

An Accurate Effort Estimation using Fuzzy Based Alternating Regression Technique (FBART)

¹E.Karunakaran, ²N.Sreenath

*Department of Computer Science and Engineering
Pondicherry Engineering College, Puducherry, India.*

Abstract

- Precise software effort estimation is considered as the most significant task in the software development life cycle. Since, software developers use effort estimate as an input for planning budgets that highlights pricing policies and investment analysis. The over estimation or underestimation of software effort can result in detrimental effect on the quality of software and may cause huge monetary loss. Hence, there arises a need for developing reliable software effort estimation techniques that uses software effort impact factors. The Least Effort Multiplier based Fuzzy Estimation Algorithm (LEMFEA) [1] is an enhanced version of FFPA-PSR (fuzzy-based function point analysis with performance metrics, security, and reliability factors) [2] algorithm that has been proposed for improving the accuracy of the software effort estimation. In LEMFEA uses fuzzy logic to frame rules based on the classification of the attributes like project type (T), programmers skill (S), software language used (L), database used (D) and criticality(C) required for the estimation. In the proposed method Fuzzy Based Alternating Regression Technique (FBART) alternates between two regression techniques, viz., i) Gauss-Newton based model and ii) Weighted Deming Model based on the size of the project estimated based on function points and lines of code. This FBART is an adaptation of LEMFEA that has been proposed for improving the accuracy of the software effort estimation. This FBART algorithm uses Re-usability (R) as an extra multiplier for effective software effort estimation. The experiments are carried out and the performance of the proposed software estimation techniques are validated and the results confirms that Fuzzy Based Alternating Regression Technique (FBART) is found better to the existing software estimation techniques.

Keywords: Estimation, Effort Estimation, Sizing the software, COSMIC full function, fuzzy-based functional point, Software Measurement, eXtreme software size Unit.

INTRODUCTION

Software development effort estimation enables the prediction of the most accurate amount of effort, expressed in terms of person-hours or money required to develop and maintain software based on inadequate, indeterminate and noisy input. Effort estimates may be used as input to project and iteration plans, budgets, investment analyses, pricing processes and bidding rounds. Moreover, accuracy in effort

estimation is used for making decision as software developers and customers need to rely on them for preplanning, budgeting, risk analysis. Further, software companies are very keen in delivering accurate and reliable software products to the end users especially by minimizing software development effort. The performance factors are mainly integrated for enabling the estimation model to estimate effort in an accurate and automated way. Since, accurate effort is considered to be more significant because inaccurate and insecure estimation of software effort estimation leads to drastic deviation in the expected and actual budget.

Software metrics give the measurements of the software development process and product used as dependent and independent variables in models for project management. The wide-spread models are used for predicting the development effort for the software system based on size, complexity, developer characteristics, and other metrics. Though developing accurate and usable models are financially feasible, there are a number of issues that are to be dealt in using the traditional techniques of formal and linear regression models like, nonlinearities and interactions in complex real-world processes, lack of stationary, over-commitment to specified values, small quantities of data available, and the inability to use the available knowledge when exact numerical values are unknown.

There is limited success in the applicability of software complexity metrics in the determination of software quality. There are many metrics that measure similar aspects of program differences. Applying these metrics in quantitative modelling scenarios is difficult due to the lack of understanding of what is being measured.

RELATED WORK

Munson and Khoshgoftaar [30] have investigated the basic issues associated with the modelling process, including problems of shared variance among metrics and the possible relationship between complexity metrics and measures of program quality. Metrics with raw complexity metrics and complexity metrics are simplified through factor analysis. The ultimate aim is to understand the applicability of complexity metrics in predictive models, so as to permit the effective use of measures in the management of complex software projects.

To deal with these challenges, the applicability of fuzzy logic is investigated and recommendations are suggested are :

Regression analysis to derive predictive equations for software metrics is recently supported by increasing numbers of studies using non-traditional methods like neural networks, fuzzy logic models, case-based reasoning systems, and regression trees. The increasing level of sophistication in the regression-based techniques used, including robust regression methods, factor analysis, and more effective validation procedures are examined in [28]. The implications of using these methods are discussed and some recommendations are also suggested.

