

A Fuzzy Logic Approach for Software Cost Prediction



Farha Masroor, Vandana Bhattacharjee, and Debjani Mustafi

Abstract Software cost prediction is the technique of accurately evaluating the amount while developing the software. Estimation involves the total time required for the completion of the software, effort required that is measured in terms of person per month (PM), and the total cost to complete the activity. Accuracy and duration are the two desirable criteria in the software estimation process. In software estimation process, there are several inputs that are being fed to the system and these inputs are used for the generation or calculation of the set of outputs. The important work of the software project managers in the present scenario is the computation of cost or effort before the absolute advancement of any particular software. There are several methods applied for software cost estimation but we will focus on the fuzzy logic which is a soft-computing method. We feel that model which is based on fuzzy logic for the software cost estimation should be able to give the uncertain values rather than other models which give precision and certain values and the results based on fuzzy logic model will be more accurate than other models. In this research paper, we have developed the fuzzy logic model by taking several inputs and membership functions, also the fuzzy rules have been deduced by expert-knowledge with the help of MATLAB's Fuzzy logic toolbox and the results are then compared with the multiple regression model using SPSS Tool upon desharnais data set which gives less accuracy than fuzzy-based model.

Keywords Fuzzy logic · Fuzzy inference system · Effort · Estimation · Desharnais data set · Linear regression

F. Masroor (✉) · V. Bhattacharjee · D. Mustafi
Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi,
Jharkhand, India
e-mail: farhamasroor01@gmail.com

V. Bhattacharjee
e-mail: vbhattacharya@bitmesra.ac.in

D. Mustafi
e-mail: debjani.mustafi@bitmesra.ac.in

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2021

V. Nath and J. K. Mandal (eds.), *Proceedings of the Fourth International Conference on Microelectronics, Computing and Communication Systems*, Lecture Notes in Electrical Engineering 673, https://doi.org/10.1007/978-981-15-5546-6_31

1 Introduction

Accuracy and duration are the most important tasks for the software projects. In the past history, several researchers have already worked in the field of software computation to advance the efficiency and several techniques have been developed. This paper deals with soft-computing technique which are used for approximate models. Soft computing replaces the hard computing technique as it is less time consuming with more intelligent processing systems. Fuzzy-based model is one of the soft-computing techniques which we will focus in this paper.

1.1 Fuzzy Logic

It is a path for computing based on “degree of truth.” It follows the concept of truth that is partial. It is not based on modern computer that gives values either “true” or “false” (1 or 0). Fuzzy systems have the edge of better performance, great productivity, simplicity, and lesser cost. Membership function plays vital role in fuzzy system as it represents the “degree of truth” in fuzzy logic. It defines whether all the information and elements in the fuzzy set are discrete or continuous. Basically, there are four membership functions: Triangular, Gaussian, Trapezoidal, and Singleton. It is very difficult to choose the correct set of membership functions for a particular fuzzy model. In this research work, we will use trapezoidal and triangular membership functions and have taken five membership values as very low, low, medium, high, and very high.

1.2 Fuzzy Rules

Fuzzy rules are the rules that are produced by human expertise brains. It may vary from person to person as different person will think differently and with their own brains, they can generate fuzzy rules accordingly. Fuzzy rules are basically ‘IF’, ‘THEN’ rules.

1.3 Fuzzy Logic System

It has four important parts:

1. Fuzzification: It is the process of converting the crisp value that is non-fuzzy input values to fuzzy data.
2. Rule-Based System: Rule-based system mean the rules that are used for the process (IF-THEN rules). Fuzzy IF-THEN rules basically connect m-conditional

variables to n -consequent variables in the form: IF (X_1 is A_1 and ... X_m is A_m) THEN (Y_1 is B_1 and ... Y_n is B_n) Where: Linguistic terms are A_1 ..., A_m and B_1 ..., B_n and Linguistic variables are (X_1 ..., X_n and Y_1 ..., Y_n) "Ancedent" is the IF part, and "Consequent" is THEN part.

