Fuzzy Based Approach for Predicting Software Development Effort

Prasad Reddy P.V.G.D prasadreddy.vizag@gmail.com
Dept. of CSSE
Andhra University
Visakhapatnam, 530003, INDIA

Sudha K. R arsudhaa@gmail.com
Dept. of EE
Andhra University
Visakhapatnam, 530003, INDIA

Rama Sree P ramasree_p@rediffmail.com
Dept. of CSE
Aditya Engineering College
JNTUK
KAKINADA, 533003, INDIA

Ramesh S.N.S.V.S.C ramesh_snsvsc@yahoo.co.in
Dept. of CSE
Sri Sai Aditya Institute of Science & Technology
JNTUK
KAKINADA, 533003, INDIA

Abstract

Software development effort prediction is one of the most significant activities in software project management. The literature shows several algorithmic cost estimation models such as Boehm's COCOMO, Albrecht's' Function Point Analysis, Putnam's SLIM, ESTIMACS etc, but each do have their own pros and cons in estimating development cost and effort. This is because project data, available in the initial stages of project is often incomplete, inconsistent, uncertain and unclear. The need for accurate effort prediction in software project management is a challenge till today. Fuzzy logic-based estimation models are more apt when vague and inaccurate information is to be used. In the present paper software development effort prediction using Fuzzy triangular and GBell membership functions is presented and compared with COCOMO. A case study based on the COCOMO81 database compares the proposed fuzzy model with the Intermediate COCOMO. The results were analyzed using five different criterions VAF, MARE, VARE, Prediction and BRE. It is observed that the fuzzy model using triangular membership function provided better results.

Keywords: Development Effort, EAF, Cost Drivers, Fuzzy Identification, Membership Functions, Fuzzy Rules, COCOMO81 53 projects

1. INTRODUCTION

In algorithmic cost estimation [1], costs and efforts are predicted using mathematical formulae. The formulae are derived based on some historical data [2]. The best known algorithmic cost model called COCOMO (COnstructive COst MOdel) was published by Barry Boehm in 1981[3]. It was developed from the analysis of sixty three (63) software projects. Boehm projected three levels of the model called Basic COCOMO, Intermediate COCOMO and Detailed COCOMO [3,5]. In the present paper we mainly focus on the Intermediate COCOMO.

1.1 Intermediate COCOMO

The Basic COCOMO model [3] is based on the relationship: Development Effort, DE =a*(SIZE) ^{b}; where, SIZE is measured in thousand delivered source instructions. The constants a, b are dependent upon the 'mode' of development of projects. DE is measured in man-months. Boehm proposed 3 modes of projects [3]:

1. **Organic mode** – simple projects that engage small teams working in known and stable environments.
2. **Semi-detached mode** – projects that engage teams with a mixture of experience. It is in between organic and embedded modes.
3. **Embedded mode** – complex projects that are developed under tight constraints with changing requirements.

The accuracy of Basic COCOMO is limited because it does not consider the factors like hardware, personnel, use of modern tools and other attributes that affect the project cost. Further, Boehm proposed the Intermediate COCOMO[3,4] that adds accuracy to the Basic COCOMO by multiplying 'Cost Drivers' into the equation with a new variable: EAF (Effort Adjustment Factor) shown in Table 1.

Development mode	Intermediate Effort Equation
Organic	DE = EAF * 3.2 * $(SIZE)^{1.05}$
Semi-detached	DE = EAF * 3.0 * $(SIZE)^{1.12}$
Embedded	DE = EAF * 2.8 * $(SIZE)^{1.2}$

TABLE 1 : DE for the Intermediate COCOMO

The EAF term is the product of 15 Cost Drivers [5,11] that are listed in Table 2 .The multipliers of the cost drivers are Very Low, Low, Nominal, High, Very High and Extra High. For example, for a project, if RELY is Low, DATA is High , CPLX is extra high, TIME is Very High, STOR is High and rest parameters are nominal then EAF = 0.75 * 1.08 * 1.65 *1.30*1.06 *1.0. If the category values of all the 15 cost drivers are "Nominal", then EAF is equal to 1.

