



## Software Effort Prediction - A Fuzzy Logic Approach

By Sanjay Kumar, Jaya Pal & Vandana Bhattacharjee

*Birla Institute of Technology, India*

*Abstract* - Accuracy in the estimation of software Effort/Cost is one of the desirable criteria for any software cost estimation model. The estimation of effort or cost before the actual development of any software is the most crucial task of the present day software development project managers. Software project attributes are often measured in terms of linguistic values such as very low, low, Average, high and very high. The imprecise nature of such attributes constitutes uncertainty and vagueness in their subsequent interpretation. In this paper we propose a Fuzzy logic based model for software effort prediction. We feel that fuzzy Software cost estimation Model should be able to deal with imprecision and uncertainty associated with various parameter values. Fuzzy analogy model has been developed and validated upon student data.

*Keywords* : software cost estimation, effort prediction, fuzzy logic, linear regression.

*GJCST-C Classification* : D.2.8



*Strictly as per the compliance and regulations of:*



# Software Effort Prediction - A Fuzzy Logic Approach

Sanjay Kumar<sup>α</sup>, Jaya Pal<sup>σ</sup> & Vandana Bhattacharjee<sup>ρ</sup>

**Abstract** - Accuracy in the estimation of software Effort/Cost is one of the desirable criteria for any software cost estimation model. The estimation of effort or cost before the actual development of any software is the most crucial task of the present day software development project managers. Software project attributes are often measured in terms of linguistic values such as very low, low, Average, high and very high. The imprecise nature of such attributes constitutes uncertainty and vagueness in their subsequent interpretation. In this paper we propose a Fuzzy logic based model for software effort prediction. We feel that fuzzy Software cost estimation Model should be able to deal with imprecision and uncertainty associated with various parameter values. Fuzzy analogy model has been developed and validated upon student data.

**Keywords** : software cost estimation, effort prediction, fuzzy logic, linear regression.

## 1. INTRODUCTION

Accurate and timely prediction of the development effort and schedule required to develop a software system is one of the most critical activities in managing software projects. In addition software estimation has been identified as one of the three great challenges for half-century-old computer science. [19] In the last 30 years many different studies have been done in the area of Software Cost Estimation to improve the estimation accuracy and so many models are introduced. The rest of the paper contains the following sections as follows: section II represents Research Method, section III represents Experimental Results, and section IV represents Conclusion and Future Scope.

### a) Fuzzy Logic

Intelligent Systems provide alternative paradigms aimed at facilitating the representation and manipulation of uncertain, incomplete, imprecise or noisy data. Fuzzy logic is a form of many-valued logic or probabilistic logic; it deals with reasoning that is approximate rather than fixed and exact.

The traditional approach to building system controllers requires a prior model of the system. The quality of the model, that is, loss of precision from linearization and/or uncertainties in the system's parameters negatively influences the quality of the

resulting control. It is well known that the fuzzy theory not only provides natural tool for describing quantitative data but also generally produces good performance in many applications. In addition, fuzzy rules allow us to effectively classify data having non-axis-parallel decision boundaries, which is difficult for the conventional attribute-based methods. However, one of the difficulties with fuzzy decision trees is determining an appropriate set of membership functions representing fuzzy linguistic terms. Usually membership functions are given manually, however, it is difficult for even an expert to determine an appropriate set of membership functions when the volume and dimensionality of data are large.

At the same time, methods of soft computing such as fuzzy logic possess non-linear mapping capabilities, do not require an analytical model and can deal with uncertainties in the system's parameters. Although fuzzy logic deals with imprecise information, the information is processed in sound mathematical theory [40]. Based on the nature of fuzzy human thinking, Lofti Zadeh originated the "fuzzy logic" or "fuzzy set theory", in 1965. Fuzzy logic deals with the problems that have fuzziness or vagueness. In fuzzy set theory based on fuzzy logic a particular object has a degree of membership in a given set that may be anywhere in the range of 0 (completely not in the set) to 1 (completely in the set) [41].

