Software Quality Assesment using COCOMO-II Metrics with ABC and NN

Naveen Malik, Sandip Kumar Goyal, Vinisha Malik

Abstract: Time, cost and quality predictions are the key aspects of any software development system. Loses that result due to wrong estimations may lead to irresistible damage. It is observed that a badly estimated project always results into a bad quality output as the efforts are put in the wrong direction. In the present study, author proposed ABC-COCOMO-II as a new model and tried to enhance the extent of accuracy in effort quality assessment through effort estimation. In the proposed model author combined the strengths of COCOMO-II (Constructive Cost Model) with the Artificial Bee Colony (ABC) and Neural Network (NN). In the present work, ABC algorithm is used to select the best solution, NN is used for the classification purpose to improve the quality estimation using COCOMO-II. The results are compared and evaluated with the pre-existing effort estimation models. The simulation results had shown that the proposed combination outperformed in terms of quality estimation with small variation of 5-10% in comparison to the actual effort, which further leads to betterment of the quality. More than 90% projects results into high quality output for the proposed algorithmic architecture.

Keywords : Effort estimation, Artificial Bee Colony (ABC), Neural Network (NN), Constructive Cost Model (COCOMO-II)

I. INTRODUCTION

The effort estimation is the most significant parameter in any software development process. In today's competing technical world, this estimation offers highly robust results with high dimensions of reliability and accuracy before starting up of a project [1, 2]. A software project gradually advances and an ideal and exact prediction is not possible in real sense [3, 4]. It is also important to understand the issues concerning such predictions that may end up in both underestimation as well as overestimation of the efforts. In such cases, the resultant estimation predications are erroneous causing a need to develop more precise software that can offer accurate software estimations. The effort estimation is multifaceted job and these estimation models that deal with such works tasks are classified as algorithm based COCOMO model, non-algorithm based trained model and those that employ the strengths of machine learning architecture [5-6]. With the growing need numerous effort estimations models developed from time to time but none of them have achieved perfection yet. The most popular models among algorithmic ones are Boehm's COCOMO [7], Park's PRICE-S model [8], Albrecht's Function Point [9] and Putnam and Myers SLIM [10].

The models have some limitations adjoining the achievement of perfect estimation that it requires the input parameters in terms of complexities, Line Of Code (LOC), etc. that couldn't be precisely achieved at the early stages of development process. This leads to inability of the models to offer solutions to withstand complex relationships, categorized data along with profound deficiency in interpretation capability [11]. In this study ABC (artificial bee colony algorithm) algorithm is employed along with COCOMO to offer more robust effort estimation in terms of both accuracy and sensitivity. ABC-COCOMO model is believed to deal with the shortcoming of the various existing COCOMO combinations.

In addition to the introduction section rest of the paper is divided in following sections:

Section II: Details the Quality Assessment and role of effort estimation in quality assessment

Section III: Discusses the COCOMO framework

Section IV: Describes the proposed model work methodology and algorithms used.

Section V: Evaluation parameters and formula are mentioned.

Section VI: Summarises the results and discussion.

Section VII: Concludes the paper.

Section VIII: Cites the reference work

II. QUALITY ASSESSMENT AND ROLE OF EFFORT ESTIMATION

The quality of a software depends upon multiple aspects including the implemented effort in the direction of the development of the software. If the skill oriented people are not deployed to the correct product, the effort will be high and the productivity will be least which finally results into bad quality product as shown in Fig.1.

Assessment of the quality has always been an area of interest for the researchers and this research work aims to classify the quality based on the effort estimation.

Revised Manuscript Received on February 20, 2020.