Given a small subset of factual data on cost drivers, cost estimation involves predicting man power, material, utilities or other costs over time. Statistical models based on regression assists this type of estimation. Artificial Neural Networks (ANNs), the non-parametric statistical estimators can be effectively used in cost estimation modelling. The performance, stability and ease of cost estimation modelling using regression versus NNs to develop Cost Estimating Relationships (CERs) are examined in [29]. From the results, it is evident that the NNs deal with data that does not adhere to the generally chosen low order polynomial forms, or data for which there is little knowledge of the appropriate CER to select for regression modelling. When an appropriate CER can be identified, regression models yield better results in terms of accuracy, variability, model creation and model examination.

Reliable software is mandatory for complex mission-critical systems. Finding fault-prone modules is a valuable technique for guiding development processes, so as to focus the resources on those parts of a system that are most likely to have faults. Logistic regression is better when compared to other classification modelling techniques as it has the capability to deal with interpretable coefficients. In the literature, prior probabilities and costs of misclassification are dealt the least. They are applied to a logistic regression-based classification rule for a software quality model in [31]. It offers an integrated method for using logistic regression in software quality modelling, including interpretation of coefficients, prior probabilities, and costs of misclassifications.

Software estimation is an uphill task for the developers. Paradigms like fuzzy logic may serve as an alternative for software effort estimation. Martin et al. [26] has given an application, the results of which are compared with those of a multiple regression. From the results, it is evident that the value of MMRE applying fuzzy logic is slightly higher than MMRE applying multiple regressions. Similarly, the value of Pred(.20) applying fuzzy logic is found to be slightly higher than Pred(0.20) applying multiple regression. Moreover 6 out of 41 MRE was equal to zero when fuzzy logic was applied.

Early in the development life cycle, algorithmic effort prediction models are not capable to deal with uncertainties and imprecision present in software projects. Ahmed et al. [27] has presented an adaptive fuzzy logic framework for software effort prediction. The training and adaptation algorithms in the framework have the capability to tolerate imprecision, explain prediction rationale through rules, incorporate experts knowledge, offer transparency in the prediction system, and adapt to new environments with the

availability of new data. The system was validated on artificial datasets as well as the COCOMO public database.

These effort multipliers can be analyzed based on lines of code or function point analysis. This research also focuses on a fuzzy-based estimation algorithm for identifying the uncertainty of the software size with the help of a triangular fuzzy set and function point analysis by considering an extra effort multiplier called Criticality of the software in addition to the multiplier factors is named as Least Effort Multipliers based on Fuzzy Estimation Algorithm (LEMFEA). In addition, LEMFEA algorithm tolerates imprecision, offers transparency in the prediction process and possess the capability of adapting to changing environment depending on the dynamic availability of new data.

Regression Analysis

Regression analysis is a statistical process that estimates the relationships among variables. It includes many techniques for modelling and analyzing several variables and relationship between a dependent variable and one or more independent variables. It helps in understanding how the typical value of the dependent variable changes with respect to variation in any one of the independent variables, while the other independent variables remain fixed. It estimate the conditional expectation of the dependent variable given the independent variables.

The estimation target is a function of the independent variables called the regression function. The variation of the dependent variable around the regression function is described by a probability distribution. It is widely used in prediction and forecasting, overlapping with the field of machine learning. It is also used to understand which independent variable relates to the dependent variable and explore relationships.

Many techniques are developed for carrying out regression analysis like linear regression and ordinary least squares regression, in that the regression function is defined in terms of a finite number of unknown parameters that are estimated from the data. Non-parametric regression refers to techniques that permit the regression function to lie in a specified set of functions which may be infinite-dimensional.

Regression refers to the estimation of continuous response variables, as opposed to the discrete response variables used in classification [23]. The continuous output variable is termed as metric regression to differentiate it from related problems [24].

Researchers have proposed number of techniques for accurate estimation of cost and effort involved in software project by considering the system security [4]. In many of the previous research works, authors have proved that the improvement of the accuracy of the highly dependent on software estimation and also the fuzzy-based function point analysis is incorporated to overcome the unpredictability and riskiness in effort estimation [5]. Some of the software estimation approaches are detailed below.

