International Journal of Pure and Applied Mathematics
Volume 119 No. 16 2018, 4035-4049
ISSN: 1314-3395 (on-line version)
url: http://www.acadpubl.eu/hub/
Special Issue
http://www.acadpubl.eu/hub/

An Efficient Fuzzy-Based Software Cost Prediction Model for Software Product or Application Requirement Specification and Effort Evaluation

Maharasan.K.S[1], V.Saravanan[2]

[1]Research Scholar – Computer Applications, Faculty of Science and Humanities, Anna University, Chennai, Tamil Nadu, India
[2]Professor and Dean, Department of Computer Applications, Sri Venkateswara College of Computer Applications and Management, Coimbatore, Tamil Nadu, India

Corresponding Author's E-mail: maharasan@gmail.com

Abstract: Software project management is an interpolation of software project planning, project monitoring and project termination. The objectives of software planning are investigating the future, to analyze the attributes that are essentially done for the consummation of the project successfully, specify the requirement of product development, animate the proper scheduling and allocate resources for the attributes. Software cost evaluation is an essential task in preeminent software project decisions such as resource allotment and bidding. The software cost analysis is totally depending on clarity of requirement analysis. There many software cost evaluation model introduced like COCOMO, DELHI, SLIM and FPA etc. However, none of technique helps to non-technical background clients to specify their requirement. It is challengeable task to understand the requirement specification like hardware, software, environment and man power with timing bound. To bring better solution of above problems, An Efficient Fuzzy-Based Software Cost Prediction Model (EFSCPM) is proposed for software cost evaluation. The proposed model carries a few desirable attributes of an efficient fuzzy-based software cost prediction model, while maintaining the qualities of the EFSCP model. The cost, effort evaluates of software projects done by a variety of associations are assembled, the outcomes are separated with the present software cost approaches, and the MRE (Mean Relative Error) is enumerated. Unlike the standard neural network approach, the proposed model can be interpreted and validated by software professionals and has good generalization ability. The model manages efficiently with inaccurateness of uncertain input of software application and improves the consistency of software applications cost evaluations. It permits input of software application information to have continuous rating variables and linguistic variables, thus avoiding the issue of similar software projects having large different evaluated costs. Based on Experimental evaluations, proposed method minimizes 15.22 % Mean Magnitude of Relative Error (MMRE) and improves the 10.05 % Prediction (PRED) and 3.32 % Evaluation Function (EF), compared to previous methods.

Keywords: MRE (Mean Relative Error), Efficient Fuzzy-Based Software Cost Prediction Model (EFSCPM), evaluation function, prediction, requirement specification.

1. INTRODUCTION

Nowadays, the software project development has an important investment for numerous associations. The software engineering professionals are involved in the process of predicting the price and quality of a software product. Software project management evaluation is including cost, quality evaluation and risk investigation which are affected by the accuracy of the software product. Software project accuracy is the most important goal of software engineering management. The software evaluation accuracy is offering support for software project management decisions to clear users. For example, many associations can support the accurateness of cost evaluations. It is investigating the feasibility of product and efficiently controls the software product development method, consequently reducing the risk.

In the past, numerous software evaluation approaches are developed, and software evaluation cost developing quickly. However, in software project engineering, the stake holders are expecting low cost with high quality of software product. COCOMO81, COCOMOII, SLIM, FP and Delphi models are previous software cost evaluation methods. Previous methods have limitations to help non-tech client to understand the requirement of their product or applications. The system does not have proper requirement specifications to clarify the requirement. The system is unable to evaluate the actual product cost in software engineering management, for instance, managing with a lot of inaccurateness and high compound nonlinear relations among variables. The current systems are predicting more error during effort evaluation.