* Correspondence Author

Naveen Malik*, PhD Research Scholar, Department of Computer Science & Engineering, Maharishi Markandeshwar (Deemed To Be University) Mullana, Ambala , India. E-mail: Naveenmalik317@gmail.com

Sandip Kumar Goyal, Professor, Department of Computer Science & Engineering, Maharishi Markandeshwar (Deemed To Be University) Mullana, Ambala, India. E-mail: skgmmec@gmail.com

Vinisha Malik, PhD Research Scholar, Department of Computer Science & Engineering, Maharishi Markandeshwar (Deemed To Be University) Mullana, Ambala , India. E-mail: Vinishamalik317@gmail.com

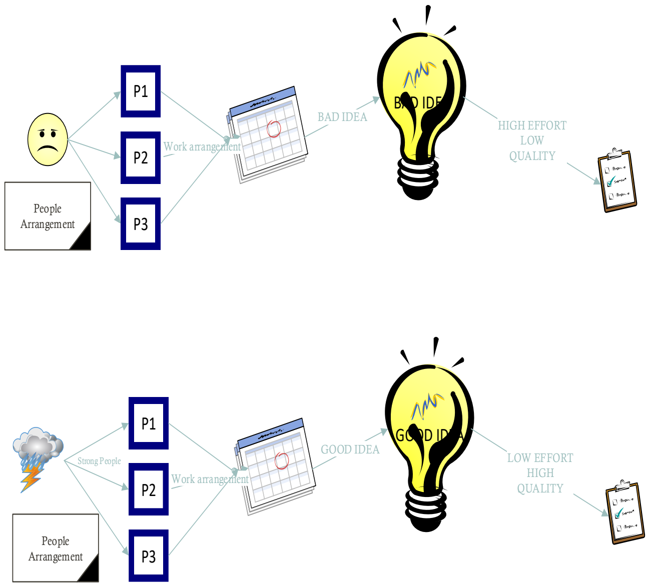

Fig. 1. Quality in accordance to Effort Estimation

III. COCOMO STRUCTURE

COCOMO was developed by Boehm and offers algorithm based cost estimation. The estimation functionality of this model depends on the regression formula. The input parametric values of this model are retrieved from the feature of the active project and the historical data of the project. The model works in three modes: First one is the Organic mode that includes the uncomplicated projects involving teams that wok in a well informative and steady situations. Second one is the Semi-detached mode that involves teams that exhibit varied knowledge. Third one is the embedded mode that is an advanced mode where multifaceted projects are worked on under strict checks to match the varying needs. The basic effort estimation is performed as follows:

$$E_{months} = p_1 * (KLOC)\, p_2 \qquad (1)$$

A. COCOMO-II Model

This model version corresponds to 81's update that deals with the practices related to software development strategies of 1990 and 2000 [7]. The estimation model works on the basis of regression analysis and is one of the most satisfactory models as compared to the available prediction models. The model exhibits the following architecture [12-13]:

- **Application Composition Model** works on the assumption that recyclable parameters like record encodings and scripts design the basic architecture of any system. It also considers the sample efforts to deal with the issues concerning performance, UI (user interface), system and software relationship. Effort estimation is performed in the early stages and size estimations depends on the application or the object points like UI, project reports and screens, etc.

- **Early Design Model** deals with the prediction of the duration and involved cost of the project sooner than the whole design could be concluded. In order to compute the size, the model takes the advantage of the Unadjusted Function Points along with novel prediction equations and expense drivers.

- **Post Architecture Model** deals with the authentic design and maintenance of the software to achieve remarkably accurate prediction of size of the software product. It has been observed that the model is very cost effective when system task, risk and perceptions are considered while designing software lifecycle framework. The size prediction is done on the basis of LOC or the function points. In COCOMO-II 17 cost-drivers have been summarized that are used in Post Architecture Model in a very low to very high scale. At this point effort estimation is done as follow:

$$E_{est} = C * (S_{size})^F * \prod_{i=1}^{17} E_m \qquad (2)$$

Where, E_{est} is effort estimation in months, S_{size} is the size of the project in KLOC, F is the scale factor, E_m is effort for 17 effort multiplier. The C and F are the constant value that depends on the values of the dataset considered for the estimation.

Table-I. Rating Scheme of important Post Architecture Cost Drivers.

