A Model for Estimation of Efforts in Development of Software Systems

Parvinder S. Sandhu, Manisha Prashar, Pourush Bassi, and Atul Bisht

Abstract**—**Software effort estimation is the process of predicting the most realistic use of effort required to develop or maintain software based on incomplete, uncertain and/or noisy input. Effort estimates may be used as input to project plans, iteration plans, budgets. There are various models like Halstead, Walston-Felix, Bailey-Basili, Doty and GA Based models which have already used to estimate the software effort for projects. In this study Statistical Models, Fuzzy-GA and Neuro-Fuzzy (NF) Inference Systems are experimented to estimate the software effort for projects. The performances of the developed models were tested on NASA software project datasets and results are compared with the Halstead, Walston-Felix, Bailey-Basili, Doty and Genetic Algorithm Based models mentioned in the literature. The result shows that the NF Model has the lowest MMRE and RMSE values. The NF Model shows the best results as compared with the Fuzzy-GA based hybrid Inference System and other existing Models that are being used for the Effort Prediction with lowest MMRE and RMSE values.

Keywords**—**Neuro-Fuzzy Model, Halstead Model, Walston-Felix Model, Bailey-Basili Model, Doty Model, GA Based Model, Genetic Algorithm.

I. Introduction

In recent years, software has become the most expensive component of computer system projects. Accurate software cost estimates are critical to both developers and customers. Underestimating the costs may result in management approving proposed systems which can exceed their budgets, with underdeveloped functions and poor quality, and failure to complete on time. Overestimating may result in too many resources committed to the project, or, during contract bidding, result in not winning the contract, which can lead to loss of jobs. So, accurate cost estimation is important. In the last three decades, many quantitative software cost estimation models have been developed. They range from empirical models such as Boehm's COCOMO models [2] to analytical models such as those in [5, 7, 10]. An empirical model uses data from previous projects to evaluate the current project and derives the basic formulae from analysis of the particular database available. An analytical model, on the other hand, uses formulae based on global assumptions, such as the rate at which developer solves problems and the number of problems available.

Typical major models that are being used as benchmarks for software effort estimation are:

- Halstead,
- Walston-Felix
- Bailey-Basili
- Doty (for KLOC > 9).

These models have been derived by studying large number of completed software projects from various organizations and applications to explore how project sizes mapped into project effort. But still these models are not able to predict the Effort Estimation accurately.

As Neuro-fuzzy based system is able to approximate the non-linear function with more precision and non of the researcher have explored Neuro-fuzzy approach for the Effort Estimation and there is still scope of exploring more statistical modeling approaches. So, in this proposed study, it is tried to use Soft Computing Techniques and statistical techniques to build a more accurate model that can improve accuracy estimates of effort required to build a software system.

The remainder of this paper can be described as follows: Section II outlines the literature review about the various techniques that are used for the effort and cost estimation. Section III discusses the methodology adopted for generating and comparing a number of models. Section IV highlights results of implementation. It discusses the results of the various models used for the effort estimation and Section V is all about conclusions of this research work.

II. Basic Cost Estimate Models

There are two major types of cost estimation methods:

A. Algorithmic Models

These models vary widely in mathematical sophistication. Some are based on simple arithmetic formulas using such summary statistics as means and standard deviations [15]. Others are based on regression models [4] and differential equations [7]. To improve the accuracy of algorithmic models, there is a need to adjust or calibrate the model to local circumstances. These models cannot be used off-the-shelf. Even with calibration the accuracy can be quite mixed.

The existing algorithmic methods differ in two aspects: the selection of cost factors, and the form of the function. Firstly, the cost factors used in these models are discussed, then characterize the models according to the form of the functions and whether the models are analytical or empirical. The

Manisha Prashar is working with Govt. Shivalik College, Naya Nangal, Punjab, India.

Parvinder S. Sandhu, Pourush Bassi and Atul Bisht are associated with Rayat Bahra Institute of Engineering & Bio-Technology, Sahauran, Mohali (India).

following are algorithmic methods discussed as under.

