International Journal of Advanced Computational Engineering and Networking, ISSN(p): 2320-2106, ISSN(e): 2321-2063
Volume-5, Issue-12, Dec.-2017, http://iraj.in

ESTIMATING SOFTWARE COST USING MULTILAYER NEURAL NETWORK

[1]KOMAL DHANOPIYA, [2]ANKUR GOYAL, [3]NIDHI MISHRA, [4]MOHIT DAYMA

[1]PG Student, [2]Assistant Professor, [3,4]Associate Professor
[1,2,3,4]Computer Science Engineering, [1,2]Yagyavalkya Institute of Technology, Jaipur, India
[3]Poornima University, Jaipur, India
[4]Anand International College of Engg., Jaipur, India
E-mail: [1]komaldhanopiya021@gmail.com, [2]ankur_gg5781@yahoo.co.in, [3]nidhi.mishra@poornima.edu.in, [4]mohitkdayma@gmail.com

Abstract - At Present, investing the cost in software projects are the most challenging task in the procedure of project management. Software cost estimation is used to evaluate the time and cost which are important to building software structure [1]. There are the different type of cost estimation methods, each method has their own pros and cons during the evaluation of development time and cost. The general question arising in our mind during the cost estimation is how to make them correct, as customers do not explicitly share their needs. No procedure is better or worse than others, in fact, their strengths and weaknesses are often admired for each other [3]. The accuracy of the assessment directly affects the success or failure of the project [4]. Accurate estimation not only saves time and resources when applications are updated or developed but also accelerates the update or development process [5]. This paper deals with different number of research papers, and it was concluded that researchers used different methods to estimate the software cost where calculated cost is not too near to the actual cost. So this paper focused on multilayer neural network to estimate the software cost with efficient result. The COCOMO 81 dataset has been used to train and test the network. The outcomes of the trial of trained neural networks are compared with that of COCOMO model. The determination of our research is to increase the accuracy of the estimation of the COCOMO model by introducing it to the Multilayer neural network [6]. From our experimental results, it was resolved that the proposed Multilayer Neural Network model diminish the Magnitude Relative Error (MRE) between actual cost and calculated cost. This paper also demonstrates the design and implementation of software cost estimation using multilayer neural network and its dream to extend this work using different dataset.

Keywords - Software Cost Estimation; COCOMO Model; Multilayer feed-forward Neural Network; Activation Function; Magnitude Relative Error(MRE).

I. INTRODUCTION

Software cost estimation is one of the most important activities in project management. The accurate cost estimate is important because it can help projects to categorize and prioritize to determine how well these resources will be utilized. The accurateness of management decisions depends on the correctness of the software growth parameters. These parameters include cost estimation, team size estimation, development time etcetera [2].

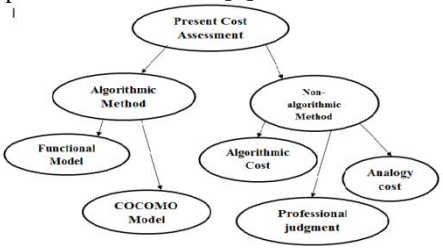


Figure1. Categorization of Software Cost Assessment Method

In our work, we have proposed a multilayer feed forward artificial neural network technique to calculate software cost with minimum error, in other words, the actual cost is closer to calculated cost. the proposed multilayer neural network (MNN) model consist input layer, one hidden layer and output layer. We have compared our proposed neural network model with COCOMO model and finds the improvement of 96.32% based on MMRE 1.062 of COCOMO and MMRE 0.039 of proposed model.

The rest of the paper is organized as Section 2 presents the COCOMO model. Section 3 describes the concepts of artificial neural network. Section 4 discusses the related research of the proposed work. Section 5 describes the proposed model. Section 6 discusses the experimental detail of the proposed work. Finally section 7 concludes the proposed work.

II. COCOMO Model

The COCOMO model stands for constructive cost model planned by Dr. Barry Boehm in 1981. It makes use of mathematical equations to calculate these parameters [6].

