Fuzzy Clustering and Optimization Model for Software Cost Estimation

Sweta Kumari [1], Mohammed Ali [2], Shashank Pushkar [3]
[1, 3] Department of Computer Science & Engineering, Birla Institute of Technology
Mesra, Ranchi - 835215, India
[1] swetak44@gmail.com
[3] shashank_pushkar@yahoo.com
[2] Department of Computer Science & Engineering, Birla Institute of Technology
Mesra, Ranchi - 835215, India
[2] mdali619@gmail.com

***Abstract* - Financial health of many organizations now-a-days is being affected by investment in software and their cost estimation. Therefore, to provide effective cost estimation models are the most complex activity in software engineering fields. This paper presents a fuzzy clustering and optimization model for software cost estimation. The proposed model uses Pearson product-moment correlation coefficient and one-way ANOVA analysis for selecting several effort adjustment factors. Further, it applies fuzzy C-means clustering algorithm for project clustering. Then, parameters of COCOMO model have been optimized using Multi-objective Genetic Algorithm (MOGA). Here, two objectives are considered. One is to minimize the Mean Magnitude of Relative Error (MMRE) and other is to maximize the Prediction (PRED). This model has been tested on the COCOMO dataset. The optimization result has also been compared with Multi-objective Particle Swarm Optimization (MOPSO) algorithm. The result has proved superiority of MOGA in parameter optimization for getting strength back the accuracy of software cost estimation.**

Keyword- Software Cost Estimation, Multi-objective Genetic Algorithm, Multi-objective Particle Swarm Optimization, Fuzzy c-means Clustering Algorithm, Constructive Cost Model (COCOMO).

I. INTRODUCTION

Cost estimation for a software project is vital for all its stakeholders. The cost of a project is a function of many parameters. Size is a most important cost factor in most models and can be measured using lines of code (LOC) or thousands of delivered lines of code (KDLOC) or function points [20]. Generally, two types of estimation methods have been derived: algorithmic and non-algorithmic [21]. First method is based on mathematical formula for calculating cost of software projects whereas non-algorithmic methods require data about previous completed similar projects to perform the estimation. A study published by the Standish Group's chaos states that in 30,000 applications development projects, 23% of projects failed, 49% of the projects were being challenged, and only 28% of the projects have been successful [22]. In this context, both overestimates and underestimates of the software effort are harmful to software companies [17]. One of the most important causes of such failures has been inaccurate cost estimates of software projects [1]. Therefore, it is very important to investigate novel models that can provide better estimation capabilities as compared to conventional models to improve the accuracy.

A number of models have been proposed for software cost estimation that are based on the three modes of COCOMO as the cluster, namely organic mode, semi- detached mode, embedded mode [3]. In recent years, some soft computing techniques were explored to construct efficient estimation models. Presently, some researchers were used neural networks [19], genetic algorithms, differential evolution algorithm [24], and intelligent gray theory [15] for software cost estimation and calculation of parameters optimization but their approaches are still based on the clusters of three modes of COCOMO software projects. In this research, an enhanced cost estimation model has been proposed that use a fuzzy c-means clustering algorithm and a MOGA based technique for parameter optimization. Experiment has been done using COCOMO software projects datasets [3]. The clustering result has been found better than the earlier clustering results obtained using the same projects datasets. Further, parameters of COCOMO model have been optimized using MOGA. The optimization result has also been compared with MOPSO algorithm. The result has proved superiority of MOGA in parameter optimization for getting better the accuracy of software cost estimation. The remainder of the paper is divided in different sections as follows: Section II includes a brief literature review about the concepts and techniques used in current model. Section III presents the proposed model based on Fuzzy Clustering and Optimization for software cost estimation. Experiments and results are described in section IV and conclusion of the paper is described in section V and in the last; Appendix shows the result of clustering for three and four clusters.

II. LITERATURE REVIEW

A. COCOMO

[3] Proposed a new cost estimation model called COCOMO. This model is well known mathematical representation for software cost evaluation. It is mainly based on the past experience of software projects and uses LOC as the unit of measure for software size. It is a collection of three variants, namely Basic model, Intermediate model, and Semidetached model. The basic COCOMO model evaluate efforts to make software development and cost as a function of program size articulated in estimated LOC [5], [9]. The effort is calculated using the following equation:-

$$Effort \ = \ a^{*} \ (KLOC)^{b} \tag{1}$$

Where, effort evaluated in person-month and KLOC is estimated number lines of code for the project. The value of the parameters a and b based on the project type. Software projects are classified into three categories based on the complexity of the projects namely organic, semi-detached and embedded. (for organic projects *a*=2.4, *b*=1.05, for semi-detached *a*=3.0, *b*=1.12 and for embedded *a*=3.6, *b*=1.20).

Intermediate COCOMO model calculates the estimation of software development effort as a function of program size and set of cost drivers that include individual assessment of the products, hardware, personnel and project attributes. Here, effort is calculated using the following equation:-

$$Effort \ = \ a^{*} \ (KLOC)^{b} * EAF \tag{2}$$

The value of the parameters a and b based on the project type (for organic projects *a*=3.0, *b*=1.05, for semi-detached *a*=3.0, *b*=1.12 and for embedded *a*=2.8, *b*=1.20) and *EAF* (Effort Adjustment Factor) is calculated using 15 cost drivers. Each cost drivers is rated from ordinal scale ranging from low to high.

Detailed COCOMO model is a more detailed classification of 15 factors for the project on each step of the software engineering process. Various phases are used in detailed COCOMO model including requirement gathering & planning, system Architecture & Design, detailed design, components & sub-component code, unit test and integration testing. The weights (W_i) are defined accordingly. Here, effort is calculated using the following equation:-

$$Effort \ = \ a^{*} \ (KLOC)^{b} \ * \ EAF * sum\,(W_i) \tag{3}$$

B. Pearson product-moment correlation coefficient

Pearson product-moment correlation coefficient [12] is a measure of the linear dependence between two variables *X* and *Y*. It is denoted by r and its value ranges between +1 and -1, where 1 is total positive correlation, 0 is no correlation and -1 is the total negative correlation. It is widely used to show the linear relationship between two sets of data.

$$r \ = \ \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2}} \tag{4}$$

where, r is the Pearson correlation coefficient for the XY co-variance divided by the standard deviation of X and Y standard deviation of the product, X and Y are the two variables, the arithmetic mean of X is $\overline{X}$, and that of Y is $\overline{Y}$.

C. One-Way ANOVA

One-Way ANOVA is a general method for studying sampled data relationship. It has been widely used in various fields for example, Chen and his colleagues have used one-way ANOVA in genetic engineering [4], Tang has used it in hotel staff job satisfaction analysis [23], Ropponen has used it to software development risks [10].