To bring solution for above problems, an Efficient Fuzzy-Based Software Cost Prediction Model (EFSCPM) method is introduced for software cost evaluation in software engineering management. The method carries out some popular attributes of an artificial neural network (ANN) with fuzzy method, while maintaining the merits of the EFSCPM model. The approach easily interpreted and authenticated by software professionals, and has generalization ability. The method manages efficiently with precise and correct input and improves the consistency of software cost evaluations in software engineering management. It permits input to maintain ranking variables and linguistic variables, consequently, avoiding the issue of similar software products having huge, diverse evaluated costs. The system explains the proper requirement specifications to clarify the product requirement. Even though, client does not technical background then also he/she can easily understand application or product requirement. The proposed model predicts actual product cost in software engineering management, for instance, managing with good accuracy. The proposed method minimizes the product cost prediction error during effort evaluation. The verification utilizing industry product information demonstrates that the method enhances evaluation accuracy in contrast. The rest of paper contribution is followed as:

- To develop Efficient Fuzzy-Based Software Cost Prediction Model (EFSCPM) for software cost evaluation in software engineering management
- To specify product or application requirement with clarity and graphical representation for understanding easily
- To contribute high and effective compound nonlinear characteristics for predicting the relation among software output variables with various types of cost driver.
- To utilize both software professionals the knowledge and statistical software product information in a single method.
- To minimize Mean Magnitude of Relative Error (MMRE) and improve the Prediction (PRED) and Evaluation Function (EF), compared to previous methods.

The rest of the paper is organized as Section 2 addresses the various literature that closest to proposed methodology. Section 3 explains the proposed methodology, implementation steps of proposed techniques with their features. Section 4 explores the implemented result with comparative analysis. Section 5 summarizes the overall work with the future work.

2. LITERATURE REVIEW

In [1] described project requirement model to construct the suite of test sequences. It mentioned test sequence, the series of inputs and outputs for software testing of the implementation. Here the test sequence obtained from tabular and systematic SCR requirements contained different data types such as Boolean, Double, Integer, Enumerated etc.,. In [2] expressed to produce proper artefacts, by documenting the software requirements as an SRS in the ideal IEEE 830:1998 format document. The issue noticed by measurement of predicting the software complexity depends on SRS which yet to be developed the software. The complexity measurement was served as a basis for the computing, and also extracted predicable evaluation of software development efforts. In [3] explored the improved requirement based complexity (IRBC) for the software project or product development. The approach was systematic, and an integrated approach to estimate software development and testing efforts. The IRBC measure primarily served to software development activities.

In [4] focused on evaluating the effort based on early steps. The primary goal of the work is to determine the correlation among the early steps effort and software size effort with the software development effort. In [5] discussed about consideration of software architectures, testing methods, and software failure manifestations. In [6] covered the topics like requirements specifications, designing, implementations, verification, validation, maintenance, software psychology and economics. The work discussed on the factors like lag, education shortfalls, rapid technology change, technology transfer inhibitions, inappropriate role models, resistance to discipline the methods and a restrictive view of software engineering. In [7] expressed an indication of current research in requirements engineering (RE). In [8] formulated a markov model for determining the consistency of a software system depended on every individual module's consistency and fixed the measurement of inter-modular transition possibilities as the user profiles. The work developed sensitivity analysis schemes that determined modules which were most critical to the consistency of the framework. In [9] explained particular fields like providing practical testing methods, tools, and processes that are helpful for the software engineers in developing high-quality software.

In [10] discussed about software product lines (SPL) testing methodology to identify effective approaches that also needs for future research. A systematic mapping study has been launched to finding many kinds of literature. In [11] described the alternative approach for estimating the use of analogies. The principle propagated the characteristics of the projects regarding parameters like development method, number of interfaces or size of the functional requirements document. In [12] summarized few sections of software cost evaluation approaches and mechanisms such as expertise-based mechanisms, learning-oriented mechanisms, dynamic based mechanisms, composite-Bayesian approaches, regression-based approaches, parametric approaches, to integrate the regression-based and expertise-based approaches. In [13] evaluated the popular algorithmic models which employed for estimating software costs like COCOMO, Function Points, ESTIMACS, and SLIM.