Linear Models have the form:

$$Effort = a_0 + \sum_{i=0}^{n} a_i x_i \tag{1}$$

Where, the coefficients $a_1, \ldots, a_n$ are chosen to best fit the completed project data. The work of Nelson belongs to this type of models [13].

Walston-Felix [4] used *Multiplicative Models* have the form:

$$Effort = a_0 \prod_{i=1}^{n} a_i^{x_i} \tag{2}$$

Again the coefficients $a_1, \ldots, a_n$ are chosen to best fit the completed project data. With each x_i taking on only three possible values equal to: -1, 0, +1. Doty model [8] also belongs to this class with each x_i taking on only two possible values either 0 or +1. These two models seem to be too restrictive on the cost factor values.

Power Function Models contains two of the most popular algorithmic models in use, as follows:

- COCOMO (Constructive Cost Model)
- Putnam's Model

COCOMO (Constructive Cost Model) model was proposed by Boehm [12, 2]. The models have been widely accepted in practice. In the COCOMO, the code-size *S* is given in thousand LOC (KLOC) and *Effort* is in person-month. The following are the various types of COCOMO models:

*a) Basic COCOMO***:** The basic COCOMO model is simple and easy to use. As many cost factors are not considered, it can only be used as a rough estimate.

b) Intermediate COCOMO and Detailed COCOMO: In the intermediate COCOMO, a nominal effort estimation is obtained using the power function with three sets of coefficients, with one coefficient being slightly different from that of the basic COCOMO. The overall impact factor (*M)* is obtained as the product of all individual factors, and the estimate is obtained by multiplying *M* to the nominal estimate. The detailed COCOMO works on each sub-system separately and has an obvious advantage for large systems that contain non-homogeneous subsystems.

Putnam's Model is based on Norden/Rayleigh manpower distribution and his finding in analyzing many completed projects [7]. The central part of Putnam's model is called software equation as follows:

$$S = E \times Effort^{1/3} t_d^{4/3} \tag{3}$$

Where, t_d is the software delivery time; *E* is the environment factor that reflects the development capability, which can be derived from historical data using the software equation. The size *S* is in *LOC* and the *Effort* is in person-year. Another important relation regarding effort estimation found by Putnam is shown below:

$$Effort = D_0 \times t_d^3 \tag{4}$$

Where, D_0 is a parameter called manpower build-up which ranges from 8 (entirely new software with many interfaces) to 27 (rebuilt software). Combining the above equation with the software equation, we obtain the power function form:

$$Effort = (D_0^{4/7} \times E^{-9/7}) \times S^{9/7} \tag{5a}$$

And

$$t_d = (D_0^{-1/7} \times E^{-3/7}) \times S^{3/7} \tag{5b}$$

Putnam's model is also widely used in practice.

B. Non-algorithmic Methods

The major non-algorithmic methods are discussed as under:

Expert Judgment method involves consulting one or more experts. The experts provide estimates using their own methods and experience. Expert-consensus mechanisms such as Delphi technique or PERT will be used to resolve the inconsistencies in the estimates. A modification of the Delphi technique proposed by Boehm and Fahquhar [2] seems to be more effective.

Parkinson's principle "work expands to fill the available volume" [6], the cost is determined (not estimated) by the available resources rather than based on an objective assessment. This method is not recommended as it may provide very unrealistic estimates. Also, this method does not promote good software engineering practice.

Many other models such as Price-S [9] and Rayleigh probability distribution [14] Model have also been used in practice.

III. Methodology Proposed

The following steps of the methodology are proposed for modeling of effort estimation:

A. Data Collection

First, Survey of the existing Models of Effort Estimation is to be performed and Secondly, Historical Data being used by various existing models for the cost estimation is collected.

B. Statistical Modeling for Effort Estimation

The following *statistical modeling* approaches are evaluated for data fitting of effort estimation data and the results are compared in terms of *RMSE* values:

Linear Model includes constant and first order terms only. Let there are sixteen inputs depicted as x_1 to x_{16}.The Equation of Linear Model can be written as:

$$y = b_0 + \sum_{i=1}^{16} b_i x_i \tag{6}$$

Where *y* will give the expected values of the response variable and b_i for i = 1,2..16 is the parameters to be estimated.