Rating Levels	Effort Multipliers						
	REL	**PCAP**	**LTEX**	**RUSE**	**PREC**	**FLEX**	**PVOL**
Very Low	0.82	1.34	1.2	NA	6.2	5.07	NA
Low	0.92	1.15	1.09	0.95	4.96	4.05	0.87
Nominal	1	1	1	1	3.72	3.04	1.00
High	1.1	0.88	0.91	1.07	2.48	2.03	1.15
Very High	1.26	0.76	0.84	1.15	1.24	1.01	1.30
Extra High	NA	NA	NA	1.24	0	0	NA

Where, RELY: Software Reliability; PCAP: Programmer Capability; RUSE: Reusability; PREC: Precedentedness; FLEX: Flexibility scale factor; PVOL: Platform volatility. Table 1 lists the rating scheme for the product factors in post architectural design [14]. The product development is a very complex task. The variation in effort estimation for software development is due to the product factors some of which are mentioned in above table.

IV. PROPOSED MODEL FRAMEWORK

A. Dataset

In the proposed model authors used the Data set consisting of 93 projects in NASA data sets.

B. Proposed Model

In the proposed combination the parametric values obtained from ABC are fed to COCOMO-II after classification carried on using NN.

The proposed workflow is shown in Figure 2. Karaboga had introduced the ABC that is inspired by the three kinds of bee colonies [15]. The responsibilities of Employed Bees (EB) is to search for the food sources, Onlooker Bees (OB) further selects and checks the searched food sources as per the fitness formula given below. The Scouts Bees (SB) further assigns the new food sources [16].

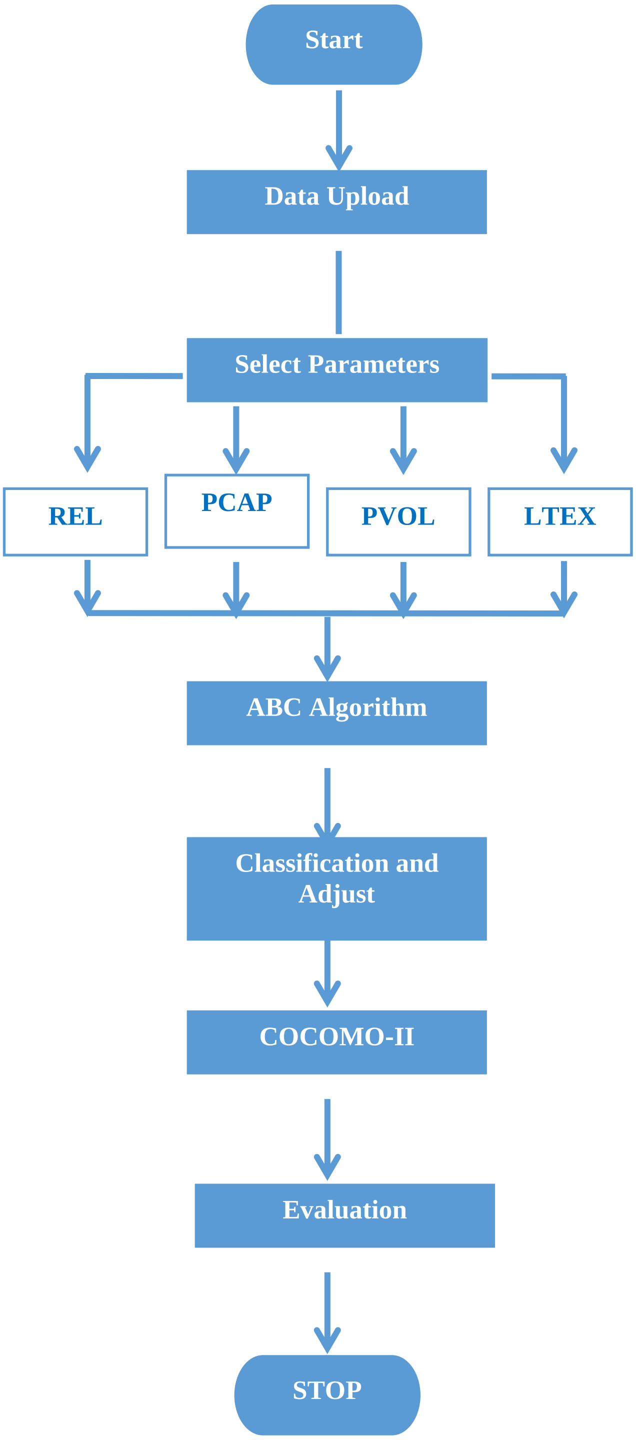

Fig. 2. Proposed model workflow.