In [14] focused on the CMM level 5 projects from multiple organizations for studying the impact of highly established procedures on effort, cycle time and quality. The higher levels of project maturity indicated from the CMM level 5 ranking, the consequences of main elements to impact software development effort, cycle time and quality. In [15] expressed several recommendations for increasing the search more broadly manner for the relevant studies. The manual search of the relevant papers completeness was essential. Conducting more research on the basic software cost estimation topics was efficient in real life situations. In [16] investigated the existing methods to support the requirements prioritization that depends on the early benefit and cost estimations which were done by software requirements (SR). The methods are empirically verified and exclusively treated the questions concerning method support.

In [17] investigated two important data-driven, software cost modelling questions. The work provided the answers for modeling methods were likely produced more results while utilizing software development cost data. In [18] surveyed verification, validation, testing methodologies, merits, demerits and life-cycle usage. The work described automated tools for performing validation, verification, and testing. In [19] studied the software reliability which included non-homogeneous poisson process (NHPP) models. It derived software reliability models, NHPP also determined environmental factors and models on costs. More applications have been presented for validating descriptive and predictive reliability in the software models. In [20] explained hybrid model to reduce the error value and proper product cost evaluation. The convergence of meta-heuristic algorithms is high in the possibility for finding the universal solution. However, the method is unable to specify the product or application requirement in clear way for effort estimation.

3. PROPOSED SYSTEM

This section illustrates the proposed strategy, implementation pre-processing steps, and implemented methodology details. The proposed Efficient Fuzzy-Based Software Cost Prediction Model (EFSCPM) method minimizes the cost evaluation in software engineering management. It can be easily interpreted and authenticated by software professionals, and has generalization ability. It utilizes both software professional knowledge and statistical software product information in a single framework. Figure 1 exhibits the working model of the proposed EFSCPM with implementation processing steps and mathematical evaluation details. The pre-processing implementation steps are described below in detail.

3.1 User Authentication

The module authenticates the user to interact with all the software cost evaluation models. If the user wants to access the software cost calculator, he/she must be a registered with centralized database. The module enables only authenticated users have access to the software cost calculators.

3.2 Software Cost Evaluations

The module offers comparative analysis of all the basic software cost and effort evaluation. The basic inputs for the module are size, mode, and cost drivers based on the model which the user selects to compute the effort. The different types of software cost evaluation calculators are designed like a COCOMO81, COCOMO2, FPA, SLIM, DELPHI. These models offer the user a wide option various types of application compatibility for product development cost and effort evaluations.

