ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2016

A Comparative Study Based on Optimization Techniques for Software Cost Estimation

D.T.R Subba Reddy, Dr. B.R Srinivas, Subhani Shaik

Professor, Dept. of CSE, St. Mary's group of Institutions Guntur, Guntur, AP, India

Professor, Dept. of Mathematics, St. Mary's group of Institutions Guntur, Guntur, AP, India

Research Scholar, Dept. of CSE, Acharya Nagarjuna University, Guntur, AP, India

ABSTRACT: Accurate effort estimation is an essential task for software development life cycle as well as for managing project cost, time and quality. In modern years, many researchers and practitioners proposed optimization and machine learning-based models for software effort estimation. In this work, a comparative study based on optimization techniques for software cost estimation is proposed. Various optimization techniques like Particle Swarm Optimization (PSO), Firefly Algorithm, Bee-Colony Optimization, BAT Algorithm, Human Opinion Dynamics, Harmony Search Algorithm, Genetic algorithm and artificial neural networks have been used. This paper improves the accuracy of software cost estimations by coupling Bayesian multi-class algorithm with existing optimization techniques. The developed model is empirically validated using different evaluation metrics through a statistical framework. It yields better results in terms VAF, MSE, MAE, MMRE, RMSE and R^2.The results of this model are also compared with COCOMO I and COCOMO II Model for optimizing the parameters. It helps project manager to provide nimble and realistic estimate for the project effort and development time that in turn gives software cost. The hypothetical results show that Bayesian model yields better results, high accuracy and has potential to become an effective method.

KEYWORDS: Software cost, Bayesian, COCOMO, PSO, BAT, Bee Colony, Firefly, GA, MMRE.

I. INTRODUCTION

Successful software project improvement not only relies on the product efficiency but also the perfect estimation of its cost. For enhanced resource utilization and project development evaluation project manager require to know the truthful cost estimation. Nowadays, a lot of companies and organization are giving impressive importance to software development and production and the major focus is on customer fulfillment and simultaneously the production cost should be reserved in consideration so that it does not guide to any financial loss or customer disappointment. The global software market has developed exponentially over the past decade. The major idea behind upward trend in the software costs is the employment accelerated nature of the software development process. Cost estimation is a forecast method to get close result of essential cost. It includes the process of considering the essential cost, experiences, time constraints, risks, schedules, assets and other elements related to the expansion of a project. Hence, cost estimation is vital in managing a project mostly to the project manager when proposing budget for secure project. In software development, mostly used term is “software project estimation” where its job is to calculate the estimation procedure.

Perfect assessment means enhanced planning and capable use of project resources such as cost, duration and effort requirements for software projects. In order to develop software effectively in competitive and complex environment several organizations use software metrics as a part of their project development. In the last two decades, many researchers and practitioners GIVEN statistical and machine learning-based models for software effort estimation. Software effort estimation has received a significant amount of attention from researchers and became a challenge for software industry. Constructive Cost Model (COCOMO) is formal effort estimation model developed by Boehm in 1981 is used as an algorithmic model to compute effort. Three basic types are: Basic COCOMO, Intermediate

COCOMO and Detailed COCOMO. Intermediate model is considered by several researchers. The values of parameters of COCOMO are flat for organic, semi-detached and embedded. But these parameters differ from organization to organization .We need to tune the parameters so as to get the optimal results.

The paper is planned in following manner: section 1 describes introduction, sections 2 and 3 presents related work and optimization techniques. Scope of work, results and discussion is described in section4 and 5. Section 6 ends the paper with a conclusion.

II. RELATED WORK

The most frequently used techniques for predicting software development effort have been based on linear-least-squares regression such as COCOMO. The models have been really susceptible to local variations in data points. In addition, the models have unsuccessful to deal with implicit nonlinearities and communications between the features of the project and intension. In recent years, a number of substitute modeling methods have been proposed and they consist of artificial neural networks, regression trees, analogy based reasoning and rule induction models. A Research on software cost estimation is going on software companies are growing most quickly so there is a requirement to find more correct estimation methods. Many Researchers worked in this area and in this section of paper illustrates some past work.

