



# Comparison of hard limiting and soft computing methods for predicting software effort estimation: In reference to Small Scale Visualization Projects

T.M Kiran Kumar\*<sup>1</sup>, M. A. Jayaram <sup>2</sup>

Department of Master of Computer Applications  
Siddaganga Institute of Technology, Tumkur-572103, Karnataka,  
\*Corresponding author E-mail: [tmkiran@yahoo.com](mailto:tmkiran@yahoo.com)\*

## Abstract:

It is a well known fact that software effort estimation is exceptionally critical in every software industry, particular during the development of projects. It is hard to estimate the parameters involved due to ambiguity and uncertainty associated with the parameters. It is exactly here the hard limiting techniques, soft computing techniques comes to play. In this unique circumstance, this paper, presents an attempt to that compare the two paradigms for effort estimation. For this, we have considered fifty real time small visualization projects thrive by post graduate students. The prototype development involves following stages:

- i) Elicitation of seven novel parameters namely Lines of Code, Cumulative Grade Point Average, New and changed code, Reused code, Cyclomatic Complexity, Algorithmic Complexity and Functional Points.
- ii) Developing of hard limiting methods and soft computing methods for prediction of software effort involved in terms of duration in minutes.

For the validation of the models error metrics namely: Mean Absolute Error (MAE), Mean Magnitude of Relative Error (MMRE), Mean of Magnitude of error Relative to the Estimate (MMER) and Root Mean Square Error (RMSE) have been used. The result showed that the models compared very well with marginal difference in terms of predict values of error matrix.

**Keywords:** Feed-Forward neural networks; Fuzzy Logic; Multiple linear regression models; multiple non-linear regression models; Software effort estimation.

## Nomenclature

MAE-Mean Absolute Error, MMRE-Mean Magnitude of Relative Error, MMER-Mean of Magnitude of error Relative, RMSE-Root Mean Square Error, SD: Student developer, LOC-Lines of code, CGPA-Cumulative Grade Point Average, N&C-New &Change, R-Reused code, CC-Cyclomatic complexity, AC-Algorithmic complexity, FP-Function point, AE: Actual effort FLM- Fuzzy Logic Models, LRM- Linear Regression Model, FLS- Fuzzy Logic System.

## 1. Introduction

Estimation of the effort is the way towards the effort enforced to build a model placed on deficient, vague or questionable guidance. It manages by the forecast of generally apparent outlay and time to undock the task. The software reckoning system can be congregate into three classes:

(a) **Expert judgment:** The result of a broad assortment of estimation nearing and it intends to get gauges rely upon an association of specialists on associated models.

(b) **Algorithmic models:** It endeavours to perform the dependency among one or more attributes of a project and effort.

(c) **Machine learning:** Large numbers of techniques are used as complement to the above two techniques they are Fuzzy models, Regression trees & CBR.

The literature survey revealed that the majority of the reported software endeavour reckoning models are based on public available data sets with input parameter not more than two-to-three [1]. In huge number of cases the effort estimation model seemed to be regression type. It is common felling among the researcher in this domain that the varied complexity of the projects is the main cause for being unable to predicate a generic estimation model.

The Fifty visualization projects used here in this work for the comparison of hard limiting technique and soft computing methods. In hard limiting techniques we have used two regression models and in soft computing method we have used neural network and fuzzy inference system with the development of the fuzzy predictive model on the proceeds of the real-time laboratory work performed by the fifty post graduate students. The domain of projects ranged in the realms of science and engineering.

The research work presented in this paper is unique and first of its kind in terms of the project development as well as the parameters elicitation. Here unlike other reported work we have not used the ready dataset which is available in the communal website, in this work we have used seven parameters have been considered.

## 2. Related Works

In by far most of the examination, the dataset identifying with the COCOMO 81 has endure for building up the estimation models like fuzzy logic model, fuzzy regression and fuzzy neural network model have been developed by the researchers [16][17][18][19]. These models vary somewhat as far as expectation exactness and significantly as far input attributes brief work by [20] has point out that the accuracy of two unlike in nature with the method of retrogression.