Table-II. Cost Drivers employed in the proposed work [14].

Sr. No.	Cost Drivers	Range
1	Reliability required (REL)	0.82-1.26
2	Programmers capability (PCAP)	1.34-0.76
3	Required reusability (RUSE)	0.87 – 1.30
4	Language & Tool experience (LTEX)	1.2-0.84

Table 2. above lists the prescribed range of 4 cost drivers used in proposed model out the 17 cost drivers used in Post Architecture Model. These cost drivers corresponds to reliability, capability and reusability of the architecture model.

Proposed Algorithm Steps

- Input dataset parameters for effort estimation.
- Read input data parameters (REL, PCAP, REUSE, and LTEX).
- Call ABC for each class.
- Initialize EB and COCOMO-II parameters.
- Find new solution for EB

$$EB_{a,b} = X_{a,b} + R * (X_{a,b} - X_{e,b}) \quad (3)$$

Where, R is random variable between -1 and +1, D are the dimensions of search.

$$a \in \{1, 2, 3, ... EB\}, \quad (4)$$

$$b \in \{1, 2, 3, ... D\}, \quad (5)$$

- Assign Hive 1; if EB > average
  Hive 2; otherwise
- Calculate fitness function

f_{fit} = 1; if EB (travel time) > OB (wait time) (6)

f_{fit} = 0; otherwise (7)

- Save the best solution and repeat the steps.
- Save the parameters from the best bee.
- Divide data for training and classification
- Classify using NN.
- Input resultant parametric values to COCOMO-II.
- Calculate effort estimation using following equation.

$$E_{est} = C * (s_{size})^F * \prod_{i=1}^{15} E_m \quad (8)$$

Output parametric values of COCOMO-II.

In the proposed algorithm, parameters are read from the dataset for effort estimation. These parameters are fed to the EB of ABC algorithm. These EB searches for the best solution in its neighbourhood and the best solution is further shared with the OB. OB checks and calculates the fitness score as per the fitness function mentioned above. Finally, SB further searches for the new solution sources and the steps are repeated till an optimal solution is not reached.

Retrieval Number: C5633029320/2020©BEIESP
DOI: 10.35940/ijeat.C5633.029320
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The solution obtained from the algorithm undergoes classification using NN to enhance the prediction accuracy of effort estimation done by COCOMO-II.

V. EVALUATION

There exist numerous criteria for the evaluation of the software effort estimation [17]. In present study evaluation of the experimental evaluation is done by comparing the accuracy of the actual effort to the estimated effort in terms of following:

- **Variance Accounted For (VAF)**

$$VAF = \left[1 - \frac{var\,(E_{measured} - E_{estimated})}{var\,(E_{mesured})}\right] * 100 \quad (9)$$

- **Mean Absolute Relative Error (MARE)**

$$MARE = mean\left[\frac{abs\,(E_{measured} - E_{estimated})}{var\,(E_{mesured})}\right] * 100 \quad (10)$$

- **Variance Absolute Relative Error (VARE)**

$$VARE = var\left[\frac{abs\,(E_{measured} - E_{estimated})}{var\,(E_{mesured})}\right] * 100 \quad (11)$$

- **Balance Relative Error (BRE)**

$$BRE = \frac{|E_{estimated} - E_{actual}|}{\min(E_{estimated}, E_{actual})} \quad (12)$$

- **Absolute Relative Error (ARE)**

$$ARE = \frac{|E_{estimated} - E_{actual}|}{E} \quad (13)$$

- **Prediction at level 'n' (P_n)**

$$P_n = [projects_{\,(ARE<n)}] * 100 \quad (14)$$

VI. RESULTS

In this section comparison of the effort estimation obtained using proposed model is compared with the actual effort and the effort obtained from COCOMO-II and Fuzzy Logic [18].

