International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:1, 2014

Software Effort Estimation Models Using Radial Basis Function Network

E. Praynlin, P. Latha

***Abstract*—**Software Effort Estimation is the process of estimating the effort required to develop software. By estimating the effort, the cost and schedule required to estimate the software can be determined. Accurate Estimate helps the developer to allocate the resource accordingly in order to avoid cost overrun and schedule overrun. Several methods are available in order to estimate the effort among which soft computing based method plays a prominent role. Software cost estimation deals with lot of uncertainty among all soft computing methods neural network is good in handling uncertainty. In this paper Radial Basis Function Network is compared with the back propagation network and the results are validated using six data sets and it is found that RBFN is best suitable to estimate the effort. The Results are validated using two tests the error test and the statistical test.

***Keywords*—**Software cost estimation, Radial Basis Function Network (RBFN), Back propagation function network, Mean Magnitude of Relative Error (MMRE).

I. Introduction

COMPUTER has conquered every aspect of our daily life. Computers are used for communication, shopping, banking etc. Sometimes computers are embedded and used in some other products like ATM, fridge, cars, ticket reservation system etc. However computers are to be told in which way the computer has to function in order to be useful to us. The set of instruction is often referred to us programs or generally software. When software fails its consequence will be very serious for example in 2011Computer system problems at one of Japan's largest banks resulted in a nationwide ATM network of more than 5,600 machines going offline for 24 hours, internet banking services being shut down for three days, delays in salary payments worth $1.5 billion into the accounts of 620,000 people and a backlog of more than 1 million unprocessed payments worth around $9 billion.

Software effort estimation is an important part of software development work and provides essential input to project feasibility analyses, bidding, budgeting and planning. The consequence of in accurate estimate can be severe [1]. Predicting the software development effort with high accuracy is still a great challenge and an unsolved problem and there is a high level of research is ongoing in this area [2]. When cost overrun or schedule overrun occurs the software quality may get affected because of resource shortage. Developing a software project with acceptable quality within budget and on planned schedule is the main goal of every software firm [3] schedule estimation is the major difficulty in managing software development projects. The effect of irrelevant and misleading information of software professionals has devastating consequences [4]. There are several methods to estimate the effort they are regression, Analogy, Expert judgment, work breakdown, function point, CART, Simulation, Neural Networks, Theory, Bayesian and combination of Estimates [5].

Among the methods available to estimate the effort neural network based methods plays the prominent role. The usage of neural network to estimate the effort was first proposed by Venkatachalam [6]. The goal of the Neural Network is to model the relationship between the input and output from the historic data so that it can be used produce the good estimate for the future projects [7]. Neural Network is compared to regression models and found Neural Network is better than regression method for estimating effort [8].

Praynlin. E is with the Department of Computer Science Engineering, Government College of Engineering Tirunelveli, Tamilnadu, India. (Corresponding author e-mail: praynlin25@gmail.com)
Latha. P is with the Department of Computer Science Engineering, Government College of Engineering Tirunelveli, Tamilnadu, India.

II. Software Cost Estimation

It is difficult to expect perfect effort estimates even in perfect estimation process because of lot of uncertainties involved in it. The software development effort of a software project is more frequently estimated by project managers. Using the estimated effort they calculate the cost and duration associated with the project. Accurate development Effort estimation at the earlier stage of a software development cycle is necessary to plan, monitor and control the allocated resources appropriately [9].

Software Development Effort Depends on:

- The amount of Implemented functionality
- Number of Errors made by programmers
- Quality of code produced
- Availability of development tools
- Probabilistic factors (absence of staff due to sickness)
- Availability of skilled person

A. Limitations of Software Effort Estimation

Most estimation model requires complete understanding of model parameters as well of certain level of expertise in order to use them effectively. Without proper understanding, team's capabilities can be overstated or the complexities of the project are can be wrongly understood. Moreover, Software projects are prone to changes in requirement, design and infrastructure as they progress in life cycle therefore; the uncertainties in resource estimation are constantly changing. Variations may be more apparent when clients are constantly

updating and changing the requirement specification. These varying environments cannot be updated automatically to software effort estimation

III. Neural Networks in Prediction

A. Back Propagation Network

The back propagation learning algorithm is one of the most widely used methods in neural network. The network associated with back-propagation learning algorithm is called as back propagation network. While training a network a set of input-output pair is provided the algorithm provides a procedure for changing the weight in BPN that helps to classify the input output pair correctly. Gradient descent method of weight updating is used [10].

