ISSN:0975-9646
Roheet Bhatnagar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (4) , 2011, 1675-1678

Predicting the Early Stage Software Development Effort using Mamdani FIS

Roheet Bhatnagar[#], Mrinal Kanti Ghose[#], Vandana Bhattacharjee[*]

[#] *Department of Computer Science and Engineering,Sikkim Manipal Institute of Technology*
Majitar, Rangpo, East Sikkim, INDIA

[*]*Department of Computer Science and Engineering, BITEC, Lalpur, BIT Mesra, Ranchi, Jharkhand, INDIA*

***Abstract*— Software development effort prediction is one of the most challenging activities in software project management. There are several cost/effort estimation models which have been proposed over the years and still no model can accurately estimate the effort required to develop the software product. Each method has their own pros and cons in estimating development cost and effort. Estimation of efforts in the early stages is all the more difficult and that is because project data, available in the early stages of project is often incomplete, inconsistent, uncertain and unclear. Although data is incomplete and inconsistent but then also there are enough information present which makes it possible to estimate the effort at early stages. The advantage is that, if we can predict the effort at the early stages say, at the design phase itself, then the project manager can provide better estimates and based on that an efficient schedule can be developed so as to complete the project well within budget and time. Soft computing techniques are gaining popularity for predicting software development effort estimations. In this paper we have tried to predict the effort estimations using Mamdani Fuzzy Inference System (FIS) using fuzzy Triangular (trimf) MF. The results were analyzed using MRE evaluation criteria. The need for accurate effort prediction in software project management is still a challenge for researchers world over.**

***Keywords*— Software Development Effort, MRE, Mamdani FIS, Membership Functions, Fuzzy Rules**

INTRODUCTION

Developing a software project with acceptable quality within budget and on planned schedule is the main goal of every software development firm. Schedule estimation has historically been and continues to be a major difficulty in managing software development projects [1]. Failure of the project mostly is attributed to failure to fulfil customers' quality expectations or the budget and schedule over-run. There are different methods of software development cost estimation proposed over the years each having their own pros and cons. No one method was found to accurately estimate the effort required to develop a software project.

Soft Computing is a new science and the fields that comprise Soft Computing are also rather new. Various characteristics of soft computing techniques make them the potential drivers for carrying out software development effort estimations. Soft computing techniques based effort estimation models were found to estimate the effort more accurately as compared to the standard statistical techniques like Regression analysis etc. [2]

I. SOFTWARE DEVELOPMENT AND SOFTWARE DEVELOPMENT EFFORT ESTIMATION

Software development project is a collection of efforts and resources in a defined time period to realize a software product which satisfies the requirements made by a client or agreed upon [3,4]. Project management focuses on suitable application of efforts and resources to achieve the constraints of Cost, Time and Quality. From very first day, the planning for efforts and resources is conducted based on estimates. Estimation is key to the planning and is made not only at the beginning but also at every single milestone. Current research in estimation is focused on issues like development of new models, metrics conversion, uncertainty, missing data, intelligent decision support and models for new life cycles [3-7]. In software development effort estimation, a large set of factors has been identified [4, 9] which affects the final effort and the productivity of the organization.

Early stage effort estimations can be defined as making software development effort estimations at the initial stages more precisely, the Design stage of SDLC. Carrying out effort estimations at the early stages is beneficial because the design stage prediction implies fewer overheads at the later stages of software development. This paper provides an approach for carrying out early stage effort estimations using fuzzy logic technique.

A. FUZZY LOGIC

Fuzzy logic is a methodology, to solve problems which are too complex to be understood quantitatively, based on fuzzy set theory [10,11]. Use of fuzzy sets in logical expression is known as fuzzy logic and a fuzzy set is characterized by a membership function. A fuzzy set is defined as the extension of a crisp (classical) set which allows only full membership or no membership to its elements [11]. Figure 1 represents the classical set showing the members and non-members of the set A.

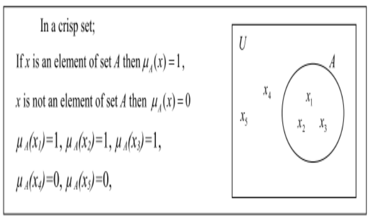


Figure 1: Representation of a crisp (classical) set

B. FUZZY MEMBERSHIP FUNCTIONS

A membership function (MF) [12] is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is also called as the universe of discourse. The main challenge of the fuzzy logic theory is the rejection of any object belonging to a single set. Instead, this approach suggests partial belongings of any object to different subsets of a universal set. The membership function may be triangular, trapezoidal, parabolic etc. However, in practical applications triangular and trapezoidal functions are preferred as simple linear functions [13] as depicted in Figure 2. Triangular membership function has been used in the present investigation.