A. Basic Model

This is a static single variable model. The Cost Estimation formula is derived as-

$E(cost) = a_b\ (size)\exp(b_b)$. (1)

WhereE is the cost applied in person-months and sizes is measured in thousands of distributed source instructions.

B. Intermediate Model

The Intermediate COCOMO Model is also known as a multivariable static model. Dr. Barry Boehm offered 15 cost drivers which are used in our proposed model.

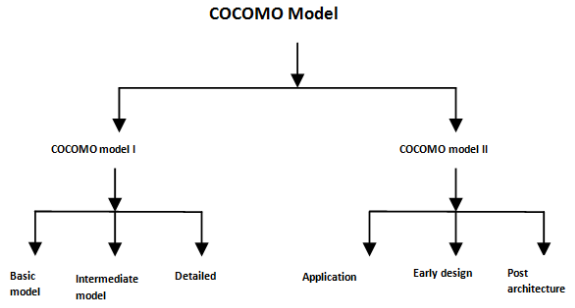


Figure 2. Structure of COCOMO model

Boehm presented 4 attributes as Product, Computer, Personal, Project Attributes which have 15 Variable as shown in the following TableI.

No.	Feature	Description
1	RELY	Required Reliability
2	DATA	Database Size
3	CPLX	Product Complexity
4	TIME	Execution Time Constraint
5	STOR	Main Storage Constraint
6	VIRT	Virtual Machine Volatility
7	TURN	Computer Turnaround Time
8	ACAP	Analyst Capability
9	AEXP	Application Experience
10	PCAP	Programmer Capability
11	VEXP	Virtual Machine Experience
12	LEXP	Programming Language Experience
13	MODP	Modern Programming Practices
14	TOOL	Use of Software Tools
15	SCED	Required Development Schedule

TABLE I. COCOMO 15 COST DERIVERS

III. ARTIFICIAL NEURAL NETWORKS

"The aim of the neural network is to copy the person with manpower to correct the surroundings of the person to change and in the current environment" [8].The Neural Network which began 50 years ago. It is based upon biological neurons. It has many nodes as the input node, output node, and hidden layer node.

A. Types of Neural Network Architecture

- ***Single Layer Artificial Neural Network***

The structure of single layer net is shown fig. 3.where I_1 and I_2 are 2 inputs with M & K weight and produce an output shown as o.

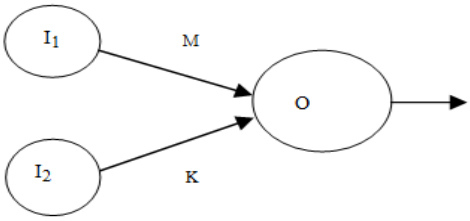
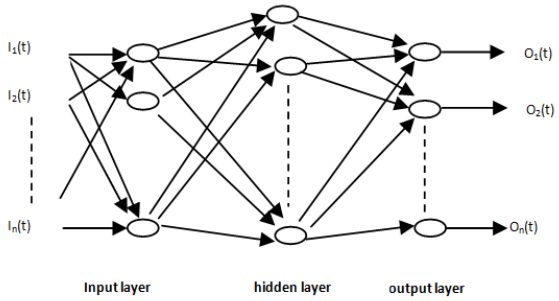


Figure 3.ASingle Layer Artificial Neural Network

- ***Multi-Layer Artificial Neural Network***

Typical Multi-layer artificial neural network, includes the input layer, the output layer and the hidden layer of neurons. Generally, MNN also known as the layered network, the three-layer MNN is visible in Figure 4.



IV. RELATEDWORK

At present, many researchers and scientists often work and implement new costs assessment process using artificial neural networks.