Considering the enhancement of software development effort estimation fuzzy logic has retain down by numerous researchers using mamdani fuzzy inference system [2][3][4][5][6][7][8], Muzaffar et al. [5] accord that the ordinary access to be utilized for exertion expectation like use of models imitative from historical data or the use of expert judgement to obtain an robustness. The work which is results with the usage of fuzzy logic which deals with deception used to outline in the beginning stages of application life cycle.

According to Ahmed et al. the author conducted set forth and fuzzified the COCOMO model in two different parts i.e., nominal effort estimation and the adjustment factor. The author contemplated fuzzy logic framework for effort prediction by consolidating using the COCOMO model [9]. The issue of this outcome utilizing the fuzzy predictive models can outperform the other techniques.

For internal product measure using a neural network approach the author Boetticher [10] evaluated the abilities of assessed the Neuro-fuzzy system in contrast with other estimation methods and models [11]. The author combines with the Neuro-fuzzy systems and valuable learning for modelling aspects of neural networks with the lingual holdings of fuzzy entity.

Lopez-Martin [12] has used the fuzzy inference system to estimate the effort of software projects inception from the short scale programs. So the author assimilated the change code as well as reused code which was parameters for the small programs programmed by engineers applying the practices of PSP. The data act as i/p for the Fuzzy model to find the effort. The precision of this model was correlated with other models to figure out the results.

## 3. Methodology

Post graduate students of master of computer applications in their third semester were allotted different kinds of projects which are visualization projects based covering different sorts of endeavours. We have tried to simulate the industry environment in the academic setting by a tight project completion and report submission deadlines. To track down this we asked the every developer to maintain the project diary. It is mere rigid to predict the estimate the software effort at the initial stages. This project enables stake holders to predict effort by providing simple parameters values. Fifty mini projects are used to predict the effort estimation. We have considered seven parameters, seven to one mapping, and effort as an output.

### 3.1 Extraction of Attributes

The seven parameters mention in the preceding paragraph were manually computed for all the projects and treated as inputs. The output is the development effort in terms of duration (minutes) consumed for through development of the project. A brief explanation of input parameters is made in the following. A detailed elaboration of the steps involved is available in ref [15]

#### LOC

This attributed is calculated manually by counting the line of code of all the fifty projects and indicate in terms of integer.

#### N&C

This parameter which consists of the code which is modified and added code. The LOC which is drafted during the development of project is termed as an added code and the code which is modified for the previous developed project is termed as modified code. This term is expressed in integer.

#### R

Reused code(R) is the segment of line of code which has been frequently used in different instance of running code. This feature is also expressed as an integer.

#### CGPA

In this case the project developer being post graduate students with the fundamental understanding of allied subjects like programming concepts, computer graphics, data structures and design of algorithms are capable enough to develop a project in the given domain making use of the knowledge that they had gain in the previous semester. In an academic setting is in the range of 5 to 10, 5 corresponds to 'D' grade and the score 10 corresponds to 'S' grade is deemed to reflect intellect capability of the student.

#### CC

This is a metric (measurement), used to indicate the intricacy of program. In essence it is measure of non sequential path taken by the number of control during evaluating source code. This cadent was connoted in terms of integer.

#### AC

This value was reckoned by diagnose using the basic related efficiency case. In this work we used the constructs such as for loop, while and do...while loops were only considered not the other terms. From among the projects AC ranged from lower complexity  $\Theta(\log n)$  to higher complexity  $\Theta(n^2)$  where n is the input size. In order to have unbiased measure of this complexity an input size of ten was considered to all the projects.

#### FP

This is a disciplined approach for problem-untangle. In this method the system is broken down into smaller components to measure the size of the software. This metric exceptionally are taking in the account for the source code errors in programming languages [13]. A function point connotes the functional aspects of program i.e. number of independent procedural units.