Pure-Quadratic Model includes constant, linear and squared terms. The Equation of Pure-Quadratic Model can be written as:

$$y = b_0 + \sum_{i=1}^{16} b_i x_i + \sum_{i=1}^{16} b_{ii} x_i^2 \tag{7}$$

Where *y* is output; x_1 to x_{16} are input parameters and other terms are fitting parameters

C. Neuro-Fuzzy, Fuzzy-GA and other Modeling Approaches

The following modeling approaches are used for effort dataset:

- Neuro-Fuzzy Model [11]

- Fuzzy-GA Hybrid Model
- Halstead Model
- Walston-Felix Model
- Bailey-Basili Model
- Doty Model
- GA Based Model [1]

The GA based model developed in [1] is used for the comparison. In case of the Neuro-fuzzy system the first Sugeno Based Fuzzy Inference System is designed that needs the initialization of the Membership Function of the different 16 attributes and linear Membership Function for the output and deducing the fuzzy rules from the data. That Sugeno Based fuzzy inference system is trained with the neural Network using the hybrid training algorithm. In the forward pass the Backpropagation learning algorithm and in the backward pass the LMS learning algorithm is used to update the non-linear and linear parameters of the Neuro-fuzzy system respectively.

The different existing models: Halstead Models, Walston-Felix Model, Bailey-Basili Model and Doty Model are also used for the comparison of results. The comparison of the results is made on the basis of:

- Mean Magnitude of Relative Error (MMRE)
- Root Mean Square Error (RMSE)
- PRED(30)
- PRED(10)

RMSE is 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:

$$\sqrt{\frac{(a_1-c_1)^2+(a_2-c_2)^2+\ldots+(a_n-c_n)^2}{n}} \tag{8}$$

The mean-squared error is one of the most commonly used measures of success for numeric prediction. This value is computed by taking the average of the squared differences between each computed value and its corresponding correct value. The root mean-squared error is simply the square root of the mean-squared-error.

The mean magnitude of relative error (MMRE) can be written as:

$$MMRE = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i} \tag{9}$$

where y_i represents the *ith* value of the effort and $\hat{y}_i$ is the estimated effort.

PRED(N) is the third criteria used for the comparison and this reports the average percentage of estimates that were within N% of the actual values [3]. PRED(N) reports the average percentage of estimates that were within N% of the actual values, as shown in the following pseudo code:

```
count = 0
for(i=1;i<=T;i++) do if MRE.i <= N/100 then count++ fi
done
PRED(N) = 100/T * count
```

For example, e.g. PRED(30)=50% means that half the estimates are within 30% of the actual.

IV. Results & Discussion

Historical NASA's Effort Dataset [3] for the effort estimation is collected and the data is polished so that the same data can be used for the modeling in MATLAB 7.4 environment. The 16 attributes are: analysts capability, programmers capability, application experience, modern programming practices, use of software tools, virtual machine experience, language experience, schedule constraint, main memory constraint, data base size, time constraint for cpu, turnaround time, machine volatility, process complexity, required software reliability and lines of source code.

In the linear fitting the RMSE value is 1.2792e+003 and coefficients of the linear model are: 1.0e+004 * (-1.2379, 0.1675, 0.6488, 0.1175, 0.1549, -0.1900, 0.1466, -0.0868, -0.2282, 0.2313, 0.0261, -0.1640, -0.1527, 0.4970, 0.0716, -0.0039, 0.0006)

The RMSE value for the linear model is: 1.2792e+003.

In the Pure Quadratic fitting the RMSE value is 1.1805e+003 and coefficients calculated are:

1.0e+005*(1.7247, 0.0944, -1.1120, 0.0530, 0.2577, 0.1783, -0.0384, -1.6070, -0.1973, 0.1898, 0.1074, 1.1388, -2.5646, -0.5188, -0.5287, 1.1846, 0.0001, -0.0397, 0.5501, -0.0298, -0.0930, 0.0687, 0.0212, 0.8252, 0.0917, 0.0986, -0.0665, -0.6006, 1.2966, 0.2831, 0.2486, -0.5211,-0.0000)

This shows that the pure quadratic fitting is better than the linear fitting means the data is of non-linear nature.