D. Fuzzy C-Means clustering algorithm

Fuzzy C-Means clustering is a method of clustering which allows one piece of data to belong to two or more clusters [7], [6]. A main objective of this algorithm is to minimize: -

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} U_{ij}^{m} \ \left\|X_i \ - \ C_j\right\|^2 \quad , \ \ 1 \ \leq m \ < \ \infty \tag{5}$$

where, $'m'$ is the fuzziness index $m \in [1\,,\infty]$, $'N'$ is the number of data points, $'C'$ denotes the number of cluster center, $'X_i'$ is the i^{th} of d-dimensional measured data, $'U_{ij}'$ is the membership of X_i in the cluster j, $'C_j'$ is the d-dimensional center of the cluster, $'X_i - C_j'$ is the Euclidean distance between i^{th} data and j^{th} cluster center.

E. Multi-Objective Genetic Algorithm

MOGA deals with solving optimization problems which involve multiple objectives such as minimizing cost and maximizing reliability and others objectives. It is different from single objective optimization in that in MOGA problem, there does not exists a single solution that simultaneously optimizes each objective. Here, the main task is to find out the trade-off surface, which is a set of non-dominated solution points, known as Pareto-optimal or non-inferior solutions. It has been seen that none of the solutions in the non-dominated set is extremely better than any other; any one of them is an acceptable solution. The choice of one solution over the other requires problem knowledge and a number of problem-related factors [2].

F. Evaluation Criteria

Evaluation criteria are essential for calculating the estimation accuracy of software cost proposed estimation model [13]. Here, MMRE and PRED (0.25) are used.

MMRE:- is average percentage of the absolute values of the relative errors over a whole dataset. It can be calculated by the following equation:-

$$MMRE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Predicted\ Value - Actual\ Value}{Actual\ Value}\right| \quad (6)$$

PRED (0.25):- It is defined as the percentage of predictions falling within 25% of the actual known value. It can be calculated by the following equation:-

$$PRED\ (0.25) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Predicted\ Value - Actual\ Value}{Actual\ Value} \leq 0.25\right) \quad (7)$$

Here, n is the number of projects. These two evaluation criteria are considered as the objective function for MOGA to search optimal parameter of COCOMO. Here, our main objective is to minimize the MMRE and other is to maximize the PRED. Generally, the optimization algorithms in this case are implemented to minimize the objectives. To maximize PRED, we take the reciprocal of PRED. Another way is to use negative sign in front of objective function to convert minimization to maximization.

$$f_1 = MMRE \quad (8)$$

$$f_2 = \frac{1}{PRED} \quad (9)$$

III. FUZZY CLUSTERING AND OPTIMIZATION BASED ESTIMATION MODEL

The fig. 1. below shows the main steps of proposed Fuzzy Clustering and Optimization Model. This model is based on the COCOMO - 81 dataset. The fifteen cost adjustment factors described in the Intermediate COCOMO model is subject to statistical analysis techniques. Pearson correlation analysis of each of the fifteen cost factors with actual effort is done. Those cost factors are selected that are more correlated with the actual effort. Similarly, one-way ANOVA analysis of the each of the fifteen cost factors with software effort is also done. The test is done to find if there exists a difference between the means of the actual effort and a cost factor. Those cost factors are selected whose means are more significantly different with the effort. The selected factors (on the basis of statistical analysis) and the effort form the basis of fuzzy clustering of the software projects. The results of fuzzy clustering are subject to defuzzification to convert the fuzzy partitions into crisp sets, so that optimization can be performed on it. Finally, parameters of COCOMO model have been optimized using MOGA.

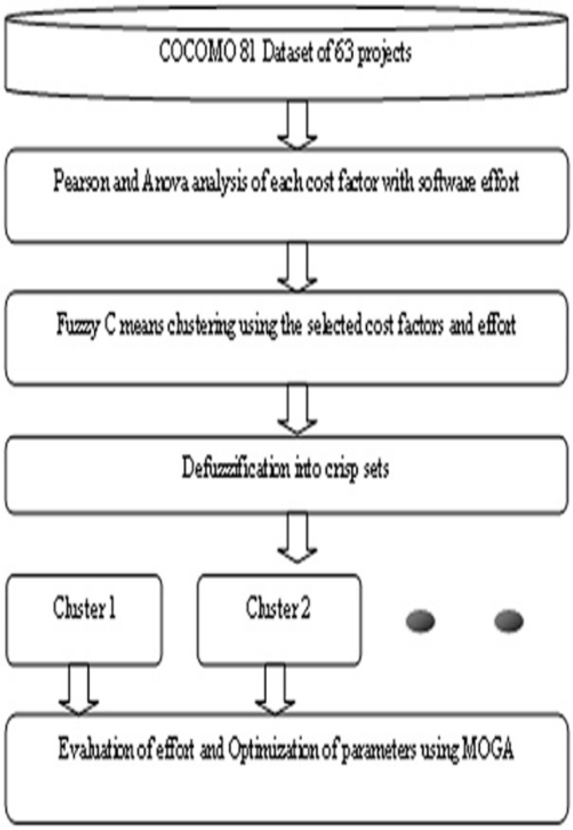

Fig.1. Fuzzy Clustering and Optimization Model.

A. Fuzzy C-Means clustering

Eight cost adjustment factors were selected by the statistical analysis techniques. These eight variables and software effort, a total of nine variables form the basis of fuzzy clustering. Thus, clustering is done using the nine variables, for all the 63 projects in the dataset. A project belongs to all groups or clusters with some membership in fuzzy clustering. In Fuzzy C means or in K means clustering the number of clusters is fixed initially. However we do not know the optimal number of clusters in the data. The research fixes the number of clusters as three and four for experimental study [11]. The Fuzzy C means algorithm implemented is as follows:

Step 1: Initialize center vector C_{jd} $[C_{j1},\ C_{j2}, \ldots, C_{jd}]$, for d - dimensional data $[d = 9]$ (the selected nine variables) and j clusters. Also, initialize an initial fuzzy membership matrix $U_{(0)} = [U_{ij}]$.

Step 2: Calculate the center vectors C_j with U_{ij} by the following equation.

$$C_j = \frac{\sum_{i=1}^{N} U_{ij}^{m} . X_i}{\sum_{i=1}^{N} U_{ij}^{m}} \tag{10}$$

Step 3: Calculate the new fuzzy membership matrix U_{K+1} by finding the Euclidean distance between the data point X_i and cluster center C_j by the following equation.

$$U_{ij} = \frac{1}{\sum_{K-1}^{C} \left(\frac{\| x_i - c_j \|}{\| x_i - c_k \|} \right)^{\frac{2}{m-1}}} \tag{11}$$

Step 4: Check if $\|U_{K+1} - U_K\| < \varepsilon$ then stop, else return to step 2. Here K is the iteration step and ε are the termination criteria between $[0, 1]$.

