ANN-Cuckoo Optimization Technique to Predict Software Cost Estimation

Vishnu Sai Desai
Dept. of Computer Science Engineering
Keshav Memorial Institute of Technology
Hyderabad, India
vishnudesai196@gmail.com

Ramakanta Mohanty
Dept. of Computer Science Engineering
Keshav Memorial Institute of Technology
Hyderabad, India
ramakanta5a@gmail.com

Abstract—Software cost is of the most complex and vital aspect in consideration when software is in its development stages. To determine the amount of time, effort and resources required to complete the project successfully translate to Software Cost Estimation (SCE). Thus far, many models have been suggested such as Fuzzy Logic, Neural Networks, Support Vector Machines, Ant Colony Optimization, Genetic Algorithms, Decision Trees, Case-Based Reasoning and Soft Computing Techniques. Such computational models have contributed to a large extent in this arena. Yet, there still lies immense scope to apply optimization methods. Neural Networks are the most utilized techniques in software cost estimation by researchers. In this paper, we propose the use of a new model, i.e. Artificial Neural Networks (ANN) trained using Cuckoo Optimization Algorithm (COA) to predict Software Cost Estimation. The key goal is to exhibit use of a novel learning procedure for ANN to better predict SCE. The proposed model is verified with the ISBSG dataset and results are compared with existing models. The results shown are in terms of Root Mean Squared Error (RMSE) and Mean Magnitude of Relative Error (MMRE).

Keywords—Software Cost Estimation, Neural Networks, Cuckoo Optimization Algorithm, RMSE, MMRE

I. INTRODUCTION

In recent years, with the rise in computing capabilities, its value has become subordinate to corporate as it keeps increasing exponentially on a yearly basis. Efficient optimization of personnel costs being primarily for software development, proper planning for SCE is a key aspect of business houses. Accurate estimation models provide the means to gauge to a fair certainty the expected Software costs. Major influences of software costs are due to human effort and thus most estimation models give estimates in terms of person-months [1]. Underestimating these costs result in poor resource management that eventually demand flexible budgets & associated poor quality performance and failure to meet deadlines. Overestimating the costs, on the other hand, leads to excess resource commitment rendering a noncompetitive evaluation for contracts, which may fail in the successful bid. An accurate estimation model reduces costs that would be incurred from inaccurate estimation and help organizations decide ways to allocate resources based on valuable predictions of the unknown future.

Software development companies have used empirical models such as Boehm's COCOMO model [2] for developing, evaluating and completing the software which uses previous projects to evaluate. As well as, Analytical Models such as [3, 4, and 5] where formulae based on an approach are used to estimate the software cost are popular. Most cost models are based on the size measure, such as Lines of Code (LOC) and Function Points (FP) derived from size estimation which directly impacts the accuracy of cost estimation. Now meta-heuristic algorithms are being used widely in hybrid optimization problems [6] due to the fact that they are efficient in solving hard and complex problems. They contribute to optimization and efficiency of optimized methods by researching the problem space for near correct solutions. So in this paper, we have used Artificial Neural Networks with Cuckoo Optimization Algorithm to present as a new model for prediction of Software Cost Estimation.

We have organized the paper as follows: In section 2, Literature survey is carried out. In section 3, Data description is given. In section 4, Methodology is described. In section 5, Results and Discussions are explored, in section 6 we discuss the threats to the validity of this research and finally section 7 finishes up the paper with the conclusion.

II. LITERATURE SURVEY

In Software Engineering, SCE is a vital aspect to be considered. Thus, a study of literature is significant. Many authors have given a range of methods from time to time in the field of Software Cost Estimation, detailed by reviews published by Mohanty et al. [1] Vekataiah et al. [7] proposed Particle Swarm Optimization using K-means to cluster the data as input to predict SCE. Idri and Azeddine [8] proposed the use of Fuzzy analogy and Classical analogy methods on the ISBSG and COCOMO datasets. Fuzzy analogy performed better than Classical analogy. They estimated accuracy and tolerance of uncertainties of cost drivers, portraying the usefulness of Fuzzy analogy in cost estimation. Manikavelan and Ponnusamy [9] used Enhancement and Expert Judgment using differential evaluation for SCE.