After the statistical fitting the Fuzzy and Neuro-fuzzy Modelling approach is experimented and the results are compared with the existing modelling approaches. In case of the Neuro-fuzzy approach first the sugeno-based Fuzzy Inference system is designed.

In order to train the Sugeno FIS, Adaptive Neuro-Fuzzy system [11] is created that makes use of the Sugeno FIS Structure as shown in Fig. 1. The following the structure parameters of the Neuro-fuzzy system:

- Number of nodes: 155
- Number of linear parameters: 68
- Number of nonlinear parameters: 128
- Total number of parameters: 196
- Number of training data pairs: 63
- Number of checking data pairs: 0
- Number of fuzzy rules: 4

The NF system is trained for 500 epoch and tested. The plot of data index v/s expected output and actual output is shown in Fig. 2.

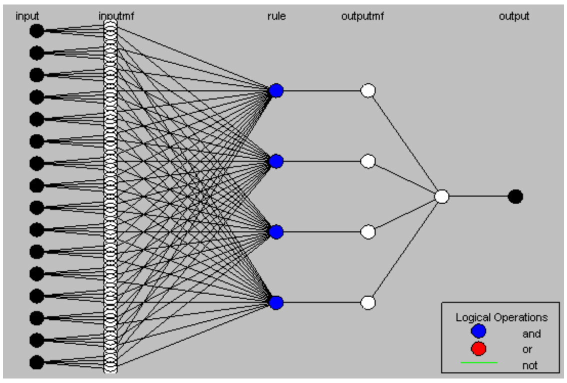

Fig. 1 Structure of the NF Inference System for Effort Dataset

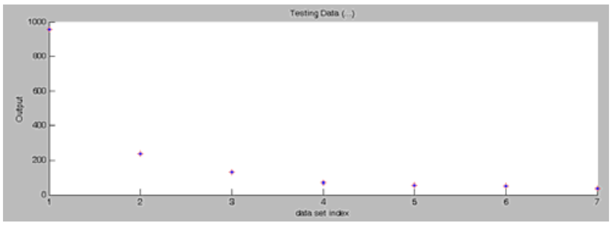

Fig. 2 Testing Results of Trained Neuro-Fuzzy System

Project-wise Results of the different Model used for the Effort dataset is shown in Table I (a).

The results of Fuzzy-GA based hybrid model, Neuro-Fuzzy Model, Halstead Model, Walston-Felix Model, Bailey-Basili Model, Doty Model and GA based Model are shown in Table I (b). The performance criteria taken are MMRE and RMSE.

The results shows that the Neuro-fuzzy Model have the lowest MMRE and RMSE values i.e. 0.014988 and 0.011651 respectively for the testing data. The PRED(30) and PRED(10) values of Neuro-fuzzy Model are maximum among all the models that are being compared i.e. 100 and 100 respectively.

TABLE I (A)
PROJECT-WISE RESULTS OF THE DIFFERENT MODEL USED FOR THE EFFORT DATASET

Actual Effort	Model Used						
	Fuzzy-GA Model	Neuro-Fuzzy Model	Halstead Model	Walston-Felix Model	Bailey-Basili Model	Doty Model	GA Based Model
958	956	958	729.54	14.049	38.897	166.7	54.608
237	237	236.98	364.48	9.2216	25.027	102.7	38.938
130	130	130.01	650	13.099	36.045	153.79	51.621
70	70	69.991	573.58	12.141	33.229	140.94	48.568
57	57	56.998	90.181	3.9521	12.131	38.743	19.712
50	50	49.981	770.44	14.522	40.336	173.17	56.08
38	38	37.985	142.75	5.2219	14.958	53.384	24.657

The Neuro-fuzzy Model shows the best results as compared with the Fuzzy-GA Model and other existing Models that are being used for the Effort Prediction.