B. Parameter Optimization using MOGA

A multi-objective optimization problem has a number of objective functions which are to be minimized or maximized. Mathematically, we can write multi-objective problem as:-

Maximize/Minimize $y = f(x) = \{f_1(x), f_2(x), \cdots, f_m(x)\}$

Subject to $g(x) = \{g_1(x), g_2(x), \cdots, g_j(x)\} \leq 0$

$h(x) = \{h_1(x), h_2(x), \cdots, h_k(x)\} = 0$

Where $x = \{x_1, x_2, \cdots, x_n\} \in X$

$$y = \{y_1, y_2, \cdots y_m\} \in Y \tag{12}$$

and x is the vector of decision variable , y is the objective vector, X is the decision space and Y is called the objective space. In this research, we use weighted sum approach [16] to combine multiple objectives into single objective for example:-

$$f(x) = w_1 f_1(x) + w_2 f_2(x) + \cdots + w_m f_m(x) \tag{13}$$

Where $f_1(x), f_2(x), \cdots, f_m(x)$ are the objective functions and $w_1, w_2, \cdots, w_m$ are the weights of corresponding objectives are normalized that satisfy the following conditions:-

$$w_i \geq 0 \quad \forall_i = 1, 2, \cdots, m$$

$$w_1 + w_2 + \cdots + w_m = 1$$

In this case $F = w_1 f_1 + w_2 f_2$, where $w_1 + w_2 = 1$ (14)

Here, we use different-different weights to obtained Pareto-optimal solutions. Generally, weights are random numbers within $(0, 1)$. The algorithmic steps of MOGA are as follows:-

Step 1. An initial population of individuals is randomly generated.

Step 2. Calculate the two objective functions f_1 and f_2 using equations (8) and (9) for every chromosome. Convert the multi-objective into single-objective by using equation (14), for every chromosome in the population this is the final fitness function to minimize.

Step 3. The roulette wheel selection is applied here to select chromosomes from the current population.

Step 4. Heuristic crossover is applied here to produce new chromosomes.

Step 5. Adaptive feasible mutation is applied here.

Step 6. The Elite strategy is used in order to keep best solution in each generation.

Step 7. The new population replaces the current one.

Step 8. If the end condition is satisfied, stop, and return the best solution in current population, if not then go to Step 2.

C. Parameter Optimization using MOPSO

A detail description of MOPSO can be found in [8], [18]. Here, MOPSO has been used to optimize the parameters of COCOMO model. This is done only for a comparison purpose. MOPSO algorithm uses same process such as parameter, velocity update and position update as in PSO except the objective function. The objective function contains multiple objectives. Here, the objectives are same as discussed in above mentioned MOGA. The algorithmic steps of MOPSO are as follows:-

Step 1. Initialize a population of i particles with random position $[p_{i1}, p_{i2}, \ldots, p_{id}]$ and velocity vectors $[v_{i1}, v_{i2}, \ldots, v_{id}]$ in d dimension space $[d = 2]$.

Step 2. Initialize all the particles as Pbest.

Step 3. Calculate the two objective functions f_1 using equation (8) and f_2 using the following equations (15) for every particle. Now, convert the multi-objective into single-objective by using equation (14) for each particle, this is the final fitness function to minimize.

$$f_2 = 1 - PRED \tag{15}$$

Step 4. For every particle if current fitness is better than Pbest then set Pbest = current fitness.

Step 5. Select the best Pbest from all the particles and set it as Gbest.

Step 6. Update velocity and position of each particle using equations (16) and (17).

Step 7. Repeat Step 3 to Step 6 until the stop condition.

$$V_{id} = w \times V_{id} + c_1 \times Rand() \times (p_{id} - x_{id}) + c_2 \times Rand() \times (p_{gd} - x_{id}) \tag{16}$$

$$x_{id} = x_{id} + V_{id} \tag{17}$$

Here, Pbest and Gbest are local and global optimum solutions, p_{id} is Pbest, p_{gd} is Gbest, population size of the swarm is taken to be 80. The cognitive and social parameter c_1 and c_2 are set to 2. Inertia weight w is 0.9 initially to promote global exploration and gradually decreased to 0.4 to get more refined solution and $Rand()$ is between 0 and 1 random numbers [8].

IV. EXPERIMENT AND RESULTS

In this research, COCOMO [3] dataset has been used to evaluate the proposed model. Here, we use Pearson product-moment correlation coefficient to analyze the linear association between each of the fifteen cost adjustment factors and the actual software effort. The correlation coefficient $'r'$ is calculated using the equation (4). X is a cost adjustment factor and Y is the actual effort for a project. Therefore, $\bar{X}$ is the arithmetic mean of X; $\bar{X}$ is the average of all the cost factors for all the 63 projects in the dataset. Similarly, $\bar{Y}$ is the arithmetic mean of Y; $\bar{Y}$ is the average of all effort values for the 63 projects in the dataset. Through Pearson correlation analysis, a comparatively strong positive association (+0.4493) is found between the cost factor Database size (data) and Effort. This means that as the value of the cost factor Database size increases, the value of software Effort also increases. Similarly, linear association is also found between each of the cost factors Modern Programming Practices (modp), Required Software Reliability (rely) and Computer Turnaround Time (turn) and the Effort. Thus these four effort or cost factors are selected for clustering. The other cost factors show weaker association, hence, they are not selected. The results are shown in the chart below. The x-axis represents the fifteen cost factors and the y-axis gives the $'r'$ value of the analysis of effort with the corresponding cost factor.