3. Inference Rule: These are the IF-THEN rules, which are carried out in fuzzy inference system and then the rules that will be fired according to the data set are the final fired rules which will be applied in the inference system to get the result.
4. Defuzzification: It is the technique to convert the fuzzy values to original values. The rest of the paper is coordinated as such: Segment 2 as Literature Review, Segment 3 consists of Proposed Framework, followed by Segment 4 as Experimental Results, Conclusion as a part of Segment 5 and Segment 6 as References.

2 Literature Review

Bedi and Singh [1] have compared the fuzzy logic approach with the COCOMO Models and have shown the accuracy by evaluating the MRE% and MMRE%. The data set that is used in this research paper is the Promise data set. The fuzzy inference system is based on Mamdani approach. Five input parameters and one output parameter are taken which is used for prediction of software effort. With different input/output parameters, different linguistic variables are taken. Based on linguistic variables and parameters, four different fuzzy rules are considered. The result with the COCOMO Model is MMRE% of 25.604 which has much higher % than fuzzy logic approach of MMRE% of 17.613. Dizaji and Gharehchopogh [2] have evaluated the price of software projects as per the meta-heuristic algorithms and the data set of NASA has been taken for further research. At first, the classification of the projects is done according to the project type. After the classification process, the ACO and the COA algorithm are used for the cost estimation of software projects. In ant colony optimization (ACO) technique, the ants are proficient enough to find the shortest path with the help of concentration level of pheromones. When an ant moves, it leaves some concentration level of pheromones in the ground, and reaches its final destination. The other ants when moves forward follow the same path with the help of pheromones and moves in the direction which has more concentration level of pheromones. This way they find the shortest path and is called ACO technique. Chaos optimization algorithm (COA) is done with the help of iterative maps. The experimental results say that the performance increased when ACO algorithm is combined with COA. Also, MARE for COCOMO Model is 0.29% and 0.078% is the MARE with proposed technique (ACO combined with COA) which proves that there is an improvement in the estimated costs of software projects. An optimized fuzzy logic framework for the research work has been used by Sharma and Verma [3]. Many research works have proved that techniques involving COCOMO Models give

poor results but in this paper, the framework is built upon COCOMO-81 and intermediate COCOMO is used. To get the exact size of software projects is very difficult in COCOMO Model as it does not consider the projects not lying exactly in any of the three categories (organic, semi-detached, and embedded), so fuzzy logic approach is used for improvement in the results. Gaussian membership function is used for software development mode and effort. For the development of fuzzy rules, the basic components of COCOMO Model are used. Triangular and Trapezoidal membership functions are used to fuzzify the cost drivers having linguistic values such as very low, low, and nominal high. For every cost drivers, independent fuzzy inference system is built. From each effort multiplier, the defuzzified value is collected by individual FISs after matching, aggregation, and subsequent defuzzification. By multiplying them all together, the total EAF is obtained. The comparison has been done of nominal effort prediction on actual real project data by FIS and COCOMO Model. Also another comparison of the comprehensive effort predicted by FIS and COCOMO Model with the addition of effort multipliers is done. The results show that the nominal effort prediction by the FIS has less than 50% error for most of the projects. Shivakumar et al. [4] have worked with the concept of adaptive neuro-fuzzy logic for the effort estimation [5] which helps in the improvement of accuracy and reliability. Ninety three instances of NASA project data were considered and 30 projects from different case studies and experiments were gathered consisting actual effort and 15 attributes together with field of work, size, and domain. After the collection of data and attributes, these 15 attributes were converted into three index values. The adaptive neuro-fuzzy technique is then built [4]. Artificial neural networks are made up of neurons which are connected in parallel. This model has an input of six-grouped attributes which leads to the development effort. The neural network and fuzzy logic principles are combined in this ANFIS framework. The MRE and MMRE are calculated and then the results have been compared with the proposed ANFIS model and other algorithmic models. Kumar et al. [6] deal with the fuzzy logic technique for software prediction taking two inputs as lines of code and adjustment difficulty level and effort as an output. The data set has been taken from BIT MCA students. Three membership functions [3] as low, medium, high have been taken. Also, according to the data set, the parameters have been taken for the inputs and output, respectively. Fuzzy inference system is built and fuzzy IF-THEN rules are applied. The predicted effort is obtained and with the help of it, the MRE and MMRE are calculated. The same data set is used for the testing in multiple regression method and the results are then compared by both the methods. By fuzzy logic technique, the results are more accurate as the MMRE is 0.1762% whereas by multiple regression, 0.5358% is the MMRE which is much higher. So fuzzy logic gives much more accurate results but the limitation in this paper is that it is difficult to determine a correct pair of M.F when the dimensionality and volume of data are broad. An approach based on fuzzy logic and optimization process to evaluate software project effort has been given by Ganesh et al. [7]. At first, Fuzzy logic approach is used for both categorical and numerical data which are specified by fuzzy sets and while generating fuzzy rules, the grouping of the optimization is done by means of particle swarm optimization (PSO), so that the rules could function better. The fitness for the