## 4. Hard Limiting Methods

### 4.1 Multiple Linear Regression models (MLR)

Pragmatically the regression analysis has been the predominant used model for software effort predictive modelling. Here, the dependent variable is effort (in terms of minutes) and independent variables are the factors which influence or carry out the effort. Specifying the nature of the regression model as to whether linear, non linear and also finite number of input parameters. This process in all involves trial and error methods to hinge upon predictive model which could foresee the most accurate predictions

In this paper we have used eight variant multiple linear regression models were refined using the data which is availed. The software we have used for the regression equation was R tool. An acceptable value of the coefficients of determination was found to be which is enough to qualify the equation to be acceptable and turns 0.9812 is adequate because the acceptable value should be greater than or equal to 0.5[21]. Accordingly the coefficient indicates that there exists a statistically significant relationship between the variables at 99% confidence level. Since the p-value is greater than 0.05 the variables are statistically significant [21].

### 4.2 Multiple Non-Linear Regression models (MNLr)

The data set which was used for the linear regression model was used even for the multiple non linear regression model and this model happens to be a second degree curve.

The analyses of results are made the very low p-value is indicative that the equation is statistically significant even for the multiple non linear regression models.

## 5. Soft Computing Methods

### 5.1 Feed-Forward neural networks

A neural system is developed to make a non linear mapping of input parameter and software effort. The anatomy comprised of accommodating 7 inputs, in the central layer having 10 neurons and output layer consists of single neuron layer comprising the effort. The extensive procedure involved majorly in arriving at optimal number of central layer neurons. A series of experiments runs of different topology of neural network were conducted starting from 10 up to 40 neurons. However a neural network model with 10 neurons in the central layer showcased encouraging results. This has shown in the below Figure-1

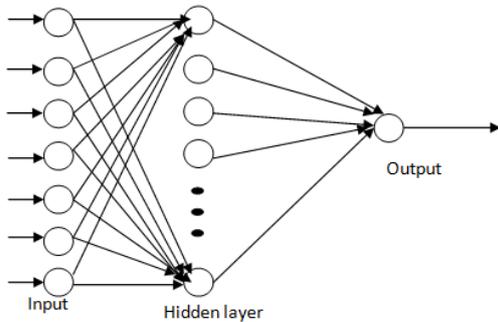


Figure-1: Topology of ANN

### 5.2 Fuzzy inference systems

A schematic view of fuzzy membership function used for input variables and output variables are represented by the below two figures (2), (3).

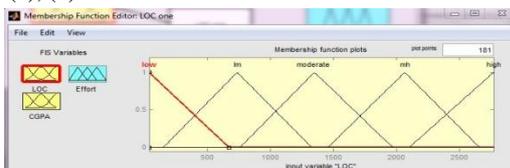


Figure 2: Fuzzy input membership functions for LOC

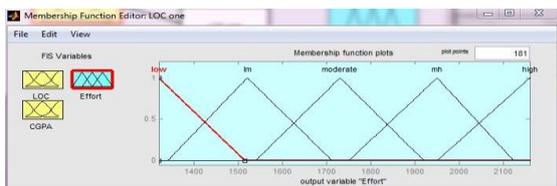


Figure 3: Fuzzy output membership functions of the Effort

Figure 2 shows, the triangular membership functions for the parameters. The corresponding output membership functions were also five in number with same linguistic names as that of inputs. Figure 3 shows the output membership functions for the LOC and CGPA as input.

A fuzzy rule base is shown in Figure 4 consists of ten rules. Similar one-to-one mappings were done for the remaining five

input parameters and an appropriate number of output membership functions and fuzzy rules were constructed.

1. If (LOC is low) and (CGPA is low) then (Effort is low) (1)
2. If (LOC is im) and (CGPA is im) then (Effort is im) (1)
3. If (LOC is moderate) and (CGPA is moderate) then (Effort is moderate) (1)
4. If (LOC is mh) and (CGPA is mh) then (Effort is mh) (1)
5. If (LOC is high) and (CGPA is high) then (Effort is high) (1)
6. If (LOC is low) and (CGPA is im) then (Effort is im) (1)
7. If (LOC is im) and (CGPA is moderate) then (Effort is mh) (1)
8. If (LOC is moderate) and (CGPA is mh) then (Effort is moderate) (1)
9. If (LOC is mh) and (CGPA is high) then (Effort is mh) (1)
10. If (LOC is high) and (CGPA is mh) then (Effort is high) (1)

Figure 4: A typical rule base for one-to-one mapping.