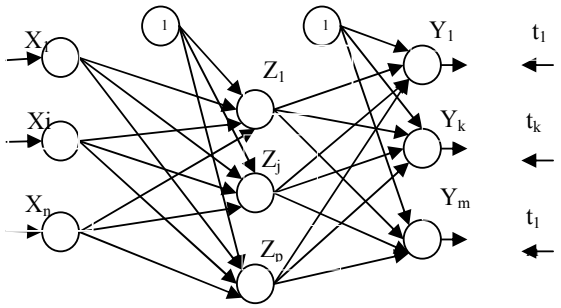

Fig. 1 Architecture of a Back propagation Network

The aim of the neural network is to train the network to achieve a balance between the net's ability to respond and its ability to give reasonable responses to the input that is similar but not identical to the one that is used in training. Back propagation algorithm differs from the other algorithm by the method of weight calculation during learning. The drawback of Back propagation algorithm is that if the hidden layer increases the network become too complex

1. Procedure For Back Propagation Algorithm:

Let the input training vector $x = (x_1, \ldots\ldots, x_i, \ldots\ldots, x_n)$ and target output vector $t = (t_i, \ldots\ldots, t_k, \ldots\ldots t_m)$ the effort multiplier and scale factor can be given as the input x and the target effort is presented as t. α represents the learning rate parameter, v_{oj} = bias on j^{th}hidden layer, w_{ok} = bias on k^{th}hidden layer, $z_j$hidden unit j, the net input to z_j is

$$Z_{inj} = v_{oj} + \sum_{i=1}^{n} x_i v_{ij} \quad (1)$$

and the output is

$$Z_j = f(Z_{inj}) \quad (2)$$

y_k= output unit k. the net input to y_k is

$$y_{ink} = w_{ok} + \sum_{j=1}^{p} z_j w_{jk} \quad (3)$$

and the output is

$$y_k = f(y_{ink}) \quad (4)$$

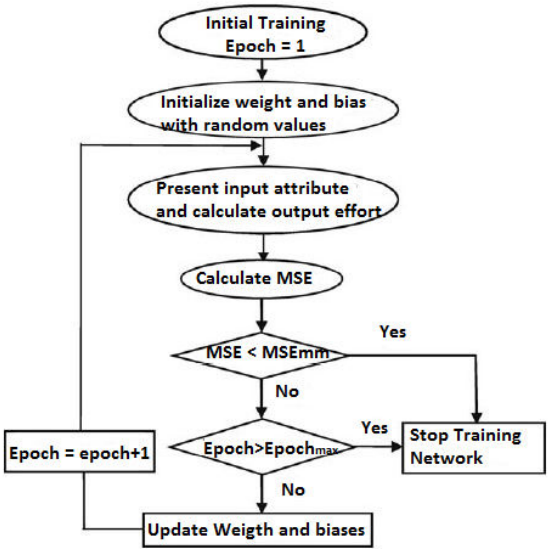

Fig.2 Back propagation flow diagram

Fig. 2 gives the detail about how the training pattern presented to the network and the how the weight and bias is get updated in each epoch.

B. Radial Basis Function Network

Radial Basis function neural Network is an Artificial Neural Network that uses Radial Basis Function as Activation Function. RBFN is the popular alternative to BPN because of its simpler structure and much faster training process. Radial Basis Function network is widely used in many application including function approximation, Time series control, and control system related application. The Radial Basis function network is proposed by Broomhead et al. [11] in 1988. The neural network is proved to be the best suitable for approximation [12]. The typical RBF network is shown in Fig. 3.

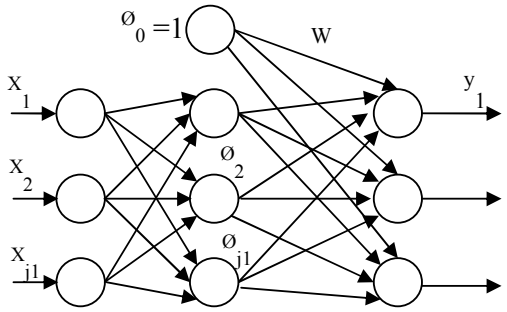

Fig. 3 Typical Radial Basis Function Network

C. Architecture of the RBFN

The input layer has J_1 nodes, the hidden and output layers have J_2 and J_3 Neurons, respectively. $\emptyset_0(x) = 1$ corresponds

to the bias in the output layer, while $\emptyset_i$(x)=$\emptyset$(x-c_i) being the center of the i th node and $\emptyset(x)$an RBF.