A.K. Soni et al., [1] used Back-Propagation Neural Networks technique to evaluate the software cost for an efficient experimental result. author used COCOMO dataset & NASA 2 dataset which consists of 63 and 93 projects respectively. Proposed approach used identity activation function and sigmoidal function. The result was acceptable according to less MRE. Pradeep Kumar Bhatia et al., [2] studied the COCOMO model and Back-Propagation Neural Network for software cost estimation. Authors merged the COCOMO & Neural Network technique into a single structure. After comparing the actual and evaluated cost, it was shown that COCOMO the evaluated cost is closer to the actual cost. R. Ponnusamy et al., [3] proposed a neural network algorithm to evaluate the software cost with accuracy because customers who not clearly share their requirements there is the lack of accuracy in software cost estimation methodology. No method is confidently better or worse than the other, so their pros and cons are often contradicted to each other. So analogy methods are used to compare the proposed project and similar ancient project. Rajani Kanta Malu et al., [4] using two layer feed forward network authors proposed neural network model to minimize the MRE between actual cost and evaluated cost and for an improved result compare to another model. The cost was estimated with 3 phases, first one as size estimation, second as cost estimation and third as time estimation. Authors used an intermediate model which gives greater accuracy than the basic and detailed model. Nevcihan Duru et al., [5] experimented with a new approach uniting k-means algorithm to estimate the software cost. Evaluated result was acceptable according to MRE and MMRE calculation. Deepak Mittal et al., [6] et al. proposed a

neural network technique using perceptron learning algorithm to evaluate the software cost based on COCOMO model. It was observed that the artificial neural network provides a well-organized way to evaluate the cost with virtually exact values. Siew Hock Ow et al., [7] Siew Hock Ow, proposed a neural network model to evaluate the software cost using desirable features of neural network approach with learning ability and good interpretability. The experimented result showed that the calculated cost was close to the actual cost. Urvashi Rahul et al., [11] worked on new COCOMO based model on soft computing approach for software effort estimation and it was observed that the experimented result gave better accuracy. D. Sivakumar et al., [12] proposed neural network model to evaluate the software effort also accommodate the model and its parameters using multilayer artificial neural network with identity activation function at the input, hidden and the output layer with acceptable result. Mukesh Kumar et al., [13] proposed a class point approach to determine the software effort to develop the software product. They used multilayer neural network (MLP) model and implemented on MATLAB tool. It was concluded that proposed work gives less MMRE. Rachna Soni et al., [14] proposed a back propagation neural network algorithm. They compared their result with COCOMO Model and it improved the result of experimentation. Venkat Ravi Kumar Dammu et al., [15] used a neural network, fuzzy logic and genetic algorithm to estimate the software effort. From the experimentation, it was concluded that no methods present the best estimation in all situations, but in an absolute sense none of the models perform the best estimation. Luiz Fernando Capretz et al., [16] using use case diagram they proposed a neural network model to estimate the software cost, productivity and complexity while the output is the predicted software effort. From the experimented result, it was concluded that the regression and UCP models also used to forecast the software effort from use case diagram.Attarzadeh, Siew Hock Ow et al., [17] improved the accurateness of software effort estimation using mathematical principles. Authors compared their result with related works using few features.It was concluded that the calculated result was acceptable with less Mean Magnitude Relative Error (MMRE).

V. PROPOSED NEURAL NETWORK MODEL

Our proposed model consists of 23 inputs, of which 15 are the multiplier and 5 scale factors, 2 bias and lines of code (LOC). We started bias B_1 with 1.00 and B_2 = 1.01, Weight W_i=1 (i=1 to 15) and Learning Rate (α) = 0.001 in our proposed network and used the identity activation function to calculate the desired output of the network.