S. No	Cost Driver Symbol	Very low	Low	Nominal	High	Very high	Extra high
1	RELY	0.75	0.88	1.00	1.15	1.40	—
2	DATA	—	0.94	1.00	1.08	1.16	—
3	CPLX	0.70	0.85	1.00	1.15	1.30	1.65
4	TIME	—	—	1.00	1.11	1.30	1.66
5	STOR	—	—	1.00	1.06	1.21	1.56

6	VIRT	—	0.87	1.00	1.15	1.30	—
7	TURN	—	0.87	1.00	1.07	1.15	—
8	ACAP	—	0.87	1.00	1.07	1.15	—
9	AEXP	1.29	1.13	1.00	0.91	0.82	—
10	PCAP	1.42	1.17	1.00	0.86	0.70	—
11	VEXP	1.21	1.10	1.00	0.90	—	—
12	LEXP	1.14	1.07	1.00	0.95	—	—
13	MODP	1.24	1.10	1.00	0.91	0.82	—
14	TOOL	1.24	1.10	1.00	0.91	0.83	—
15	SCED	1.23	1.08	1.00	1.04	1.10	—

TABLE 2 : Intermediate COCOMO Cost Drivers with multipliers

The 15 cost drivers are broadly classified into 4 categories [3,5].

1. Product : RELY - Required software reliability
   DATA - Data base size
   CPLX - Product complexity
2. Platform: TIME - Execution time
   STOR—main storage constraint
   VIRT—virtual machine volatility
   TURN—computer turnaround time
3. Personnel: ACAP—analyst capability
   AEXP—applications experience
   PCAP—programmer capability
   VEXP—virtual machine experience
   LEXP—language experience
4. Project: MODP—modern programming
   TOOL—use of software tools
   SCED—required development schedule

Depending on the projects, multipliers of the cost drivers will vary and thereby the EAF may be greater than or less than 1, thus affecting the Effort [5].

2. FUZZY IDENTIFICATION

A fuzzy model is used when the systems are not suitable for analysis by conventional approach or when the available data is uncertain, inaccurate or vague [7]. The point of Fuzzy logic is to map an input space to an output space using a list of if-then statements called rules. All rules are evaluated in parallel, and the order of the rules is unimportant. 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 Sugeno type Fuzzy Inference System.
2. Define the input variables and output variable.
3. Set the type of the membership functions (TMF or GBellMF) for input variables.
4. Set the type of the membership function as linear for output variable.
5. The data is now translated into a set of if–then rules written in Rule editor.
6. A certain model structure is created, and parameters of input and output variables can be tuned to get the desired output.

2.1 Fuzzy Approach for Prediction of Effort

The Intermediate COCOMO model data is used for developing the Fuzzy Inference System (FIS)[10]. The inputs to this system are MODE and SIZE. The output is Fuzzy Nominal Effort. The framework [8] is shown in Figure 1.

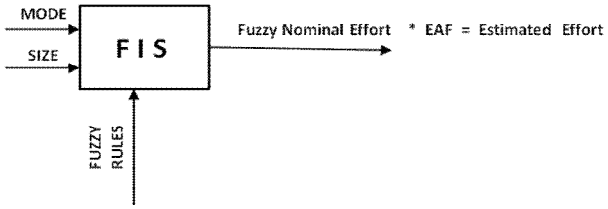

FIGURE 1: Fuzzy Framework

Fuzzy approach [9] specifies the SIZE of a project as a range of possible values rather than a specific number. The MODE of development is specified as a fuzzy range .The advantage of using the fuzzy ranges is that we will be able to predict the effort for projects that do not come under a precise mode i.e. comes in between 2 modes. This situation cannot be handled using the COCOMO. The output of this FIS is the Fuzzy Nominal Effort. The Fuzzy Nominal Effort multiplied by the EAF gives the Estimated Effort. The FIS needs appropriate membership functions and rules.

2.2 Fuzzy Membership Functions

A membership function (MF) [9] 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. For our problem, we have used 2 types of membership functions:

1. Triangular membership function
2. Guass Bell membership function

Triangular membership function (TMF):

It is a three-point function [8], defined by minimum (α),Maximum (β) and modal (m) values, that is, TMF (α, m, β), where (α ≤ m ≤β). Please refer to Figure 2 for a sample triangular membership function.