- **G Rao, et.al** proposed a MOPSO algorithm for multi-objective optimization problem. Effort estimation with MOPSO gives superior results compared to COCOMO as it computed on multi-objective crisis so MARE (mean absolute relative error) is minimized and calculation accuracy is maximized. The author has performed two experiments and the results are good for little projects of size less than 50 KDLOC and for testing two in large projects the accuracy is good in some cases only.
- **Bardsiri et.al** used Analogy Based Estimation Approach for the estimation process combined with PSO. The proposed scheme consists of testing and training stages in which an estimation model is developed and evaluated. The results proved that combination of PSO and ABE gives better results and improved the performance of existing models.
- **Dizaji et.al** proposed a bee colony optimization algorithm for effort estimation and results are compared to intermediate COCOMO and the outcome implies that the proposed approach decreases the mean absolute relative error to 0.1619.
- **Sheta et.al** applied genetic algorithm for the estimation of COCOMO model parameters. The developed GA based model is calculated with fitness function VAF (Variance Accounted For) and the modified COCOMO model considered. The developed model examined on NASA software projects and provides fine judgment capability but it can find a more advanced function so that estimated effort will be more accurate.
- **Oliveira et.al** developed a hybrid approach for parameter selection and model optimization. The Genetic Algorithms (GA) is used for optimizing a Support Vector Regression model. The authors described the impact of using GA in attribute selection and parameter optimization of the effort estimation model. The results of their approach demonstrate that GA is applicable to progress the performance of the SVR model compared to other approaches.
- **Shivani Sharma et.al** proposed a model for computing budget of project based on Top down method. The whole process will be completed by Ant colony optimization algorithm. It is compared and evaluated with K Modes algorithm and RF model and it gives better results.
- **Nazeeh Ghatasheh et.al**, Firefly Algorithm is proposed as a meta-heuristic optimization technique for optimizing the specifications of three COCOMO models. These models include the basic COCOMO model and other two models. The developed estimation models are evaluated using diverse evaluation metrics. Experimental outcomes display high accuracy and significant error minimization.

III. SCOPE OF SEARCH

Software Industry has faced many challenges of Software crisis due to time, cost and quality. The software to be developed needs to be accurate with expected quality, within the estimated cost given certain time constraint. However,

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2016

if budget exceeds the estimated budget then project fails horribly. The project cases where the cost exceeds the estimated cost the organization is left crippled with wastage of all the effort and time put into the project without any business. Since cost proves to be a vital parameter for any project being undertaken, utmost importance is required to estimate cost using precise effort. So, there is a need to have a technique that gives more accurate results in terms of effort.

IV. OPTIMIZATION TECHNIQUES FOR COST ESTIMATION

The optimization Algorithms have previously implemented in the area of software cost estimation. Significant work has been done for manipulating effort using these algorithms freshly.

A. *MULTI-OBJECTIVE PARTICLE SWAM OPTIMIZATION:*

Particle Swarm Optimization (MOPSO) was imported by Eberhart and Kennedy in 1995. Swarm Intelligence is an pioneering distributed intelligent paradigm for simplify optimization problem which took its motivation from the biological examples by swarming, flocking and herding occurrence in vertebrates [1].There are two finest values Pbest and Gbest. Each bit is stored according to its strength cost. With each iteration the bit position is updated to first to Pbest as local best then to Gbest as universal best. Single objective optimization problem defined as maximizing or minimizing. We use PSO but in some problems there is a required for optimization of two additional objectives. A multi objective Optimization is defined as $X=[X_1, X_2....X_n]$ where X is the control variable vector, and n is no. of control variables. Objective function is min/max.

$$Z= \{Z_1(X), Z_2(X)....... Z_m(X)\}$$

Every objective combined with weight is given by the formula:

$$Z_1 * Z_1(X) + Z_2 * Z_2(X) +.....+ Z_m * Z_m(X)$$

And normalize the weights using $Z_1+Z_2+.......Z_m=1$

$$F^{r+1} + \equiv F^{r+1}\ a1 * rand_1 * (Fbest - S() + a2 * rand()_2 * (Gbest - S())$$

$$S^{r-1} \equiv S^r + V^{r+1}$$

Whereas, S^r is current search point, S^{r+1} is modified search point.