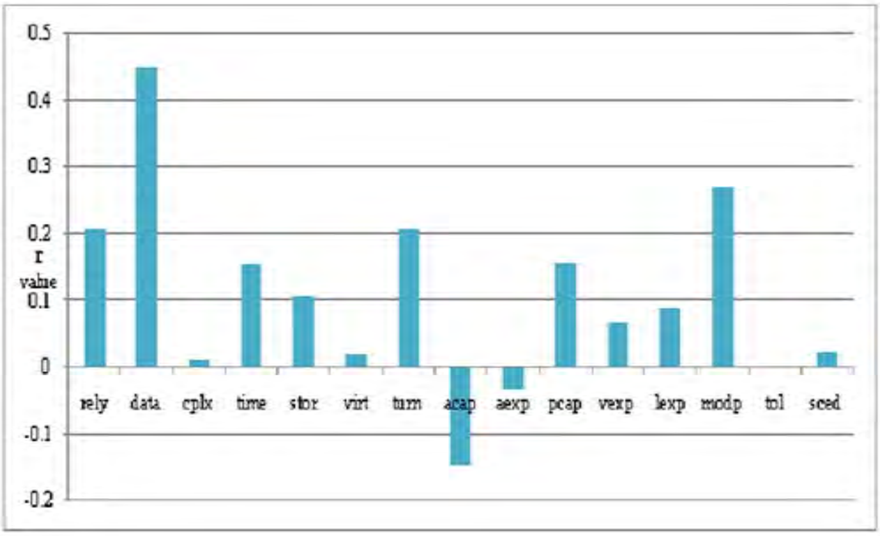

Fig. 2. COCOMO factors analyzed by Pearson correlation analysis

After that, we use one-way ANOVA analysis to test whether or not the means of several variables are equal. It is done to find out if there is a noteworthy difference among the means of two unrelated group, the group of the cost adjustment factor and the effort. A cost factor out of the fifteen cost adjustment factors is chosen to analysis with the actual effort in the COCOMO 81 dataset. The null hypothesis is that the means of the two variables are equal.

$$H_0 : \mu_k = \mu_e \tag{18}$$

Where μ_k is the k^{th} cost factor $k = 1, 2, 3, \ldots, 15$; and μ_e is the effort. The null hypothesis is rejected, if P value is less than significance level α (α =0.005). Greater F-value of the test signifies that the sample means μ_k and μ_e is more significantly different. The selected factors are Analyst Capability (acap), Applications Experience (aexp), Programmer Capability (pcap), and Programming Language Experience (lexp). Computer Turnaround Time and Modern Programming Practices, even though their F-value is high, were already selected in the correlation analysis hence they are not selected again over here. The F-values for cost factors Virtual Machine Volatility (virt), Virtual Machine Experience (vexp) and Programming Language Experience(lexp) are approximately same, but Programming language Experience is selected because it has more $'r'$ value (in correlation analysis) than virt and vexp. The result of one-way ANOVA is given in the chart below. The x-axis represents the fifteen cost factors and the y-axis gives the F value of the ANOVA analysis of effort with the corresponding cost factor.

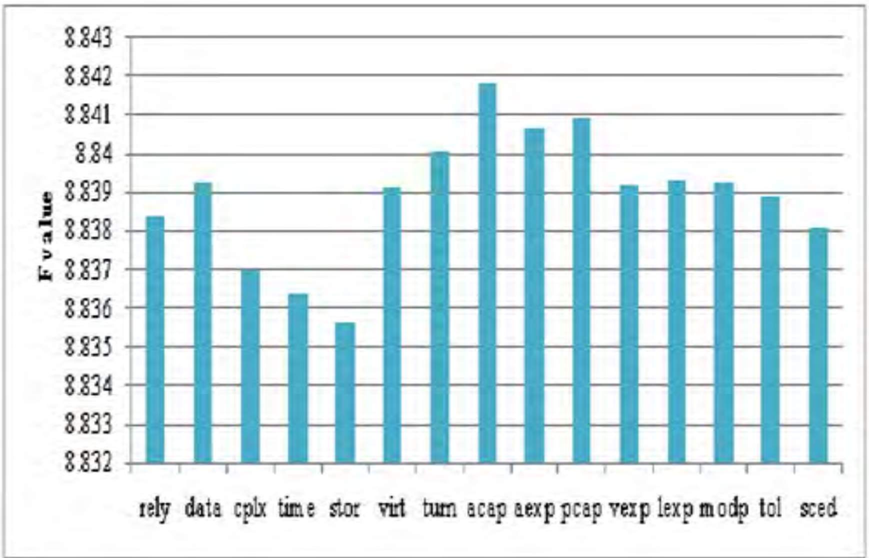

Fig. 3. COCOMO factors analyzed by One-Way ANOVA analysis

The above selected eight effort adjustment factors and effort is selected for clustering. Here, we use Fuzzy C-Means clustering algorithm for project clustering. On the other hand, it has been seen that in COCOMO dataset, the value of the effort is much higher than that of the effort factors; hence the results of clustering are easily impacted by the effort values. This makes the clustering poor. To make the clustering depend on all variables we take the logarithm conversion of effort that makes the clustering better. The resultant fuzzy membership matrix due to the process of clustering into three and four groups is given in the appendix Table II & III. In fuzzy clustering a project has some membership in all clusters. In order to do optimization, crisp sets are needed. This fuzzy to crisp conversion is done by the maximum membership defuzzification process [14]. The project goes into that cluster in which its membership is highest. The y-axis gives the cluster number and x-axis gives the membership of a project in the cluster to which it has been assigned. The result after defuzzification for three and four clusters is shown below: -

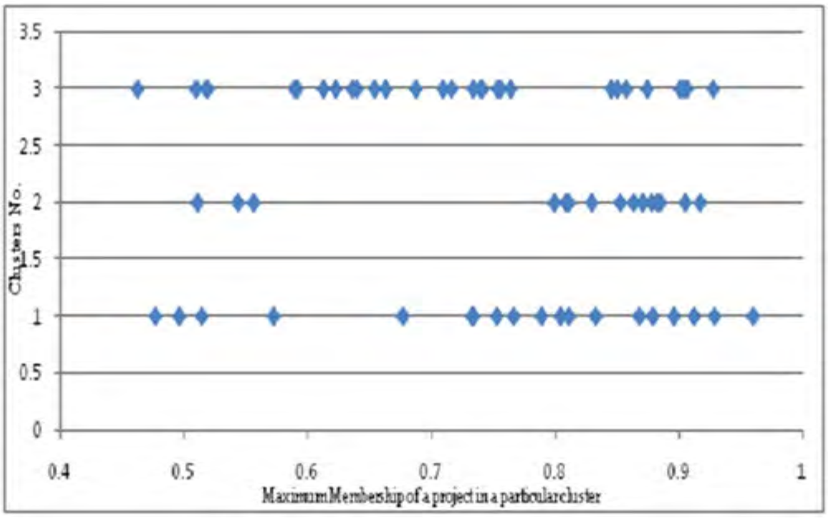

Fig. 4. Result of defuzzification of fuzzy sets into crisp sets for three clusters.