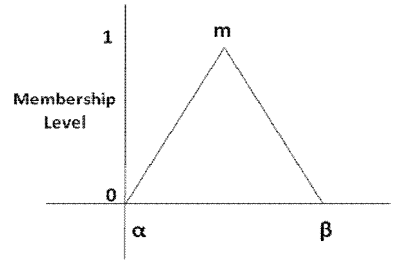

FIGURE 2: A Sample Triangular Membership Function

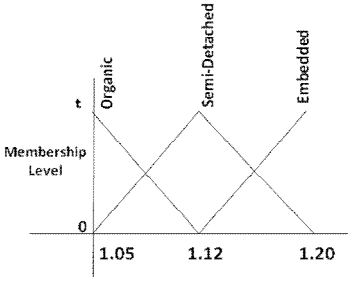

FIGURE 3: Fuzzy Set for Mode

The fuzzy set definitions for the MODE of development appear in Figure 3 and the fuzzy set [8] for SIZE appear in Figure 4.

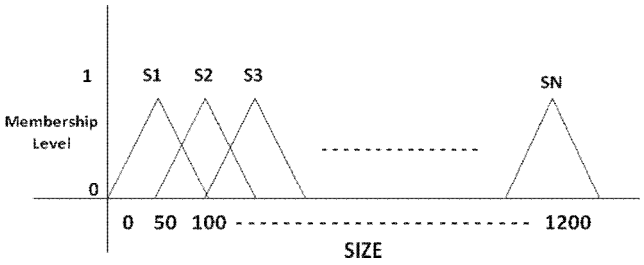

FIGURE 4: Fuzzy set for SIZE

Guass Bell membership function (GBellMF):

It is a three-point function, defined by minimum (α), maximum (β) and modal (m) values, that is, GBellMF(α, m, β), where (α ≤ m ≤β). Please refer to Figure 5 for a sample Guass Bell membership function.

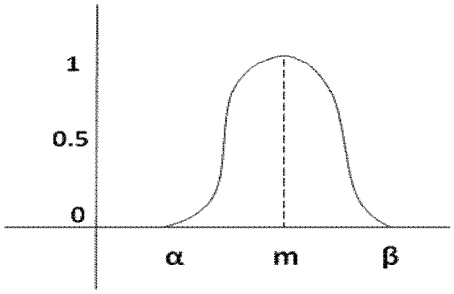

FIGURE 5: A Sample Guass Bell Membership Function

We can get the Fuzzy sets for MODE, SIZE and Effort for GBellMF in the same way as in triangular method, but the difference is only in the shape of the curves.

2.3 Fuzzy Rules

Our rules based on the fuzzy sets [9] of MODE, SIZE and EFFORT appears in the following form:

If MODE is organic and SIZE is s1 then EFFORT is EF1

If MODE is semidetached and SIZE is s1 then EFFORT is EF2

If MODE is embedded and SIZE is s1 then EFFORT is EF3

If MODE is organic and SIZE is s2 then EFFORT is EF4

If MODE is semidetached and SIZE is s2 then EFFORT is EF5

If MODE is embedded and SIZE is s3 then EFFORT is EF5

If MODE is embedded and SIZE is s4 then EFFORT is EF3

If MODE is organic and SIZE is s3 then EFFORT is EF4

If MODE is embedded and SIZE is s5 then EFFORT is EF6

If MODE is organic and SIZE is s4 then EFFORT is EF4

.....

3. VARIOUS CRITERIONS FOR ASSESSMENT OF SOFTWARE EFFORT ESTIMATION MODELS

1. Variance Accounted For (VAF)

$$\text{VAF (\%)} = \left(1 - \frac{\text{var}\left(E - \hat{E}\right)}{\text{var}E}\right) \times 100$$

2. Mean Absolute Relative Error (MARE)

$$\text{MARE (\%)} = \frac{\sum f\left(\text{R}_{\text{E}}\right)}{\sum f} \times 100$$

3. Variance Absolute Relative Error (VARE)

$$\text{VARE (\%)} = \frac{\sum f\left(R_E - meanR_E\right)^2}{\sum f} \times 100$$

4. Prediction (n)

Prediction at level n is defined as the % of projects that have absolute relative error less than n.

5. Balance Relative Error (BRE)

$$\text{BRE} = \frac{\left|E - \hat{E}\right|}{\min(E, \hat{E})}$$

Where, E = estimated effort $\hat{E}$ = actual effort

$$\text{Absolute Relative Error (RE)} = \frac{\left|E - \hat{E}\right|}{\left|E\right|}$$

A model which gives higher VAF is better than that which gives lower VAF. A model which gives higher Pred(n) is better than that which gives lower Pred(n). A model which gives lower MARE is better than that which gives higher MARE[11]. A model which gives lower VARE is better than that which gives higher VARE [6]. A model which gives lower BRE is better than that which gives higher BRE.