The RBFN is a J_1–J_2–J_3 FNN, and is shown in Fig. 3. Each node in the hidden layer uses an RBF, denoted by $\emptyset(r)$, as its nonlinear activation function. The hidden layer performs a nonlinear transform of the input, and the output layer is a linear combiner mapping the nonlinearity into a new space. The biases of the output layer neurons can by an additional neuron in the hidden layer, which has a constant activation function $\emptyset_0(r) = 1$. The RBFN, can achieve a global optimal solution to the adjustable weights in the minimum MSE sense by using the linear optimization method.

For an input pattern X, the output of the network is given by

$$Y_i(X) = \sum_{k=1}^{j2} W_{ki} \emptyset \left(||X - C_k||\right) \quad (5)$$

For i=1,…J_3, Where$Y_i(X)$ is the ith output of the RBFN, W_{ki} is the connection weigth from the kth hidden unit to the ith output unit, $C_k$is the prototype or center of the kth hidden unit, and $\| \cdot \|$ denotes the Euclidean norm. The RBF $\emptyset$(.) is typically selected as the Gaussian function.

For a set of N pattern pairs$\{(X_p, Y_p)\}$, (5) can be expressed in the matrix from as

$$Y=W^T \emptyset \quad (6)$$

where $W=[w1,\ldots,W_{J3}]$ *is a* $J_2 X\ J_3$ weight matrix, $W_i = (W_{1i}, \ldots, W_{J2i})^T$, $\emptyset=[\emptyset 1, \ldots, \emptyset_N]$ is a $J_2\, X\, N$ matrix, $\emptyset_p=(\emptyset_p, 1, \ldots, \emptyset_p, J_2)^T$ is the output of the hidden layer for the pth sample, $\emptyset_p\ , k = \emptyset(||X_p - C_k||)$, $Y=[Y_1, Y_2,\ldots Y_n]$ *is a* $J_3\, X\, N$ matrix, and$Y_p=(Y_p, 1, \ldots\ Y_p, J_{3)}{}^T$.

IV. Data Sample

Here six types of datasets are used for analysis. They are COCOMO dataset, Desharnais dataset, Maxwell dataset IKH dataset, Kitchenham dataset and Telecom dataset. While IBM, kemerer, and Hallmark datasets are combined together as IKH data set for our analysis.

A. Cocomo Datasets

The COCOMO Dataset used in the analysis and validation of the model is obtained from the historic projects of NASA. One set of dataset consists of 63 projects and other has 93 projects. The datasets is of COCOMO II format. In our experiment 93 projects are used for training and 63 projects are used for testing.

The Dataset need for training as well as testing is available in the promise data repository [13]. The dataset available is of COCOMO 81 format which is to be converted to COCOMO II format using the COCOMO II Model definition manual [14] and Rosetta stone [15]. COCOMO 81 is the earlier version developed by Barry Boehm in 1981 and COCOMO II is the next model developed by him in the year 2000. Some of the attributes like TURN are used only in COCOMO 81 which has been neglected in COCOMO II due to the vast availability of resources in the software industry during the recent years. The detailed description about how the attributes of the dataset their range and how they can be fixed [16] can be understood before analysis.

B. Desharnais Dataset

The Desharnais dataset includes eight input parameters like Team experience, Managers experience etc. and an output effort for each project. It totally consists of 77 projects of which 62 projects are used for training and 15 projects are used for testing. The Dataset is available in the promise data repository.

C. Maxwell Dataset

Maxwell dataset is the most recent among all the datasets. It was created by Kathrina D. Maxwell and is made up of categorical features. It totally consists of 62 projects among which 44 projects are used for training and 18 projects are used for testing. The Dataset is available in the promise data repository.

D.IBM, Kemerer, and Hallmark Dataset (IKH)

The IKH dataset is made up of Function point and lines of code. It consists of 24 IBM projects, 15 Kemerer projects and 28 Hallmark projects. Where 17 of IBM, 11 of Kemerer and 20 of Hallmark projects are used for training and 7 of IBM, 4 of Kemerer and 8 of Hallmark projects are used for testing

E. Kitchenham Dataset

Kitchenham dataset is the data collected from one of the biggest commercial bank in Finland. Kitchenham dataset consist of only one parameter named function point. It is used as the input and effort is the output. This dataset is made up of totally 145 projects. Among which 100 is used for training and 45 is used for testing.

