Volume 3, Issue 5, May 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Neural Network and Fuzzy Logic based framework for Software Development Effort Estimation

Shiyna Kumar, Vinay Chopra
Department of Computer Science and Engineering, D.A.V.I.E.T.
Jalandhar, Punjab, 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. These models too have difficulty in modelling the inherent complex relationships between the contributing factors, are unable to handle categorical data as well as lack of reasoning capabilities. The limitations of these models led to the exploration of the techniques which are soft computing based. In This paper we have compared neural network and fuzzy logic model for software development effort estimation. It will help us to make accurate software effort estimation by these estimation techniques

Keywords : Soft computing; Effort prediction; Neural Network; Fuzzy logic, MRE. MMRE, Prediction.

1. 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.

In this paper, we present a fuzzy logic (FL) framework for effort prediction. The paper is organized as follows. In Section 2, we discuss the eventually development of both algorithmic And non-algorithmic models; Section 3 presents our soft computing-based prediction systems. Section 4 concludes discussions of our various experiments to realize the framework and points out possible directions for future research.

2. EFFORT PREDICTION MODELS

Software effort estimation spawned some of the first attempts at meticulous software measurement, so it is the oldest, most mature aspect of software metrics. Considerable research had been carried out in the past, to come up with a variety of effort prediction models using algorithmic and non-algorithmic techniques. This section discusses the evolution of both algorithmic and non algorithmic estimation techniques eventually. We summarize the section by giving the motivation for our work in this Research.

2.1 Algorithmic Models

The algorithmic models are based on mathematical models that produce effort estimate as a function of a number of variables, which are considered to be the major effort factors. Any algorithmic model has the form:

$$\text{Effort} = f(x1, x2\ldots xn) \qquad \ldots\ldots (1)$$

Where {x1, x2, …,xn} denote the cost factors. The existing algorithmic methods differ in two aspects: the selection of cost factors, and the form of the function f. We will first discuss the cost factors used in these models, and then typify the models according to the form of the functions and whether the models are analytical or empirical.

Boehm was the first researcher to look at software engineering from an economic point of view. He came up with a cost estimation model, COCOMO-81 in 1981, after investigating a large set of data from TRW in the 1970s [4]. Putnam also developed an early model known as SLIM in 1978 [5]. COCOMO and SLIM [6] are both based on linear regression techniques, using data from past projects. Both COCOMO and SLIM take number of lines of code (about which least is known very early in the project) as the major input to their models. Albrecht's function points measures the amount of functionality in a system as described by a specification [6]. A survey on these algorithmic models and other cost estimation approaches is presented by Boehm et al. [2].Most models rely on perfect estimate of either size of software in terms of line of code (LOC), number of user screen, interfaces, convolution, etc. at a time when uncertainty is mostly present in the project [5].The most popular algorithmic estimation models include Boehm's COCOMO, Putnam-slim, and Albrecht's function point.Algorithmic models such as COCOMO, have failed to present appropriate solutions that take into consideration technological advancements. One possible reason why algorithmic models have not proven to provide such solution is because, they are often unable to detain the complex set of relationships (e.g. the effect of each variable in a model to the overall prediction made using the model) that are evident in many software development environments. They can be successful within a particular type of environment, but not flexible enough to adapt to a new environment. Their inability to handle categorical data (that is, data that are specified by a range of values) and most importantly lack of reasoning capabilities (that is, ability to draw conclusions or make judgments based on available data) contributed to the number of studies exploring non algorithmic methods (e.g. FL).

2.2 Non-Algorithmic Models

In Non Algorithmic models some information about the previous projects which are similar under estimate project is required and usually estimation process in these methods is done according to the study of the previous datasets. Algorithmic Effort Modelling, Expert Judgment, Estimation by Analogy, Parkinson's Law.Newer computation techniques to Effort estimation that are non-algorithmic were sought in the 1990s. Researchers curved concentration to a set of approaches that are soft computing-based.Many researchers have contributed towards software development effort prediction using soft computing techniques which handle the imprecision and vagueness in data aptly due to their inherent nature. The first realization of the fuzziness of several aspects of one of the best known [7], most successful and widely used model for cost estimation, COCOMO, was that of Fei and Liu [7]. Fei and Liu observed that an accurate estimate of delivered source instruction (KDSI) cannot be made before starting the project. Therefore, it is awkward to assign a determinate number for it. Jack Ryder [8] investigated the application of fuzzy modeling techniques to two of the most widely used models for effort prediction; COCOMO and the Function-Points models, respectively. Idri and Abran [9] applied FL to the cost drivers of in-between COCOMO model. The application of FL to represent the mode and size as input to COCOMO model was later presented by Musilek et al. In Ref. [10].Musilek et al. presented a two-stage functioning called simple F-COCOMO model and enlarged F-COCOMO model, respectively. A fragment evaluation scheme is given in Section 3 of this paper.