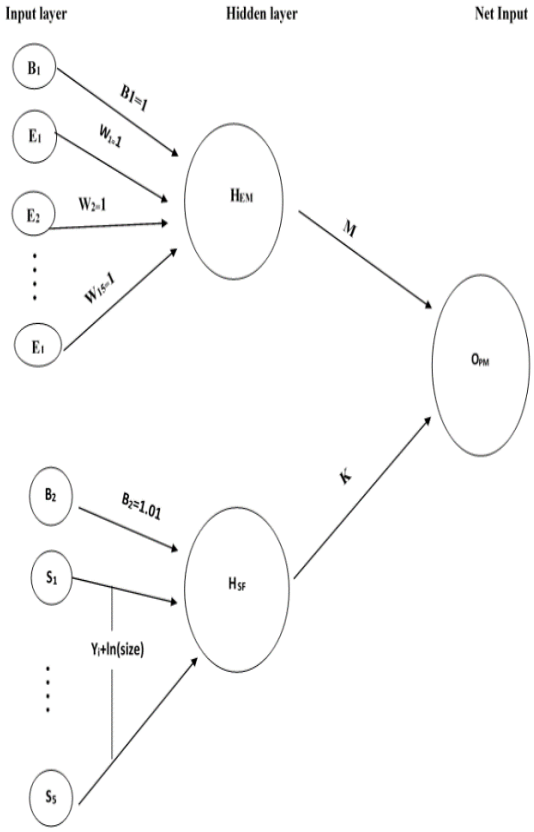


Figure 5.Multilayer feed forward neural network

Training Algorithm is as-

Step 1: Set the weights W_1 to W_{15}=1, learning rate α =0.001, Bias B_1=1,B_2=1.01,θ(Threshold)= 6

Step 2: Execute Steps 3 to12 until the stopping condition is false.

Step 3: Execute Steps 4 to 9 for all training data.

Step 4: Each input layer receives 15 cost multipliers E_1 to E_{15} and 5 scale factors SF_1 to Sf_5 sends it to the hidden layer H_{EM} and H_{SF}.

Step 5: Hidden layer H_{EM} and H_{SF} multiply by its weight to calculate net input output as--

$H_{EM} = B_1 + Ei \times Wi$ for i=1 to 15

$H_{SF} = [B_2 + Si \times (yi + \ln(size)]$ for i=16 to 20

apply identity activation function on hidden layer H_{EM} and H_{SF} and send the output from hidden layer to output layer.

Step 6: Compute the net input-output:

$O_{PM} = H_{EM} \times M + H_{SF} \times K$

Step7: Compute the error term as--

$\delta = |E_{ACT} - O_{PM}|$

where E_{ACT} is the actual cost taken from the dataset (COCOMO 81) and $O_{PM}$is the Calculated Cost computed from step 6.

Step 8: If $\delta <= \theta$

Then Go to Step 9

Step 9: Compute MRE = $(| E_{ACT} - O_{PM} |) / O_{PM} \times 100$,go to step 12

Else

Go to step 10

Step 10: Update the weights between hidden and output layer:

$M(new) = M(old) + \alpha \times \delta \times H_{EM}$
$K(new) = K(old) + \alpha \times \delta \times H_{SF}$

Step 11: Update the weights and bias between input and hidden layers as--

{Error is calculated as $\delta_{EM} = \delta \times M$, $\delta_{SF} = \delta \times K$};

$Wi(new) = Wi(old) + \alpha \times \delta_{EM} \times Si$ for (i=1 to 15)

$Si(new) = Si(old) + \alpha \times \delta_{SF} \times Si$ for (i=16 to 20)

$B_1(new) = B_1(old) + \alpha \times \delta_{EM}$

$B_2(new) = B_2(old) + \alpha \times \delta_{SF}$

Execute Step 3-9.

Step 12: Display Calculated Cost (O_{PM}) and MRE

Step 13: Stop.