F. Telecom Dataset

Telecom dataset includes attributes like ACT, ACT_DEV, ACT_TEST, CHNGS and FILES. ACT is actual effort, ACT_DEV and ACT_TEST are actual development and testing effort, CHNGS is the number of changes made as recorded by the configuration management system and files is the number files changed by the particular enhancement project. Only FILES can be used for predictive purpose. Since none of other information would be available at the time of making prediction. Telecom dataset consists of 17 projects among which 13 are used for training and 4 are used for testing.

V. The Approach

The problem can be stated using the mathematical notation as follows. The input is a given$n \times l$ matrix given by $X = (x_1, x_2, \ldots\ldots, x_n)^T$ where each of the n input vectorx_i, for $i = 1, \ldots\ldots n$ is in a l dimensional space. Let $y = (y_1, y_2, \ldots\ldots\ldots y_n)^T$ be the target vector whose element $y_i$is the output corresponding to the input vectorx_i, for$i = 1,2, \ldots\ldots n$. In short it can be given as the training dataset D

$$D = \{(x_i, y_i): x_i \in R^d, y_i \in R, i = 1, \ldots\ldots n\} \quad (7)$$

This gives the input values and the corresponding value of output. The model will map the l dimensional input space to a one dimensional Target value based on Training data D.

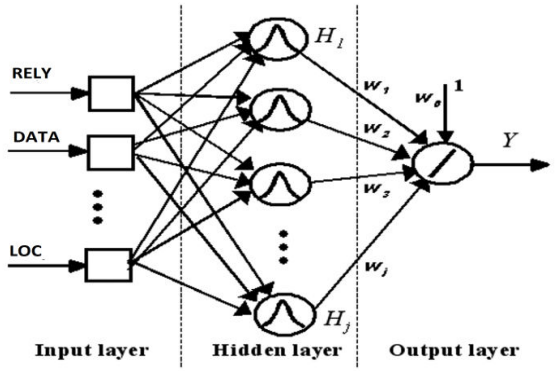

Fig. 4 RBFN Effort estimation system

The simulation is done in MATLAB 10b environment. In RBFN the weights are randomly fixed so each time there is a possibility of getting different result to avoid this problem the whole network is made to run for 50 iteration and their errors are averaged. In matlab RBFN is created by 'newrb' and maximum number of neuron is fixed as 400, goal is set as 0.01 and spread is fixed 1.0. similarly BPN is created by using 'newff' Only one hidden layer is used and Number of hidden layer neuron is set as 8, 'tansig' is used as the transfer function for hidden layer, ' Purelin' is used as the transfer function for output layer and 'trainlm' is used as the training function.

VI. Various Criteria for Assessment of Estimation Models

For evaluating the different software effort estimation models two types of test are done: Error test and statistical test

A. Error Test

The most widely accepted evaluation criteria are the mean magnitude of relative error (MMRE), Probability of a project having relative error less than 0.25, Root mean square error, and Mean and standard deviation of error.

The magnitude of relative error (MRE) is defined as follows

$$MRE_i = \frac{|actual\ effort_i - predicted\ effort_i|}{actual\ effort_i} \quad (8)$$

The MRE value is calculated for each observation whose effort is predicted. The aggregation of MRE over multiple observations (N) can be achieved through the mean MMRE as follows

$$MMRE = \frac{1}{N}\sum_i^N MRE_i \quad (9)$$

$$PRED(25) = \frac{\boldsymbol{MRE \leq 0.25}}{\boldsymbol{N}} \quad (10)$$

Consider Y is the neural network output and T is the desired target. Then Root mean square error (RMSE) can be given by

$$RMSE = \sqrt{(Y-T)^2} \quad (11)$$

B. Statistical Test

Error can be calculated by the difference between Y and T then mean and standard deviation is calculated by calculating the mean and standard deviation of the error

$$ERROR=(Y\text{-}T) \quad (12)$$

Mean of the error can be calculated by

$$\mu = \frac{\sum Error}{N} \quad (13)$$

The standard deviation can be calculated by

$$\text{Standard deviation}=\boldsymbol{\sigma} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x_i}-\boldsymbol{\mu})^2} \quad (14)$$

The **skewness** of a random variable is the ratio of its third central moment $\mu_3$to the cube of its standard deviation σ. Skewness is denoted as Y_1.