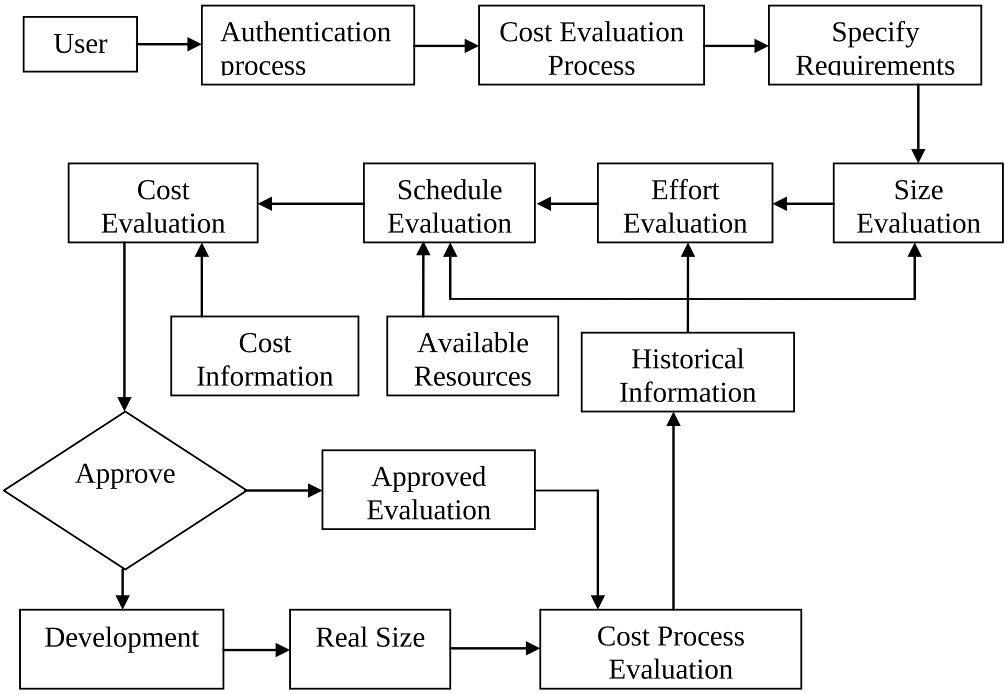

Figure 1: Workflow Diagram of Proposed Efficient Fuzzy-Based Software Cost Prediction Model

3.3 Efficient Fuzzy-Based Software Cost Prediction Model (EFSCPM)

Efficient Fuzzy-Based Software Cost Prediction Model (EFSCPM) is proposed for software cost evaluation in software engineering management. The approach easily interpreted and authenticated by software professionals, and has generalization ability. The method manages efficiently with precise and correct input and improves the consistency of software cost evaluations in software engineering management.The inputs of EFSCPM are the software project/product size and rankings of cost drivers, containing level attributes and effort multipliers. It depends on the integration of the artificial neural networks (ANN) classifier and fuzzy logic. The proposed system main basis of proposed model develops from the COCOMO2 post architectural model where scaling features and effort multipliers are applied to an efficient fuzzy-based software cost prediction sub-model. The sub-model produces new scaling features and effort multipliers which are much more proficient than the previous approaches.

The system explains the proper requirement specifications to clarify the product requirement. Even though, client does not have technical background then also he/she can easily understand application or product requirement. The proposed model predicts actual product cost in software engineering management, for instance, managing with good accuracy. Each cost driver correlates to an every ranking stage value called a multiplier variable that is a quantitative variable used in the model. Ranking of software cost drivers have many binary variables, or linguistic words such as “Very Low”, “Low”, “Nominal” “High” and “Very High” or medium complexity and high cost. The system has three main parts such as fuzzifier, fuzzy inference engine and defuzzifier. The fuzzifier transmits the input into linguistic words utilizing relationship operations that describes how much considered an estimated binary variable of a specific value fits the linguistic word. Fuzzy inference engine achieves the mapping among input relationship operations and the outcome relationship operations utilizing fuzzy rules which acquired and modeled relations of the software professional knowledge. The best input relationship degree, the stronger rule fires, the stronger the pull towards the output relationship operation. Because, the numerous outcome relationship operations included in the subsequent of rules triggered. The defuzzifier carries out a process to integrate the outcome into a single label or binary variable as per requirement. Thus an input into an EFSCPM can be either numerical or linguistic with the similar to the output. The proposed method minimizes the product cost prediction error during effort evaluation. It can easily utilized by software professionals the knowledge and statistical software product information in a single method. The method minimizes MMRE (Mean Magnitude Relative Error), and improves evaluation function, prediction (PRED25). The pseudo code of the proposed method is illustrated below in details:

```
Input: Size, Mode of the software projects and cost driver
Output: Display the Mean Magnitude of Relative Error (MMRE), Prediction (PRED25)
        and Evaluation Function (EF)with clear software or  application requirement specification
Procedure:
Start
    Perform authentication process
    Identify requirements for software project/product;
    Analyze and evaluate the all aspects of cost driver for project development;
    Apply EFSCPM cost evaluation method;
```

```
Display the historical information;
Compute entire Size of product/project requirement;
Calculate Effort of product/project requirement;
Identify available resources to perform the all SDLC process;
 Predict the project or product, Mean Magnitude of Relative Error (MMRE),
Prediction (PRED25) and Evaluation Function (EF);
If project or product cost predicted;
    Developing the cost drivers' information;
    Predict the Real size of historical information;
    Perform cost evaluation analysis;
    View product or application, MMRE, prediction and evaluation function;
    Else
    Failed to predict cost evaluation;
    Repeat the product or project requirement analysis process;
```

Pseudo Code for Proposed EFSCP Model

Report

The module displays the product or project predicted cost details with full analysis report. The module allows the user to view all the efforts which are computed during the software cost evaluations and fuzzy based software cost prediction module. It offers a unique feature for specific project to view the computed and cost for users.