Vachik S. Dave et al. [11] proposed they suggest changes needed in MMRE calculations and propose Modified MMRE algorithm for the effort estimation evaluation criterion. MMRE shows FFNN is a better estimation model than RBFNN. But when we evaluate these models using RSD, as suggested in [1] and Modified MMRE, it shows that RBFNN is more accurate model for effort estimation.

ZeeshanMuzaffar et al. [12] in this paper fuzzy logic based prediction systems could produce further better estimates provided that various parameters and factors pertaining to fuzzy logic are carefully set. This paper show that the prediction accuracy of a fuzzy logic based effort prediction system is highly dependent on the system architecture, the corresponding parameters, and the training algorithms. Modified height defuzzification, triangular membership function and relative error were shown to be performing better than height defuzzification, Gaussian membership function and normalized error, respectively.

Stanislav Berlin et al. [13] represented two types of models that have been employed to estimate project duration and effort separately: linear regression estimation models and models deriving from a more novel approach based on artificial neural networks (ANNs). In order to estimate development effort size and complexity in the early stages of a project are also estimated values and can generate additional noise in the prediction model.

Nonika Bajaj et al. [14] Studies suggest that the software companies should use Bottom up approach unless they have a vast experience from the similar projects. The goal of their research work is to extend the existing bottom up approach to achieve greater precision in the estimates. They proposed the uses and concepts of fuzzy set theory to extend the Bottom up approach to Fuzzy bottom up approach. With the productivity rate generated by fuzzy bottom up, derived values such as effort of development can be more precisely determined.

Cuauhtémoc López Martín et al. [15] describes an application whose results are comparedwith those of a multiple regression. A subset of 41modules developed from ten programs is used as data. Result shows that the value of MMRE (an aggregation of Magnitude of Relative Error, MRE) applying fuzzy logic was slightly higher than MMRE applying multiple regression; while the value of Pred (20) applying fuzzy logic was slightly higher than Pred(20) applying multiple regression. Moreover, six of 41 MRE was equal to zero (without any deviation) when fuzzy logic was applied (not any similar case was presented when multiple regression was applied).

MacDonell et al. [16] explored an expert knowledge based application of FL to effort prediction. This particular research has evolved into the development of a tool, FULSOME, to assist project managers in making predictions. MacDonell also applied fuzzy modeling to software source code sizing in Ref. [5].

3. SOFT COMPUTING-BASED TECHNIQUES

3.1 Artificial Neural Networks for Software Effort Estimation

Many different models of neural networks have been proposed [17].They may be grouped in two major categories. First, feed-forward networks where no loops in the network path occur. Second, feedback networks that have recursive loops. The feed-forward multilayer perceptron with Back-propagation learning algorithm are the most commonly used in the Effort estimation field. In these nets, neurons are arranged in layers and there are only connections between neurons in one layer to the following. Figure 1 illustrates possible network architecture configured for software development effort estimation. The network generates output (effort) by propagating the initial inputs (project attributes) through subsequent layers of processing elements to the final output layer. Each neuron in the network computes a nonlinear function of its inputs and passes the resultant value along its output.

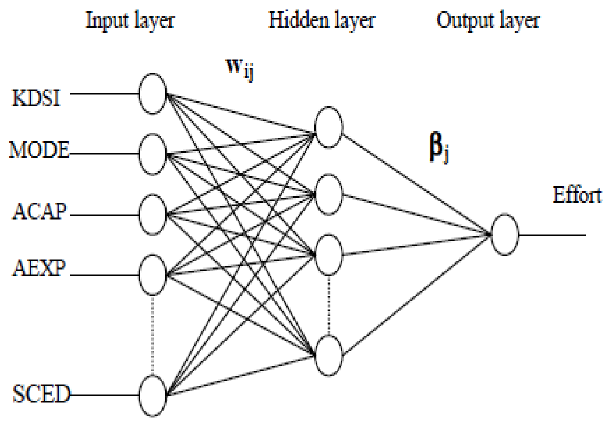

Figure 1: Neural Network Architecture for Software Development Effort

The use of the neural network approach to estimate the software effort requires certain decisions and choices about the architecture, learning algorithm and the activation functions.