4. Experimental Study

The COCOMO81 database [5] consists of 63 projects data [3,11], out of which 28 are Embedded Mode Projects, 12 are Semi-Detached Mode Projects, and 23 are Organic Mode Projects. Thus, there is no uniformity in the selection of projects over the different modes. In carrying out our experiments, we have chosen 53 projects data out of the 63, which have their lines of code (size) to be less than 100KDSI. The estimated efforts using Intermediate COCOMO, Fuzzy using TMF and GBellMF are shown in Table 3. Table 4 and Figure.6.to Figure 13. shows the comparisons of various models basing on different criterions.

SNo	MODE	SIZE	EAF	Actual Effort	COCOMO Effort	Effort using TMF	Effort using GBell
1	1.05	46	1.17	240	212	246	252
2	1.05	16	0.66	33	39	41	41
3	1.05	4	2.22	43	30	34	34

4	1.05	6.9	0.4	8	9.8	11	11
5	1.2	22	7.62	1075	869	1078	1116
6	1.2	30	2.39	423	397	484	485
7	1.2	18	2.38	321	214	239	231
8	1.2	20	2.38	218	243	287	303
9	1.2	37	1.12	201	238	280	280
10	1.2	24	0.85	79	108	138	138
11	1.12	3	5.86	73	60	63	62
12	1.2	3.9	3.63	61	52	51	50
13	1.2	3.7	2.81	40	38	37	36
14	1.2	1.9	1.78	9	10.7	10	9
15	1.2	75	0.89	539	443	534	927
16	1.12	90	0.7	453	326	453	486
17	1.2	38	1.95	523	430	502	502
18	1.2	48	1.16	387	339	380	379
19	1.2	9.4	2.04	88	89	74	75
20	1.05	13	2.81	98	133	143	143
21	1.12	2.14	1	7.3	7	7	7
22	1.12	1.98	0.91	5.9	5.8	6	6
23	1.2	50	3.14	1063	962	1063	1064
24	1.2	40	2.26	605	529	615	614
25	1.2	22	1.76	230	201	249	258
26	1.2	13	2.63	82	161	135	138
27	1.12	12	0.68	55	33	31	31
28	1.05	34	0.34	47	44	46	47
29	1.05	15	0.35	12	20	21	21
30	1.05	6.2	0.39	8	8.4	9	9
31	1.05	2.5	0.96	8	8.1	9	9
32	1.05	5.3	0.25	6	4.7	5	5
33	1.05	19.5	0.63	45	46	48	49
34	1.05	28	0.96	83	102	106	106
35	1.05	30	1.14	87	130	136	136
36	1.05	32	0.82	106	100	104	105
37	1.05	57	0.74	126	166	126	114
38	1.05	23	0.38	36	33	35	35
39	1.12	91	0.36	156	168	235	246
40	1.2	24	1.52	176	193	247	246
41	1.05	10	3.18	122	114	124	124
42	1.05	8.2	1.9	41	55	61	61
43	1.12	5.3	1.15	14	22	23	23
44	1.05	4.4	0.93	20	14	16	16
45	1.05	6.3	0.34	18	7.5	8	8
46	1.2	27	3.68	958	537	673	673
47	1.2	15	3.32	237	239	234	210
48	1.2	25	1.09	130	145	185	184
49	1.05	21	0.87	70	68	72	72
50	1.05	6.7	2.53	57	60	66	66
51	1.05	28	0.45	50	47	50	50
52	1.12	9.1	1.15	38	42	40	40