4. RESULTS AND DISCUSSION

4.1 Experimental Setup

The experimental work is conducted on a laptop with Intel Dual Core processor with 2GB RAM, 500 GB memory and Window 7 Ultimate system. The proposed techniques is developed in JAVA programming environment and deployed in NetBeans 8.0, Appache Tomcat Server 8.0.3 and MYSQL database.

4.1.1 Dataset

In order to evaluate proposed EFSCPM, the NASA93 software projects dataset (http://promise.site.uottawa.ca/SERepository/datasets/cocomonasa_2.arff) is used for software product cost and effort evaluation. The cocomonasa93 contains 93 instances, 15 EM factors with 24 attributes.

4.2 Simulation Result

The proposed method represent a mathematical model to evaluate the performance of Mean Magnitude of Relative Error (MMRE), Prediction (PRED25) and Evaluation Function (EF) to finds the efficiency of cost and effort estimations. The proposed model expresses the product or application requirement specification in efficient way to perform the actual cost with actual effort.

4.2.1 Mean Magnitude of Relative Error (MMRE)

The proposed method describes mathematical model for mean magnitude of relative error in equation (1). The method computes the mean value for the indicator over all observed data. It is also average estimation accuracy of MRE. The MRE calculates the difference between actual effort and estimated effort relation to the actual effort in each observation of i.

$$MMRE = \frac{1}{n}\sum_{i=1}^{n} MRE_i \qquad (1)$$

$$MRE = \frac{|Acual\ Effort_i - Estimated\ Effort_i|}{Actual\ Effort_i} X100$$

4.2.2 Prediction (PRED)

The method describes PRED is an important criterion in software cost evaluation accuracy. The general approach of analysis of prediction accuracy is MMRE and PRED. The PRED(x) is described in equation (2).

$$\mathrm{PRED}\ (\mathrm{x}) = \frac{1}{\mathrm{n}} \times \sum_{\mathrm{i}=1}^{\mathrm{n}} \begin{Bmatrix} 1, & \text{if MRE} \leq \mathrm{x} \\ 0, & \text{otherwise} \end{Bmatrix} \qquad (2)$$

Where MRE is magnitude of relative error

4.2.3 Evaluation Function (EF)

The proposed method describes mathematical model for Evaluation Function in equation (3). In order to illustrate the superiority of hybrid estimation models two criteria MMRE and PRED is defined as Evaluation Function (EF).

$$\mathrm{EF} = \frac{\mathrm{PRED}(25)}{1+\mathrm{MMRE}} \qquad (3)$$

Table 1 represents mean magnitude of relative error (MMRE), Prediction (PRED25) and evaluation function (EF) with NASA93 software projects dataset. The proposed technique displays their average values for a respective parameter with the respective dataset. The proposed EFSCPM is evaluated with following existing models such as COCOMO (Constructive Cost Model) [20], GA (Genetic Algorithm) [20], and FA (Firefly Algorithm) [20]. The proposed EFSCPM is utilized to evaluate the software cost. The model also supports proposed methodology for minimizing the magnitude of relative error (MMRE), prediction (PRED25) and improves the evaluation function (EF). Along with Table 1, it noticed that EFSCPM has the best score on every particular constraint for the respective constraint.