For this reason fuzzy logic is often defined as multi-valued logic (0 to 1), compared to bi-valued Boolean logic [42]. Specifically, Fuzzy Logic offers a particularly convenient way to generate a keen mapping between input and output spaces thanks to fuzzy rules' natural expression [2]. Fuzzy logic has been used in [36][37][38][39]. Fuzzy set theory and fuzzy logic are a highly suitable and applicable basis for developing knowledge-based systems This paper presents a fuzzy rule based system having two fuzzy inputs, namely Line of code (LOC) and Adjusted difficulty level (Adj.diff.level) and one output Development time (Devtime) as shown in Figure 1.

**Authors α σ ρ** : Department of CSE, Birla Institute of Technology Ranchi, India. E-mails : k\_sanjay71@yahoo.co.in, jayapal@bitmesra.ac.in, vbhattacharya@bitmesra.ac.in

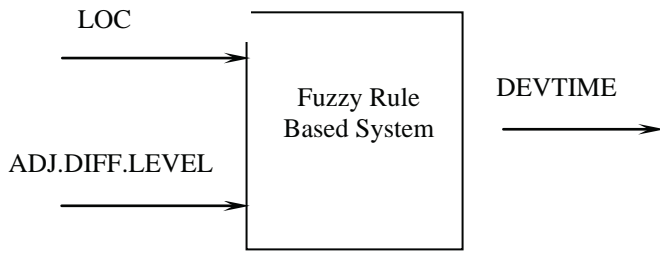
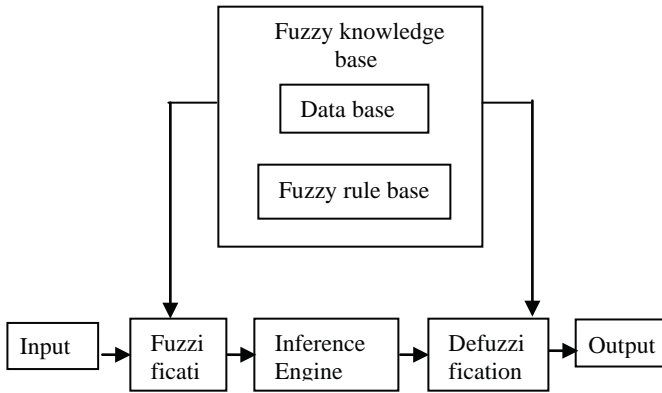


Figure 1 : Fuzzy Rule Based System

b) Fuzzy Rule Based System

A typical fuzzy logic system consists of four main components as shown in Figure.



i. Fuzzification

That contains predefined set of linguistic values. It converts non-fuzzy (Deterministic) inputs of fuzzy system into fuzzy inputs for inference mechanism.

ii. Knowledge Base

That consists of two parts: database that defines linguistic variables conditional statements (Fuzzy sets, and rule base that represents the mapping of fuzzy input set into a fuzzy output set. Rules are fuzzy implications). Fuzzy sets, and rule base that represents the mapping of fuzzy input set into a fuzzy output set. Rules are fuzzy conditional statements (implications).

iii. Decision Logic

That simulates human decision making based on fuzzy concepts. Conclusion of certain condition is derived by decision making logic.

iv. Defuzzification

That converts rule base fuzzy outputs into non-fuzzy (.numerical) values.

Central mechanism of knowledge base and decision making logic considers the fuzzy extension of conventional rule inferencing concept to fuzzy rules inferencing. Premises and conclusions of rules now contain fuzzy values. These facts by definition describe practically continual input set of characteristics. In this manner, one rule can replace more conventional rules. Fuzzy inferencing rules generally connect m conditional

variables  $X_1, \dots, X_m$  to n consequent variables  $Y_1, \dots, Y_n$  in form of:

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  $A_1, \dots, A_m$  and  $B_1, \dots, B_n$  are linguistic terms of linguistic variables  $X_1, \dots, X_m$  and  $Y_1, \dots, Y_n$ , respectively.