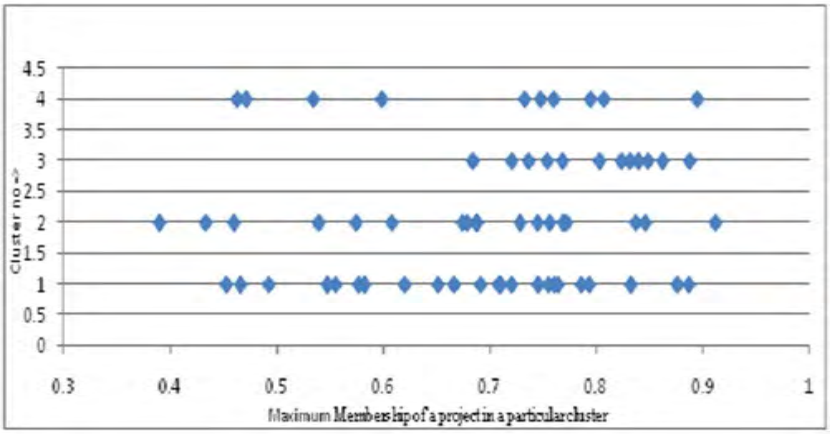

Fig. 5. Result of defuzzification of fuzzy sets into crisp sets for four clusters.

After obtaining crisp sets, each group has been identified by MOGA and MOPSO to optimize COCOMO parameters. One of the most important advantages of the multi-objective optimization is that it gives several non-dominated solutions in which a user can choose any one of the best solution from a set of solutions according to his or her preference. Here, two objectives are considered. One is to minimize the MMRE and other is to maximize the Prediction. Fig. 6 to 11 represents the average values of the above mentioned optimization algorithms for three and four clusters with different-different weights. A comparative result is presented in Table I. By making a comparison between the obtained values in Fig. 6 to 11 and Table I, it has been seen that the performance of MOGA are better than the MOPSO algorithms in terms of maximum prediction. Here, we focus on finding maximum prediction with lower MMRE. The best compromised solution of the proposed model shows that this model is effective for estimation of software cost. However, generally research on software cost estimation is based on three project type of COCOMO. It is little related research that using project database of the effort adjustment factor and software effort to clustering project and estimation software effort.

TABLE I. Comparative results on related research data.

Estimation Model	**MMRE**	**PRED(0.25)**
MOPSO result with three group	0.28	0.58
MOGA result with three group	**0.28**	**0.61**
MOPSO result with four group	0.25	0.57
MOGA result with four group	**0.26**	**0.65**
COCOMO	0.26	0.54

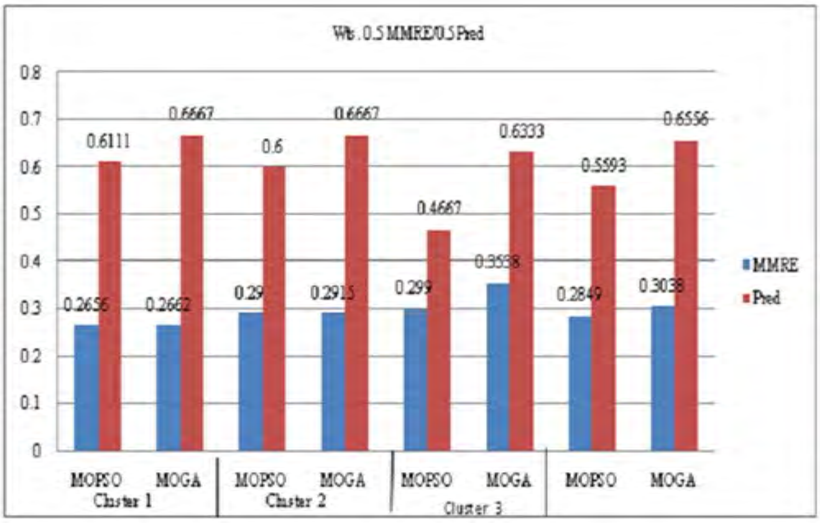

Fig. 6. Average number of Pareto-optimal solutions obtained by Wts. 0.5 MMRE/0.5 Pred.

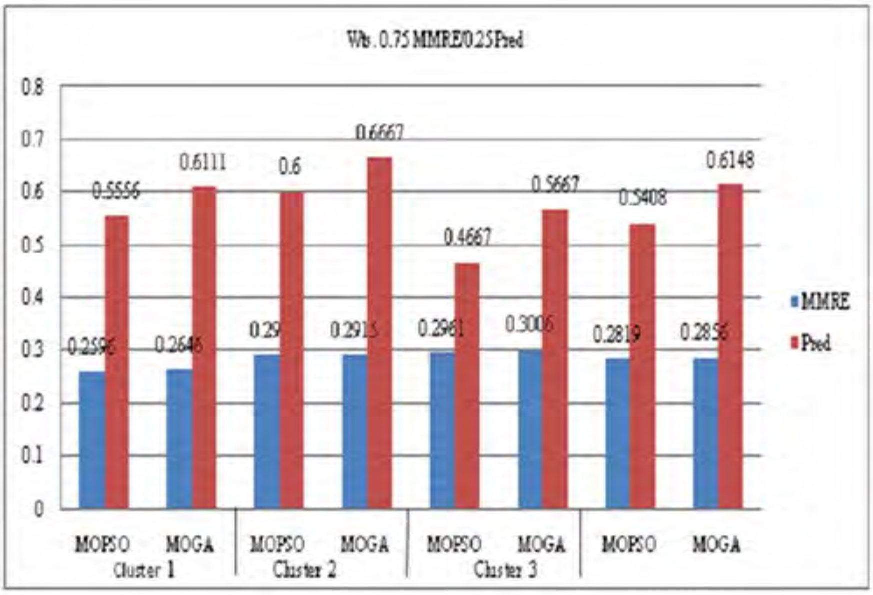

Fig. 7. Average number of Pareto-optimal solutions obtained by Wts. 0.75 MMRE/0.25 Pred.