Patil et al. [10] proposed a hybrid model where a Feed Forward Neural Network trained with delta rule is based on Principal Component Analysis (PCA). A. Kaushik et. al. [11] Proposed modeling of COCOMO using Feed Forward Backpropagation Neural Network, where they used COCOMO and NASA2 datasets to train and test the network, which takes an identity function at input layer and signed at the hidden and output layers. Singh and Johri trained Neural Network with Bayesian Regularization producing reduced condition and found it to be more consistent than the Fuzzy model having membership functions. Ch. S. Reddy et al. [13] found that using Radial

978-1-5386-8215-9/18/$31.00 ©2018 IEEE

Basis Function Neural Network (RBFN) for functional approximation with K-means clustering gave better results in cost estimation. Venkatachalam. [14] used ANN approach for SCE and compared results with the COCOMO model.

Attarzadeh et al. [15] used adaptive ANN in COCOMO-II to handle imprecise attributes. Bardsiri et al. [16] used C-Means clustering by using Fuzzy logic and compared it with ANN, ABE, CART, SWR, and MLR. It was evaluated using the metrics MMRE and PRED (0.25). Li et al. [17] used a nonlinear adjustment to ABE with the ability to approximate complex relationships, on a number of datasets. Attarzadeh and Hock [18] again used Neuro-Fuzzy techniques to handle uncertainties in SCE. Hari et al. [19] used a hybrid method by constructing clustered data using CAK-Means clustering. The PSOIW algorithm applied to clustered data for cost estimation. Jiang et al. [20] used grey associating degree analysis of driver factors of SCE. Hari et al. [21] also proposed Interval Type-2 Fuzzy in a better way to handle uncertainties and imprecision.

Dejaeger et al. [22] argued the better performance of Data Mining techniques in SCE, but not as a replacement for Expert judgment. Zhang and Zhang [23] used Fuzzy-Grey evaluation method for SCE. Malathi and Lijin [24] employed a hybrid model, by integrating an Analogy based reasoning with Fuzzy and logistic variables to handle uncertainty. Mittas et al. [25] provided a framework for statistical comparison of SCE models and visualization using Automation tool like startREC. Zhang et al. [26] employed Bayesian Regression modeling and Expectation Maximization method for SCE. Miandoab and Gharehchopogh [27] used the Cuckoo Optimization Algorithm and KNN with several datasets. Most recently, Mohanty et al. [28] used Ant Colony Optimization Techniques to obtain the best results in SCE in the literature as of late.

III. DATA DESCRIPTION AND PREPARATION

In this research paper, we have chosen to use the International Software Benchmarking Standards Group (ISBSG) dataset [29] to verify the proposed model. This dataset has data collected over a decade and is based on suitable metrics. About 4106 projects, with each project having a total number of 105 attributes divided into 18 sub-attributes. This paper shows a prediction in terms of work effort. The time and software cost can be derived from the work effort.

Dimensionality reduction is essential for faster processing and higher accuracy of the data. Thus, we've carried out preprocessing of the ISBSG dataset, where we've employed data transformation and selection. Data transformation is required to remove noise and missing values from the data, which requires the following transformation methods. Handling missing values and Normalization. Missing data can be handled either by removing null values or by data imputation. Here, data cleaning is done to remove any projects containing null values for the effort summary as it was found that data imputation didn't provide any remarkable change in the overall result.

During data selection, we need to identify features that have the highest potential in providing good work effort estimates. Hence, here we consider the following 5 attributes as relevant for the effort prediction; Input Count, Output Count, Enquiry Count, Master File and Interface File. The five attributes taken contain 1531 project values which are used for training the proposed model for cost estimation. Then we must normalize the data as some features may show a larger range of values than others which may have a negative impact on the overall computation. All numeric values are ranged in the scale of (0-1).

IV. METHODOLOGY

A. *Artificial Neural Networks*

Artificial Neural Networks [30] are a model that is based on the basic anatomy of the human nervous system, where highly interconnected nodes between neurons transmit information and uncover a relation using computational resources. The main intent of neural networks is to design a computational machine that can secure knowledge faster than outdated systems and make it useful. In the present days Neural Networks have become widely popular in use both in the industry as well as in research; foundationally this arises as it can model around almost any function. They can do different tasks, such as classification, approximation, pattern detection, regression, and clustering. Neural networks have been the most popular choice of models for Software Cost Estimation by researchers.