In [6], authors presented a fuzzy-based function point analysis method for handling ambiguous and linguistic inputs for estimating the effort and cost involved in software project. In contrast, the homogeneous data sets were considered for software effort estimation in [7] which results in accurate and reliable software estimates. In this work, the comparative analysis is done by considering both homogeneous and irrelevant or disordered data sets to prove the significance of the ordered set of input data for effort estimates. In reference [8] authors have proposed a novel fuzzy-based framework for managing imprecision and uncertainty problem in effort estimation process.

Similarly, in [9] authors have investigated a hybrid methodology by integrating neuro-fuzzy and SEER-SEM techniques that can effectively functions with various other algorithmic models for effort estimation. Further, the work presented in reference [10], is Enhancing Software Sizing Adjustment Factors which effectively estimates the software efforts by predicting size of the software. Moreover, in [11], authors contribute an enhanced analogy-based approach based on extensive dimension weighting and this method shows experimentally evaluated results for project efforts. Recently, a work suggested by the researchers in [12] proves that the change in standard function points analysis may improve the accuracy in effort estimation by reducing the ambiguity. Recently, in work [13], authors demonstrated that effort estimation can be performed by incorporating soft computing technique to handle uncertainty in input dataset. The main limitation associated with all the aforementioned existing works is that, they all focus of eliminating uncertainty and complexity involved in input data with least concentration with development process involved in effort estimates. Many of the existing research works concentrates on accuracy issue than the other performance related issues like security.

Extract of the Literature

From the literature review conducted, the following shortcomings are identified as below:

- a) Generally, the similarities between these studies focus on the data sets or the initial phase of the estimation but do not concentrate on the development phase of the effort.
- b) Most of the methods use the fuzzy logic to handle imprecision in the data sets but does not focus on performance factors.

These limitations induced us in proposing a Least Effort Multiplier based Fuzzy Estimation Algorithm for effort estimation.

PROPOSED WORK

The proposed Fuzzy Based Alternating Regression Technique (FBART) alternates between two regression techniques viz., i) Gauss-Newton based model and ii) Weighted Deming Model based on the size of the project estimated based on function points and lines of code.

FBART Algorithm and Model

This FBART is an adaptation of FFPA-PSR (fuzzy-based function point analysis with performance metrics, security, and reliability factors) algorithm that has been proposed for improving the accuracy of the software effort estimation. The proposed FBART algorithm uses fuzzy logic to frame rules based on the classification of the attributes like project type (T), programmer's skill (S), software language used (L), database used (D) and criticality(C) and Re-usability (R) essential for estimation. These performance factors are integrated for estimating effort in an accurate and automated way.

Fuzzy Based Alternating Regression Technique Algorithm (FBART)

1. Identify the size of the project based on requirement specification with the aid of functional points and Lines of code (LoC).
2. Estimate effort through regression models (linear and non-linear)
3. If the number of functional points and LoC is low, Estimate the effort based on Gauss-Newton based regression model.
4. Else if, the number of functional points and LoC is maximum, estimate the effort based on Weighted Deming regression model.
5. Categorize the effort by generating fuzzy rule using cross over mechanism.
6. Improve the effort estimation by integrating the impacts factors (T, S, L, D, C, R)
7. Fine tune Value Adjustment Factor (VAF) with respect to the considered least multiplier factors (T, S, L, D, C, R).
8. Manipulate the fuzzy function point through Unadjusted Function Point (UFP) and Value Adjustment Factor (VAF)
9. Determine the performance metrics by estimating the precision value.
10. Modify the precision value based on the estimation.
11. Examine the accuracy of FBART through real time data.
12. The model is implemented if the results are superior with these fuzzy rules. Else, Create new fuzzy rules.

Linear Regression Model for Small Size Projects

In FBART, the small organic project is estimated through a process that takes functional points and Line of Codes (LoC) as inputs. The main goal of the estimation is achieved by fitting a curve to data from actual software projects based on Gauss-Newton based Model.