The Flow Chart of our proposed algorithm is as-

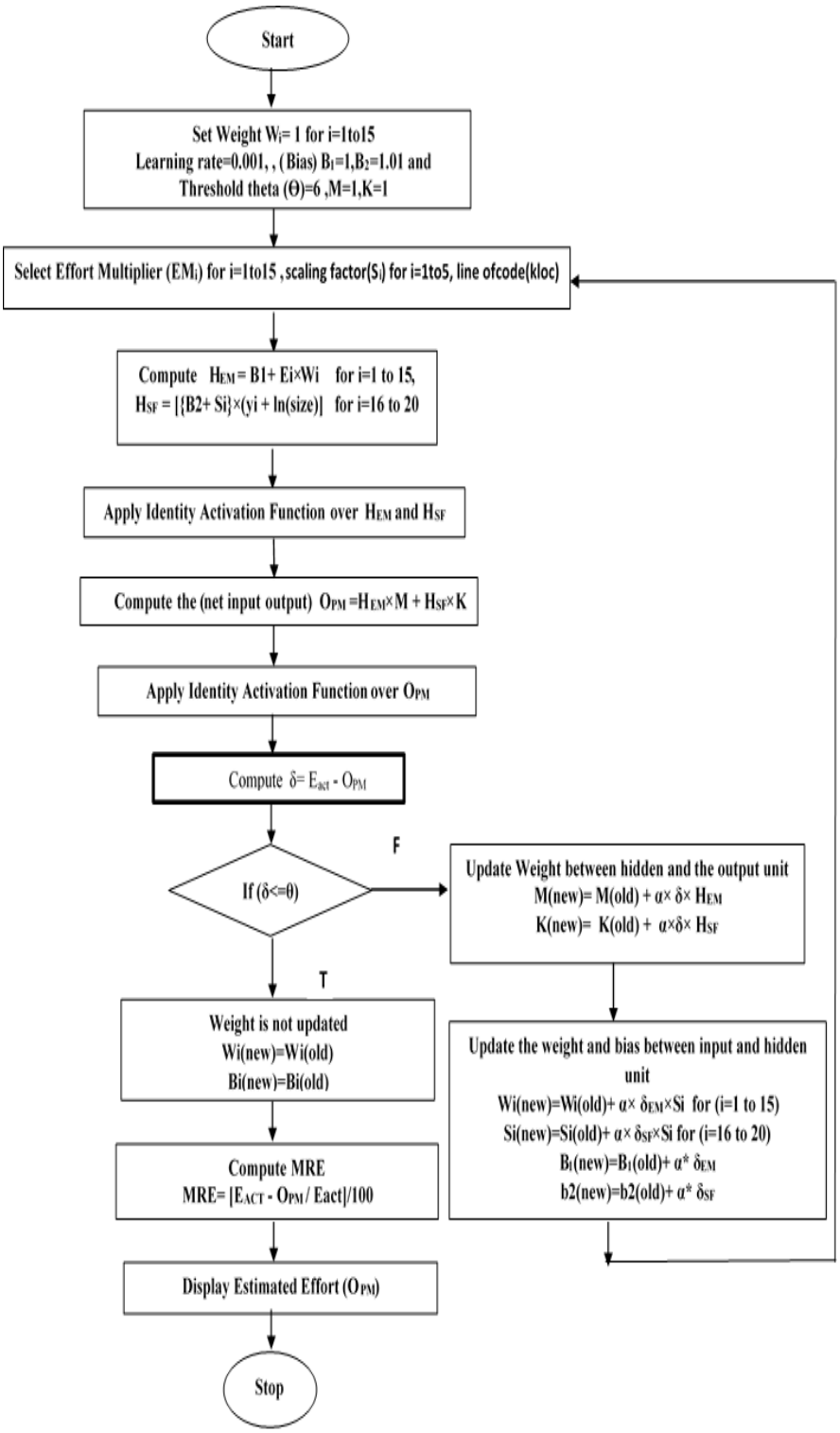


Figure6.Flow chart showing the algorithm of our proposed model

VI. EVALUATION CRITERIA AND RESULTS

Our research is done with feed forward multilayer neural network model through 18 projects among 63 projects which are existed with actual cost in COCOMO 81 dataset [21], which is publicly available.