- F^r is current velocity, F^{r+1} is modified velocity.
- C_j is weighting factors.
- Rand () is consistently distributed random number.

B. *FIREFLY ALGORITHM:*

The meta-heuristic algorithm developed by Dr. Xin-Shi Yang based on flashing characteristics of fireflies. It is a multimodal optimization algorithm, belongs to behaviour of fireflies or lightning bugs. FA (Firefly Algorithm) has three basic rules:

- All Fireflies are attracted to each other with disrespect to gender.
- Attractiveness is associated with light discharge or brightness such that bright flies attract to less bright ones and in their absence the movement become chance.
- Last rule is that brightness is proportional to objective function.

In the simplest form, the light intensity J(s) changes according to the inverse square law $J(s) = \frac{J_r}{r^2}$ where J(r) is the intensity at the source. For a given medium with a fixed light absorption coefficient β, the light intensity J varies with the distances.

i.e. $$J = J_0 e^{-\beta s}$$

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2016

Where J_0 is the original light intensity. As a firefly's attractiveness is proportional to the light intensity seen by adjacent fireflies, we can now define the attractiveness β of a firefly by

$$\gamma = \gamma 0 \ e^{\beta s^2}$$

C. ***BEE-COLONY OPTIMIZATION:***

This Optimization Algorithm was initially introduced by Teodorivic. This Algorithm is based upon the natural occasion of getting food by bees which are performed in two stages as moving rearward and moving forward. In the first stage bees discover many basic solutions and at the second stage they present that solution in a meeting and then they prefer according to food quality .The probability of selecting a solution is computed by:

$$V(j) = (\text{Max}(Z) - Z_j) / (\text{Max}(Z) - \text{Min}(Z)), \ J = 1, 2....., N$$

Whereas N denotes the number of solutions, Z denotes all solutions and Z_j denotes the current solution.

D. ***BAT ALGORITHM:***

In this algorithm search is motivated by social behaviour of bats and phenomenon of echolocation. It is a novel meta-heuristic technique for global numerical optimization problems. BA is used to optimise the weights of the parameters. These optimised weights can then be used for test effort estimation of new projects of a similar kind. In Bat algorithm, the spot of each bat is defined by and velocity, frequency, intensity, and the emission pulse rate in a D-dimensional search space. The two factors loudness and rate of pulse emission, i.e., A, r are also initialised with a constant value of 0.5 each.

The loudness is inversely proportional to the solution and the rate of pulse emission is directly proportional. Generate local solutions Y (t) and velocities V (t) at time step t by

$$F = F\ min\ (F\ max - F\ min)$$
$$V(t) = V(t-1) + (Y(t) - Y^*)$$
$$Y(t) = Y(t-1) + V(t)$$

Here, Y^* is the current global best location (solution) located after comparing all the solutions among all the n bats at each iteration.

E. ***HUMAN OPINION DYNAMICS:***

It is an inspiration to solve complex optimization problems based upon human creative problem solving process. As human beings are considered the most intelligent social animal in the world, the algorithm is based upon opinion formation of human beings. Opinion formation is an Evolutionary process. A real valued Optimizer CODO (Continuous Opinion Dynamics Optimizer) is developed and henceforth it is also called as CODO Algorithm. The Algorithm has four basic essential elements mainly:

- Social Structure
- Social Influence
- Opinion Space
- Updating Rule

F. ***BAYESIAN OPTIMIZATION ALGORITHM:***

The Bayesian optimization structure has two key factors. The primary factor is a probabilistic model, which consists of a prior distribution that confiscation our attitude about the performance of the unfamiliar objective function and a inspection model that describes the data generation method. The next factor is a loss function that describes how best arrangement of queries are; in practice, these loss functions often take the form of regret, either simple or collective. Ideally, the conventional loss is then reduced to select an optimal sequence of queries. After observing the result of every query of the objective, the prior is updated to produce a more informative posterior distribution over the space of objective functions. The proposed algorithm is called the Bayesian optimization algorithm (BOA). The permutation of prior information and the set of positive solutions are used to estimate the distribution. Prior information about the construction of a problem as well as the information performed by the set of optimal solutions can be incorporated into the algorithm. The ratio among the prior information and the information acquired during the run used to generate novel solutions can be controlled. The pseudo-code of the BOA follows:

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2016

Step-1: Set j← 0 randomly generate initial population E (0)
Step-2: Select a set of promising strings G (j) from E (j)
Step-3: Construct the network using a select metric and constraints
Step-4: Generate a set of novel strings K(s) according to the joint distribution encoded by A
Step-5: Create a novel population E(s+1) by replacing some strings from E(s) with K(s) set j← j+ 1
Step-6: If the termination criteria are not met, go to (2).

V. SIMULATION RESULTS

The experiment is applied to BMCA, FA, GA,HOD, HAS, ANN, BAT, BCO and PSO algorithms for optimizing the coefficients of the basic COCOMO model, COCOMO Model I and COCOMO Model II based on the training part of NASA data set. In each run, the optimized models are evaluated based on the testing data using VAF, MSE, MAE, MMRE, RMSE and R2 evaluation metrics. The experiments, results are shown in following Figures respectively for the three variations of COCOMO model. This research considers a famous and public data set in order to produce comparable results; namely NASA projects' effort data set. The data set is challenging due to the small number of instances and limited number of analyzed variables. The data set is split into two parts; training and testing set. NASA data set consists of 18 software projects for which this research considers three main variables that are the project size in thousand Lines of Code (KLOC), Methodology (ME), and Actual Effort (AE). Training data set has 13 instances and the records from 14 till 18 are for testing the model. The statistical results are show in table, table 2, and table 3.

Table 1: Basic COCOMO Model

	Training						Testing					
	BMCA	**BAT**	**BCO**	**FA**	**GA**	**PSO**	**BMCA**	**BAT**	**BCO**	**FA**	**GA**	**PSO**
VAF	94.63%	93.72%	93.56%	93.82%	93.72%	93.73%	98.89%	98.63%	98.15%	98.17%	97.98%	97.97%
MSE	104.66	106.23	105.35	104.89	107.25	107.16	55.23	58.69	59.63	59.15	63.95	63.69
MAE	7.01	7.03	0.74	7.05	7.03	7.03	5.23	5.60	5.63	5.66	6.05	6.03
MMRE	0.23	0.24	0.24	0.25	0.25	0.25	0.9	0.10	0.10	0.11	0.13	0.12
RMSE	10.23	10.24	10.24	10.25	10.37	10.36	7.63	7.36	7.86	7.67	8.00	7.98
R^2	0.9325	0.9652	0.9635	0.9367	0.9352	0.9353	0.9652	0.9865	.0.9667	0.9781	0.9763	0.9765

Table 2: COCOMO Model I

	Training						Testing					
	BMCA	**BAT**	**BCO**	**FA**	**GA**	**PSO**	**BMCA**	**BAT**	**BCO**	**FA**	**GA**	**PSO**
VAF	93.63%	92.45%	94.65%	95.78%	93.94%	96.96%	98.23%	97.56%	97.25%	98.62%	97.97%	98.52%
MSE	104.55	106.23	107.45	56.05	127.70	54.16	54.36	57.89	69.63	47.74	98.17	60.07
MAE	6.01	7.03	0.84	5.42	8.94	5.16	6.32	5.56	5.53	5.56	7.70	5.63
MMRE	0.33	0.24	10.12	0.41	0.53	0.39	0.8	0.11	0.11	0.24	0.29	0.23
RMSE	11.13	11.24	10.67	7.48	10.95	7.36	6.63	8.25	6.76	6.82	9.39	7.72
R^2	0.9645	0.9256	0.9635	0.9662	0.9229	0.9673	0.9652	0.9865	.0.9556	0.9823	0.9637	0.9778