Gauss-Newton based Model

The method named after the mathematicians Carl Friedrich Gauss and Isaac Newton, solves the non-linear least squares problems. It is a modification of Newton’s method for finding a minimum of a function as it can only be used to minimize a sum of squared function values. The challenging computation of the second derivative is not required. This is applicable in situations where, non-linear least squares problems arise for instance in non-linear regression.

Traditionally, majority of the research and practice portrays that linear relationship exists between functional points, lines of code with effort is linear. Hence from the historical real data sets obtained from the projects, we model a linear regression analysis to fit our model. The equation that depicts a relation function between LoC, functional points with effort is shown through

$$x = g(y) \quad \dots \quad (3.1)$$

where, 'x' is the estimated effort based on 'y' that represents functional points or lines of code.

Similarly, the COCOMO is adjusted based on functional points and LoC by

$$PH = a \times LoC^b \quad \dots \quad (3.2)$$

where, 'PH' is represented for the human resource needed (person hour), "a" and "b" represents parameters of functional points and lines of code. Now, for instance, consider a historical project data as portrayed in Table 3.1 which is for the type of small organic project.

Table 3.1 Historical data sets of small sized projects

| No | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------------------------|----|-----|-----|-----|-----|-----|-----|-----|-----|
| LoC | 47 | 135 | 148 | 179 | 198 | 260 | 298 | 386 | 598 |
| Functional points | 7 | 9 | 14 | 16 | 18 | 19 | 22 | 26 | 31 |
| Person hour | 6 | 12 | 13 | 17 | 23 | 32 | 36 | 43 | 46 |

Now consider, 'x_i' and 'y_i' as the input values of functional points or LoC and effort in person-hours respectively. Then, the Gauss-Newton model is incorporated to compute 'a' and 'b' such that the sum of squares of the residuals represented by equation (3.3) is minimized.

$$M_i = x_i - a \times x_i^b, \text{ Minimize under the iterations, } i=1 \dots 9 \quad (3.3)$$

The incorporation of this linear regression model, after seven iterations of the Gauss-Newton algorithm obtains the optimal values as a=0.3709 and b=0.7547 and the effort of this considered historical data set is estimated as

$$E = 0.3709 \times LoC^{0.7547} \quad \dots \quad (3.4)$$

3.1.4 Linear Regression Model for Large Size Projects

In FBART, the large sized projects are estimated through functional points and Line of Codes (LoC) as inputs similar to small organic projects. But, the main goal of the estimation is achieved by fitting a curve to data from actual

software projects based on Weighted Deming Model.

Weighted Deming Model

Deming regression, named after W. Edwards Deming is an errors-in-variables model which finds the line of best fit for a two-dimensional dataset. It accounts for errors in observations on both the x- and the y- axis, thus differing from the simple linear regression model. It is a special case of total least squares with a more complicated error structure, allowing any number of predictors. It is similar to the maximum likelihood estimation of an errors-in-variables model in which the errors for the two variables are assumed to be independent and normally distributed, and the ratio of their variances (δ) is known.

Generally the research and practice proves that non-linear relationship may also exist between functional points and lines of code. Hence from the historical real data sets obtained from the projects, we model a linear regression analysis to fit our model. As a result both the dependent effort and size variables are transformed to a natural logarithmic scale providing the new variables ln E and ln LoC, functional points respectively.

Finally, the form of weighted Deming regressions is given by

$$\ln E = a_0 + a_1 \ln LoC \quad \dots \quad (3.5)$$

Further, based on the estimated parameters, we construct the regression co-efficient through

$$E = 0.511 \times LoC^{0.653} \quad \dots \quad (3.6)$$

The Integrated Multiplier Model

In order to improve the accuracy in estimation, the factors that have direct influence over the software effort evaluation are considered. Therefore the equation of the proposed alternative fuzzy based regression model is calculated using

$$E_{eff} = E_{op} * F_{lm} \quad \dots \quad (3.7)$$

where

E_{op} –The effort estimation calculated through regression model.

F_{lm} - The multiplier factors that highly influences effort estimation.

Further, 'F_{lm}' is expressed by

$$F_{lm} = \sum_{i=1}^n (W_i * CED_i) * \pi (CED_i) \dots \quad (3.8)$$

where

CED_i- Cost effort drivers.