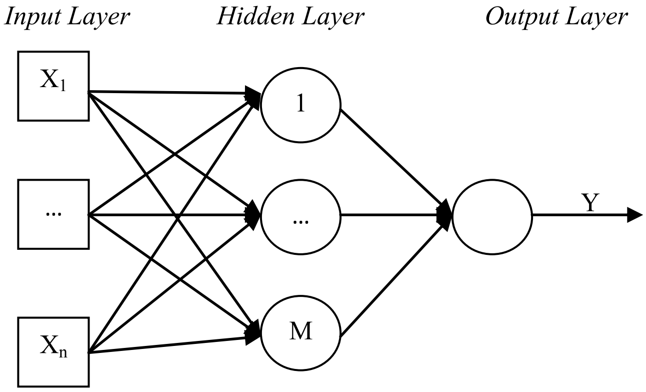

Fig1. The architecture of an Artificial Neural Network

B. *Cuckoo Optimization Algorithm*

The Cuckoo Optimization algorithm is a meta-heuristic algorithm proposed by Rajabioun [31] in 2011. It was first expanded by Yang and Deb in 2009 [32] where levy flights were used. Later, Rajabioun [31] described the algorithm in detail.

In this method, the essential factor that makes this algorithm work is due to the property of egg laying by the cuckoos. Each cuckoo has some eggs, which they lay in other birds' nests, as is the characteristic of the cuckoo's special lifestyle for production. They lay eggs that look similar in design and size to that of the host bird's eggs to avoid detection by the host. Many bird species learn to recognize the cuckoo eggs in their nests and their either desert the nest to start afresh or throw out the strange egg. So cuckoo constantly improves its mimicry in laying eggs to avoid detection. These cuckoo

eggs hatch earlier than the host eggs and kill them by their faster, mature development and food resource requirement.
The greater number of surviving eggs in a zone, the greater the suitability of that zone, thereby leading to more attention to that zone. This is the parameter that the Cuckoo Optimization Algorithm wants to optimize.
Like any other Evolutionary algorithm, this begins with an initial population of cuckoos. Terminologies in GA by Holland [33] and PSO by Kennedy and Eberhart [34], the array of problem variables is referred to as Chromosome and Particle Position respectively. Similarly, here in the Cuckoo Optimization Algorithm, it is called Habitat.

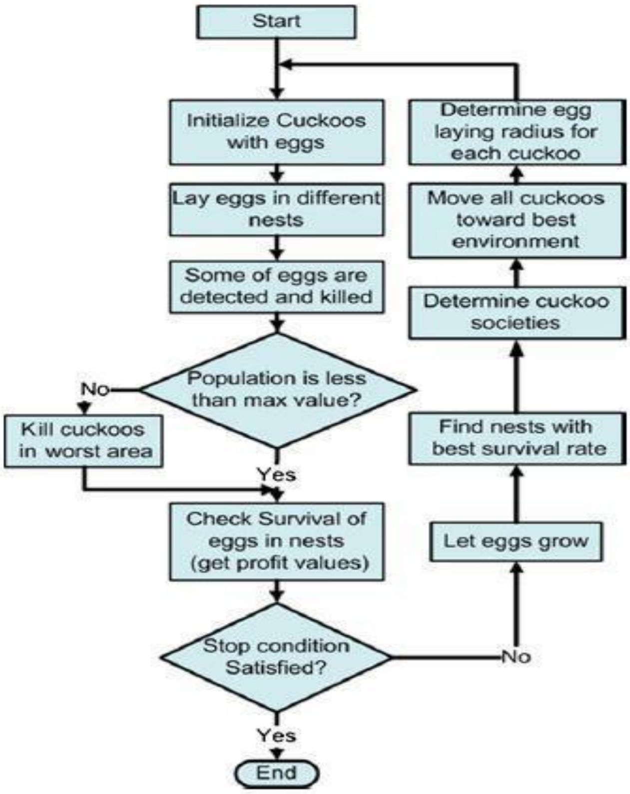

Fig 2. COA flowchart. [30]