Table-III. Effort estimation for actual, ABC with NN, COCOMO-II and Fuzzy Logic for first 18 project ID's.

Project ID	Post Architecture Parameters				Effort Estimation			
	RELY	PCAP	PVOL	LTEX	ACTUAL	ABC with NN	COCOMO-II	FUZZY LOGIC
1	1.15	1	0.87	0.91	117.6	111.72	78.204	99.9894
2	1.15	1	0.87	0.91	117.6	111.72	78.204	99.9894
3	1.15	1	0.87	0.91	31.2	29.64	20.748	26.5278
4	1.15	1	0.87	0.91	36	34.2	23.94	30.609
5	1.15	1	1	0.95	25.2	23.94	16.758	21.4263
6	1.15	1	0.87	0.91	8.4	7.98	5.586	7.1421
7	1.15	1	0.87	0.91	10.8	10.26	7.182	9.1827
8	1.15	1	0.87	0.91	352.8	335.16	234.612	299.9682
9	1.4	0.7	1	0.95	72	68.4	47.88	61.218
10	1	0.74	0.87	0.91	72	68.4	47.88	61.218
11	1	0.87	0.87	0.91	24	22.8	15.96	20.406
12	1	0.74	0.87	0.91	360	342	444.6	275.31
13	1	1	0.87	1.1	36	37.8	49.14	30.429
14	1	0.87	1.15	1.22	215	225.75	293.475	181.7288
15	1	0.87	0.87	0.91	48	50.4	65.52	40.572

Table-IV. Quality Assessment using Estimated Effort

Inputs	Actual Effort(AE) , Estimated Effort (EE) Input Value = $\frac{EE-AE}{AE} * 100$
Member Ship Function to Input	Radical High , Radical Medium , Radical Low
a) Radical Low b) Radical Medium c) Radical High	a) >=35 % b) >15% and <35% c) <15%
Output	Estimated Quality
Output Membership Function	Low, Medium , High
RULE SET	
INPUT RULE	OUTPUT
If the Input is (a)	Output is (c)
If the Input is (b)	Output is (b)
If the Input is (c)	Output is (a)

Retrieval Number: C5633029320/2020©BEIESP
DOI: 10.35940/ijeat.C5633.029320
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table 4. takes both the input as Actual Effort and the estimated effort applied with three membership function for both the input and the output. If the input value is radically low, the evaluated quality is high. If the input value is radically is radically medium then the output is radically medium else the output is low. The data is undertaken from Promice set which is a free available data architecture at https://www.promice.org/PromiceDataPortal/.

The evaluation was done over 92 projects out of which 15 values are listed here.

The evaluation of the input for the quality assessment is listed in Table 5

Table-V. Evaluated Value for Quality Assessment

Project ID	Evaluated Input Value in % ABC With NN	Evaluated Input Value in % COCOMO II	Evaluated Input Value in % Fuzzy
1	4.149659864	32.64965986	11.57363946
2	4.149659864	31.79931973	11.57363946
3	-5.384615385	30.29487179	-1.050641026
4	-8.666666667	30.72222222	3.863888889
5	-6.904761905	29.53174603	-0.898015873
6	-6.904761905	21.5952381	3.070238095
7	-32.03703704	14.98148148	5.715740741
8	4.716553288	32.93310658	14.12465986
9	3.611111111	30.72222222	13.58611111
10	-0.555555556	30.72222222	8.030555556
11	-3.333333333	25.16666667	10.80833333
12	4.722222222	-24.05555556	22.13611111
13	-16.11111111	-39.27777778	12.69722222
14	-5.930232558	-36.96511628	15.00986047
15	-11.25	-40.66666667	11.30833333

It is observed that in most of the cases, the proposed architecture results about 90% low input values which further results into 90% high quality output. Fuzzy logic architecture results into input value which is less than 35%. The result trend for Fuzzy logic also is the same but it is comparatively less in comparison to the proposed work. The most lag is observed with COCOMO II where for most of the projects, the evaluated quality is on the medium or low side.