$$\boldsymbol{Y_1} = \frac{\boldsymbol{\mu_3}}{\boldsymbol{\sigma^3}} \quad (15)$$

The **kurtosis** of a random variable is the ratio of its fourth central moment μ_4 to the fourth power of its standard deviation σ. Kurtosis is denoted asY_2. Thus

$$\boldsymbol{Y_1} = \frac{\boldsymbol{\mu_4}}{\boldsymbol{\sigma^4}} \quad (16)$$

VII. Experimental Results

The results for the RBFN can be determined by having two types of tests: The error test and the statistical test. In the Error test MMRE, RMSE, and PRED parameters are calculated and in statistical test mean, standard deviation, skewness and kurtosis are calculated for each of the methods under consideration.

The model with the lower MMRE and smaller standard deviation will be considered the best method. The mean should be such that it is closer to zero in order for the effort to be accurate.

TABLE I
Comparative Results of BP And RBFN for NASA Dataset

	BP		RBFN	
Cocomonasa	Training	Testing	Training	Testing
MMRE	0.356	3.2741	0.2128	2.3539
RMSE	636.207	1651	248.428	1847
PRED	60.2151	22.2222	69.8925	9.5238
Mean	27.0726	-482.83	-17.121	-410.51
Std.Dev	639.076	1592	249.181	1815
skewness	-3.4799	-4.4913	-2.5154	-4.3766
kurtosis	32.4337	23.6482	28.8554	23.1341

TABLE II
COMPARATIVE RESULTS OF BP AND RBFN FOR DESHERNAIS DATASET

	BP		RBFN	
Deshernais	Training	Testing	Training	Testing
MMRE	0.413	0.7069	0.1916	0.6841
RMSE	1929	9644	647.3	13.33
PRED	58.0645	6.6667	74.193	13.3333
Mean	20.8217	4775	0	2533.3
Std.Dev	1944	8673	652.7	6566
skewness	0.29	0.3099	0.3245	-1.6849
kurtosis	7.7623	2.8664	3.9978	6.2727

TABLE III
COMPARATIVE RESULTS OF BP AND RBFN FOR MAXWELL DATASET

	BP		RBFN	
Maxwell	Training	Testing	Training	Testing
MMRE	0.2443	1.0143	0.7869	0.6088
RMSE	6956	7963	2978	9761
PRED	77.2727	11.1111	72.7273	11.1111
Mean	-1246.5	-1995	0	5411
Std.Dev	6923	7933	3012	8359
skewness	-4.8787	-2.1439	-1.6006	-2.8122
kurtosis	26.9826	8.3647	12.7886	11.1066

TABLE IV
COMPARATIVE RESULTS OF BP AND RBFN FOR IKH DATASET

	BP		RBFN	
IKH	Training	Testing	Training	Testing
MMRE	2.5849	1.9688	0.6356	1.6837
RMSE	7.5341	13.07	17.5852	10.36
PRED	35.4167	21.0526	22.9167	26.3158
Mean	0	-2.1263	6.4582	0.9834
Std.Dev	7.6138	13.256	16.5294	10.599
skewness	-0.5985	-0.2491	2.3598	0.9914
kurtosis	8.7735	2.9761	11.0916	3.431

TABLE V
COMPARATIVE RESULTS OF BP AND RBFN FOR KITCHENHAM DATASET

	BP		RBFN	
Kitchenham	Training	Testing	Training	Testing
MMRE	0.3984	0.6462	0.395	0.6482
RMSE	1450	12640	1453	12658
PRED	42	8.8889	42	13.3333
Mean	0.0061	685.0911	-0.0235	745.3743
Std.Dev	1457	12764	1460.8	12779
skewness	-1.1504	-6.2843	-1.0497	-6.2759
kurtosis	10.8134	41.3514	10.3327	41.2814

TABLE VI
COMPARATIVE RESULTS OF BP AND RBFN FOR TELECOM DATASET

	BP		RBFN	
Telecom	Training	Testing	Training	Testing
MMRE	0.1543	0.1999	0.1268	0.1753
RMSE	45.028	6.12	50.742	5.49
PRED	84.615	50	84.615	50
Mean	0	-2.30	0.2535	-2.70
Std.Dev	46.866	6.54	52.813	5.52
skewness	0.0699	0.3633	-1.3373	0.1926
kurtosis	2.0907	1.5021	5.0531	1.318

VIII. CONCLUSION

Prime concern of the effort predictor model is the closeness of the estimated effort to the actual effort. In this paper, we have used RBFN to estimate the software development effort. The results are tested using 6 datasets. From the Error test and statistical test it can be clearly understood that RBFN has the lower MMRE than BPN. As far as we know our research initiative is the first to undertake the effort prediction models using large number of datasets and make separate statistical tests like standard deviation, skewness, and kurtosis. Finally it is concluded that for the software industry RBFN is best suited to effort prediction compared to BPN.