The IF part is called the "antecedent" and the THEN part is called the "consequent". To make a decision based on a set of rules, a rules-based system follows these steps:

1. All the rules that apply are invoked, using the membership functions and truth values obtained from the inputs (by a process called fuzzification), to determine the result of the antecedent.
2. This result in turn will be mapped into a membership function and truth value controlling the output variable. This process is known as implication. Two of the more common implication functions are: clipping (the fuzzy set is clipped to a value given by the level of activation of the input variables) and scaling (the fuzzy set is multiplied by a value given by the level of activation of the input variables).
3. These results are combined by a process called aggregation. One common approach for the aggregation involves using the "maximum" of the implicated sets.
4. Finally, a process known as defuzzification is used to compute a single value that is representative of the aggregated fuzzy set.

c) Multiple Regressions

A linear equation with three independent variables (multiple regressions) may be expressed as:

$$y = b_0 + b_1x_1 + b_2x_2 \tag{1}$$

Where  $b_0, b_1,$  and  $b_2$  are constants;  $x_1,$  and  $x_2$  are the independent variables, and  $y$  is the dependent variable. The values of  $b_0, b_1,$  and  $b_2$  of the multiple regression. Equation may be obtained solving the following system of linear equations

$$\sum y = nb_0 + b_1(\sum x_1) + b_2(\sum x_2) \tag{2}$$

$$\sum x_1y = b_0(\sum x_1) + b_1(\sum x_1^2) + b_2(\sum x_1x_2) \tag{3}$$

$$\sum x_2y = b_0(\sum x_2) + b_1(\sum x_1x_2) + b_2(\sum x_2^2) \tag{4}$$

d) Evaluation Criteria

A common criterion for the evaluation of software effort models is the Magnitude of Relative Error (MRE) which is defined as follows:

$$MRE = \left| \frac{\text{Actual devtime} - \text{Predicted devtime}}{\text{Actual devtime}} \right|$$

The MRE value is calculated for each observation whose devtime is predicted. The

aggregation of MRE over multiple observations (N) can be achieved through the Mean MRE (MMRE) as follows:

$$MMRE = \frac{1}{N} \sum_i^N MRE$$

A complementary criterion is the prediction at level *l*, Pred (*l*) = *k*/*N*, where *k* is the number of observations where MRE is less than or equal to *l*, and *N* is the total number of observations. Thus, Pred (25) gives the percentage of development time of software which were predicted with a MRE less or equal than 0.25.

## II. RESEARCH METHOD

### a) Metrics Used

The following metrics have been used Line of Code (LOC), and Adjusted Difficulty Level (ADJ.DIFF.LEVEL) which is served as input to the Fuzzy Logic System. And one output Development Time (DEVTIME).

#### Description of Metrics

1. *Line of Code (LOC)*: Loc is the total number of lines of code used to develop the software excluding the comment lines. This metric was measured on the scale of 0-60.
2. *Adjusted Difficulty Level (ADJ.DIFF.LEVEL)*: This is the difficulty level of the programmers to develop the software which is further adjusted with the help of expert judgments. This metric was measured on the scale of 0-6.
3. *Development Time (Devtime)*: It is the time taken to develop the software. This metric was measured on the scale of 0-24.

### b) Data Gathered

The proposed model was validated by a data set collected from the BIT, students of MCA. This data set consists of 10 project data. The data set is applied to the proposed fuzzy model is shown in the Table 1.