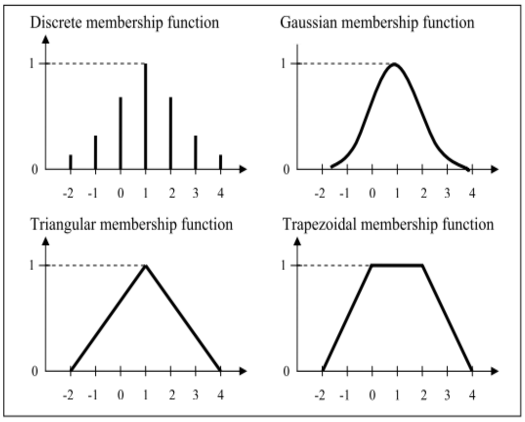


Figure 2: Membership functions for “x is close to 1”

II. METHODOLOGY

A. Student Dataset and Mamdani FIS discussion

Experiments have been conducted on a dataset (as given in Appendix - I) prepared by us based on the Major Project reports of the B.Tech. students of Sikkim Manipal Institute of Technology. [14] Mamdani type fuzzy inference system was implemented using Fuzzy Logic Toolbox, a GUI tool of Matlab. For writing the rules, the inputs and outputs of the system are to be identified. To obtain a fuzzy model from the data available, the steps to be followed are,

1. Select a Mamdani type Fuzzy Inference System.
2. Define the input variables and output variable.
3. Set the type of the membership functions (triangular mf) for input variables.
4. Set the type of the membership function (triangular mf) for output variable.
5. The data is now translated into a set of if–then rules written using Rule editor.
6. A certain model structure is created, and parameters of input and output variables can be tuned to get the desired output.

So from the dataset, we have taken TCOE and CGPA as two input variables and RDE as the output variable for preparing Mamdani FIS for estimating the effort required for software development.

Two inputs and one output Mamdani FIS is as given in Figure 3 while the FIS specifications are as listed in the Table 1.

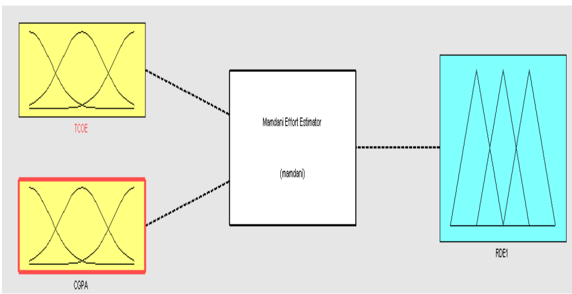


Figure 3: Mamdani Model for effort prediction

TABLE 1
FIS Specifications

Fuzzy Inference System Type	Mamdani
And Method	Min
Implication Method	Min
Aggregation Method	Max
Defuzzification	Centroid
Membership Function Type	triangular

Figures 4, 5, and 6 shows the membership function for two input variables namely Total count of Entities (TCOE), Cumulative Grade Point Aggregate (CGPA) and one output variable as Redistributed Development Effort (RDE) respectively. TCOE has three linguistic values as Low, Medium and High, CGPA has Poor, Average and Excellent while RDE has Low, Medium and High.