The 1 x N_{var} array of problem variables is shown as follows,

$$Habitat = [X1, X2, X3, Xn] \quad (1)$$

The suitability of the habitat is evaluated using the profit function f_p,

$$Profit = fp(Habitat) = fp(X1, X2, X3Xn) \quad (2)$$

$$Profit = cost(Habitat) = -fc(X1, X2, X3, ., Xn) \quad (3)$$

A candidate size of N_{pop}*N_{var} is used to begin the optimization problem. Each habitat is given a random number of eggs. The egg laying radius (ELR) is calculated by taking into account the number of eggs and the distance between the current cuckoo positions and the current optimized zone. The ELR is calculated as

$$ELR = \alpha \cdot \frac{Number\ of\ current\ cuckoos\ eggs}{Total\ number\ of\ eggs} \cdot (varHig - varLow) \quad (4)$$

When cuckoos migrate, they travel λ% of the way with a deflection, shown as follows.

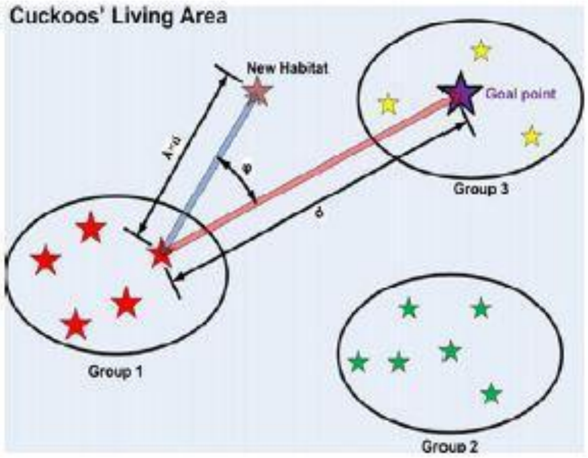

Fig 3. Egg Laying of Cuckoos. [30]

After Egg laying, P% of eggs that is less profitable (usually 10%) are destroyed and the remaining eggs hatch and the chicks grow into mature cuckoos. Each cuckoo lives in their zones till they grow up and when the time for egg laying arrives, they seek more habitable zones for laying eggs. To find the most habitable zones amongst the ones that the cuckoos are living in, K-means clustering is done to solve this problem to get all cuckoos from different groups to the best location

Migration Formula is:

$$X(\text{Next Habitat}) = X(\text{Current Habitat}) + F(X\ \text{Goal Point} - X\ \text{Current Habitat}) \quad (5)$$

C. *Proposed ANN-COA model*

Estimation of parameters for nonlinear modeling can be referred to as a function optimization problem. The best parameters to be chosen are those that optimally relays to the specific function to be optimized. According to Saati and Kareem [35] in this scenario, learning algorithms such as Gradient descent tend to find the local minimum solutions, whereas stochastic search algorithms such as Evolutionary strategies present a more efficient functionality in estimating the models' parameters. Keeping this in mind, we used an ANN to make predictions of effort on the Software Cost Estimation dataset using the Cuckoo optimization technique.
As a parametric optimization method, we input parameters of ANN to the meta-heuristic model to get the ideal values to make the best predictions on the dataset. We used the sigmoid activation function for the hidden layer of our ANN model. As meta-heuristics don't make any assumptions about the parameters to be optimized, but search for the best values to produce the least cost in case of cost minimization problems. Hence, they help reach the global minima quickly with respect to cost function.

1. ***Initialize Cuckoo habitats with some random points on the profit function***
2. ***Dedicate some eggs to each cuckoo***
3. ***Define ELR for each cuckoo***
4. ***Let cuckoos lay eggs inside their corresponding ELR***
5. ***Kill those eggs that are recognized by host birds***
6. ***Let eggs hatch and chicks grow***
7. ***Evaluate the habitat of each newly grown cuckoo***
8. ***Limit cuckoo's maximum number and kill those who live in worst habitats***
9. ***Cluster cuckoos and find best group and select goal habitat***
10. ***Let new cuckoo population immigrate toward goal habitat***
11. ***If stop condition is satisfied stop, if not then go to step 2.***

Pseudo Code for COA

In the proposed model, RMSE and MMRE are used as the fitness functions. MMRE is the most widely employed performance metric for software cost estimation in literature. The objective is to minimize these values. These metrics are defined in (6) and (7).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Ai - Ei)^2} \tag{6}$$

$$MMRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Ai-Ei|}{Ai} \tag{7}$$

V. Results and Discussions

We verified this technique on the ISBSG licensed dataset, which contains 1531 projects. In this dataset, we used 5 input variables which resulted in the output effort estimation. Normalization of the dataset was done as discussed in section 3.