Table 1

PROG ID	LOC	ADJ.DFF.LEVEL	DEVTIME(ACT)
1	16	1.5	3
2	32	1.8	4
3	23	2.3	10
4	18	2.8	15
5	16	1.5	3
6	32	1.3	4
7	16	1.3	4
8	18	1.4	4
9	19	1.8	7
10	22	1.9	5

### c) Fuzzy Rules

The term fuzzy identification usually refers to the techniques and algorithms for constructing fuzzy models from data. The expert knowledge in a verbal

form is translated into a set of if-then rules. A certain model structure can be created, and parameters of this structure, such as membership functions and weights of rules, can be tuned using input and output data.

This paper is based on five fuzzy rules as follows:

1. If ADJ.DIFF.LEVEL is (Very Low) then (DEVTIME (Very Low)). (1).
2. If (loc is Low) and (ADJ.DIFF.LEVEL is (Average)) then (DEVTIME is (Low)) (1).
3. If (loc is Average) and (ADJ.DIFF.LEVEL is (Average)) then (DEVTIME is (Average)) (1).
4. If (loc is High) and (ADJ.DIFF.LEVEL is (Average)) then (DEVTIME is (High)) (1).
5. If (loc is Very High) and (ADJ.DIFF.LEVEL is (High)) then (DEVTIME is (Very High)) (1).

The weight of all the rule is 1.

Input and output Membership Functions (MF) are depicted in Table 2. All are triangular and their scalar parameters (a, b, c) are defined as follows:

Table 2 : Membership Function Characteristics

Variable Name	Range	MF	Parameters		
			a	b	c
LOC	0-60	VL	0	10	20
		L	10	20	30
		AV	20	30	40
		H	30	40	50
		VH	40	50	60
ADJ.DIFF.LEVEL	0-6	VL	0	1	2
		L	1	2	3
		AV	2	3	4
		H	3	4	5
		VH	4	5	6

### Output

Variable Name	Range	MF	Parameters		
			a	b	c
DEVTIME	0-24	VL	0	4	8
		L	4	8	12
		AV	8	12	16
		H	12	16	20
		VH	16	20	24

Where MF is membership function.

The membership function plots corresponding to Table 2 are shown in figures2 (a), 2(b) and 2(c).

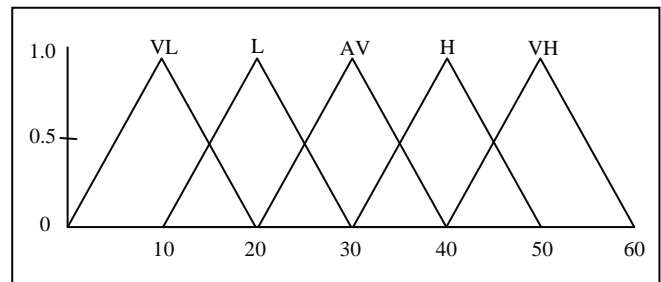


Figure 2(a) : LOC (Input)

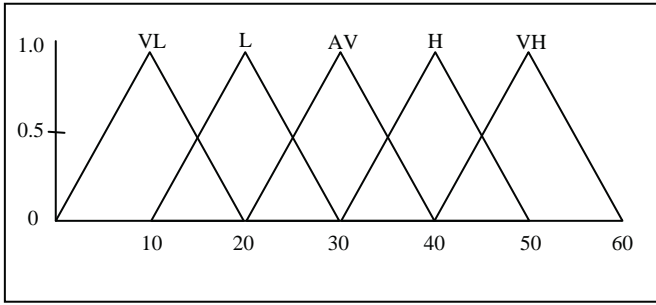


Figure 2(b) : ADJ.DIFF.LEVEL (input)

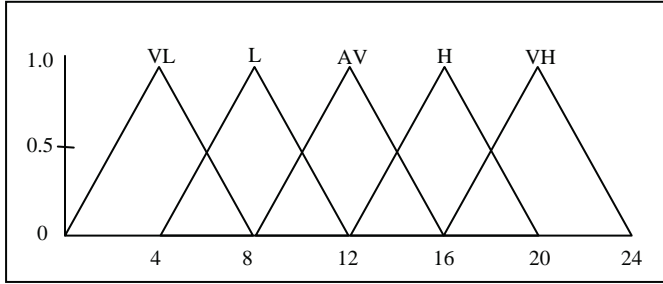


Figure 2(c) : DEVTIME (output)

d) Multiple Regressions

The same dataset has been used in Multiple Regression model to estimate the development time, which is to be used to develop a software. The estimated value of development time is compared with the actual value of development time, and with the help of this, using the evaluation criterion the MRE, MMRE and the Pred(25) value is also calculated.