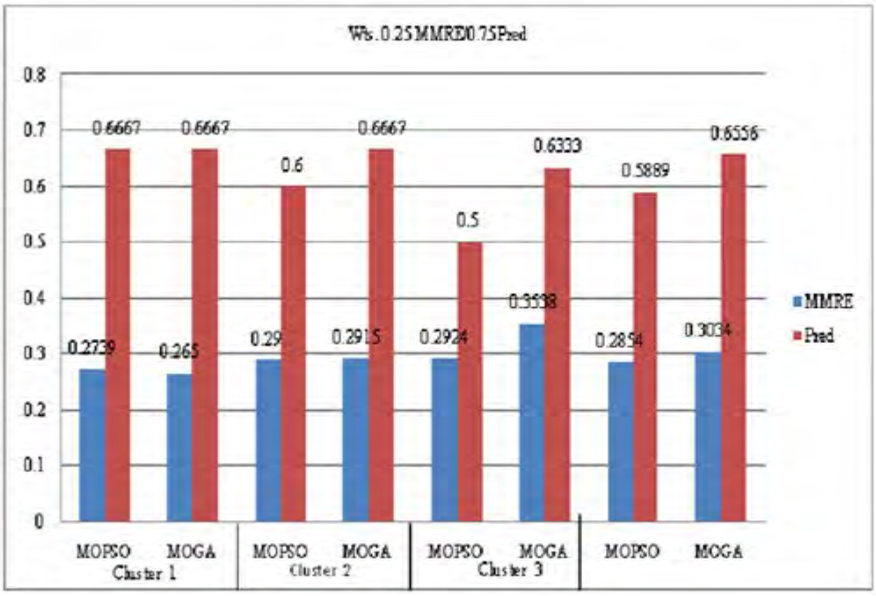

Fig. 8. Average number of Pareto-optimal solutions obtained by Wts. 0.25 MMRE/0.75 Pred.

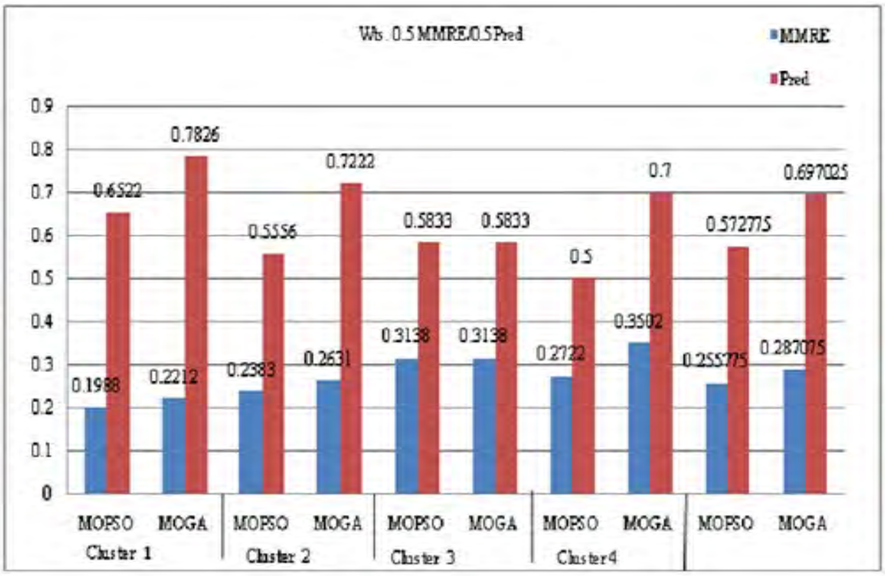

Fig. 9. Average number of Pareto-optimal solutions obtained by Wts. 0.5 MMRE/0.5 Pred.

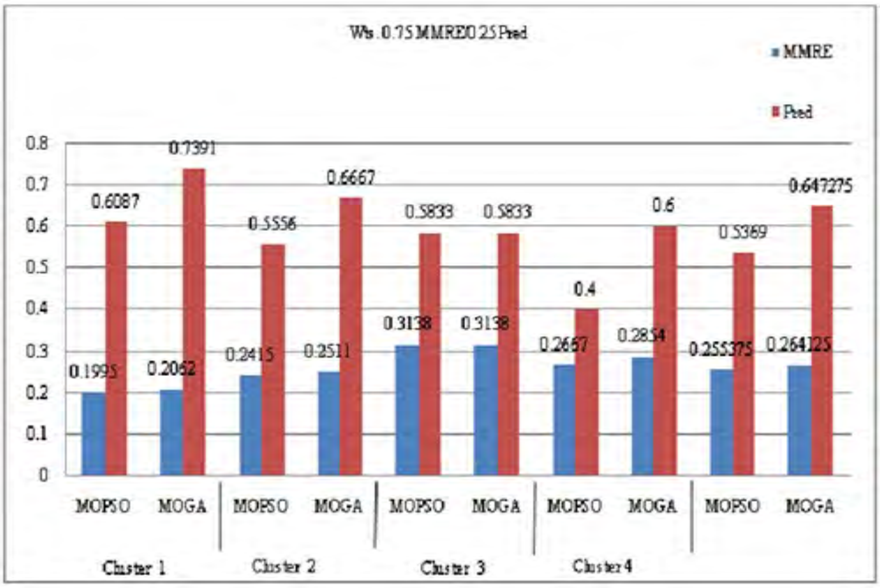

Fig. 10. Average number of Pareto-optimal solutions obtained by Wts. 0.75 MMRE/0.25 Pred.

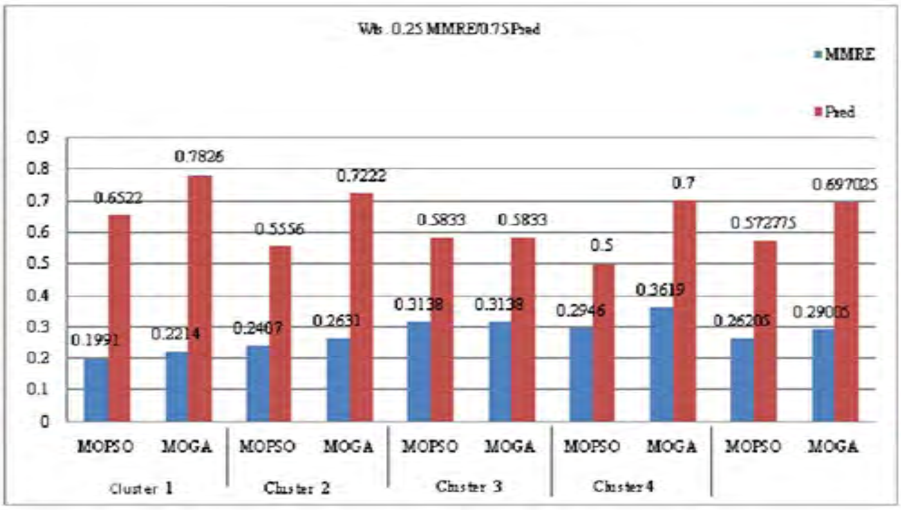

Fig. 11. Average number of Pareto-optimal solutions obtained by Wts. 0.25 MMRE/0.75 Pred.

V. CONCLUSION

This paper proposes an efficient Fuzzy Clustering and Optimization Model for Software Cost Estimation. This model aims to utilize some statistical technique with clustering and optimization algorithms that help to improve the accuracy of software cost estimation. Here, the focus is finding maximum prediction with minimum MMRE. The model has been tested using COCOMO dataset. This paper also provides a comparative result between the performance of MOGA and MOPSO to optimize the COCOMO parameter for estimation of software cost. It has been observed that the results obtained from MOGA are better than the MOPSO in terms of accuracy.