53	1.2	10	0.39	15	17	15	15

TABLE 3: Estimated Effort in Man Months of Various Models

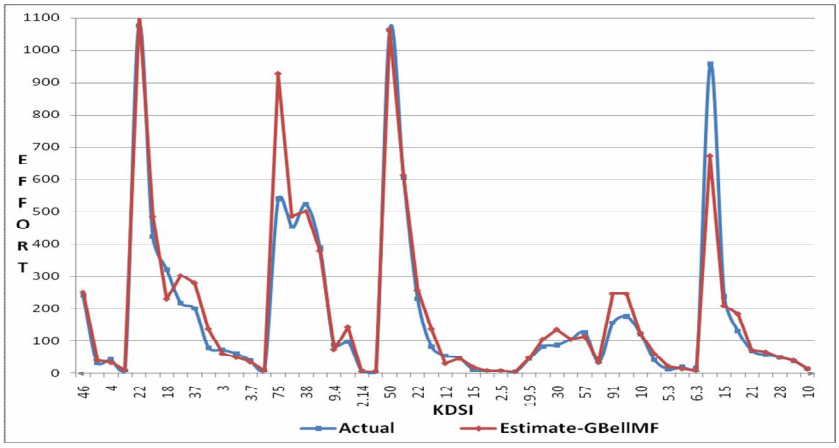

FIGURE 6 : Estimated Effort using Fuzzy GBellMF versus Actual Effort

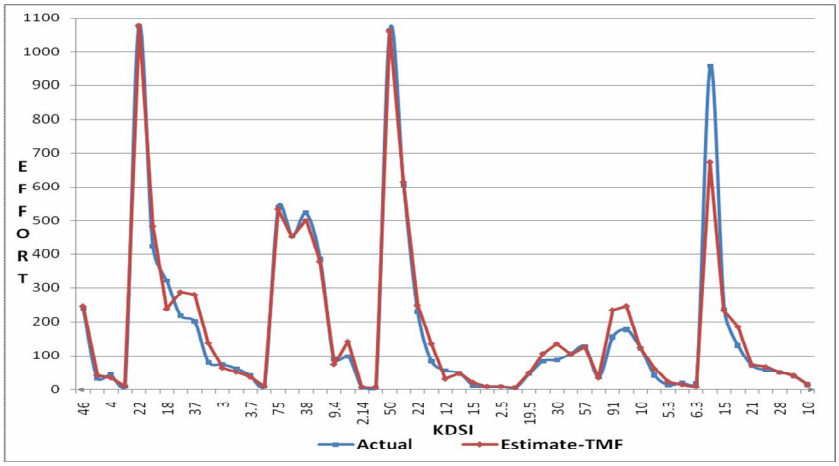

FIGURE 7 : Estimated Effort using Fuzzy TMF versus Actual Effort

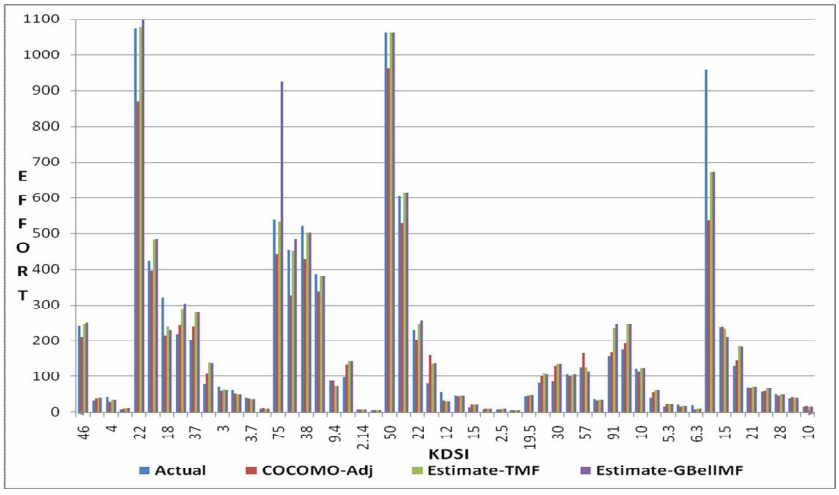

FIGURE 8: Estimated Effort of various models versus Actual Effort

Model	VAF(%)	MARE(%)	VARE(%)	Mean BRE	Pred (25)(%)
Intermediate COCOMO Model	87.16	21.41	5.48	0.25	72
Fuzzy using TMF	95.83	18.63	4.35	0.23	68
Fuzzy using GBellMF	92.25	20.35	4.24	0.26	62