## 6. Accuracy Criterion

Literature survey revealed that MAE(1), MMRE(2), MMR(3) and RMSE(4) are necessary and sufficient yard sticks for the evaluation of any effort estimation models. The value of MMR<-0.25 has been considered to be acceptable limit [14]. Formulations are presented below:

$$MAE = \frac{1}{TP} \left( \sum_{i=1}^{TP} |AE_i - PE_i| \right) \tag{1}$$

$$MMER = \frac{1}{TP} \left( \sum_{i=1}^{TP} \frac{|AE_i - PE_i|}{PE_i} \right) \tag{2}$$

$$MMRE = \frac{1}{TP} \left( \sum_{i=1}^{TP} \frac{|AE_i - PE_i|}{AE_i} \right) \tag{3}$$

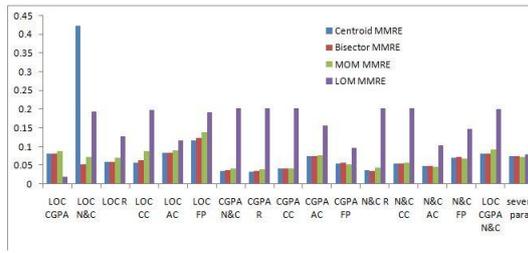
$$RMSE = \sqrt{\frac{\sum_{i=1}^{TP} (AE_i - PE_i)(AE_i - PE_i)}{TP}} \tag{4}$$

## 7. Comparisons of Hard Limiting and Soft Computing Methods

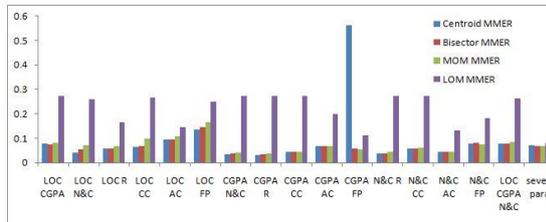
Table 1 presents a relative comparison of models results in terms of errors. It can be seen from the table that all the methods excluding the fuzzy inference system have shown negligible MMER, very low p-value and very high regression coefficients, while being fuzzy inference system prove to be little erroneous in terms of predication accuracy, where as in fuzzy the model behaviour in terms of MAE, RMSE, MMRE and MMER for all the fifty projects presented from figure 5 (a)- figure 5 (d). These figures bring out the values of the error measures under different defuzzification methods

Table 1: Comparative model

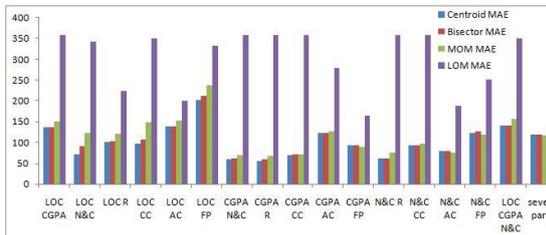
Model	MMER	p-Value	r <sup>2</sup>
MLR	0.00188	0.00025	0.9841
MNLr	0.000577	0.00021	0.9924
NN	0.00457	0.00021	0.9821
Fuzzy	0.0401	0.00041	0.8252



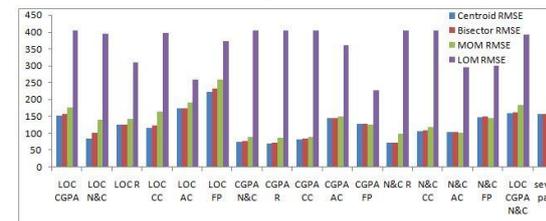
MMRE  
Figure 5(a)



MMER  
Figure 5(b)



MAE  
Figure 5(c)



RMSE  
Figure 5(d)

Figure 5: Comparison of MMRE for various Defuzzification methods

## 8. Results and Discussions

The modelling involved intensive computational experiments. Among 50 records availed, 70% of the data were used to train the system and the remaining 30% were used for testing the system prediction. The harvests in almost all the cases were found to be highly correlating and matching with the actual values.