e) Evaluation criteria

For this model the same evaluation criterion is used. The criterion which is used to evaluate the fuzzy model.

III. EXPERIMENTAL RESULTS

The results show more accuracy in case of effort estimation by the proposed fuzzy model. The result is shown in the Table 3, Table 4 and Table 5.

Table 3

PROG_ID	DEVTIME (ACT)	DEVTIME (PRED) CALCULATED USING FUZZY LOGIG	MRE
1	3	4	0.3333
2	4	4	0.0000
3	10	10	0.0000
4	15	8	0.4667
5	3	4	0.3333
6	4	4	0.0000
7	4	4	0.0000
8	4	4	0.0000
9	7	4	0.4285
10	4	4	0.0000

Table 4

PROG_ID	DEVTIME (ACT)	DEVTIME (PRED) CALCULATED USING LINEAR REGRESSION	MRE
1	3	4.2048	0.4016
2	4	5.7181	0.4295
3	10	9.8191	0.1808
4	15	13.7431	0.0838
5	3	4.2048	0.4016
6	4	2.0155	0.4961
7	4	2.7237	0.3191
8	4	3.3757	1.5606
9	7	6.2935	1.0092
10	4	6.9013	0.4753

Table 5 : Prediction Results

	Multiple Regression	Fuzzy Logic
Min(MRE)	0.0838	0.0000
Max(MRE)	1.5606	0.4667
MMRE	0.5358	0.1762
Pred(25)	0.3	0.6

We have compared the actual development time with the predicted development time given by the model for each data set and found that difference between the actual devtime and predicted devtime. Then we calculated the MRE of each project and MMRE =0.1762 and pred(0.25%) which is 0.6. The same dataset has been tested using multiple regression model and the calculated MMR was used to further find out the MMRE=0.5358 and Pred(0.25%)=0.3. after going through the results we conclude that the proposed fuzzy model gives the better accuracy.

IV. CONCLUSION AND FUTURE SCOPE

The main benefit of this model is its good interpretability by using fuzzy rules and another great advantage of this research is that it can put together expert knowledge (Fuzzy rules) and project data into one general framework that may have a wide range of applicability in software estimation.

Further the comparison with multiple regression model to fuzzy logic model, and the results support the fuzzy logic model.

In our future work we will test this model upon different real-time datasets. Datasets have been collected from the software engineering data repository [34]. These datasets have been used by various researchers in their work [4] [34] [35].

V. ACKNOWLEDGMENT

This research work has been partially funded by UGC [F.No.:33-61/2007(SR)] under financial grants for Major Research Project.