TABLE 4: Comparison of various models

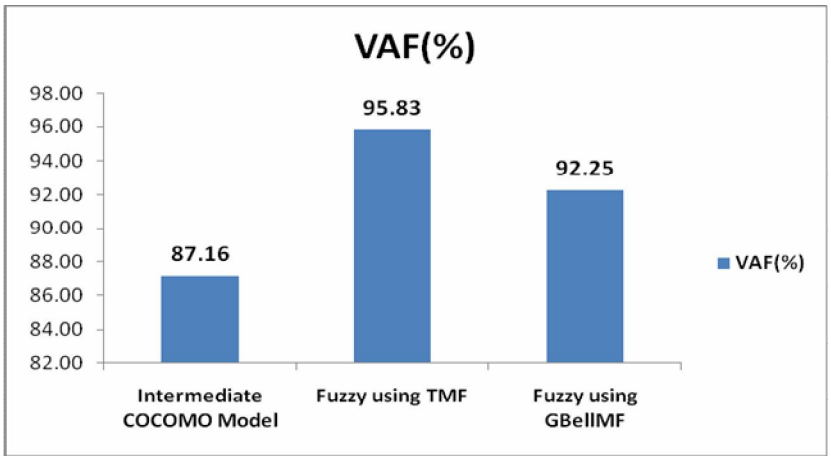

FIGURE 9: Comparision of VAF against various models

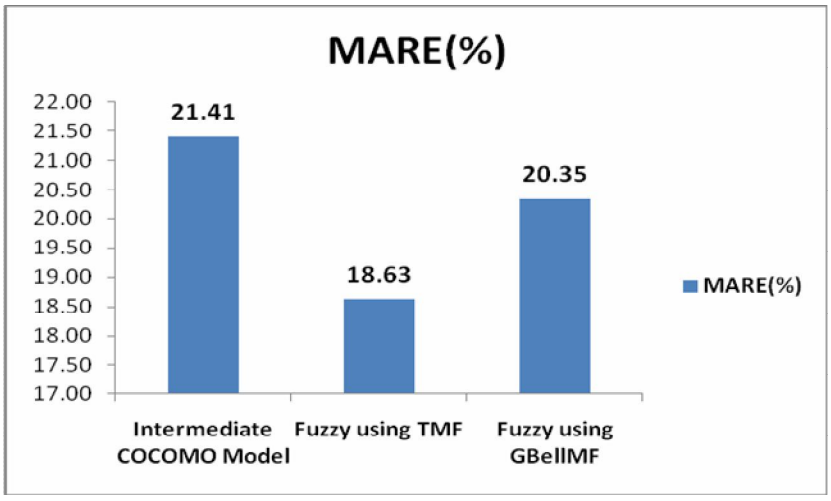

FIGURE 10: Comparision of MARE against various models

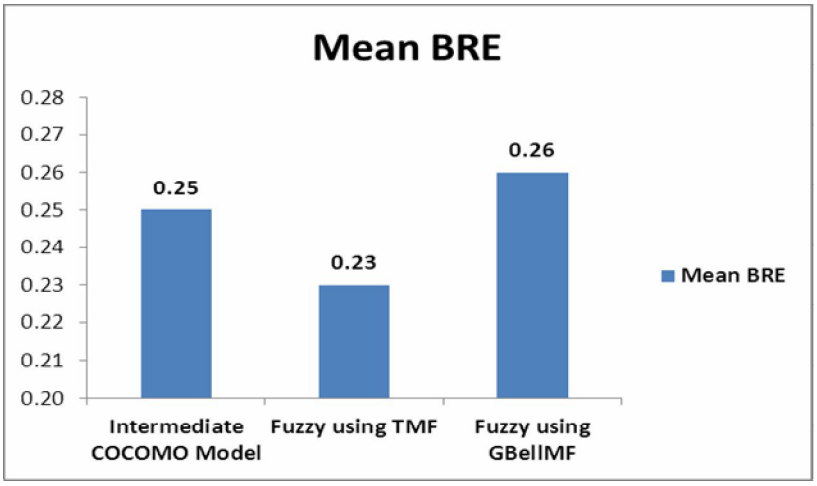

FIGURE 11: Comparision of Mean BRE against various models

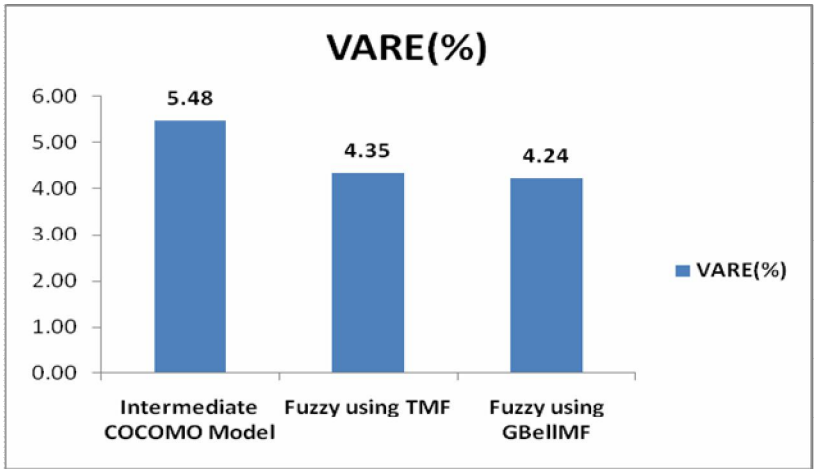

FIGURE 12 : Comparision of VARE against various models

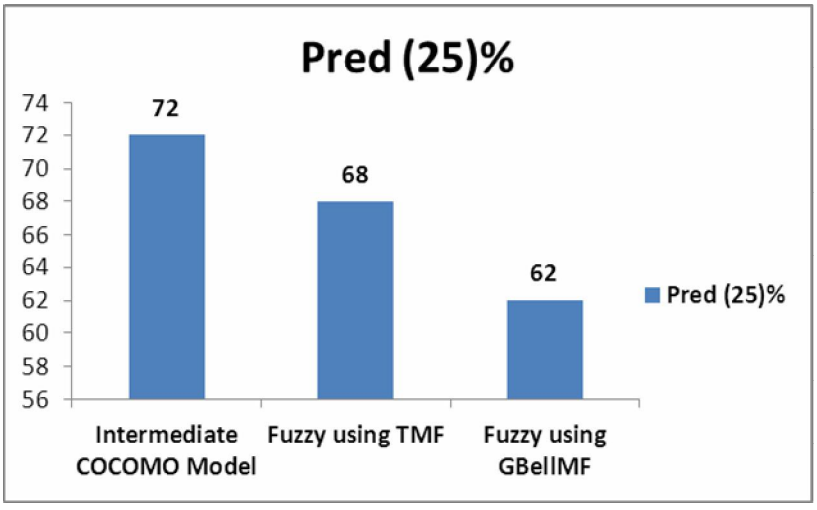

FIGURE 13 : Comparision of Pred(25) against various models

5. CONCLUSION

Referring to Table 4, we see that Fuzzy using TMF yields better results for maximum criterions when compared with the other methods. Thus, basing on VAF, MARE & Mean BRE, we come to a conclusion that the Fuzzy method using TMF (triangular membership function) is better than Fuzzy method using GBellMF or Intermediate COCOMO. It is not possible to evolve a method, which can give 100 % VAF. By suitably adjusting the values of the parameters in FIS we can optimize the estimated effort.

6. REFERENCES

[1] Ramil, J.F. , Algorithmic cost estimation for software evolution, Software Engg. (2000) 701-703.

[2] Angelis L, Stamelos I, Morisio M, Building a software cost estimation model based on categorical data, Software Metrics Symposium, 2001- Seventh International Volume (2001) 4-15.