The two figures 6 & 7 shows the relationship at intervals for predicted effort and actual effort and de-fuzzification methods are considered. Polynomial was found to be the best fit. Figure 6(a) shows the prediction behaviour of the system when two inputs are considered, here again, polynomial curve was in good agreement. Figure 6(b) shows the prediction behaviour of the system when three inputs are considered and figure 6(c) shows the prediction behaviour of the system when seven inputs are considered using centroid method. Figure 7 shows the relationship between actual effort and predicated effort for two input parameters using bi-sector method here again polynomial curve was in good agreement.

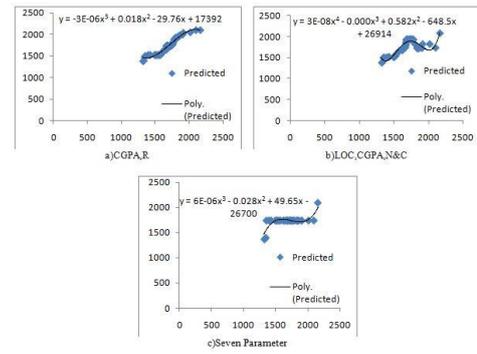


Figure 6: Scatter Plots showing actual effort v/s predicted effort for seven parameters using Centroid method

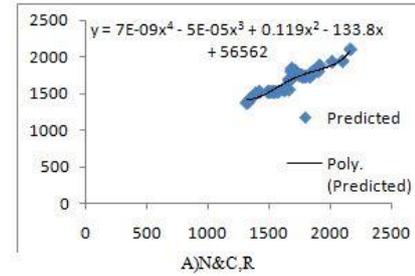


Figure 7: Scatter Plots showing actual effort v/s predicted effort for Bisector method

Figure 8 shows a 3D view of the variation of yield when three input parameters (LOC, CGPA, N&C) are considered. The high magnitude of errors in the predicted values was noticed for other combinations of input-output. Such high errors may be attributed to non-interrelationships among the input parameters when they are considered in combinations.

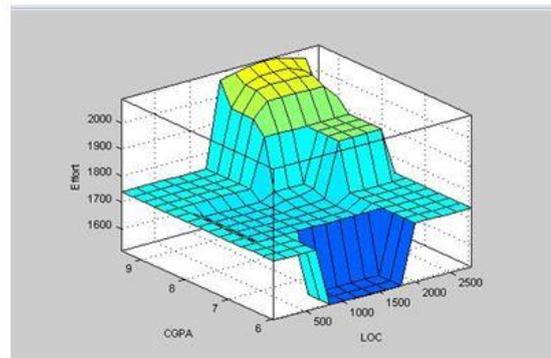


Figure 8: 3D view of three-to-one mapping.

## 9. Conclusions

This paper presents the comparison approach adapted to predict software effort involved while developing fifty varied small visualization projects attempted by post graduation students in almost industry setting. This paper also brought out a comparative analysis of hard limiting effort estimation technique such as linear, nonlinear regression and two soft computing techniques namely neural network and fuzzy inference system.

The results almost are similar even if the methods are different as far as magnitude of errors metrics are concern. However fuzzy inference system showed the errors at a higher level which may be attributed to the method itself. That is fuzzy inference system is based upon approximation reasoning. The work presented was also novel in the sense that the data use for the developing models are garnered through realistic computational experiments involving post graduation students, unlike other works found in

vast literature which have involved data drawn from the public dataset for developing models.