W_i- Weight of cost drivers estimated by expert judgment. ('i' from 1 to 6)

Thus, CED₁- Software Type

CED₂- Skill of the developer

- CED₃ - Database Used
- CED₄ - Programming language used
- CED₅ - Criticality of the software
- CED₅- Re-usability of the software

Furthermore these six factors are represented by the six tuples <T, P, L, D, C, R>. In addition, the other factors are either included in any one of these five factors or it has negligible impact in the software development process. Finally, the ratings of these factors are designated as 'LEAST SIGNIFICANT', 'TOLERABLE', 'SIGNIFICANT' and 'SENSITIVE'. In order to handle the dependencies and precision of the factors, they are fuzzy analyzed to improve the accuracy. But, this fuzzification is difficult to achieve. Thus, rectangular fuzzy if – then rules are generated to tackle this situation.

The fuzzy rules used in FBART are:

- IF the size of the project is SMALL, complexity of development is MINIMUM and the assigned weight is also LOW, THEN the fuzzy function point is LEAST SIGNIFICANT;
- IF the size of the project is SMALL, complexity of development is MODERATE and the assigned weight is also MODERATE, THEN the fuzzy function point is TOLERABLE;
- IF the size of the project is LARGE, complexity of development is MINIMUM and the assigned weight is also HIGH, THEN the fuzzy function point is SIGNIFICANT;
- IF the size of the project is LARGE, complexity of development is MAXIMUM and the assigned weight is also HIGH, THEN the fuzzy function point is SENSITIVE.

EVALUATION RESULTS

The proposed FBART estimation technique is developed in Java script under windows environment and validated with real project data sets. The effort estimation data of twenty implemented software projects of 2014 is used for testing.

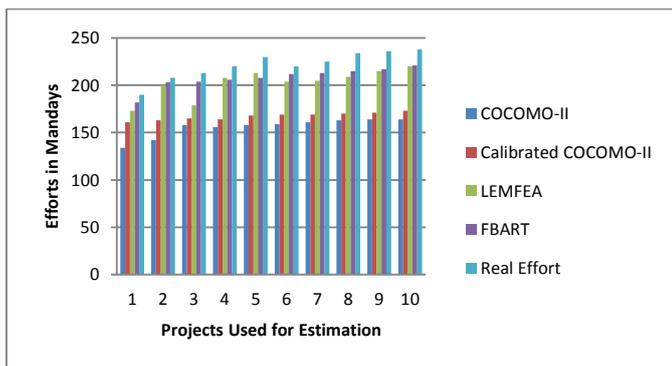


Figure 4.1 Effort Estimation Chart based on Models with FBART

At the same time, the proposed FBART estimation technique is compared with COCOMO-II and Calibrated COCOMO-II [15-16]. The comparative results for estimated efforts are presented in Figure 4.1.

From Figure 4.1, it is transparent that the proposed FBART algorithm estimates effort values that are very closer to the real value estimations. It also infers that the accuracy of the FBART algorithm is high but practically it is proved based on the popularly known performance evaluation metrics like Mean Magnitude Relative Error (MMRE) and Prediction rate (PRED). The Mean Magnitude Relative Error (MMRE) and Prediction rate (PRED) are calculated based on the ratio of actual number of observations to the number of estimates generated by the model as presented through Figure 4.2 and Figure 4.3 respectively.

From Figure 4.2, it is evident that the PRED value of the proposed FBART technique is comparatively low when compared to the benchmark estimation techniques like COCOMO-II, Calibrated COCOMO-II

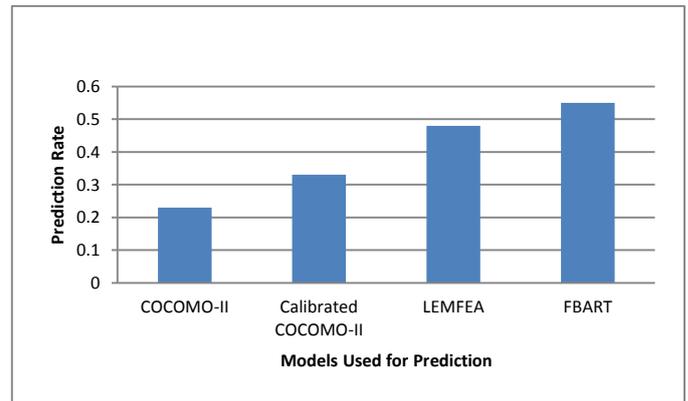


Figure 4.2 Effort Estimation Chart based on PRED

From Figure 4.3, it is evident that the MMRE value of the proposed FBART technique is comparatively low when compared to the benchmark estimation techniques like COCOMO-II and Calibrated COCOMO-II.