optimization function is assumed to be the effort of the software. The outcome of these fitness functions that is the fitness values are further carried for optimization of fuzzy rules. The comparison by Bhatnagar et al. [8] has been done between fuzzy logic and neural network models for the development of software effort estimation. Radial basis neural network (RBNN), FFNN, and fuzzy logic models are built and the results are evaluated based upon parameters like MRE, MMRE, BRE, and prediction. Neural network as a tool and backpropagation (learning algorithm) method for training the networks are used. At first, the fuzzy inference system is built with fuzzy rules having Gaussian membership functions, mode, and size as inputs and effort as output. For each cost drivers, fuzzy inference system is defined. EAF is calculated by multiplying the values of each cost drivers. The final effort is then evaluated by combining the two components, i.e., nominal effort and EAF. After the designing of the fuzzy system, the FFNN and RBNN with ten hidden layers are designed and these neural networks are trained with 50 randomly chosen projects and 2500 epoch value is taken and are saved as FFNN and RBNN. The evaluation of these saved networks with the value of effort is done. At last, the comparison between both the neural networks, i.e., FFNN and RBNN and fuzzy logic are done on the basis of MMRE and prediction. Kumar and Chopra [9] have focussed on the literature of fuzzy logic and other algorithmic models. The basic purpose of this paper is to take a review on the studies of software estimation using fuzzy and other models in the past years to improve the accuracy. The theoretical part is discussed in this paper, so that this could help for the development of new framework for estimation or to do the changes in the existing one. The comparison between different techniques for software effort/cost prediction is also shown in the paper. Singh and Sahoo [5] have introduced the ANN structure and the performance analysis of different ANNs for software effort estimation is done. ANN is structured between the independent (cost drivers) variables and dependent (effort) variable. Four types of ANN is assumed, MATLAB 10NNTool is used using NASA data set. Kushwaha and Suryakant [10] have developed fuzzy logic technique for software cost estimation and the comparison in the performance has been done with the COCOMO Model in this research paper. The basic difference in this paper with the others is in the membership function. Generally, in the fuzzy logic approach, many researchers have used triangular MF's but in this research work, Gaussian mfs are used in fuzzy logic technique. According to the results, 13 GMF gives better and is closer to the actual effort than 11 GMF and COCOMO Model. Also, by analyzing the results, it is clear that higher the membership functions better will be the results.

3 Proposed Framework

3.1 Data Analysis

The proposed model is validated on desharnais data set. The data set consists of several parameters/metrics from which four inputs and one output parameter/metrics have been used which are important for software modelling and contains 81 program data values.

