical Studies of Software Cost Estimation: Training of EffortEstimation Uncertainty Assessment Skills", 11th IEEE International Software Metrics Symposium, IEEE, 2005.
[4] Mrinal Kanti Ghose, Roheet Bhatnagar and Vandana Bhattacharjee, "Comparing Some Neural Network Models for Software Development Effort Prediction", IEEE conference , 2011.
[5] J. Jantzen, "Neuro- Fuzzy modeling", Reort no 98-H-874, 1998.
[6] Xishi Huang et al., "A Neuro-Fuzzy Model for Software Cost Estimation", Proceedings of the Third International Conference on Quality Software (QSIC'03), IEEE, 2003.
[7] Nassif Bou Ali, Ho Danny, Capretz Fernando Luiz, "Towards An Early Software Estimation Using Log-Linear Regression and A Multilayer Perceptron Model", 2013, pp. 144-160.
[8] Shina Dhingra, Palvinder Singh Mann, "Design and Implementation of Neuro Fuzzy model for Software Development Time Estimation", International Journal of Computer Applications (0975 – 8887), Vol. 86, pp. 8-10, January 2014.
[9] Divya Kashyap, Ashish Tripathi,Prof. A.K.Mishra[29],"Software Development Effort and Cost Estimation: Neuro-Fuzzy Model",IOSR Journal of Computer Engineering(IOSRJCE), Vol. 2,pp. 12-14, July-Aug. 2012.
[10] Urvashi Rahul Saxena, S.P.Singh, "Software Effort Estimation Using Neuro-Fuzzy Approach", CSI 6th International Conference, pp.1-6, Sept.2012.
[11] Vachik S. Dave, Kamlesh Dutta, "Neural Network based Software Effort Estimation & Evaluation criterion MMRE", International Conference on Computer & Communication Technology (ICCCT), pp.347-351, 2011.
[12] N. Sharma1, A. Bajpai and M. R. Litoriya, "A Comparison of Software Cost Estimation Methods: A Survey", The International Journal of Computer Science and Applications (TIJCSA), Vol.1, no. 3, ISSN – 2278 – 1080, May 2012.
[13] T. R. Benala, S. Dehuri and R. Mall, "Computational Intelligence in Software Cost Estimation: An Emerging Paradigm", ACM SIGSOFT Software Engineering Notes Page, Vol. 37, no.3, 2012.
[14] J.S. Pahariya, V. Ravi and M. Carr, "Software Cost Estimation using Computational Intelligence Techniques", World Congress on Nature & Biologically Inspired Computing (NaBIC 2009) IEEE, 2009.
[15] C.J. Burgess, M. Lefley, " Can genetic programming improve software effort estimation? A comparative evaluation", Information and Software Technology, pp.863–873, 2001.
[16] Anish Mittal, Kamal Parkash and Harish Mittal "Software Cost Estimation Using Fuzzy Logic", ACM SIGSOFT Software Engineering Notes Page , Vol 35 no. 1, November 2010.
[17] Harsh Kumar Verma, Vishal Sharma, "Handling Imprecision in Inputs using Fuzzy Logicto Predict Effort in Software Development" , IEEE conference, 2010.
[18] G. D. Boetticher, "An assessment of metric contribution in the construction of a neural network-based effort estimator", Proceedings of Second International Workshop on Soft Computing Applied to Software Engineering, 2001.
[19] A.R.Venkatachalam, "Software cost estimation using artificial neural networks", in: Proceedings of the International Joint Conference on Neural Networks, pp. 987–990, 1993.
[20] J. Ryder, "Fuzzy modeling of software effort prediction", Proceedings of IEEE Information Technology Conference, Syracuse, NY, 1998.
[21] Allidri et al., "COCOMO Cost Model Using Fuzzy Logic", 7th International Conference On RFuzzy Theory & Technology Atlantic City, New Jersey, February27-March3, 2000.

[22] Heejum Park and Seung Baek, "An Empirical validation of a neural network model for software effort estimation", Expert System with Application,Elsevier, pp.929-937, 2007.

[23] Dinesh C.S. Bisht and Ashok Jangid, "Discharge Modelling using Adaptive Neuro-Fuzzy Inference System", International Journal of Advanced Science and Technology,Vol. 31,pp.99-114, June 2011.

[24] Mrinal Kanti Ghose, Roheet Bhatnagar and Vandana Bhattacharjee, "Software Development Effort Estimatiom- Neural Network Vs. Regression Modelling Approach", International Journal of Engineering and Technology, Vol. 2(7), pp. 2950-2956, 2010.

[25] Divya Kashyap, Ashish Tripathi,Prof. A.K.Mishra, "Software Development Effort and Cost Estimation: Neuro-Fuzzy Model", IOSR Journal of Computer Engineering (IOSRJCE), Vol. 2, pp. 12-14, July-Aug. 2012.

[26] M. Chemuturi, "Software Estimation Best Practices, Tools & Techniques: A Complete Guide for Software Project Estimator" , 2009.

[27] Vachik S. Dave, Kamlesh Dutta, "Comparison of Regression model, Feed-forward Neural Network and Radial Basis Neural Network for Software Development Effort Estimation", ACM SIGSOFT Software Engineering, Vol. 36, No.5, Sept. 2011.