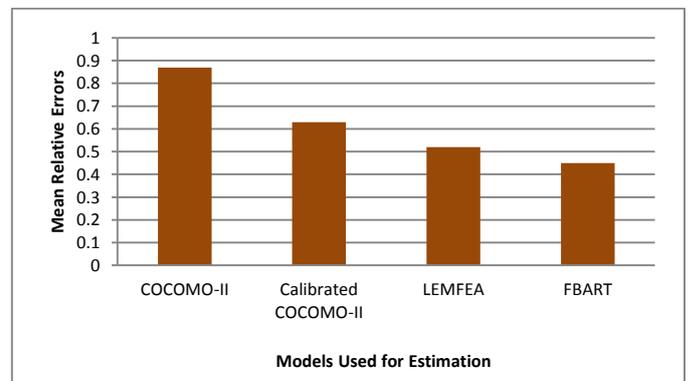


Figure 4.3 Effort Estimation Chart based on MMRE

Likewise, the proposed estimation technique is compared with the existing models in terms of accuracy that predicts a lower MMRE value and higher prediction value. This analysis based

on accuracy is depicted through Figure 4.4.

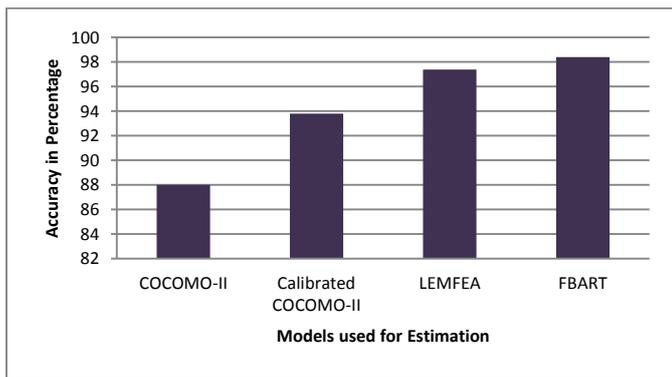


Figure 4.4 Effort estimation chart based on Accuracy

Furthermore the estimation models are incorporated for effort estimation either through training or through testing. Figure 4.5 presents the comparative analysis of estimation through training and testing respectively. It also portrays that the proposed FBART technique is comparatively minimum in terms of MSE, MAE and RMSE but possesses high VAF than the compared benchmark estimation techniques like COCOMO-II and Calibrated COCOMO-II.

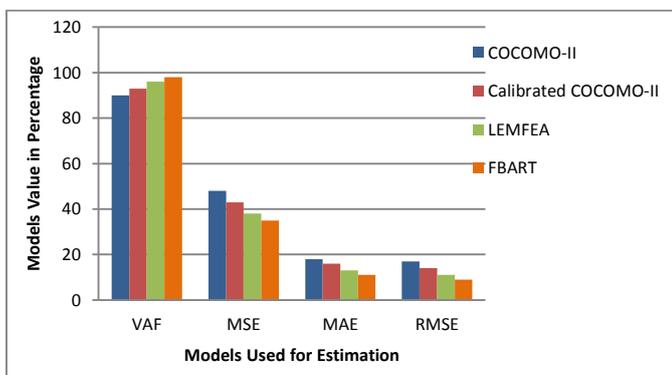


Figure 4.5 Effort Estimation Chart based on VAF, MSE, MAE and RMSE for FBART

In addition to the proposed estimation model is also validated through chi-square test and the evaluation reveals that LEMFEA, COCOMO II model and the calibrated COCOMO II model as worst fit as presented in Table 4.