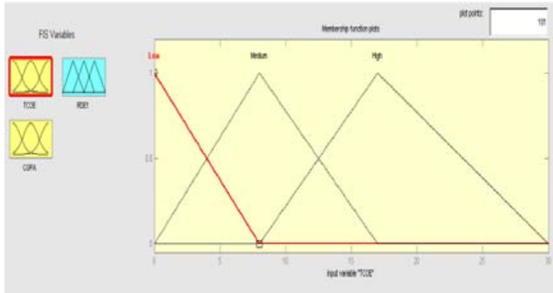


Figure 4: Membership Function for TCOE

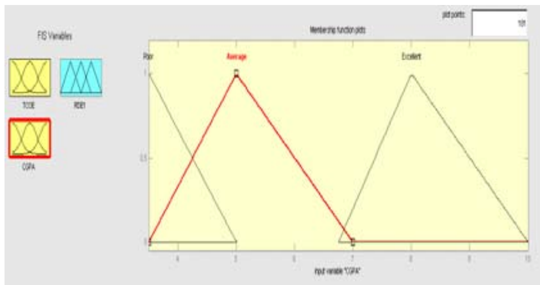


Figure 5: Membership Function for CGPA

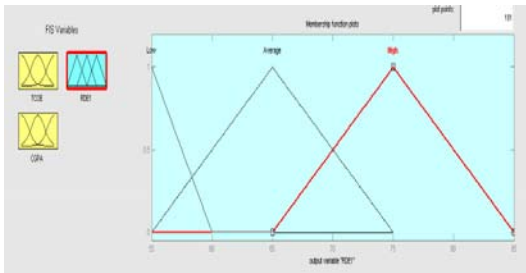


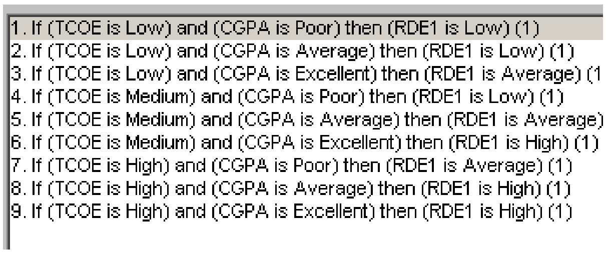
Figure 6: Membership Function for RDE1

Once the inputs are fuzzified, the fuzzy rule base are applied to arrive at the fuzzy output. Fuzzy Inference Rule Base as shown in Table 2, comprises different Fuzzy Rules (nine in numbers) which are shown in Figure 7. Fuzzy inference process which is performed automatically is as depicted in Figure 8.

Figure 9 visualizes the rule surface of the fuzzy system which has been developed to predict the effort at the early stages of software development.

TABLE 2
Fuzzy Rule base for Mamdani Effort Estimator

		CGPA		
		Poor	**Average**	**Excellent**
TCOE	**Low**	L	L	M
	Medium	L	M	H
	High	M	H	H



1. If (TCOE is Low) and (CGPA is Poor) then (RDE1 is Low) (1)
2. If (TCOE is Low) and (CGPA is Average) then (RDE1 is Low) (1)
3. If (TCOE is Low) and (CGPA is Excellent) then (RDE1 is Average) (1
4. If (TCOE is Medium) and (CGPA is Poor) then (RDE1 is Low) (1)
5. If (TCOE is Medium) and (CGPA is Average) then (RDE1 is Average)
6. If (TCOE is Medium) and (CGPA is Excellent) then (RDE1 is High) (1)
7. If (TCOE is High) and (CGPA is Poor) then (RDE1 is Average) (1)
8. If (TCOE is High) and (CGPA is Average) then (RDE1 is High) (1)
9. If (TCOE is High) and (CGPA is Excellent) then (RDE1 is High) (1)

Figure 7: Fuzzy Rules for Mamdani Effort Estimator

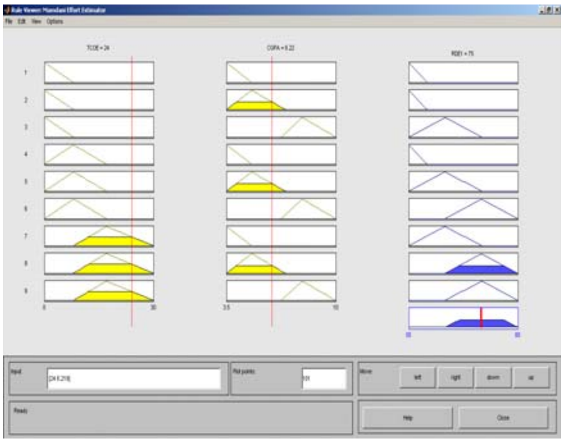


Figure 8: Fuzzy Inference Process for Mamdani Effort Estimator

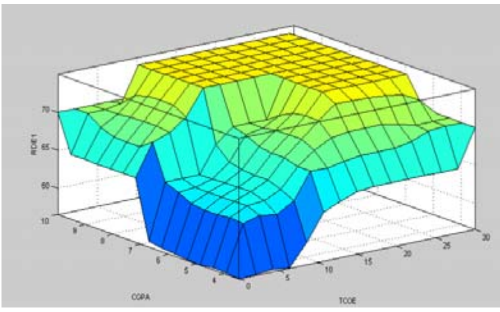


Figure 9: Rule surface for Mamdani Effort Estimator

B. Evaluation Criteria Used

There are many evaluation criteria to evaluate the accuracy of the software development effort in literature. The Mean Magnitude Relative error (MMRE) is a widely accepted criterion in the literature and is based on the calculation of the magnitude relative error (MRE). Eq. (1) below shows an equation for computing the MRE value that is used to assess the accuracies of the effort estimates. Here, the Yj represents the actual effort while $\hat{Y}_j$ is the estimated effort for the project j.