Table 1: Mean Magnitude of Relative Error (MMRE), PRED25 and Evaluation Function (EF) for NASA93 software projects dataset

Algorithm	MMRE	PRED(25)	EF
COCOMO	58.80	51.61	0.86
GA	38.31	77.41	1.96
FA	30.84	80.64	2.53
Hybrid Model	22.53	88.17	3.74
EFSCPM	7.31	98.12	6.42

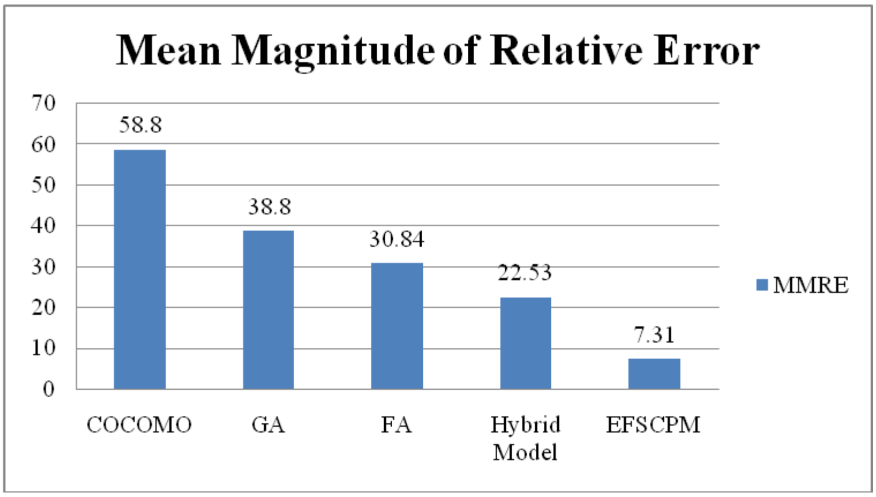

Figure 2: Comparison of Mean Magnitude of Relative Error (MRE) for NASA93 software projects dataset