The metrics that are used are Transactions, Entities, PointNonAdjusted, and Language which are inputs to the fuzzy system and effort as the only output to the fuzzy system.

- Transactions—It is the number of necessary transactions in the data model and is measured in the range of 0–1000.
- Entities—Entities are the total number of objects to represent the software or systems and is measured in the range of 0–400.
- PointNonAdjust—It is used to measure the size of the project in adjusted function points and is measured in the range of 0–1200.
- Language: How many programming languages are used in the scheme and are expressed as 1, 2, or 3 measured in the range of 1–3.
- Effort—Actual effort is evaluated in person per hour and is deliberated in the range of 500–24,000.

3.2 Fuzzy Rules

Fuzzy Rules are the IF-THEN rules which is used for constructing fuzzy model. The high expertise knowledge in oral form is being converted to a set of IF-THEN Rules. The membership functions and weights of the rules are coordinated with the help of input and output data. This paper consists of 14 fuzzy rules:

1. If Transaction is LOW, Entity is MEDIUM, and PointNonAdjust is LOW, then Effort is LOW.
2. If Transaction is LOW, Entity is HIGH, and PointNonAdjust is LOW, then Effort is LOW.
3. If Transaction is LOW, Entity is LOW, and PointNonAdjust is MEDIUM, then Effort is LOW.
4. If Transaction is LOW, Entity is MEDIUM, and PointNonAdjust is MEDIUM, then Effort is LOW.
5. If Transaction is MEDIUM and PointNonAdjust is MEDIUM, then Effort is LOW.
6. If Transaction is LOW, Entity is LOW, and PointNonAdjust is LOW, then Effort is VERY LOW.

7. If Transaction is LOW and PointNonAdjust is HIGH, then Effort is VERY LOW.
 8. If Transaction is MEDIUM, Entity is MEDIUM, and PointNonAdjust is MEDIUM, then Effort is LOW.
 9. If Transaction is LOW and PointNonAdjust is HIGH, then Effort is MEDIUM.
 10. If Transaction is MEDIUM, Entity is MEDIUM, PointNonAdjust is MEDIUM, and Language is MEDIUM, then Effort is HIGH.
 11. If Transaction is MEDIUM, Entity is LOW, PointNonAdjust is LOW, and Language is MEDIUM, then Effort is HIGH.
 12. If Transaction is MEDIUM, Entity is HIGH, and PointNonAdjust is MEDIUM, then Effort is VERY HIGH.
 13. If Transaction is LOW, Entity is HIGH, and PointNonAdjust is HIGH, then Effort is VERY HIGH.
 14. If Transaction is HIGH, Entity is HIGH, and PointNonAdjust is MEDIUM, then Effort is VERY HIGH.
- All the membership functions are triangular in the inputs but for the output, we have used both triangular and trapezoidal membership functions as shown in Tables 1 and 2 with all the scalar parameters(a, b, c)-input and (a, b, c, d)-output

Table 1 Membership function characteristics (input)

Variable name	Range	MF	Parameters		
			a	b	C
Transactions	0–1000	L	0	250	500
		M	250	500	750
		H	500	750	1000
Entities	0–400	L	0	100	200
		M	100	200	300
		H	200	300	400
PointNonAdjust	0–1200	L	0	300	600
		M	300	600	900
		H	600	900	1200
Language	1–3	L	1	1.5	2
		M	1.5	2	2.5
		H	2	2.5	3

Table 2 Output

Variable name	Range	MF	Parameters			
Effort	500–24,000		<i>a</i>	<i>b</i>	<i>c</i>	<i>D</i>
		VL	500	500	2000	3500
		L	2000	3500	5000	
		M	4000	8000	12,000	
		H	10,000	14,000	18,000	
		VH	16,000	18,000	24,000	24,000

4 Evaluation Criteria

4.1 Magnitude of Relative Error (MRE)

It is the common criteria for the evaluation of software effort models.