Table 3: COCOMO Model II

	Training						Testing					
	BMCA	**BAT**	**BCO**	**FA**	**GA**	**PSO**	**BMCA**	**BAT**	**BCO**	**FA**	**GA**	**PSO**
VAF	93.53%	93.72%	93.48%	96.95%	92.42%	97.48%	98.53%	98.56%	97.32%	98.43%	98.25%	97.56%
MSE	103.53	105.76	105.35	53.74	129.37	45.28	56.42	58.69	59.63	45.28	45.02	114.79
MAE	7.81	7.91	0.74	5.36	8.20	4.43	6.23	5.60	5.63	4.43	5.57	7.83
MMRE	0.22	0.34	0.24	0.38	0.40	0.30	0.80	0.10	0.10	0.30	0.24	0.27
RMSE	11.23	11.32	10.24	7.26	11.05	6.72	6.89	7.36	7.86	6.72	6.62	9.86
R^2	0.9322	0.9063	0.9635	0.9676	0.9219	0.9727	0.9325	0.9865	.0.9667	0.9727	0.9833	0.9575

The Graphical Representation of evaluating parameters using basic COCOMO model during training and testing phase are shown in figure 1 and figure 2.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2016

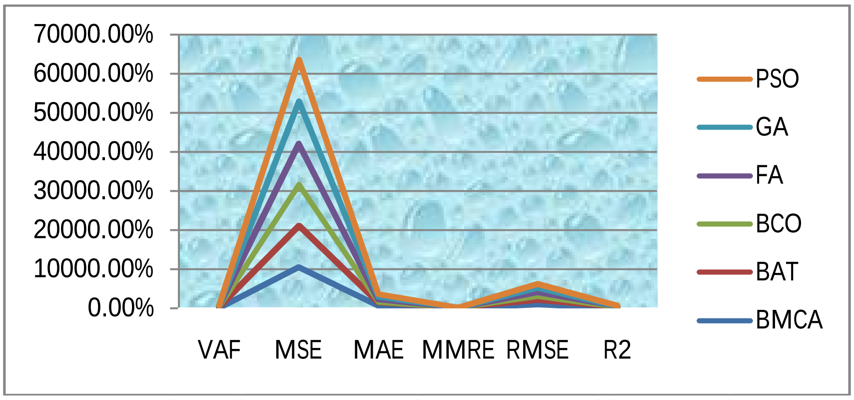

Fig1. Evaluating the Parameters using basic COCOMO model in training phase

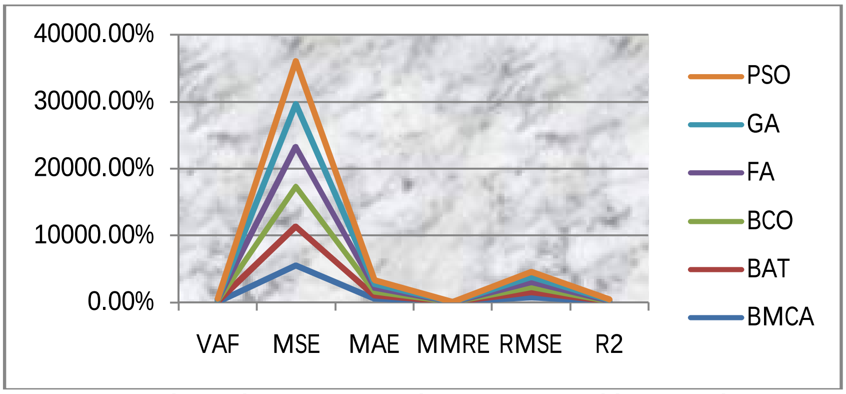

Fig 2. Evaluating the Parameters using basic COCOMO model in testing phase

In order to check the performance of the developed models, the computed measures are the Correlation Coefficient (R^2),

$$R^2 = \frac{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2 - \sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$$

The Mean Squares Error (MSE),

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2$$

The Mean Absolute Error (MAE),

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \bar{y}_i|$$

The Mean Magnitude of Relative Error (MMRE),

$$MMRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \bar{y}_i|}{y_i}$$

And the Variance-Accounted-For (VAF)

$$VAF = \left[1 - \frac{\mathrm{var}(y(t) - \hat{y}(t))}{\mathrm{var}(y(t))}\right] \times 100\%$$

These performance criteria are used to measure how close the predicted effort to the actual values, where y is the actual value, y^ is the estimated target value, and n is the number of instances. The following figure 3 and figure 4 shows comparative study.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2016