TABLE I (B)
RESULTS OF THE DIFFERENT MODEL USED FOR THE EFFORT DATASET

Performance Criteria	Models Used						
	Fuzzy-GA	Neuro-Fuzzy Model	Halstead Model	Walston-Felix Model	Bailey-Basili Model	Doty Model	GA Based Model
MMRE	2.0722	0.014988	887.78	83.584	60.447	126.75	66.528
RMSE	1.6036	0.011651	26575	1880.9	1691.6	1382.1	1815.7
PRED(30)	100	100	3.1746	0	22.222	25.397	14.286
PRED(10)	100	100	1.5873	0	11.111	6.3492	1.5873

V. CONCLUSION

In this study Statistical Models, Fuzzy-GA and Neuro-Fuzzy Inference Systems are experimented to estimate the software effort for projects. The performances of the developed models is tested on NASA software project data presented in [3] and results are compared with the Halstead, Walston-Felix, Bailey-Basili, Doty and Genetic Algorithm Based models as renowned algorithms mentioned in the literature. On comparison, the results shows that the Neuro-fuzzy Model has the lowest MMRE and RMSE values as error values of the developed system during testing i.e. 0.014988 and 0.011651 respectively. During testing the PRED(30) and PRED(10) values of Neuro-fuzzy Model are maximum among all the models that are being compared i.e. 100 and 100 respectively.

Hybrid Fuzzy-GA model, Linear Statistical Models and Pure Quadratic Statistical Model are also developed. But the Neuro-fuzzy Model shows the better results as compared with the Fuzzy-GA based hybrid Inference System, Linear Statistical Models and Pure Quadratic Statistical Model for the Effort Prediction. Hence, the developed Neuro-Fuzzy model is able to provide good estimation capabilities. It is suggested to use of Neuro-Fuzzy technique to build suitable model structure for the software effort.

REFERENCES

[1] Alaa F. Sheta, "Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects", Journal of Computer Science 2 (2): 118-123, 2006.

[2] B. W. Boehm, Software engineering economics, Englewood Cliffs, NJ: Prentice-Hall, 1981, pp. 50-100.

[3] J. W. Bailey and V. R. Basili, “A meta model for software development resource expenditure,” in Proceedings of the International Conference on Software Engineering, pp. 107–115, 1981.

[4] C. E. Walston, C. P. Felix, A method of programming measurement and estimation, IBM Systems Journal, vol. 16, no. 1, pp. 54-73, 1977.

[5] G. Cantone, A. Cimitile, U. De Carlini, A comparison of models for software cost estimation and management of software projects, Computer Systems: Performance and Simulation, Elisevier Science Publishers.

[6] G.N. Parkinson, Parkinson's Law and Other Studies in Administration, Houghton-Miffin, Boston, 1957.

[7] L. H. Putnam, A general empirical solution to the macro software sizing and estimating problem, IEEE Trans. Soft. Eng., pp. 345-361, July 1978.

[8] J. R. Herd, J.N. Postak, W.E. Russell, K.R. Steward, Software cost estimation study:. Study results, Final Technical Report, RADC-TR77-220, vol. I, Doty Associates, Inc., Rockville, MD, pp. 1-10, 1977.

[9] R. E. Park, PRICE S: The calculation within and why, Proceedings of ISPA Tenth Annual Conference, Brighton, England, pp. 231-240, July 1988.

[10] N. A. Parr, An alternative to the Raleigh Curve Model for Software development effort, IEEE on Software Eng., pp. 77-85, May 1980.

[11] R. Jang, Neuro-Fuzzy Modeling: Architectures, Analyses and Applications, Ph.D. Thesis, University of California, Berkeley, 1992.

[12] R.K.D. Black, R. P. Curnow, R. Katz, M. D. Gray, BCS Software Production Data, Final Technical Report, RADC-TR-77-116, Boeing Computer Services, Inc., March, pp. 5-8, 1977.

[13] R. Nelson, Management Hand Book for the Estimation of Computer Programming Costs, AD- A648750, Systems Development Corp., pp. 20-34, 1966.

[14] R. Tausworthe, Deep Space Network Software Cost Estimation Model, Jet Propulsion Laboratory Publication 81-7, pp. 67-78, 1981.

[15] W. S. Donelson, Project Planning and Control, Datamation, pp. 73-80, June 1976.