$$MRE = \frac{\text{Actual Effort} - \text{Predicted Effort}}{\text{Actual Effort}}$$

MRE will be calculated for each data value whose effort is predicted. So for the given data set, there are 81 data values and for each data values, the MRE will be calculated. The cumulative of MRE for all the observations(M) can be calculated through Mean MRE(MMRE).

$$MMRE = 1/M \sum_i^M MRE$$

4.2 Multiple Regression

With four independent variables, multiple regression can be expressed as:

$$y = a + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4$$

where *y* is the dependent variable; *a*, *b*₁, *b*₂, *b*₃ and *b*₄ are constants and *x*₁, *x*₂, *x*₃ and *x*₄ are the four independent variables. We can deduce the values of constants by solving these equations for multiple regression:

Table 3 Predicted effort with MRE using fuzzy logic technique

Project ID	Actual effort	Predicted effort using fuzzy logic	MRE
1	5152	1803.077549	0.650024
2	5635	2051.513135	0.635934
3	805	1882.352941	1.338327
4	3829	1970.810351	0.485294
5	2149	1743.221877	0.188822
–	–	–	–
77	1400	2844.228408	1.031592
78	2800	1688.431373	0.396989
79	9520	3380.234657	0.644933
80	5880	4863.416623	0.172888
81	23,940	1852.796053	0.922607

$$\begin{aligned} \sum x_1y &= b_1\left(\sum x_1^2\right) + b_2\left(\sum x_1x_2\right) + b_3\left(\sum x_1x_3\right) + b_4\left(\sum x_1x_4\right) \\ \sum x_2y &= b_2\left(\sum x_2^2\right) + b_1\left(\sum x_1x_2\right) + b_3\left(\sum x_2x_3\right) + b_4\left(\sum x_2x_4\right) \\ \sum x_3y &= b_3\left(\sum x_3^2\right) + b_1\left(\sum x_1x_3\right) + b_2\left(\sum x_2x_3\right) + b_4\left(\sum x_3x_4\right) \\ \sum x_4y &= b_4\left(\sum x_4^2\right) + b_1\left(\sum x_1x_4\right) + b_2\left(\sum x_2x_4\right) + b_3\left(\sum x_3x_4\right) \end{aligned}$$

5 Experimental Results

To proclaim the feasibility of the proposed framework, the experimentation has been done with multiple regression and fuzzy logic (proposed methodology) methods by taking the large amount of data from the data set.

The snapshot of predicted effort and MRE using fuzzy logic is shown in Table 3.

Table 4 shows the snapshot of predicted effort along with MRE through linear regression.

Table 5 shows the comparison between both the techniques and the screenshot of the final output has been attached.

6 Conclusion

For every data from the given model, we have analyzed the results of the actual and the predicted efforts and then evaluated the mean relative error (MRE) of each

Table 4 Predicted effort with MRE using linear regression technique

Project ID	Actual effort	Predicted effort using linear regression	MRE
1	5152	5915.0551	0.148109
2	5635	6427.91261	0.140712
3	805	2465.08731	2.06222
4	3829	6377.22194	0.665506
5	2149	4851.76832	1.257687
–	–	–	–
77	1400	3941.1818	1.81513
78	2800	5900.69183	1.10739
79	9520	11187.66875	0.175175
80	5880	8153.78173	0.386698
81	23,940	20,490.10925	0.144106

Result showing MMRE for both the techniques

Table 5 Comparative results of both the techniques in terms of MRE and mean MRE

	Multiple regression	Fuzzy logic
Min (MRE)	0.0171	0.0049
Max (MRE)	3.4712	2.2984
MMRE	0.5743	0.5089

project and mean MRE (MMRE). The same data set has also being tested for multiple regression model and evaluated the results in the same manner. Table 4 shows the comparison where MMRE% of 0.5089 by proposed technique is much superior than MMRE% of 0.5743 by multiple regression. So, after analyzing the outcomes, we came to the conclusion that the multiple regression technique gives less accuracy than the proposed fuzzy logic method.