$$\mathrm{MRE}_j = \frac{|Y_j - \hat{Y}_j|}{Y_j} \qquad \text{Eq. (1)}$$

The MMRE aggregates the multiple MRE's. The model with the lowest MMRE is considered the best [15]. As shown in Eq. (2), the estimation accuracy of the MMRE is the mean of all the MREs among n software projects.

$$MMRE = \frac{1}{n}\sum_{j=1}^{n} MRE_j \qquad \text{Eq. (2)}$$

In this study, the MRE, and MMRE were adopted as the indicators of the accuracy of the established software effort estimation models since they are the ones most widely used in the literature, thereby rendering our results more comparable to those of other work. [14]

III. RESULTS DISCUSSION

Appendix 2 contains the result table which shows the Redistributed Effort Estimations (RDE) using the mamdani FIS and the MRE values for each instances were calculated as shown in the table. MMRE for the given dataset is obtained by summing up the MRE's and dividing the sum by 41 i.e. the total number of instances in the dataset. The MMRE value obtained is 0.0628 (MMRE % is 6.28) and lower the MMRE better is prediction accuracy of the model.

Previous work, carried out by the author's where the same dataset was used to train and compare 3 different neural network models namely Feed Forward Backpropagation Neural Network (FFBPNN), CascadeFFBPNN and Layer Recurrent Neural Network (LRNN) reports that LRNN is the best among the three models as it has got the lowest MMRE of 11.45. In that work the data from serial number 31 to 41 (part of the main dataset as given in Appendix – I) were taken for interpretation purpose. [14]

Now when we took the same subset and calculated the MMRE % we got a value of 5.072 which is quite less than that obtained for LRNN. It shows that fuzzy logic can predict the effort more accurately.

IV. CONCLUSIONS

We know that one of the most critical and extremely crucial activities of any software development project is software effort & cost estimation. Accurate estimation of a software development effort is critical for good management decision making. The development costs tend to increase as the complexities in software project development increases. Therefore an important objective of the software engineering community has been to develop an appropriate model to estimate the software effort accurately.

From the results and discussion it is evident that soft computing techniques such as ANN, Fuzzy Logic etc. provides an alternative and can be utilized to develop models for predicting efforts at the early stages of development. Our future work will focus more on the applications of these techniques to the real project dataset from the software development industry. [16]

REFERENCES

[1] Steve McConnell. Rapid development: taming wild software schedules. Microsoft Press, 1996.

[2] Roheet Bhatnagar, Mrinal Kanti Ghose, Vandana Bhattacharjee; "Comparing Some Neural Network Models for Software Development Effort Prediction" in International Journal of Engineering Science and Technology (IJEST) Vol. 2(7), 2010, 2950-2956

[3] C. Larman, "Agile and Iterative Development: A Manager's Guide", Addison Wesley, 2003

[4] Bohem B. et al. 1995, “Cost models for future life cycle processes: COCOMO2.0”, Annals of Software Engineering, Vol 1.

[5] Jingzhou Li, Guenther Ruhe “Decision Support Analysis for Software Effort Estimation by Analogy”, Third International Workshop on Predictor Models in Software Engineering (PROMISE'07)

[6] Walker Royce, “ Software Project Management, A Unified Frame work” Pearson Education, 2000

[7] Mohammad Azzeh et al. “Software Effort Estimation Based on Weighted Fuzzy Grey Relational Analysis”, ACM 2009

[8] Andrew R. Gray, Stephen G. MacDonell, A comparison of techniques for developing predictive models of software metrics, Information and Software Technology 39 (1997) 425-437

[9] Boehm B., C. Abts and S. Chulani 2000, “Software development cost estimation approaches—A survey”, Annals of Software Engineering 10, pp. 177–205.

[10] L.A. Zadeh, 2002, From Computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions, Int. J. Appl. Math.Comut.Sci.,2002, Vol.12,No.3,307-324.

[11] L.A. Zadeh, 1965, Fuzzy Sets, Information and Control, 8, 338-353.

[12] Iman Attarzadeh and Siew Hock Ow, Software Development Effort Estimation Based on a New Fuzzy Logic Model, IJCTE, Vol. 1, No. 4, October2009

[13] Tayfur, G., S. Ozdemir and V.P. Singh. 2003. Fuzzy logic algorithm for runoff-induced sediment transport from bare soil surfaces. Advances in Water Resources Journal 26: 1249-1256.