3.2 FUZZY LOGIC SYSTEMS

According to the Oxford English Dictionary, the word Fuzzy is defined as blurred, indistinct, imprecisely defined, confused or vague. Fuzzy systems are knowledge based or rule based systems [18]. The heart of a fuzzy system is a knowledge base consisting of the so called fuzzy IF-THEN rules. A fuzzy IF-THEN rule is an IF-THEN statement in which some words are characterized by continuous membership functions. Thus fuzzy logic can be used to handle the imprecision and uncertainty present in the early stages of the project to predict the effort more accurately by incorporating total transparency in the prediction system.

3.3 Intermediate COCOMO Model

COCOMO model was proposed by Boehm in 1981.This model estimates the total effort in terms of "person months" of the technical project staff. It was developed from the analysis of sixty three (63) software projects. The COCOMO model is a set of three models: Basic, Intermediate, and Detailed. The basic COCOMO model computes software development effort(and cost) as a function of program size expressed in estimated lines of code(LOC).The Intermediate COCOMO model computes software development effort as a function of program size and set of "cost drivers" that include subjective assessments of product, hardware, personnel and project attributes. Detailed COCOMO model incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc) of the software engineering process.

4. PROPOSED METHOD

The proposed framework developed an optimized fuzzy logic based framework and reconstruct the Neural Network model for software effort estimation. To evaluate development effort we have used COCOMO NASA data set on proposed developed models. This research is used to handle the imprecision and uncertainty present in the early stages of the project to predict the effort more accurately by incorporating total transparency in the prediction system. This model computes effort as a function of program size and a lot of cost drivers that includes subjective assessment of product attributes, hardware attributes and project attributes. To estimate the effort more accurately we have compared the neural network model and fuzzy logic framework. The evaluation of the models is based upon the following parameters i.e. Mean Magnitude Of Relative Error (MMRE) and Prediction Accuracy (Pred).

$$\text{MMRE} = \frac{1}{n}\sum_{i=1}^{i=n}\frac{|\text{Actual Effort} - \text{Predicted Effort}|}{\text{Actual Effort}} \quad \ldots (2)$$

Where mean magnitude of relative error (MMRE) computes the average of MRE over 'n' projects and 'n' is the number of projects. Generally, the acceptable target value for MMRE is 25%.

$$\text{PRED (l)} = \frac{k}{n} \times 100 \quad \ldots\ldots (3)$$

Prediction (n): Prediction at level n is defined as the % of projects that have absolute relative error less than n. Where, k is the number of projects and n is the number of all predictions.
A model which provides higher value of Pred (n) is better than one with lower value of Pred (n); while model with lower MMRE is better than one with higher MMRE.

5. RESULTS

For experimental analysis, we have chosen 7random projects from COCCOMONASA data set. It shows that the proposed model has MMRE less than FFNN model as shown in Figure 5.1. The comparisons between the results is shown in table 1.

Table1: COMPARISONS OF RESULTS

LOC	**Actual effort**	**FFNN**	**MRE**	**Fuzzy Logic**	**MRE**
423	2300	2765.49	.20	545.0492	.76
79	400	271.05	.32	365.8719	.08
284.7	973	1642.41	.68	854.5155	.12
282.1	1368	1615.92	.18	860.6137	.37
78	571.4	266.97	.53	361.5493	.36
11.4	98.8	96.51	.02	109.6088	.10
19.3	155	108.18	.30	134.6465	.13

Table2: THE MMRE AND PRED COMPARISON BETWEEN ESTIMATION MODELS

Performance Criteria	**FFNN**	**Fuzzy Logic**
MMRE (%)	31.88	27.50
Pred (25 %)	42.88	57.14

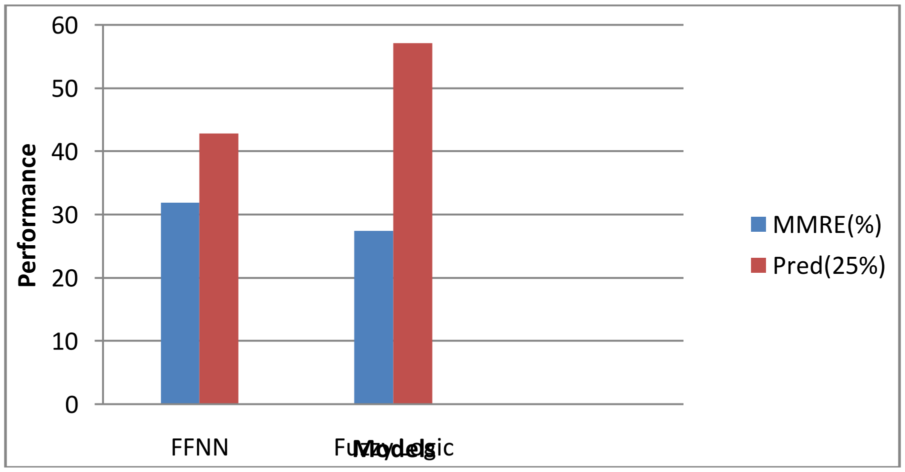

Figure 2: Comparison of Different models

6. CONCLUSIONS

A model which gives lower MMRE is better than that which gives higher MMRE. A model which gives higher Pred(n) is better than that which gives lower Pred (n). Hence from the Table II we can observe that Fuzzy Logic Model is better.