The proposed model cannot minimize the MMRE beyond a certain point. On increasing the weights for MMRE, the MMRE does not decrease but the PRED decreases. Modifications have to be done to the model such that the trade-offs in the weights decreases the MMRE even further. The future work is to investigate some more optimization and data clustering algorithms that are able to provide accurate cost with maximum PRED and minimum MMRE.

REFERENCES

[1] A.L.I. Oliveira, Estimation of software project effort with support vector regression, Neurocomputing 69 (13–15) (2006) 1749–1753. <http:// dx.doi.org/10.1016/j.neucom.2005.12.119>.
[2] A. Osyczka, "Multicriteria optimization for engineering design", in Design Optimization, J. S. Gero, Ed. New York: Academia, pp. 193-227, 1985.
[3] Boehm, B.W., "Software Engineering Economics", Englewood Cliffs, NJ: Prentice-Hall, 1981.
[4] Dechang Chen, Dong Hua, Jaques Reifman, Xiuzhen Cheng, "Gene Selection for Multi-Class Prediction of Microarray Data", IEEE Proceedings of the Computational Systems Bioinformatics (CSB'03), 2003.
[5] F.K. Chris, "An Empirical Validation of Software Cost Estimation Models", Management of Computing-Communications of ACM, Vol: 30, No. 5, pp. 416-429, May 1987.
[6] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algoritms", Plenum Press, New York, 1981.
[7] J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", Journal of Cybernetics 3: pp. 32-57, 1973.
[8] J. Kennedy and R. C. Eberhart, "Particle swarm optimization", IEEE International Conference on Neural Network, 4, 1942-1948, 1995.
[9] J. Magne and S. Martin, " A Systematic Review of Software Development Cost Estimation Studies", IEEE Transactions On Software Engineering, Vol. 33, No. 1, pp. 33-53, January 2007.
[10] Janne Ropponen and Kalle Lyytinen, "Components of Software Development Risk: How to Address Them? A Project Manager Survey", IEEE Transactions on Software Engineering, Vol.26, NO.2, 2000.
[11] Jin-Cherng Lin and Han-Yuan Tzeng , "Applying Particle Swarm Optimization to Estimate Software Effort by Multiple Factors Software Project Clustering" , Computer Symposium (ICS), 2010 International, pp 1039 - 1044 , 16-18 Dec. 2010.
[12] Joseph Lee Rodgers and W. Alan Nicewander, "Thirteen Ways to Look at the Correlation Coefficient", The American Statistician, Vol. 42, No. 1 (Feb., 1988), pp. 59-66, 1988.
[13] K. Moløkken-Østvold et al., "Project Estimation in the Norwegian Software Industry - A Summary", 2004, Simulation Research Laboratory.
[14] Mogharreban, N. and Dilalla, L.F., "Comparison of Defuzzification Techniques for Analysis of Non-interval Data", pp. 257 – 260, Fuzzy Information Processing Society, 2006. NAFIPS 2006. Annual meeting of the North American, 2006.
[15] Mohammad Azzeh, Daniel Neagu, Peter I. Cowling, "Fuzzy grey relational analysis for software effort estimation", Empirical Software Engineering, Vol.15(1), pp.60-90, Feb. 2010.
[16] Pei-Chann Chang, Jih-Chang Hsieh, Yen-Wen Wang, "Development of Multi-objective Genetic Algorithms for Scheduling", Proceedings of the Fifth Asia Pacific Industrial Engineering and Management Systems Conference 2004.
[17] R. Agarwal, M. Kumar, Yogesh, S. Mallick, R.M. Bharadwaj, D. Anantwar, Estimating software projects, SIGSOFT Software Engineering Notes 26 (4) (2001) 60–67. <http://doi.acm.org/10.1145/505482.505491>.
[18] S.Lalwani, S. Singhal, R. Kumar and N. Gupta, "A Comprehensive Survey: Applications of Multi-objective Particle Swarm Optimizations (MOPSO) Algorithm", Transactions of Combinatorics, Vol. 2, No. 1, pp 39-101, 2013.
[19] Sun-Jen Huang , Nan-Hsing Chiu "Applying fuzzy neural network to estimate software development effort" Applied Intelligence, Springer Science+Business Media, Vol. 30(2), LLC,pp.73-83, 2007.
[20] Sweta Kumari and Shashank Pushkar, "Comparison and Analysis of Different Software Cost Estimation Methods", International Journal of Advanced Computer Science and Applications, Vol. 4, No.1, pp. 153-157, 2013.
[21] Sweta Kumari and Shashank Pushkar, "Performance Analysis of the Software Cost Estimation Methods: A Review", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, Issue 7, ISSN: 2277 128X, pp. 229-238, July 2013.
[22] The Standish Group Interaction, "EXTREME CHAOS Report 2001".
[23] Xiuli Tang, Yingkang Gu, " Research on Hotel Staff Job Satisfaction: the Case of Shanghai", IEEE Information Management and Engineering (ICIME), pp. 118-120,April 2010.
[24] Yucheng Kao, Jin-Cherng Lin , Jian-Kuan Wu "A Differential Evolution Approach for Machine Cell Formation" IEEE International Conference on Industrial Engineering and Engineering Management, pp.772-775, 2008.

APPENDIX

The below table II shows the result of clustering for three clusters. The first column denotes the project number and the second, third and fourth column gives the fuzzy membership of the particular project in the groups or clusters 1, 2 and 3 respectively.