Finally Table 4.2 shows the performance comparison based on percentage of improved accuracy. Besides the choice of factors like project type (T), programmer’s skill (S), software language (L), database used (D) and criticality(C) of the software has greater influence on the accuracy of the proposed FBART estimation model.

It also improves the accuracy of estimation by an average of 9.34 % than the function point model like LEMFEA, COCOMO-II and Calibrated COCOMO-II, while it improves the accuracy by 12.33% than the existing functional point’s model. Thus, it is inferred that the proposed FBART

estimation model is superior in estimating the efforts than any other functional point analysis model available in the literature.

Table 4.2 Performance Comparison based on percentage of improved

| Model | Accuracy (in %) | Improved accuracy (in %) |
|-----------------------------|-----------------|--------------------------|
| COCOMO II | 85.12 | 6.72 |
| Calibrated COCOMO II | 95.67 | 9.34 |
| Proposed (LEMFEA) | 97.65 | 12.33 |
| FBART | 98.35 | 14.12 |

CONCLUSION

In this Chapter FBART estimates the effort involved in software project in an accurate and reliable way. The proposed hybrid model is evaluated by means of integrated datasets with ten different project domains. The proposed hybrid model shows better results than the existing models in terms of accuracy in effort estimation, reliable and secures estimation factors which may be utilized in highly critical applications. This model is highly efficient to handle uncertainty input data set for effort estimation which is obtained due to incomplete requirement information for software project. The proposed model efficiently estimates the effort within the time and budget frame which is one of the much needed requirements of the software organizations. Our future plan of research is to extent this work for assessing risks involved in software project development.

REFERENCES

- [1] E.Karunakaran and N.Sreenath, “A Method to Effort Estimation for XP Projects in Clients Perspective”, International Journal of Applied Engineering Research, ISSN-0973-4562, Vol.10,No.7, 2015, pp.18529-18550.
- [2] Sommerville Ian, “Software Engineering”, Prentice Hall PTR Upper Saddle River, ISBN:0321210263, 2004.
- [3] E.Karunakaran and N.Sreenath, “An Accurate Effort Estimation using Least Effort Multipliers based on Fuzzy Estimation Algorithm for Software Development ”, International Journal of Applied Engineering Research, ISSN-0973-4562, 2015.
- [4] Thomas Cutting, “Estimating Lessons Learned in Project Management - Traditional”, Retrieved on May 28, 2010.
- [5] Molokken, K. and Jorgensen, M, “A review of software surveys on software effort estimation. In Empirical Software Engineering”, Proceedings of