Table 6: Evaluate Quality through Models

Evaluated Quality Through ABC and Neural	Evaluated Quality Through COCOMO II	Evaluated Quality Through Fuzzy
High	Low	High
High	Low	High
High	Low	High
High	Low	High
High	Medium	High
Low	High	Medium
High	Low	High
High	Low	High
High	Low	High
High	Low	High
High	Medium	Medium
Medium	Low	High
High	Low	Medium
High	Low	High

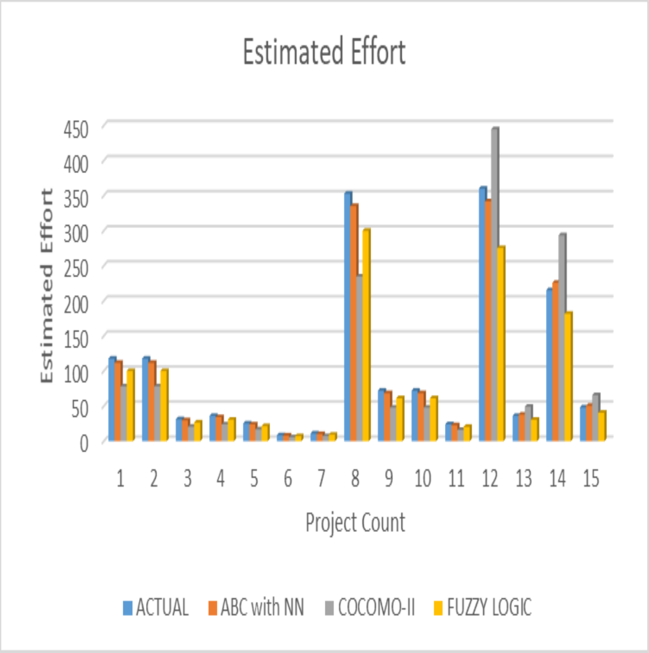

Fig. 2. Comparison of effort estimation for first 15 project ID's.

Actual effort estimation values are compared with the effort values obtained using proposed ABC with NN architecture, COCOMO-II and Fuzzy Logics for first 15 project ID's.. In both graphs it is observed that the effort estimation for the proposed combination of ABC with NN shows only 5-10% variation which will result into high quality, while COCOMO-II showed an average variation of 25-30% which will result into low quality whereas Fuzzy Logics showed a variation of 15% in comparison to the actual effort which will result into average quality.

VII. CONCLUSION

Over the time, efforts and quality estimation are being done by various authors to overcome the degradation of the developed software. It is really difficult to obtain a satisfactory prediction well before the starting of the project as on the way project gets evolved and influence by numerous factors on the go. In the present work, authors tried to reach a near ideal effort estimation with the combination of ABC algorithm to search for the produced quality at the end. The solution obtained from ABC is then fed to NN for classification and training to enhance the quality of results. Finally, COCOMO-II is employed to predict the effort estimation parameters. The effort estimation for the proposed model shows only 5-10% variation from the referee work and hence proved that the proposed work is able to reach the ideal effort estimations and high quality. The tabular representation demonstrates that more than 95% project's quality is evaluated to be high using ABC and Neural Networks. The story remains of similar kind with Fuzzy logic but the input value is comparatively low as compared to the proposed model. The least quality producing model is observed to be COCOMO II among the tested three architecture with almost 65% projects with low quality.