S. NO	Project ID	Actual cost (E_{ACT})	Calculated Cost Using COCOMO model	Calculated Cost Using proposed model(O_{PM})	$\delta(E_{ACT}-O_{PM})$
1	P1	113	329.93	113.31	0.31
2	P3	132	32.29	131.59	0.41
3	P4	60	28.29	56.4	3.6
4	P18	320	2084	318.23	1.77
5	P20	299	1084.5	294.45	4.54
6	P21	252	174.42	246.28	5.71
7	P22	118	6.51	118.19	0.19
8	P23	77	28.63	74.83	2.16
9	P24	90	29.46	89.71	0.28
10	P25	38	14.55	35.84	2.15
11	P26	48	13.31	43.87	4.13
12	P46	73	9.67	74.25	1.25
13	P47	23	1.29	22.50	0.50
14	P49	91	7.12	90.20	0.79
15	P56	27	21.1	24.05	2.95
16	P58	25	24.68	21.41	3.58
17	P59	23	3.31	20.28	2.72
18	P61	28	1.47	27.47	0.52

Parameters	Initialize by value
Cost Multiplier(E_i)	(15) from COCOMO 81 dataset
W_i (for i=1to15)	1
Bias (B)	B1=1.B2=0.01
Learning rate (α)	0.001
Scaling Factors(S_i)	5 scaling factors
Threshold theta (Θ)	6

TABLE II. ASSESSMENT OF CALCULATED COST

Calculated Cost of our proposed model has been compared to the COCOMO model, which is shown in Table II. For example, the cost to calculate project id 1 is 329.93 for COCOMO model and 113.31 for our proposed model, where there is an actual cost is 113 from COCOMO 81 dataset.

TABLE III.: DIFFERENT PARAMETERS USED IN OUR PROPOSED MODEL

S. NO	Project ID	Actual cost (E_{ACT}) (person month)	Calculated Cost of Proposed model(O_{PM}) (person months)	MRE using COCOMO model(%)	MRE using proposed model(%)
1	P1	113	113.31	191.97	0.27
2	P3	132	131.59	75.53	0.31
3	P4	60	56.4	53.05	6.00
4	P18	320	318.23	551.25	0.55
5	P20	299	294.45	262.60	1.51
6	P21	252	246.28	30.78	2.26
7	P22	118	118.19	94.48	0.16
8	P23	77	74.83	62.81	2.80
9	P24	90	89.71	67.26	.31

10	P25	38	35.84	61.69	5.65
11	P26	48	43.87	78.22	8.60
12	P46	73	74.25	86.75	1.72
13	P47	23	22.50	94.36	2.17
14	P49	91	90.20	92.17	0.86
15	P56	27	24.05	21.85	10.92
16	P58	25	21.41	1.26	14.33
17	P59	23	20.28	85.60	11.82
18	P61	28	27.47	94.75	1.85

TABLE IV. ASSESSMENT OF MRE

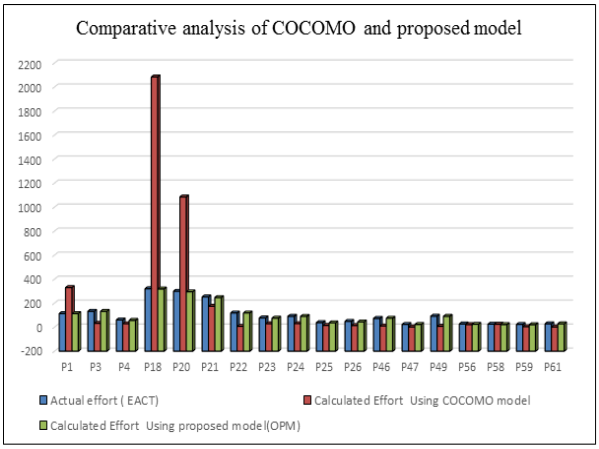


Figure 7.Comparative analysis of COCOMO and proposed model.