## REFERENCES RÉFÉRENCES REFERENCIAS

1. Ali Idri, Alain Abran, Laila Kjiri(2000), "COCOMO Cost Model Using Fuzzy Logic" in Proc. 7th Intl. Conf. On Fuzzy theory & Technology, 2000.
2. Ali Idri, Alain Abran, T.M. Khoshgoftaar (2002), "Fuzzy Case-Based Reasoning Models for Software Cost Estimation", to appear in the book *Soft Computing in Software Engineering: Theory and Applications* Published by Springer- Verlag, 2002.
3. Imran Attarzadeh, Siew Hock Ow (2010), "Soft Computing Approach for Software Cost Estimation", In international Journal of Software Engineering, IJSE Vol.3 No.1 January 2010.
4. M. Shepperd and C. Schofield. (1997), "Estimating Software Project Effort Using Analogies", IEEE Transactions on Software Engineering, vol. 23, no. 12, pp. 736-743, November 1997.
5. Sarah Jane delany, Pdraig Cunningham and Wolfgang Wilke (1998), "The Limits of CBR in Software Projects Estimation". Presented at 1998 German Workshop on Case-Based Reasoning.
6. Magne Jargensen (2005), "Practical Guidelines for Expert-Judgment-Based Software Effort Estimation", In Proceedings of the 7th ACIS International Conference on Software Engineering Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'06), pp. 57-64, IEEE Computer Society.
7. A. Aamodt and E. Plaza (1994), "Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches", AI Communications, IOS Press, vol. 7:1, pp. 39-59.
8. Hareton Leung, Zhang Fan, "Software Cost Estimation".
9. Krishnamoorthy Srinivasan and Douglas Fisher (1995), "Machine Learning approaches to Estimating Software Development Effort", In IEEE transactions on Software Engineering, Vol.21, No.2, February 1995.
10. Lionel C. Briand, "On the many ways Software Engineering can benefit from Knowledge Engineering".
11. Martin Auer, Adam Trendowicz, Bernhard Graser, Ernst Haunschmid, and Stefan Biffl (2006), "Optimal Project Feature weights in Analogy – Based Cost Estimation: Improvement and Limitations", In IEEE transactions on Software Engineering, Vol.32, No.2, February 2006.
12. M. Wasif Nisar, Yong-Ji WANG, Manzoor Elahi (2008), "Software Development Effort Estimation Using Fuzzy Logic – A Survey", In 5th International Conference on Fuzzy Systems and Knowledge Discovery, IEEE Computer Society.
13. Martin Auer, Stefan Biffl (2004), "Increasing the Accuracy and Reliability of Analogy – Based Cost Estimation with Extensive Project feature Dimention Weighting", In International Symposium on Empirical Software Engineering (ISESE'04), IEEE Computer Society.
14. Christopher Rush and Rajkumar Roy, "Expert Judgement in cost estimating: Modelling reasoning process".
15. G. Kadoda, M. Cartwright, L. Chen, and M. Shepperd. (2000), "Experiences Using Case-Based Reasoning to Predict Software Project Effort", In Proceedings of EASE, p.23-28, Keele, UK.
16. Marcio Rodrigo Braz, and Silvia Regina Vergilio (2006), "Software Effort Estimation Based on Use Cases", In Proceedings of the 30th Annual International Computer Software and Applications Conferences (COMPSAC'06), IEEE Computer Society.
17. Cuauhtemoc Lopez Martin, Jerome Leboeuf Pasqueir, Cornelio Yanez M., Agustin Gutierrez T., "Software Development Effort Estimation using Fuzzy Logic: A Case Study", In Proceedings of the 6th International conference on Computer Science (ENC'05, IEEE Computer Society).
18. Marcio Rodrigo Braz, and Silvia Regina Vergilio (2004), "Using Fuzzy Theory for Effort Estimation of Object – Oriented Software", In Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004), IEEE Computer Society.
19. Cuauhtemoc Lopez Martin, Cornelio Yanez-Marquez, Agustin Gutierrez-Tornes(2006), "A Fuzzy Logic Model Based Upon Reused and New & Cahnged Code for Software Development Effort Estimation at Personal Level", In Proceedings of the 15th International conference on Computing (CIC'06), IEEE Computer Society.
20. A. Idri and A. Abran. (2001), "A Fuzzy Logic Based Measures for Software Project Similarity: Validation and Possible Improvements", In Proceedings of the 7th International Symposium on Software Metrics, pp. 85-96, England, UK, IEEE Computer Society.
21. Ali. Idri, Alain Abran, and Taghi M. Khoshgoftaar, "Computational Intelligence in Empirical Software Engineering".
22. V. Bhattacharjee and S. Kumar,(2004),"Software Cost Estimation and its Relevance in The Indian Software Industry", In Proceedings of the International Conference on Emerging Technologies IT Industry, PCTE, Ludhiana, India.
23. S. Kumar and V.Bhattacharjee, (2005), "Fuzzy Logic Based Model for Software Cost Estimation", In Proceedings of the international Conference on information Technology, Nov'05, PCTE, Ludhiana, India.
24. V. Bhattacharjee and S .Kumar, (2006), "An Expert-Case Based Frame work for Software Cost