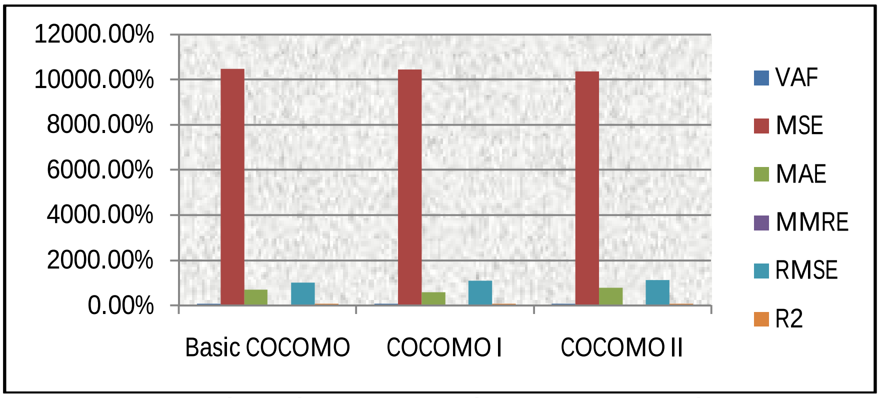

Fig3. Evaluating the Parameters using basic BMCA in training phase

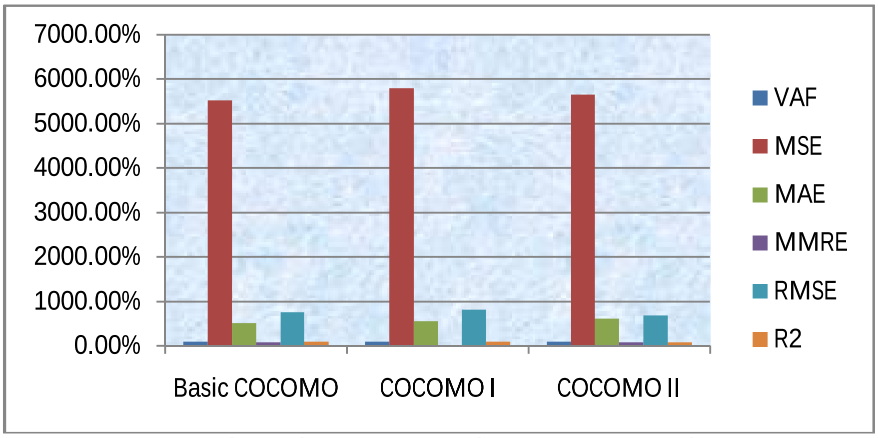

Fig4. Evaluating the Parameters using basic BMCA in testing phase

VI. CONCLUSION

Accurate effort estimation is an essential task for software development life cycle as well as for managing project cost, time and quality. In modern years, many researchers and practitioners proposed optimization and machine learning-based models for software effort estimation. In this work, a comparative study based on optimization techniques for software cost estimation is proposed. This work investigated the efficiency of applying the Bayesian multi optimization Algorithm to optimize the parameters of different effort estimation models. Various optimization techniques like Particle Swarm Optimization (PSO), Firefly Algorithm, Bee-Colony Optimization, BAT Algorithm, Human Opinion Dynamics, Harmony Search Algorithm, Genetic algorithm and artificial neural networks have been used. Although these techniques find their applications in the areas of social sciences and global numerical optimization. This paper improves the accuracy of software cost estimations by coupling Bayesian multi-class algorithm with existing optimization techniques. The developed model is empirically validated using different evaluation metrics through a statistical framework. It yields better results in terms VAF, MSE, MAE, MMRE, RMSE and R2.The results of this model are also compared with COCOMO I and COCOMO II Model for optimizing the parameters. It helps project manager to provide nimble and realistic estimate for the project effort and development time that in turn gives software cost. The hypothetical results show that Bayesian model yields better results, high accuracy and has potential to become an effective method.