REFERENCES

[1] Martin Shepperdand Michelle Cartwright,Predicting with Sparse Data, *IEEE Transactions on Software Engineering,* Vol. 27, n. 11, pp. 987-998, 2001.
[2] IngunnMyrtviet, Erik Stensrudand Martin Shepperd, Reliability and Validity in Comparative Studies of Software Prediction Models, *IEEE Transactions on Software Engineering,* Vol. 35, n. 5, pp. 380-391, 2005.
[3] Steve McConnell, Rapid Development: Taming Wild Software Schedules, *Microsoft Press*, 1996.
[4] Magne Jorgensen, The Impact of Irreverent or Misleading Information on Software Development Effort Estimate, *IEEE Transaction on Software Engineering*, Vol.37, n.5, pp. 695-707, 2011.
[5] Magne Jorgensen and Martin Shepperd, A Systematic Review of Software Development Cost Estimation Studies, *IEEE Transaction on Software Engineering*, Vol.33, n.1, 2007.
[6] Venkatachalam, Software Cost Estimation Using Artificial Neural Networks, *International Joint Conference on Neural Networks*, Nogoya, IEEE, 1993.
[7] MiyoungShin and AmritL.Goel, Emprical Modelling in Software Engineering Using Radial Basis Functions, *IEEE Transaction on Software Engineering*, Vol.26, n.6, 2000
[8] Abbas Heiat, Comparison of Artificial Neural Network and Regression Models for Estimating Software Development Effort, *Information and Software Technology*, vol.44, pp.911-922, 2002.
[9] M.Ochodek, J.Nowrocki, K.Kwarciak, *Simplifying Effort Estimation Based On Use Case Points*, Information and Software Technology, vol.53, no.3,pp.200-213, 2011.
[10] Dr.S.N.Sivanandam, Dr.S.N.Deepa, *Principles of softComputing*(Wiley - India, 2004).
[11] Broomhead.D.S, D.Lowe, Radial Basis Function, Multivariable Function Interpolation and Adaptive Networks*, Royal Signals and Radar Establishment*, Memorandum. 4148, 1988
[12] Yue Wu, Huiwang, Biaobiaozhang, K.L.Du, Using Radial Basis Function Networks for Function Approximation and Classification, ISRN Applied Mathematics, pp.1-34, Vol.2012
[13] Sathyananda Reddy, KVSVN Raju, "Improving the Accuracy ofEffort Estimation through Fuzzy Set Combination of Size and Cost Drivers," *WSEAS Transaction on Computers*," Vol.8, no.6, PP.926-936, June 2009.
[14] Barry Boehm, "COCOMO II: Model Definition Manuel. Version 2.1,"*Center for Software Engineering*, USC, 2000.
[15] Donald J. Reifer, Barry W. Boehm and Sunithachulani, The Rosetta Stone: Making COCOMO 81 Estimates Work with COCOMO II, *CROSSTALK The Journal of Defense Software Engineering*, pp. 11 – 15, Feb.1999.
[16] E.Praynlin, P.Latha, "Performance Analysis of Software Effort Estimation Models Using Neural Network", International Journal of Information Technology and Computer Science, PP. 101-107, August 2013.

Praynlin,E. received his master's degree in Applied Electronics from Noorul Islam University, ThuckalayTamilnadu, India in the year 2011. He graduated Bachelors degree from Sun college of Engineering and Technology, Nagercoil, Tamil Nadu, India in Electronics and communication engineering in the year 2009.

He has published Papers in Journals like IJITCS, IJCIS etc., his research interests include software cost Estimation and Neural Networks.

Latha,P.gotmaster's degree in computer science from Bharatiyar University, Coimbatore, Tamil Nadu, India in the year 2001. She graduated Bachelor's degree from Madurai KamarajUniversity, Madurai, Tamil Nadu, and India in Electronics and communication engineering in year 2009.

She has publishedpapers in reputed journals her research interest includes Image processing and soft computing. Latha. P is the secretary of ISTE Staff chapter.