## Reference

- [1] E. Mendes, N. Mosley, I. Watson, A comparison of case-based reasoning approaches to web hypermedia project cost estimation, in: Proceedings of the 11th international conference on World Wide Web, ACM, (2002), pp.272-280.
- [2] Ali Bou Nassif, Danny Ho, Luiz Fernando Capretz, Towards an early software estimation using log-linear regression and a multilayer perceptron model, *Journal of Systems and Software*, Volume 86, Issue 1, January (2013), Pages 144-160, ISSN 0164-1212,10.1016/j.jss.2012.07.050.
- [3] Lopez-Martin C.,( 2011). A fuzzy logic model for predicting the development effort of short scale programs based upon two independent variables. *Applied Soft Computing*. Pp 724-732.
- [4] Bathla R., Singh S., Vishav M., Gupta R.,( 2010). Innovative scenario in software development effort estimation based on a new fuzzy logic model. *International Journal of Information Technology and Knowledge Management*. July-December 2010, Volume 2, No. 2, pp. 361-364.
- [5] Muzaffar Z. And Ahmed M.A., (2010). Software development effort prediction: A study on the factors impacting the accuracy of fuzzy logic systems. *Information and Software Technology* 52 (2010) 92–109.
- [6] Ahmed M.A., Muzaffar Z., (2009) Handling imprecision and uncertainty in software development effort prediction: A type-2 fuzzy logic based framework. *Information and Software Technology* 51. Pp. 640–654.
- [7] Attarzadeh I. and Hock O. S., (2009). Software Development Effort Estimation Based on a New Fuzzy Logic Model. *International Journal of Computer Theory and Engineering*, Vol. 1, No. 4.
- [8] Nisar M.W., Wang Y-J., Elahi M., Khan I.A., (2009).Software development effort estimation using fuzzy logic. *Information Technology Journal*.
- [9] M. A. Ahmed, M. O. Saliu, and J. AlGhamdi, “Adaptive fuzzy logic-based framework for software development effort prediction”, *Information and Software Technology Journal*, Volume 47 (2005), pp. 31-48.
- [10] 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).
- [11] A. C. Hodgkinson, and P. W. Garratt, “A neuro fuzzy cost estimator”, *Proceedings of the Third International Conference on Software Engineering and Applications—SAE* (1999), pp. 401–406.
- [12] Cuauthemoc Lopez-Martin, A Fuzzy Logic model for predicting the development effort of short scale programs based upon two independent variables, *Applied Soft Computing* 11(2011) 724-732.
- [13] PichaiJodpimai, PeraphonSophsathit, and ChidchanokLursin- sap, ”Analysis of Effort Estimation based on Software Project Models”, *IEEE*, 2009.
- [14] S.D. Conte, H.E. Dunsmore and V.Y. shen, “Software engineering metrics and models” *European Journal of Operational Research* 28(2), (1987), PP.235-236.
- [15] M.A.Jayaram,T.M.KiranKumar,H.V.Raghavendra. “Models for predicting development effort of small – scale visualization projects”, *Journal of Intelligent Systems*, 20160247, ISSN (Online) 2191-026X, ISSN (Print) 0334-1860, DOI: <https://doi.org/10.1515/jisys-2016-0247>.
- [16] E. Mendes, N. Mosley, I. Watson, A comparison of case-based reasoning approaches to web hypermedia project cost estimation, in: Proceedings of the 11th international conference on World Wide Web, pp.272-280,ACM(2002)
- [17] S.G. MacDonell, A.R. Gray, Alternatives to regression models for estimating software projects, in: Proceedings of the IFPUG Fall Conference, Dallas TX, IFPUG,(1996)
- [18] Crespo, F.J., Sicicila, M.A., Cuadrado, J.J., On the use of fuzzy regression in parametric software estimation models: integrating imprecision in COCOMO cost drivers. *WSEAS Transactions on Systems*, (2004).
- [19] Yunsik Ahn, Jungseok Suh, Seungryeol Kim, Hyunsoo Kim, “The software maintenance projects effort estimation model based on function points”, *Journal of Software maintenance and evolution*, Volume 15, Issue 2, pp. 71–85, ( 2003)
- [20] Heiat, A.: Comparison of artificial neural network and regression models for estimating software development effort. *Journal of Information and Software Technology*, 44(15), pp.911-922, (2002).
- [21] W. Humphrey W., “A Discipline for Software Engineering. Addison Wesley”(1995).