ACKNOWLEDGEMENT

Authors are thankful to Dr.D.Nagaraju, Dr.B.V.S.TSai, Professors, and Shaik Subhani, for giving continues support and encouragement to carry out this work. Authors are also thankful to the reviewer for critically going through the manuscript and giving valuable suggestions for the improvement of manuscript.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2016

REFERENCES

1. A New Hybrid For Software Cost Estimation Using Particle Swarm Optimization And Differential Evolution Algorithms, Majid Ahadi , Ahmad Jafarian, Informatics Engineering, An International Journal (IEIJ), Vol.4, No.1, March 2016.
2. Software Cost Estimation using Hybrid Algorithm Shivani Sharma , Aman Kaushik , AbhishekTomarInternational Journal of Engineering Trends and Technology (IJETT) – Volume 37 Number 2- July 2016.
3. Novel Meta-Heuristic Algorithmic Approach for Software Cost Estimation, RuchiPuri, Iqbaldeepkaur, International Journal of Innovations in Engineering and Technology (IJIET), Volume 5 Issue 2 April 2015, ISSN: 2319 – 1058.
4. Optimizing Software Effort Estimation Models Using Firefly Algorithm, Nazeeh Ghatasheh, Hossam Faris, Ibrahim Aljarah, Rizik M. H. Al-Sayyed, Journal of Software Engineering and Applications, 2015, 8, 133-142 Published Online March 2015 in SciRes.
5. Software Development Effort Estimation Using Fuzzy Logic: A Review Shifali, Naveen Bilandi, International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 2841-2845.
6. Automation of Software Cost Estimation using Neural Network Technique, Gaurav Kumar, Pradeep Kumar Bhatia, International Journal of Computer Applications (0975 – 8887) Volume 98 – No.20, July 2014.
7. A New Approach for Software Cost Estimation with Hybrid Genetic Algorithm and Ant Colony Optimization Isa Maleki , Ali Gaffer , and Mohammad Masdari, International Journal of Innovation and Applied Studies ISSN 2028-9324 Vol. 5 No. 1 Jan. 2014, pp. 72-81 © 2014.
8. Maleki, I., Ghaffari, A. and Masdari, M. (2014) A New Approach for Software Cost Estimation with Hybrid Genetic Algorithm and Ant Colony Optimization. International Journal of Innovation and Applied Studies, 5, 72-81
9. Artificial Neural Networks for Software Effort Estimation: A Review Dharmesh Santani, Mahesh Bundele, Poonam Rijwani, International Journal of Advances in Engineering Science and Technology, ISSN: 2319-1120.
10. Yang, X.-S. (2010) A New Meta-heuristic Bat-Inspired Algorithm. In: Gonzlez, J., Pelta, D., Cruz, C., Terrazas, G. and Krasnogor, N., Eds., Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Vol. 284 of Studies in Computational Intelligence, 65-74. Springer, Berlin and Heidelberg.
11. Wang, F.S. and Chen, L.-H. (2013) Heuristic Optimization. In: Dubitzky, W., Wolken hauer, O., Cho, and K.-H. And Yokota, H., Eds., Encyclopedia of Systems Biology, Springer, New York, 885-885.
12. Uysal, M. (2010) Estimation of the Effort Component of the Software Projects Using Heuristic Algorithms. INTECH Open Access Publisher, Croatia.

BIOGRAPHY

D.T.R Subba Reddy currently working as Professor, St.Marys Group of Institution, Guntur, Andhra Pradesh, India. His research areas include Data Mining, Pattern Recognition, Machine Learning, Soft Computing Techniques, Computer Network and Design and analysis of algorithm. He published many papers in National and International conferences and journals.

Dr. B.R Srinivas is currently working as professor in dept.of Mathematics, St.Marys Group of Institution, Guntur, Andhra Pradesh, India. His research areas include Data Mining, Pattern Recognition, Machine Learning, Soft Computing Techniques, Graph Theory and Partial Differential Equations. He published many papers in National and International conferences and journals.

Subhani Shaik is a Research Scholar in the Dept.of CSE, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. He received Bachelor of Technology (B.Tech) degree from Andhra University, in 2002. M.Tech from Jawaharlal Nehru Technological University Hyderabad (JNTUH) in 2011. His research interests are Computer Networks (wireless Networks), Image Processing, Cloud Computing, Data Mining. He published many papers in National and International conferences and journals.