In this paper, software effort is estimated as follows

$$Effort = C \times Size \tag{8}$$

Where C is a constant and effort is measured in terms of Person-Hours also size is measured in terms of Kilo Lines of Code (KLOC). Functional Point analysis is carried out to calculate the size factor.

$$FP = UFP \times VAF \tag{9}$$

Where VAF is a value adjustment factor which is the complexity that is in processing the project and the Unadjusted Function Point is as follows:

$$UFP = External\ Input \times 4 + External\ Output \times 5 + External\ Inquiry \times 4 + Log\ File \times 10 + Interface\ File \times 7 \tag{10}$$

The experimentation was carried out in MATLAB programming environment, the simulation of Cuckoo optimization algorithm and ANN. The dataset was divided into ratio of 90:10 and followed by 10 fold cross-validation was employed. The results were measured in RMSE and MMRE metric values as defined in section 4.3.

While experimenting we chose the following parameter values to produce the best optimization results in COA shown in table 1. The stop condition associated with the COA to get the ideal results were based on early stopping rounds of a 100 iterations, wherein training is continued only if the cost keeps improving after every 100 iterations. It was found that the ideal cost value converged within 500 iterations. We've carried out multiple experiments with different maximum cuckoo values such as 10, 20, 30, 40, and 50. We found that 20 maximum numbers of cuckoos give the best results in an appropriate duration. Higher the value of the maximum cuckoo count, the greater was the duration observed for the neural network parameter optimization. We also experimented with various activation functions for the ANN model such as Sigmoid, ReLu, and Tanh, to find that the model made better predictions with the sigmoid activation function. It was also found that a slight change in cuckoo population variance value proved in better accuracy.

We presented the results of the RMSE values and MMRE values of our proposed model in Tables 2 and 3 respectively, in comparison with previous techniques employed from literature.

Table 1. Cuckoo Parameter Values

S. L No	Name of the Parameter	Value
1	No. of Cuckoos	5
2	Min No of Eggs	2
3	Max No of Eggs	4
4	Motion Coefficient	9
5	Max No of Cuckoos	20
6	Radius Coefficient	0.05
7	Cuckoo Population Variance	1e-10

Table 2. RMSE values comparison on ISBSG dataset.

S.L No	Method	RMSE (Test)
1	ACOT	0.00817
2	**ANN-COA**	**0.03578**
3	GP	0.03794
4	CPNN	0.04499
5	CART	0.04561
6	TREENET	0.04565
7	MLP	0.04817
8	MLR	0.04833
9	DENFIS	0.04837
10	MARS	0.04871
11	SVR	0.04922
12	RBF	0.05167
13	POLYNOMIAL REGRESSION	0.05327

Table 3. MMRE values comparison on ISBSG dataset

S.L No	Method	MMRE (Test)
1	**ANN-COA**	**0.1482**
2	DTF	0.1700
3	MT-EBA	0.2010
4	DT	0.4900
5	MLR	0.5800
6	NABE	0.5700
7	OLS	0.6700
8	ABE	0.7100
9	ANN	0.7500
10	RABE	0.8100
11	SABE	0.8500

VI. Threats to Validity

The threats to validity of this research are as follows,

- We have used MMRE as one of the evaluation metrics, for verifying our proposed model and comparing the results with other models from literature. MMRE has been criticized, as a performance criterion by Shepperd and MacDonell [36] because it favors models that underestimate and is therefore biased, so that can be considered as a threat.
- We have used meta-heuristics to optimally select parameters for our Neural Network model based on the intrinsic relation of the data provided; implying that the learning methodology is sensitive to the data provided, thus data collection must be certifiable or only standard or licensed datasets may be used.
- We used the ISBSG dataset to conduct our experiments, which is cross-company data. Not using within company data in this research can be considered a threat.