- International Symposium on Empirical Software Engineering, pp. 223-230, ISBN: 0-7695-2002-2, 2003.
- [6] Verner, J. and Tate, G., "A Software Size Model", IEEE Transaction on Software Eng., pp. 265-278, vol. 18, no. 4, ISSN : 0098-5589, 2003.
- [7] Karpowich, M. F., Sanders, T. R., and Verge, R. E., "An Economic Analysis Model for Determining the Custom versus Commercial Software Trade-off". Analytical Methods in Software Engineering Economics, pp. 237-252, ISSN: 978-3-642-77797-4, 1993.
- [8] Abts, C. M., and Boehm, B., COCOTS (Constructive COTS), "Software Integration Cost Model: An Overview". USC Centre for Software Engineering, University of Southern California, briefing, ISSN: 3540455884, 1998.
- [9] Stutzke, R. "Costs impact of COTS volatility", Proceedings of Workshop on COCOMO, Vol. 2, 1995.
- [10] Ellis, T. COTS, "Integration in Software Solutions-A Cost Model, 1995", Proceedings of the NCOSE International Symposium Systems Engineering in the Global Marketplace, pp. 174-182, Vol.5 (1), 2014.
- [11] Albrecht, A. J., and Gaffney Jr, J. E., "Software function, source lines of code, and development effort prediction" A software science validation Software Engineering, IEEE Transactions on, pp. 639-648, Vol.9 (6), ISSN: 0098-5589, 1983.
- [12] Aoyama, M., "A component-based software development methodology", Proceeding of IPSJ SIG Notes, pp. 33-40, Vol.96 (84), 1993.
- [13] Aoyama, M., "Process and economic model of component-based software development." A study from software CALS next generation software engineering program, Proceedings Fifth International Symposium on Assessment of Software Tools and Technologies, pp.100-103, ISBN: 0-8186-7940-9, 1997.
- [14] Hakuta, M., Tone, F., and Ohminami, M. "A Software Size Estimation Model and Its Evaluation" J. Systems and Software, pp. 253-263, vol. 37(3), doi:10.1016/S0164-1212(96)00020-9, 1997.
- [15] Yakimovich, D., Bieman, J. M., and Basili, V. R "Software architecture classification for estimating the cost of COTS integration. In Software Engineering", Proceedings of the 1999 International Conference on Software engineering, pp. 296-302, ISBN:1-58113-074-0, 1999.
- [16] Dolado, J. J. "A validation of the component-based method for software size estimation. Software Engineering", IEEE Transactions on Software Engineering, pp. 1006-1021, Vol. 26(10), ISSN: 0098-5589, 2000.
- [17] Abts, C., Boehm, B. W., and Clark, E. B. COCOTS, "A COTS software integration lifecycle cost model-model overview and preliminary data collection findings", In the proceeding of the ESCOM-SCOPE Conference, pp. 18-20, 2000.
- [18] Abts, C., Boehm, B., and Clark, E. B "Empirical observations on COTS software integration effort based on the initial COCOTS calibration database" ICSE COTS Workshop, 2000.
- [19] Abts, C. M. "Extending the COCOMO II software cost model to estimate effort and schedule for software systems using commercial-off-the-shelf (cots) software components: the COCOTS model", ACM Digital Library", 2004.
- [20] Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., and Selby, R., "Cost models for future software life cycle processes: COCOMO 2.0", Annals of software engineering, pp. 57-94, Vol.1(1), ISSN: 1022-7091, 1995.
- [21] Banker, R., H. Change, and C. Kemerer, "Evidence on Economies of Scale in Software Development," Information and Software Technology, vol. 36, no. 5, 1994, pp.275-828.
- [22] Shepperd, M. and Schofield, M., "Estimating Software Project Effort Using Analogies", IEEE Transactions on Software Engineering, pp. 736-743, vol. 23, no. 12, 1994.
- [23] Christopher M. Fraser , "Neural Networks: A Review from a Statistical Perspective", Hayward Statistics (2000).
- [24] Waegeman, Willem; De Baets, Bernard; Boullart, Luc "ROC analysis in ordinal regression learning". Pattern Recognition Letters 29:, pp. 1-9., 2008.
- [25] Gray, A., and MacDonell, S, " Applications of fuzzy logic to software metric models for development effort estimation", Fuzzy Information Processing Society, 1997. NAFIPS'97, 1997 Annual Meeting of the North American pp. 394-399, ISSN: 0-7803-4078-7, 1997, September
- [26] Martin, C. L., Pasquier, J. L., Yanez, C. M., and Tornes, A. G., "Software development effort estimation using fuzzy logic: a case study", Computer Science, 2005. ENC 2005. Sixth Mexican International Conference, pp. 113-112, ISBN: 10.7763, 2005, September.
- [27] Ahmed, M. A., Saliu, M. O., and AlGhamdi, J., "Adaptive fuzzy logic-based framework for software development effort prediction", Information and Software Technology, pp.31-48, Vol.47(1), 2005
- [28] Gray, A. R., and MacDonell, S. G., "A comparison of techniques for developing predictive models of software metrics", Information and software technology, pp.425-437, Vol.39(6), 1997.

- [29] Smith, A. E., and Mason, A. K. , “Cost estimation predictive modelling: Regression versus neural network” The Engineering Economist, pp.137-161, Vol. 42(2), 1997.
- [30] Munson, J. C., and Khoshgoftaar, T. M., “ Regression modelling of software quality: empirical investigation”, Information and Software Technology, pp. 106-114, Vol.32(2), 1990.
- [31] Khoshgoftaar, T. M., and Allen, E. B., “ Logistic regression modelling of software quality”, International Journal of Reliability, Quality and Safety Engineering, pp. 303-317, Vol.6(04), 1999.