[3] B.W. Boehm, Software Engineering Economics, Prentice-Hall, Englewood Cli4s, NJ, 1981

[4] Kirti Seth, Arun Sharma & Ashish Seth, Component Selection Efforts Estimation– a Fuzzy Logic Based Approach, IJCSS-83, Vol (3), Issue (3).

[5] Zhiwei Xu, Taghi M. Khoshgoftaar, Identification of fuzzy models of software cost estimation, Fuzzy Sets and Systems 145 (2004) 141–163

[6] Harish Mittal, Harish Mittal, Optimization Criteria for Effort Estimation using Fuzzy Technique, CLEI Electronic Journal, Vol 10, No 1, Paper 2, 2007

[7] R. Babuska, Fuzzy Modeling For Control, Kluwer Academic Publishers, Dordrecht, 1999

[8] Moshood Omolade Saliu, Adaptive Fuzzy Logic Based Framework for Software Development Effort Prediction, King Fahd University of Petroleum & Minerals, April 2003

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

[10] Xishi Huang, Danny Ho,Jing Ren, Luiz F. Capretz, A soft computing framework for software effort estimation, Springer link, Vol 10, No 2 Jan-2006

[11] Prasad Reddy P.V.G.D, Sudha K.R , Rama Sree P & Ramesh S.N.S.V.S.C,Software Effort Estimation using Radial Basis and Generalized Regression Neural Networks, Journal of Computing, Vol 2, Issue 5 May 2010, Page 87-92