VII. Conclusion

In this paper, we employed the proposed ANN-COA model on the ISBSG dataset to predict Software Cost Estimation. The use of meta-heuristics for parameter optimization is portrayed with this technique particularly for an ANN model. With a higher likeliness to reach global maxima or minima than traditional training methods, this technique is subject to further research in other domains. With the select values for COA parameters to optimize the ANN, we get the values as shown in Tables 2. and 3. in terms of RMSE and MMRE values, respectively. In comparison to the other techniques deployed from literature, we have found that ANN-COA model has outperformed most models except ACOT. Particularly this model has proven to outperform other Neural Network based techniques. Hence we can conclude that the ANN-COA model is the relatively the best amongst the Neural Network predictors.

References

[1] R. K. Mohanty, V. Ravi, and M. R. Patra, "The Application of Intelligent and Soft-computing Technique to Software Engineering Problems", A state of the art Report. International Journal of Information and Decision Sciences, Vol. 2, Number 3, pp. 232–272, 2009.

[2] B. W. Boehm, "Software Engineering Economics", Vol. 197, Englewood Cliffs (NJ): Prentice-Hall, 1981.

[3] L. H. Putnam, "A general empirical solution to the macro software sizing and estimating problem", IEEE Trans. Soft. Eng., July 1978, pp. 345-361.

[4] N. A. Parr, "An alternative to the Raleigh Curve Model for Software development effort", IEEE on Software Eng, May 1980.

[5] G. Cantone, A. Cimitile, and U. De Carlini, "A comparison of models for software cost estimation and management of software projects", in Computer Systems, Performance and Simulation, Elsevier Science Publishers B.V., 1986.

[6] J.S. Pahariya, V. Ravi, and M. Carr, "Software Cost Estimation using Computational Intelligence Techniques", IEEE Conference on World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), pp. 849–854, 2009.

[7] V. Venkataiah, R.K. Mohanty, and M. Nagaratna, "Application of Practical Swarm Optimization to predict Software Cost Estimation" 6th IEEE International Conference on Communication Systems and Network Technologies, 05–07, March 2016.

[8] A. Idri, and A. Zahi, "Software Cost Estimation by Classical and Fuzzy Analogy for Web Hypermedia Applications", A replicated study. IEEE Symposium on Computational Intelligence and Data Mining (CIDM), pp. 117–121, 2013.

[9] D. Manikavelan, and R. Ponnusamy, "To Find the Accuracy Software Cost Estimation Using Differential Evaluation Algorithm", IEEE International Conference on Computational Intelligence and Computing Research, 2013.

[10] V. Lalit Patil, M. Nitin Shivale, D. JoshiJ, V. Khanna, "Improving the Accuracy of CBSD Effort Estimation using Fuzzy Logic", IEEE International Advance Computing Conference, pp. 1395–1391, 2014.

[11] A. Kaushik, A. K. Soni, and Rachna Soni, "A Simple Neural Network Approach to Software Cost Estimation", Global Journals of Computer Science & Technology, Vol. 13, Issue 1, Version 1, 23-30, 2013.

[12] S. P. Singh, and P. Johri, "A Review of Estimating Development Time and Efforts of Software Projects by Using Neural Network and Fuzzy", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, Issue 10, 306-310, Oct. 2012.

[13] Ch. S. Reddy, P. S. Rao, KVSVN Raju, and V. V. Kumari, "A New Approach For Estimating Software Effort Using RBFN Network", International Journal of Computer Science and Network Security, Vol. 8, No.7, pp. 237-241, July 2008.

[14] A.R. Venkatachalam, "Software cost estimation using artificial neural networks", Proc. of International Joint Conference on Neural Networks (IJCNN), Vol. 1, pp. 987-990, 25-29 Oct. 1993.

[15] I. Attarzadeh, A. Merhanzadeh, and A. Barati, "Proposing an Enhanced Artificial Neural Network Prediction Model Improve the Accuracy in Software Effort Estimation", IEEE Fourth International Conference on Computational Intelligence, Communication Systems and Networks, pp. 167–172, (2012)

[16] V. K. Bardsiri, D. N. A. Jawawi, S. Z. M. Hashim, and E. Khatibi, "Increasing the accuracy of software development effort estimation using project clustering", The Institution of Engineering and Technology Journal, Vol.6, Iss.6, pp. 461–473, 2012.

[17] Y. Li, M. Xie, and T. N. Goh, "A study of the non-linear adjustment for analogy based software cost estimation", Emperical Software Engineering, pp. 603-643, 2008.