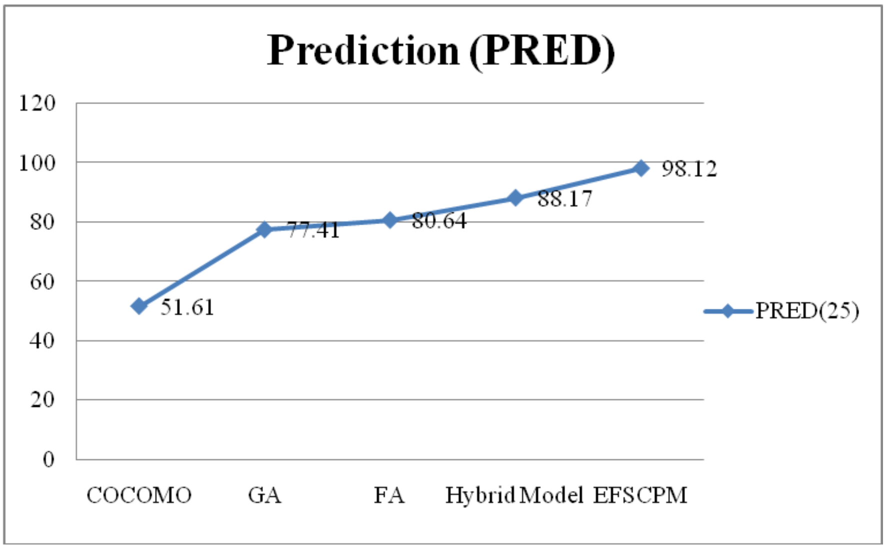

Figure 3: Comparison of PRED(25) for NASA93 software projects dataset

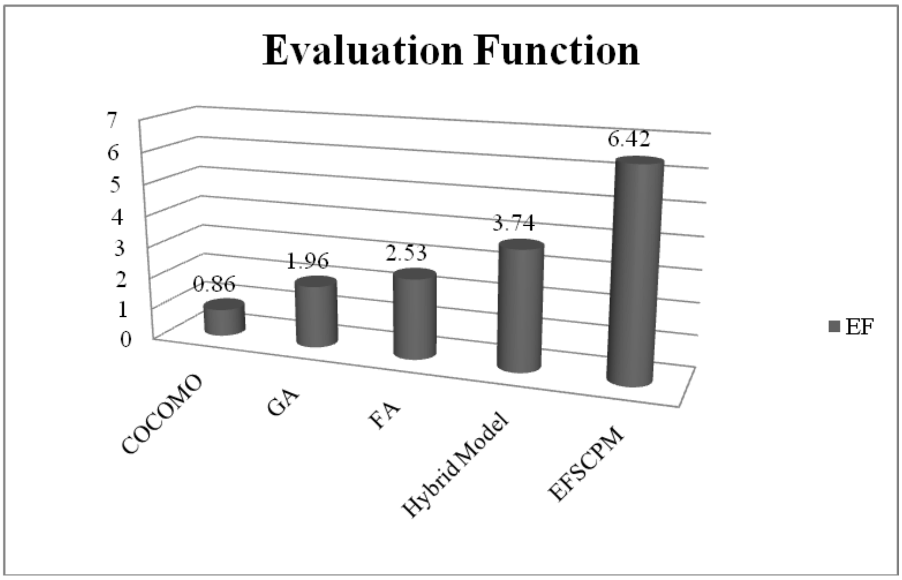

Figure 4: Comparison of Evaluation Function for NASA93 software projects dataset

According to Figure 2 to 4 observations, the proposed EFSCPM computes Mean Magnitude of Relative Error (MMRE), Prediction (PRED25) and Evaluation Function (EF) for identifying the effectiveness of the technique. The proposed EFSCPM is evaluated with COCOMO, GA and FA existing models behalf of Mean Magnitude of Relative Error (MMRE), Prediction (PRED25) and Evaluation Function (EF). The COCOMO model utilized to estimate the costs more precisely in software projects. However, it does not give accurate cost evaluations for the software projects because of the linearity of the mathematical functions and imprecision of project features. The proposed EFSCPM is providing accurate cost evaluations for software projects. GA is a meta-heuristic approach which is developed to discover optimum solution and hybrid optimization issues. However, it fails to maintain optimum solutions and does not discover optimization issues. The proposed EFSCPM maintains optimum solutions and determines issues of optimization. FA method is good convergence of solutions and accomplishing absolute optimum cost solutions for software projects. However, it fails to maintain effort evaluation issues. Hybrid model is the nearest competitor on overall constraints. The convergence of meta-heuristic algorithms is high in the possibility for finding the universal solution. However, the method is unable to specify the product or application requirement in clear way for effort estimation. The proposed EFSCPM minimizes 15.22 % Mean Magnitude of Relative Error (MMRE) and improves the 10.05 % Prediction (PRED25) and 3.32 % Evaluation Function (EF), compared to previous methods. Finally, the paper declares the proposed EFSCPM performs best on every parameter & respective input constraints.

5. CONCLUSION

An Efficient Fuzzy-Based Software Cost Prediction Model (EFSCPM) is a computing software cost evaluation. Various desirable attributes such as learning capability and good interpretability present in an Efficient Fuzzy-Based Software Cost Prediction Model. Therefore, the model is capable of generalization, an essential criterion for successful applications of neural networks and fuzzy logic techniques. The model permits for inputs as continuous-rating values and reduces the issue of similar projects with extensive different software management cost evaluations. The fuzzy based software cost prediction model permits the combination of numerical data and software professional knowledge. The system explains the proper requirement specifications to clarify the product requirement. Even though, client does not have technical background then also he/she can easily understand application or product requirement. The proposed model predicts actual product cost in software engineering management, for instance, managing with good accuracy. It can be a more useful tool to tackle important issues in software engineerings such as software cost evaluation and quality prediction. The proposed EFSCPM minimizes 15.22 % Mean Magnitude of Relative Error (MMRE) and improves the 10.05 % Prediction (PRED) and 3.32 % Evaluation Function (EF), compared to previous methods. Finally, the paper declares the proposed EFSCPM performs best on every parameter & respective input constraints.

In future, the paper work can be extended with cost and effort prediction for upgrade of existing product for client. It will expose for reusability of existing application with new feature for reliable client.

References

[1]. Gargantini, A., & Heitmeyer, C., “Using model checking to generate tests from requirements specifications”, In ACM SIGSOFT Software Engineering Notes, Springer-Verlag, Vol. 24, No. 6, pp. 146-162, 1999.