After this, the Relative Error (MRE) for project id 1 is 191.97% for COCOMO model and 0.27% for our proposed Multilayer Neural Network Model. This result of Magnitude of Relative Error (MRE) has shown in table IV for all 18 project IDs, which shows the MRE between COCOMO model and our proposed model. Fig. 7 represents the graphical representation of actual cost and calculated cost of COCOMO and proposed model and Fig. 8 showing the difference between the Magnitude of Relative Error (MRE) of COCOMO model and our proposed model in the form of line graph.

Therefore, from the Table IV, V and Fig. 7, 8. we can say that our proposed model gives the most efficient estimated results compared to any other model.

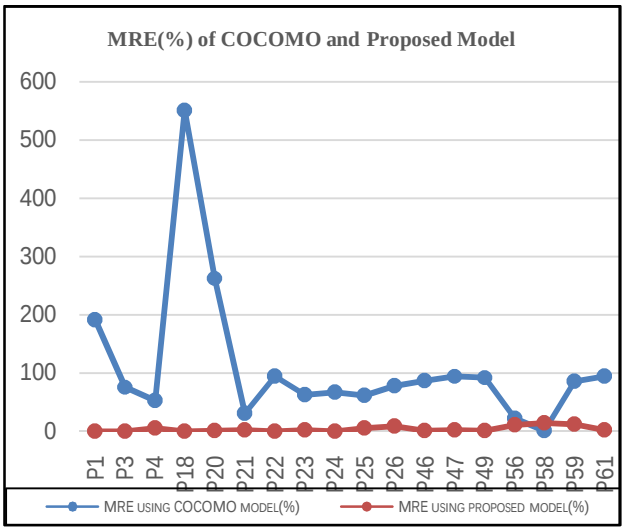


Figure 8. Graph showing less MRE(%) in compare to COCOMO model.

Model	Calculation	MMRE
Proposed Model vs. COCOMO	COCOMO	1.062
	Proposed Model	0.039
	(Accuracy) Improvement %	96.32

Table V. MMRE of the COCOMO model vs. Proposed Model

CONCLUSION

Software cost estimation is an inspiring task for both the engineering as well as proposed communities. The exact computations during the primary stages of software development lifecycle can significantly benefit the software cost development team, in other words, it may help for project succession [1]. Keeping every possible convenient information in use can help managers to overcome the probable problems and give them more detailed evaluations, which will move to the success of the project [5]. There are a number of cost estimation methods that can be used to calculate the software development cost. One of these several methods, the Multilayer Feed Forward Neural Network technique is one of the better cost estimation models that has been used in this paper because of its capability to compute the cost with more exactness. we have used identity activation function at the input, hidden and an output layer.To increase the performance of the network, our aim is to use minimum levels and nodes, as in our proposed model there are two nodes in the hidden layer.

We have tested on the COCOMO 81 dataset which has 63 elements for training and testing the network. Computations have been applied for cost estimation, and output is organized using PHP. Based on the experimental results, it is found that the proposed model outscored COCOMO model and adequate and acceptable as per MRE and MMRE calculation. MMRE of proposed model is 0.039 and MMRE of COCOMO model is 1.062 and overall accuracy of proposed model is 96.32%. In the future, this model can be expanded from another one and customize the cost estimation using genetic algorithms, and the use of other neural network structures can also be applied to software cost estimates.

REFERENCES

[1] A.K. Soni, AnupamaKaushikRachnaSoni.: “A Simple Neural Network Approach to Software Cost Estimation” Global Journal of Computer Science and Technology Neural & Artificial Intelligence Volume 13 Issue 1 Version 1.0 Year (2013).

[2] Pradeep Kumar Bhatia, GauravKumar,.: ‘‘Automation of Software Cost Estimation using Neural Network Technique’’ International Journal of computer application ,Vol. 98.,No.20, july (2014).

[3] R. Ponnusamy, Manikavelan, D.,.:"Software cost estimation by analogy using feed forward neural network." International Conference on Information Communication and Embedded Systems (ICICES), IEEE, pp. 1-5 (2014).