[18] I. Attarzadeh, and O. S. Hock, "Proposing a New Software Cost Estimation Model Based on Artificial Neural Networks", IEEE 2nd International Conference on Computer Engineering and Technology, Vol. 3, pp. 287–291, 2010.

[19] CH. V. M. K. Hari, S.S. Tegjyot, B.S.S. Kaushal, and S. Abhishek, “CPN-a hybrid model for software cost estimation”, IEEE International Conference on Recent Advances in Intelligent Computational Systems (RAICS), 902–906, Sep 22, 2011.
[20] Jiang, G. Wang,Y., Haitao., Research on Software Evolution Model on Case Based Reasoning. IEEE 2nd International Conference on WRI World Congress on Software Engineering, 338–341, 2010.
[21] CH. V. M. K. Hari, P. V. G. D. Prasad Reddy, M. Jagadeesh, and G. SriRam Ganesh, “IntervalType-2 Fuzzy Logic for Software Cost Estimation Using TSFC with Mean and Standard Deviation”, IEEE International Conference on Advances in Recent Technologies in Communication and computing, pp. 40–44, 2010.
[22] K. Dejaeger, W. Verbeke, M. David, and B. Bart, “Data Mining Techniques for Software Effort Estimation, A Comparative Study”, IEEE Transactions on Software Engineering, Vol. 38, No. 2, March/April 2012.
[23] B. Zhang, and R. Zhang, “Evolution Model of Software cost estimation methods based on Fuzzy-Grey Theory”, IEEE Fourth International Conference on Internet Computing for Science and Engineering, pp. 52–55, 2009.
[24] S. Malathi, and B.S. Lijin, “An Efficient Method for the Estimation of Effort in Software Cost”, International Journal of Advance Research in Computer Science and Management Studies Volume 2, pp. 330-335, February 2014.
[25] M. Nikolaos, I. Mamalikidis, and L. Angelis, “A framework for comparing multiple cost estimation methods using an automated visualization toolkit”, Information and Software Technology Vol. 57, pp. 310–328, 2015.
[26] W. Zhang, Y. Yang, and Q. Wang, “Using Bayesian Regression and EM algorithm with missing handling for software effort prediction”, Information and Software Technology, pp. 58–70, February 2015.
[27] E. E. Miandoab, and F. G. Gharehchopogh, “A Novel Hybrid Algorithm for Software Cost Estimation Based on Cuckoo Optimization and K- Nearest Neighbors Algorithms”, International Journal of Engineering, Technology & applied Science Research, Vol. 2, No. 3, pp. 1018–1022, 2016.
[28] V. Venkataiah, R. K. Mohanty, J.S. Pahariya, and M. Nagaratna, “Application of Ant Colony Optimization Techniques to Predict Software Cost Estimation” In, S. Satapathy, V. Bhateja, K. Raju, B. Janakiramaiah (eds) Computer Communication, Networking and Internet Security. Lecture Notes in Networks and Systems, Vol. 5, Springer, Singapore 2017.
[29] International Software Benchmarking Standards Group(ISBSG) , http://isbsg.org/software-project-data/
[30] E. Praynlin, and P. Latha, “Performance analysis of software effort estimation models using neural networks”, International Journal of Information Technology and Computer Science (IJITCS), Vol.5, No. 9, pp.101, 2013.
[31] R., Rajabioun, Cuckoo Optimization Algorithm. Applied Soft Computing journal, 11, 5508-5518. R., 2011.
[32] X.S. Yang, S. Deb, “Cuckoo search via Lévy flights”. In, Proc. of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), December 2009, India. IEEE Publications, USA, pp. 210-214, 2009.
[33] J. Holland, “Adaptation in Natural and Artificial Systems”, University of Michigan, Michigan, USA, 1975.
[34] J. Kennedy and R. Eberhart, "Particle swarm optimization," Neural Networks, 1995. Proceedings, IEEE International Conference on, Perth, WA, 1995, pp. 1942-1948 vol.4.
[35] N. A. AL-Saati, M. A. AlKareem, “The Use of Cuckoo Search in Estimating the Parameters of Software Reliability Growth Models”, International Journal of Computer Science and Information Security, Vol. 11, No. 6, 2013.
[36] M. Shepperd and S. MacDonell, “Evaluating prediction systems in software project estimation,” Information and Software Technology 54, 820–827, 2012.