- Estimation”, In Proceedings of the National Conference on Soft Computing Techniques for Engineering Application (SCT-2006), NIT Rourkela.
25. S. Kumar and V. Bhattacharjee, (2007), “Analogy and Expert Judgment: A Hybrid Approach to Software Cost Estimation”, In Proceedings of the National Conference on information Technology: Present practice and Challenge, Sep’07, New-Delhi, India.
  26. V. Bhattacharjee, S. Kumar and Ekbal Rashid (2008), “Estimation of Software Development Effort in University Setting: A Case Study”, Presented at National Conference on Architecturing Future IT Systems (NCAFIS ’08), Indore, October 2008.
  27. V. Bhattacharjee, S. Kumar and E. Rashid, (2008), “A Case Study on Estimation of Software Development Effort.”, In Proceedings of the & Technology, Hyderabad.(A National Conference ICACT-2008, Gokaraju Rangaraju Institute of Engineering.P.), India.
  28. V. Bhattacharjee, S. Kumar and Ekbal Rashid (2009), “Case Based Estimation Model using Project Feature Weights”, ISM, Dhanbad. Jharkhand.
  29. V. Bhattacharjee, P. K. Mahanti and S. Kumar, (2009), “Complexity Metric For Analogy Based Effort Estimation”, Journal of Theoretical and Applied Information Technology, Vol. 6, No. 1, 31st july 2009.
  30. B.W. Boehm.(1981), “Software Engineering Economic”, Prentice –Hall.
  31. Internal Reports, Birla Institute of Technology Ranchi.
  32. R. Fairley, (1997) “Software Engineering Concepts”, T M H.
  33. S. Rajasekaran and G. A. Vijayalakshmi Pai (2003), “Newral Networks Fuzzy logic, and Genetic Algorithms Synthesis and Applications”, PHI.
  34. Software Engineering Data Repository. <http://promisedata.org/repository/data/desharnais/desharnais.arff>.
  35. J. M. Desharnais, “Analyse statistique de la productivite des projets informatique a partie de la technique des point des foncti\ on,” University of Montreal, Masters Thesis, 1989.
  36. V. Bhattacharjee, and J. Pal, “Fuzzy Rule Based System for Fine Prediction ction in Accidents” in proceedings of Conference on information Science, Technology and Management, CISTM 2009, MDI Gurgaon, India, July 13-15, 2009.
  37. J. Pal and V. Bhattacharjee, “A Fuzzy Logic System for Software Quality Estimation”, in proceedings ICIT 2009, pp 183-187, 2009.
  38. J. Pal and V. Bhattacharjee., “Movie Analyzer using Fuzzy Rule based System”, in proceedings ICCS 2010, pp 16-21, November 19-20, 2010.
  39. Bhattacharjee, V. and Pal, J. “Applying Fuzzy Clustering To Predict Software Usability”, International Journal of Applied Research on Information Technology and Computing (IJARITAC), Volume: 1, Issue: 3, Sept-Dec, 2010, pp.279-289.
  40. Bose B. K., “Expert systems, fuzzy logic, and neural network application in power electronics and motion control”, Proceeding of the IEEE, vol. 82, Aug. 1994.
  41. L. H. Tsoukalas and R. E. Using, “Fuzzy and Neural Approaches in Engineering”, John Wiley, NY, 1997.
  42. Math Works, Fuzzy Logic Toolbox User’s Guide, Jan., 1998.