TABLE II. Result of clustering for 3 clusters:

Project no.	cluster 1	cluster 2	cluster 3
1	0.80444	0.054572	0.14099
2	0.91255	0.022408	0.06504
3	0.2505	0.086137	0.66336
4	0.33369	0.15628	0.51003
5	0.053515	0.55656	0.38992
6	0.13812	0.39907	0.46281
7	0.025092	0.885	0.089908
8	0.92888	0.016686	0.054431
9	0.57274	0.065371	0.36189
10	0.4288	0.09385	0.47735
11	0.25799	0.1054	0.63661
12	0.1604	0.07531	0.76429
13	0.076302	0.17	0.7537
14	0.059166	0.20676	0.73407
15	0.091327	0.25428	0.6544
16	0.073439	0.40813	0.51843
17	0.044209	0.79956	0.15623
18	0.73415	0.087873	0.17798
19	0.78916	0.066446	0.14439
20	0.76666	0.074502	0.15884
21	0.89628	0.028644	0.075078
22	0.86824	0.026644	0.10511
23	0.75304	0.04365	0.20331
24	0.51471	0.094087	0.3912
25	0.67718	0.059562	0.26326
26	0.49654	0.073021	0.43044
27	0.048889	0.093378	0.85773
28	0.038442	0.057311	0.90425
29	0.043213	0.80898	0.14781
30	0.044411	0.81121	0.14438
31	0.87902	0.029062	0.091917
32	0.81145	0.040003	0.14855
33	0.73298	0.055478	0.21154
34	0.21274	0.07747	0.70979
35	0.083275	0.17524	0.74148
36	0.079943	0.29702	0.62303
37	0.05231	0.35608	0.59161
38	0.016071	0.9174	0.066525
39	0.028462	0.87106	0.10048
40	0.030172	0.86348	0.10635
41	0.034488	0.85275	0.11276
42	0.050307	0.35996	0.58974
43	0.022933	0.049133	0.92793
44	0.042493	0.08296	0.87455

45	0.038374	0.054751	0.90688
46	0.047292	0.051503	0.90121
47	0.067557	0.54406	0.38838
48	0.83292	0.042121	0.12496
49	0.13442	0.10927	0.7563
50	0.09743	0.057054	0.84552
51	0.19228	0.19462	0.6131
52	0.062181	0.41806	0.51976
53	0.021946	0.8823	0.095759
54	0.018577	0.87807	0.10335
55	0.029675	0.82976	0.14056
56	0.96017	0.008765	0.031066
57	0.21053	0.07293	0.71654
58	0.16014	0.15216	0.6877
59	0.036832	0.11273	0.85044
60	0.054052	0.2058	0.74014
61	0.051617	0.30836	0.64002
62	0.060858	0.51125	0.4279
63	0.017264	0.90527	0.077465

The table III shows the result of clustering for four clusters. Here also the first column denotes the project number and the second, third, fourth and fifth column gives the fuzzy membership of the particular project in the groups or clusters 1, 2, 3 and 4 respectively.

TABLE III. Result of clustering for 4 clusters:

Project no	cluster 1	cluster 2	cluster 3	cluster 4
1	0.067706	0.16868	0.030388	0.73323
2	0.042805	0.14418	0.017815	0.7952
3	0.15586	0.75658	0.033674	0.053885
4	0.32765	0.3905	0.10858	0.17326
5	0.58303	0.083947	0.30552	0.0275
6	0.45317	0.18435	0.2815	0.080978
7	0.092545	0.030988	0.8627	0.013767
8	0.076023	0.35977	0.029382	0.53482
9	0.038978	0.91229	0.01064	0.038087
10	0.11769	0.76936	0.033725	0.07922
11	0.20262	0.67478	0.050744	0.071859
12	0.19149	0.72906	0.035847	0.043599
13	0.62048	0.20895	0.12807	0.042502
14	0.74582	0.11231	0.11554	0.026329
15	0.5774	0.19516	0.17843	0.049003
16	0.55577	0.13809	0.26697	0.03918
17	0.17121	0.064626	0.73695	0.027216
18	0.069202	0.14433	0.038279	0.74819
19	0.048998	0.11761	0.025831	0.80756
20	0.062673	0.14341	0.033433	0.76049
21	0.022815	0.071594	0.010387	0.8952
22	0.072898	0.68872	0.025004	0.21338

23	0.051406	0.84659	0.015711	0.086297
24	0.18177	0.57488	0.056916	0.18644
25	0.080597	0.77245	0.025634	0.12132
26	0.079303	0.83752	0.020284	0.06289
27	0.71046	0.17372	0.083717	0.0321
28	0.72079	0.18362	0.064709	0.030879
29	0.16166	0.057684	0.75449	0.026165
30	0.14927	0.056033	0.76864	0.026055
31	0.086392	0.41548	0.034624	0.4635
32	0.10975	0.53997	0.038041	0.31224
33	0.093893	0.6878	0.033233	0.18507
34	0.16981	0.7453	0.033495	0.051397
35	0.70903	0.1464	0.10534	0.039231
36	0.6669	0.12554	0.16979	0.037772
37	0.76446	0.071097	0.14549	0.018953
38	0.078271	0.023991	0.88815	0.009585
39	0.09956	0.036024	0.84896	0.01546
40	0.11731	0.040601	0.82411	0.017974
41	0.10819	0.04119	0.83193	0.018695
42	0.78633	0.061504	0.13532	0.016851
43	0.88744	0.064797	0.035342	0.012417
44	0.8328	0.094182	0.052561	0.020457
45	0.76103	0.15384	0.056542	0.028594
46	0.65183	0.24651	0.062244	0.039424
47	0.49301	0.11586	0.35292	0.03821
48	0.08956	0.27508	0.036261	0.5991
49	0.39275	0.46036	0.081502	0.065383
50	0.29961	0.60854	0.04533	0.046521
51	0.46642	0.29124	0.13796	0.10439
52	0.69192	0.084248	0.19753	0.0263
53	0.14091	0.039878	0.80357	0.015638
54	0.21565	0.04674	0.7211	0.016511
55	0.2342	0.05863	0.68455	0.022619
56	0.073811	0.4274	0.0269	0.47189
57	0.21883	0.67942	0.038852	0.062899
58	0.38368	0.4339	0.10707	0.075349
59	0.87657	0.060138	0.050303	0.012988
60	0.79398	0.086228	0.09876	0.021034
61	0.75554	0.081324	0.14281	0.020331
62	0.548	0.10845	0.31118	0.032367
63	0.11431	0.033065	0.84014	0.012483

AUTHOR PROFILE

Sweta Kumari received her M.Tech degree in Information Security from Ambedkar Institute of Technology, Delhi, and pursuing Ph.D. in Computer Science & Engineering from Birla Institute of Technology, Mesra, Ranchi. In 2010, she joined the Master of Computer Application of Rajiv Academy for Technology & Management as an Assistant Professor. Her current research interest includes computing intelligence techniques, machine learning, data mining and applications of these techniques to information systems and software engineering.

Mohammed Ali received his ME(SE) degree from Birla Institute of Technology, Mesra, Ranchi. His current research interest includes computing intelligence techniques and software engineering.

Shashank Pushkar received his Ph.D. degree from National Institute of Technology, Jamshedpur, India, in the field of IT project management. He is currently working as Assistant Professor in the Department of Computer Science & Engineering, Birla Institute of Technology, Mesra, Ranchi, India. His current topics of interest are Software Cost Estimation Techniques and Application of Optimization Techniques in Software Project Management and Data Mining.