References

1. Bedi RPS, Singh A (2017) Software cost estimation using fuzzy logic technique. *Indian J Sci Technol* 13(10)
2. Dizaji ZA, Gharehchopogh FS (2015) A hybrid of ant colony optimization and chaos optimization algorithms approach for software cost estimation. *Indian J Sci Technol* 8(2)
3. Sharma V, Verma HK (2010) Optimized fuzzy logic based framework for effort estimation in software development. *IJCSI Int J Comput Sci Issues* 7(2)
4. Shivakumar N, Balaji N, Ananthakumar K (2016) A neuro fuzzy algorithm to compute software effort estimation. *Glob J Comput Sci Technol* 16(1)
5. Singh J, Sahoo B (2011) Software effort estimation with different artificial neural network
6. Kumar S, Pal J, Bhattacharjee V (2013) Software effort prediction-a fuzzy logic approach. *Glob J Comput Sci Technol* 13(10)

7. Ganesh MKS, Thanushkodi K (2015) An efficient software cost estimation technique using fuzzy logic with the aid of optimization algorithm. *Int J Innov Comput Inf Cont* 11(2)
8. Bhatnagar R, Bhattacharjee V, Ghose MK (2010) A proposed novel framework for early effort estimation using fuzzy logic techniques. *Glob J Comput Sci Technol* 10(14)
9. Kumar S, Chopra V (2013) To design and implement neural network and fuzzy logic for software development effort prediction. *Int J Comput Appl* 84(11)
10. Kushwaha N, Suryakant (2014) Software cost estimation using the improved fuzzy logic framework. In: 2014 conference on IT in business, industry and government (CSIBIG). IEEE, pp 1–5
11. Ochodek M, Nawrocki J, KwarciaK K (2011) Simplifying effort estimation based on use case points. *Inf Softw Technol* 53(3):200–213
12. Thamarai I, Murugavalli S (2016) An evolutionary computation approach for project selection in analogy based software effort estimation. *Indian J Sci Technol* 9(21)
13. Singh H, Singh G, Bhatia N (2012) Election results prediction system based on fuzzy logic. *Int J Comput Appl* 53(9)
14. Mittal A, Parkash K, Mittal H (2010) Software cost estimation using fuzzy logic. *ACM SIGSOFT Softw Eng Notes* 35(1):1–7
15. Sheta A, Rine D, Ayesh A (2008) Development of software effort and schedule estimation models using soft computing techniques. In: 2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence). IEEE, pp 1283–1289
16. Pahariya JS, Ravi V, Carr M (2009) Software cost estimation using computational intelligence techniques. In: 2009 world congress on nature and biologically inspired computing (NaBIC). IEEE, pp 849–854
17. Verma HK, Sharma V (2010) Handling imprecision in inputs using fuzzy logic to predict effort in software development. In: 2010 IEEE 2nd international advance computing conference (IACC). IEEE, pp 436–442
18. Ghose MK, Bhatnagar R, Bhattacharjee V (2011) Comparing some neural network models for software development effort prediction. In: 2011 2nd national conference on emerging trends and applications in computer science. IEEE, pp 1–4
19. Johanyák ZC, Tikk D, Kovács S, Wong KW (2006) Fuzzy rule interpolation Matlab toolbox-FRI toolbox. In: 2006 IEEE international conference on fuzzy systems. IEEE, pp 351–357
20. Idri A, Hosni M, Abran A (2016) Improved estimation of software development effort using classical and fuzzy analogy ensembles. *Appl Soft Comput* 49:990–1019
21. Capers JT (2007) Estimating software costs. McGraw-Hill Inc, New York
22. <https://www.kaggle.com/toniesteves/desharnais-dataset>