[2]. Sharma, A., Vardhan, M., & Kushwaha, D. S., “A versatile approach for the estimation of software development effort based on SRS document”, International Journal of Software Engineering and Knowledge Engineering, Vol. 24, No. 01, pp. 1-42, 2014.

[3]. Sharma, A., & Kushwaha, D. S., “APPLYING REQUIREMENT BASED”, ITCS, SIP, JSE-2012, CS & IT 04, pp. 393–415, 2012.

[4]. Tsunoda, M., Kamei, Y., Toda, K., Nagappan, M., Fushida, K., & Ubayashi, N., “Revisiting software development effort estimation based on early phase development activities”, In Mining Software Repositories (MSR), 2013 10th IEEE Working Conference, IEEE, pp. 429-438, 2013.

[5]. Lyu, M. R., “Software reliability engineering: A roadmap”, In 2007 Future of Software Engineering, IEEE Computer Society, pp. 153-170, 2007.

[6]. Boehm, B. W., “Software engineering economics”, IEEE Transactions on Software Engineering, Vol. 1, pp. 4-21, 1984.

[7]. Nuseibeh, B., & Easterbrook, S., “Requirements engineering: a roadmap”, In Proceedings of the Conference on the Future of Software Engineering, ACM, pp. 35-46, 2000.

[8]. Cheung, R. C., “A user-oriented software reliability model”, IEEE Transactions on Software Engineering, Vol. 2, pp. 118-125, 1980.

[9]. Harrold, M. J., “Testing: a roadmap”, In Proceedings of the Conference on the Future of Software Engineering, ACM, pp. 61-72, 2000.

[10]. Engström, E., & Runeson, P., “Software product line testing–a systematic mapping study”, Information and Software Technology, Vol. 53, No. 1, pp. 2-13, 2011.

[11]. Shepperd, M., & Schofield, C., “Estimating software project effort using analogies”, IEEE Transactions on software engineering, Vol. 23, No. 11, pp. 736-743, 1997.

[12]. Boehm, B., Abts, C., & Chulani, S., “Software development cost estimation approaches—A survey”, Annals of software engineering, Vol. 10, No. 1-4, pp. 177-205, 2000.

[13]. Kemerer, C. F., “An empirical validation of software cost estimation models”, Communications of the ACM, Vol. 30, No. 5, pp. 416-429, 1987.

[14]. Agrawal, M., & Chari, K., “Software effort, quality, and cycle time: A study of CMM level 5 projects”, IEEE Transactions on software engineering, Vol. 33, No. 3, pp. 145-156, 2007.

[15]. Jorgensen, M., & Shepperd, M., “A systematic review of software development cost estimation studies”, IEEE Transactions on software engineering, Vol. 33, No. 1, 2007.

[16]. Herrmann, A., & Daneva, M., “Requirements prioritization based on benefit and cost prediction: an agenda for future research”, In International Requirements Engineering, 2008. RE'08, 16th IEEE, pp. 125-134, 2008.

[17]. Briand, L. C., El Emam, K., Surmann, D., Wieczorek, I., & Maxwell, K. D., "An assessment and comparison of common software cost estimation modelling techniques", In Proceedings of the 21st international conference on Software engineering, ACM, pp. 313-322, 1999.

[18]. Adrion, W. R., Branstad, M. A., & Cherniavsky, J. C., “Validation, verification, and testing of computer software”, ACM Computing Surveys (CSUR), Vol. 14, No. 2, pp. 159-192, 1982.

[19]. Pham, H., “Software reliability and cost models: Perspectives, comparison, and practice”, European Journal of Operational Research, Vol. 149, No. 3, pp. 475-489, 2003.

[20]. Maleki, I., Ebrahimi, L., & Gharehchopogh, F. S., “A hybrid approach of firefly and genetic algorithms in software cost estimation”, *MAGNT Research Report,* Vol. *2,* No. 6, pp.372-388, 2014.