REFERENCES

1. Gharehchopogh, F. S., Talebi, A., & Maleki, I. (2014). Analysis of use case points models for software cost estimation. International journal of academic Research, 6(3).
2. Naveen Malik, Sandip Kumar Goyal, Vinisha Malik, "A New Pattern for Software Engineering as a Service " in international journal –" IJSDR ",Volume 4 , Issue 6, pages 408-412 , ISSN:2455-2631.
3. Maleki, I., Ebrahimi, L., & Gharehchopogh, F. S. (2014). A hybrid approach of firefly and genetic algorithms in software cost estimation. MAGNT Research Report, 2(6), 372-388.
4. Jorgensen, M., & Shepperd, M. (2006). A systematic review of software development cost estimation studies. IEEE Transactions on software engineering, 33(1), 33-53.
5. Srichandan, S. (2012). A new approach of software effort estimation using radial basis function neural networks. International Journal on Advanced Computer Theory and Engineering (IJACTE), 1(1), 113-120.
6. Barry, B. (1981). Software engineering economics. New York, 197.
7. Park, R. E. (1988). Parametric Software Cost Estimation with an Adaptable Model. AACE International Transactions, G-11.
8. Attarzadeh, I., & Ow, S. H. (2009). Software development effort estimation based on a new fuzzy logic model. International Journal of Computer Theory and Engineering, 1(4), 473.
9. Vinisha Malik, Sandip kumar goyal & Naveen Malik (2019 , October) A Hybrid Model For android Malware Detection, scopus journal International journal of Innovative Technology and Exploring Engineering. IJITEE (Vol. 8, Issue 12, pp 2656-2662).
10. Saliu, M. O., Ahmed, M., & AlGhamdi, J. (2004, June). Towards adaptive soft computing based software effort prediction. In IEEE Annual Meeting of the Fuzzy Information, 2004. Processing NAFIPS'04. (Vol. 1, pp. 16-21). IEEE.
11. Boehm, B., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, E, Steece, B. (2000). Cost estimation with COCOMO II. ed: Upper Saddle River, NJ: Prentice-Hall.
12. Boehm B.W, B. Clark, E. Horwitz, R. Madachy, C. Abts, S.Chulani, A.W.Brown and B. Steece,"COCOMO II model definition manual, University of South California Center for Software Engineering, 2000.
13. Karaboga, D., & Basturk, B. (2007, June). Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In International fuzzy systems association world congress (pp. 789-798). Springer, Berlin, Heidelberg.
14. Gao, W., & Liu, S. (2011). Improved artificial bee colony algorithm for global optimization. Information Processing Letters, 111(17), 871-882.
15. Naveen Malik, Ontology based test case reused- A Semantic Web Technique National Conference RAPS- 2014 ,PEC University Of Technology , Chandigarh.
16. Naveen Malik, Sandip Kumar Goyal, Vinisha Malik, "Quality Assessment Of Semantic Web Based Applications and Saas" presented in ISPCC- 2017 4th IEEE international conference JUIT solan and published in IEEE Xplore.
17. Yadav, R. K., & Niranjan, S. (2017). Optimized model for software effort estimation using cocomo-2 metrics with fuzzy logic. International Journal of Advanced Research in Computer Science, 8(7).

AUTHORS PROFILE

Naveen Malik, is a PhD Research Scholar in Department of Computer Science & engineering , M.M Engineering College, Maharishi Markandeshwar (Deemed to be University) Mullana, Ambala, Haryana, India. Mr Malik Received M.Tech Degree from Kurukshetra University, Kurukshetra. Mr. Malik is in teaching and Research & Development from 2014. He has published about 14 Research papers in International Journals, Scopus Journal and Refereed International Conferences. His Current area of research is Semantic Web, Ontology & Software Engineering.

Dr. Sandip Kumar Goyal is working as Professor & Head, Computer Science & engineering Department , M.M Engineering college, Maharishi Markandeshwar (Deemed To Be University) Mullana, Ambala, Hrayana, India.Dr. Goyal received PhD from M.M University Mullana and M.Tech from Kurukshetra University, Kurukshetra. Dr. Goyal is in teaching and Research & Development from 2002. He has published about 46 Research papers in International, national Journals, scopus Journal and Refereed International Conferences. His Current area of research is Load Balancing in Distributed system and simulation & Modeling.

Vinisha Malik is a PhD Research Scholar in Department of Computer Science & Engineering, M.M Engineering College, Maharishi Markandeshwar (Deemed to be University) Mullana, Ambala, Haryana, India. Vinisha received M.Tech Degree in 2013 with Goldmedal from DCRUST Govt. University Murthal, Haryana. Vinisha is in Teaching and Research Development since 2013. She has published about 5 research papers in International Journals and refereed Conferences. One paper is also Published in Scopus journal IJITEE. She has also Industry work experience of two years in TCS Company. Her current research interests are in Mobile Malwares In Smartphones.