ACKNOWLEDGEMENT

The authors are greatly indebted to the Department of Computer Science and Engineering, D.A.V Institute of Engineering and Technology, Jalandhar for providing excellent lab facilities that make this work possible.

REFERENCES

[1] Harsh Kumar Verma, Vishal Sharma "Handling Imprecision in Inputs using Fuzzy Logicto Predict Effort in Software Development" , IEEE 2010.

[2] ImanAttarzadeh, Siew Hock Ow "Improving Estimation Accuracy of the COCOMO II Using an Adaptive Fuzzy Logic Model" 2011 IEEE International Conference on Fuzzy Systems June 27-30, 2011, Taipei, Taiwan

[3] S.G. MacDonell, Software source code sizing using fuzzy logic modeling, Information and Software Technology 45 (2003) 389–404.

[4] Zonglain, F. and xihui L., "F-COCOMO:FuzzyConstrutive Cost Model In Software Engineering" , In Proc. Of Ieee International Conference On Fuzzy Systems.1992,pp.331-337

[5] C. Schofield, Non-algorithmic effort estimation techniques, Technical Reports, Department of Computing, Bournemouth University, England, TR98-01, March 1998.

[6] 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 ComputingApplied to Software Engineering, 2001.

[7] Z. Fei, X. Liu, f-COCOMO: fuzzy constructive cost model in software engineering, Proceedings of the IEEE International Conference on Fuzzy Systems, IEEE Press, New York, 1992, pp. 331–337.

[8] J. Ryder, Fuzzy modeling of software effort prediction, Proceedings of IEEE Information Technology Conference, Syracuse, NY, 1998.

[9] Allidri And Alain Arban , LailaKjiri "COCOMO Cost Model Using Fuzzy Logic"7th International Conference On Fuzzy Theory & Technology Atlantic City,New Jersey,February27-March3,2000

[10] P. Musilek, W. Pedrycz, G. Succi, and M. Reformat, " Software cost estimation with fuzzy models", Applied Computing Review, 8(2)2000 24–29.

[11] Vachik S. Dave KamleshDutta, "Neural Network based Software Effort Estimation & Evaluation criterion MMRE" International Conference on Computer & Communication Technology (ICCCT)-2011.

[12] ZeeshanMuzaffar , "Moataz A. Ahmed, Software development effort prediction: A study on the factors impactingthe accuracy of fuzzy logic systems" Information and Software Technology 52 (2010) 92–109,2009.

[13] Stanislav Berlin, TzviRaz , "ChananGlezer, Moshe Zviran, Comparison of estimation methods of cost and duration in IT projects" Information and Software Technology 51 (2009) 738–748, 2008.

[14] Nonika Bajaj, AlokTyagi and RakeshAgarwal, "Software Estimation – A Fuzzy Approach", ACM SIGSOFT Software Engineering Notes, May 2006.

[15] Cuauhtémoc López Martín, "Software Development Effort Estimation Using Fuzzy Logic: A Case Study" Proceedings of the Sixth Mexican International Conference on Computer Science (ENC'05), 0-7695-2454-0/05 $20.00 © IEEE 2005.

[16] S.G. MacDonell, A.R. Gray, M.J. Calvert, FULSOME: "A fuzzy logic modeling tool for software metrician" in: Proceedings of the 18th International Conference of the North American Fuzzy Information Processing Society—NAFIPS, IEEE Press, New York, 1999,pp. 263–267.

[17] G. Wittig, G. Finnie, Estimating software development effort with connectionist models, Information and Software Technology 39 (1997) 469–476.

[18] S.G. MacDonell, A.R. Gray, M.J. Calvert, FULSOME: a fuzzy logic modeling tool for software metricians, in: Proceedings of the 18th International Conference of the North American Fuzzy Information Processing Society—NAFIPS, IEEE Press, New York, 1999,pp. 263–267.

[19] A.C. Hodgkinson, P.W. Garratt, A neurofuzzy cost estimator, in: Proceedings of the Third International Conference on Software Engineering and Applications—SAE, 1999, pp. 401–406.