[4] RajaniKantaMalu, SoumyabrataMukherjee ,.: “Optimization of Project Cost Estimate Using Neural Network” International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), IEEE, pp. 406-410 (2014).

[5] NevcihanDuru, Saraç, ÖmerFaruk,.: "A novel method for software cost estimation: Estimating with boundaries."International Symposium onInnovations in Intelligent Systems and Applications (INISTA), IEEE, pp. 1-5 (2013).

[6] Deepak Mittal, AnupamaKaushik, SachinGupta.: “COCOMO Estimates Using Neural Networks”International Journal of Intelligent Systems and Applications 4.9 (2012).

[7] Siew Hock Ow, Attarzadeh, Iman,.: "A novel soft computing model to increase the accuracy of software development cost estimation." The 2nd International Conference onComputer and Automation Engineering (ICCAE), Vol. 3. IEEE, (2010).

[8] Shivanandan, S. N., S. N. Deepa.: Introduction to neural networks using Matlab 6.0. Tata McGraw-Hill Education, (2006).

[9] Reddy ChSatyananda, K. V. S. V. N. Raju.: "A concise neural network model for estimating software cost." International Journal of Recent Trends in Engineering 1.1:188-193, (2009).

[10] GeShuzhi Sam, Chang C. Hang, Tong H. Lee, Tao Zhang.: Stable adaptive neural network control. Vol. 13. Springer Science & Business Media, (2013).

[11] Urvashi Rahul Saxena, S. P. Singh.: "Software cost estimation using Neuro-fuzzy approach" CSI Sixth International Conference onSoftware Engineering (CONSEG), IEEE, pp. 1-6 (2012).

[12] d.sivakumar, M.madheswaran,.: “Enhancement of prediction accuracy in cocomo model for software project using neural network ” International Conference on Computing Communication and Networking Technologies (ICCCNT), IEEE, pp. 1-5, (2014).

[13] Mukesh Kumar, ShashankMouliSatapathy.: Santanu Kumar Rath.: "Fuzzy-class point approach for software cost estimation using various adaptive regression methods." CSI transactions on ICT 1.4: 367-380, (2013).

[14] RachnaSoniKaushik, Anupama, A. K. Soni,.: "An adaptive learning approach to software cost estimation." National Conference onComputing and Communication Systems (NCCCS), IEEE, pp. 1-6, (2012).

[15] Venkat Ravi Kumar Dammu, Merugu, R. Raja Ramesh,.: "Cost estimation of software project." International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 1.10: pp-33, (2012).

[16] Luiz Fernando Capretz, Nassif, Ali Bou, , Danny Ho.: "Estimating software cost using an ANN model based on use case points." 11th International Conference on Machine Learning and Applications (ICMLA), Vol. 2. IEEE, pp. 779-784, (2015).

[17] Attarzadeh, Siew Hock Ow, Iman ,.: “Proposing a New Software Cost Estimation Model Based on Artificial Neural Networks”2nd International Conference on Computer Engineering and Technology (ICCET), Vol. 3. IEEE, pp. V3-487-V3-491, (2010).

[18] Chidchanok Lursinsap, Jodpimai, Pichai, Peraphon Sophatsathit,.: "Estimating software cost with minimum features using neural functional approximation." International

Conference on Computational Science and Its Applications (ICCSA), IEEE, (2010).

[19] Cuauhtemoc Lopez-Martin, Kalichanin-Balich, Ivica,.: "Applying a feed forward neural network for predicting software development cost of short-scale projects." Eighth ACIS International Conference on Software Engineering Research, Management and Applications (SERA), IEEE, pp. 269-275, (2010).

[20] Johannes Sidabutar, Sarno, Riyanarto.: "Comparison of different Neural Network architectures for software cost estimation." International Conference on Computer, Control, Informatics and its Applications (IC3INA), IEEE, pp. 68-73, (2015).

[21] http://promise.site.uottawa.ca/SERepository/datasets/COCOMO81.arff.

★★★