[14] Roheet Bhatnagar, Mrinal Kanti Ghose, Vandana Bhattacharjee; 'A Novel Approach to the Early Stage Software Development Effort Estimations using Neural Network Models: A Case Study', IJCA Special Issue on “Artificial Intelligence Techniques - Novel Approaches & Practical Applications”, Number 3 - Article 5, pp. 27-30, AIT, 2011.

[15] Shepperd M., Schofield C., 1997 “Estimating software project effort using analogies”, IEEE Transactions on Software Engineering 23 (12) 736–743.

[16] "Comparing Some Neural Network Models for Software Development Effort Prediction"; Roheet Bhatnagar, Vandana Bhattacharjee and Mrinal Kanti Ghose, IEEE Xplore digital library, Print ISBN: 978-1-4244-9581-8/11 ,Digital Object Identifier: 10.1109/NCETACS.2011.5751391, Date of Current Version: 15 April 2011

APPENDIX – I

Student Project Dataset & Attributes for Early Stage Development Effort Estimations, TCOE: Total count of entities, TCOA: total count of attributes, TCOR: total count of relationships, CGPA: cumulative Grade Point Aggregate (parameter for judging academic excellence of students), RDE: Recalculated Development Effort in number of days.

Serial Number	TCOE	TCOA	TCOR	CGPA	RDE
1	24	70	29	6.219	75
2	24	70	29	8.012	75
3	24	70	29	7.733	75
4	10	56	9	7.564	70
5	5	44	5	5.519	55
6	19	47	11	7.507	70
7	8	33	9	6.171	75
8	8	33	9	6.705	75
9	17	53	7	7.629	75
10	9	37	7	8.130	70
11	10	36	8	8.083	65
12	10	36	8	8.126	65
13	10	36	8	7.202	65
14	5	17	5	8.417	65
15	5	16	7	7.757	70
16	4	26	4	7.431	70
17	4	26	4	7.121	70
18	4	26	4	7.660	70
19	7	34	6	8.017	75
20	7	34	6	9.076	75
21	7	27	5	7.550	70
22	6	37	5	6.583	65
23	6	27	12	7.276	65
24	6	27	12	8.124	65
25	5	26	4	6.530	75
26	5	26	4	6.685	70
27	6	28	6	7.843	65
28	7	38	9	9.160	70
29	7	38	9	8.617	75
30	6	18	3	8.719	80
31	4	22	3	8.860	65
32	5	18	5	7.664	75
33	16	85	15	6.795	65
34	16	85	15	6.757	65
35	9	36	9	6.207	70
36	9	36	9	6.636	70
37	9	36	9	6.790	70
38	8	24	7	8.095	65
39	20	115	22	7.990	75
40	20	115	22	8.095	75
41	15	60	9	6.340	75

APPENDIX – II

RDE using Mamdani FIS and corresponding MRE values

Serial Number	TCOE	CGPA	RDE	RDE using Mamdani FIS	MRE
1	24	6.219	75	75	0.000
2	24	8.012	75	75	0.000
3	24	7.733	75	75	0.000
4	10	7.564	70	75	0.071
5	5	5.519	55	64.3	0.169
6	19	7.507	70	75	0.071
7	8	6.171	75	65	0.133
8	8	6.705	75	65	0.133
9	17	7.629	75	75	0.000
10	9	8.13	70	75	0.071
11	10	8.083	65	75	0.154
12	10	8.126	65	75	0.154
13	10	7.202	65	75	0.154
14	5	8.417	65	71	0.092
15	5	7.757	70	71	0.014
16	4	7.431	70	70	0.000
17	4	7.121	70	70	0.000
18	4	7.66	70	70	0.000
19	7	8.017	75	73.4	0.021
20	7	9.076	75	72.8	0.029
21	7	7.55	70	73.2	0.046
22	6	6.583	65	64.4	0.009
23	6	7.276	65	71.3	0.097
24	6	8.124	65	72.1	0.109
25	5	6.53	75	64.4	0.141
26	5	6.685	70	64.5	0.079
27	6	7.843	65	72.1	0.109
28	7	9.16	70	72.7	0.039
29	7	8.617	75	73.3	0.023
30	6	8.719	80	71.9	0.101
31	4	8.86	65	70	0.077
32	5	7.664	75	71	0.053
33	16	6.795	65	70	0.077
34	16	6.757	65	70.4	0.083
35	9	6.207	70	67.1	0.041
36	9	6.636	70	68.6	0.020
37	9	6.79	70	70	0.000
38	8	8.095	65	75	0.154
39	20	7.99	75	75	0.000
40	20	8.095	75	75	0.000
41	15	6.34	75	71	0.053