Volume 6, Issue 9, September 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Novel Approach Based on Bayesian Multi-Scale Optimization for Software Cost Estimation

D.T.R. Subba Reddy
Professor, Dept. of CSE,
SMGIG, Guntur, AP, India

Dr.B.R. Srinivas
Professor, Dept. of CSE,
SMGIG, Guntur, AP, India

Shaik Subhani
Assist. Professor, Dept. of CSE,
SMGIG, Guntur, AP, India

Abstract— Software Cost Estimation is very important challenging task for completing the project successfully. The estimation in software development depends on various factors particularly managing project cost, time and quality and effort factors. Therefore, accurate assessment is a consequential factor in projects success and reducing the risks. In the last two decades, many researchers and practitioners presented statistical and machine learning-based models for software effort estimation. In this paper, a novel approach based Bayesian multi-class algorithm is proposed for software cost estimation. It helps project manager to provide nimble and realistic estimate for the project effort and development time that in turn gives software cost. The proposed work is carried in two steps, in first phase known as training phase, optimizing the parameters and second step known as validating phase, the prediction process. The Parameters SSE, RME, MAD and $R^2$are calculated for COCOMO-II dataset. Statistical results show that our method could significantly improve accuracy, error minimization and has potential to become an effective method for software cost estimation.

Keywords— Software cost, Software quality, Bayesian Optimization, Gaussian, JMP, COCOMO-II.

I. INTRODUCTION

Software development has become an essential activity for many modern organizations. In fact the quality, cost, and timeliness of developed software are often crucial elements of an organization’s success. Effort assessment of software development has been a critical task for software engineering community. The global software market has grown exponentially over the past decade. Significantly, the cost of developing software has grown. The main idea behind increasing trend in the software costs is the employment intensive nature of the software development process. To manage software projects effectively, it is important to have accurate assessments of cost and effort involved in software development. The number of project deficiencies and the cases of cost and schedule overwhelm have been a significant issue for software project managers. Poor estimates not only led projects to exceed budget and go overscheduled but also in many cases to be terminated entirely. Software cost assessment is the set of techniques and procedures that organizations use to arrive at an estimate for proposal request, project planning and probability estimates. Accurate assessment means better planning and proficient use of project resources such as cost, duration and effort requirements for software projects. Efficient software development effort estimation is one of the most demanding tasks in software industry. Unfortunately software industry suffers from the crisis of inaccurate estimate for projects and in many cases inability to set exact release date, leads to low quality of delivered product.

Software cost estimation is most critical task in managing and completing the software projects successfully. Development costs leads to increase with project complexity and hence accurate cost estimates are highly desired during the early stages of development. A major problem of the software cost estimation is first obtaining an accurate size estimate of the software to be developed. An important objective of the software engineering community is to develop useful models that constructively give the details of software development life cycle and accurately estimate the cost of software development. In order to develop software effectively in competitive and complex environment many organizations use software metrics as a part of their project development. In this paper, a novel approach based Bayesian multi-class algorithm is proposed for software cost estimation. It helps project manager to provide nimble and realistic estimate for the project effort and expansion time that in turn gives software cost.

The paper is organized in following manner: section 1 describes introduction, sections 2 and 3 presents related work and software cost estimation process. Proposed work, results and discussion is described in section4 and 5. Section 6 ends the paper with a conclusion.

II. RELATED WORK

The most frequently used methods for predicting software development effort have been based on linear-least-squares regression such as COCOMO. As such, the models have been extremely susceptible to local variations in data points. In addition, the models have failed to deal with implicit nonlinearities and interactions between the characteristics of the project and effort. In recent years, a number of alternative modeling techniques have been proposed and they consist of artificial neural networks, analogy based reasoning, regression trees and rule induction models. Gray and MacDonnell applied fuzzy logic to software metric models for enhancement effort estimation. They outlined the use of fuzzy logic for define software metrics as linguistic variables and for modeling processes. They made comparison of results obtained from a basic fuzzy inference system with other techniques such as linear regression and neural network techniques and found that it outperformed them well. He also described a simulation-based study on the performance of these experiential modeling techniques using size and effort software metric dataset and observed that M-estimation regression outperformed of every parametric and non-parametric techniques. Xu and Khoshgoftaar presented an innovative fuzzy identification software cost estimation modeling technique, which is an advanced fuzzy logic technique that integrates fuzzy clustering, space projection, fuzzy inference and defuzzification. Based on their case study on the COCOMO'81 database it was observed that the fuzzy recognition model provided significantly better cost estimations.

Many of Machine Learning (ML) techniques in the literature have been applied to progress the software effort estimation. ML optimization algorithms that are encouraged from nature have received much awareness to find more truthful estimation for software effort. Nature-encouraged ML algorithms include Cuckoo Search, Particle Swarm Optimization (PSO), Bat Algorithm, Firefly Algorithm, and many others. The following are Research papers reviewed for this study.

- A hybrid approach was developed by Oliveira, A.L.I., Braga, P.L., Lima, R.M.F. and Cornélio, M.L. for parameter selection and model optimization. The authors make use of Genetic Algorithms (GA) for optimizing a Support Vector Regression model. The authors define the impact of using GA in attribute selection and parameter optimization of the effort estimation model. The results of their approach show that GA is applicable to progress the performance of the SVR model compared to other approaches.
- Software cost estimation is the vital step to start any project. It gives us the outline of effort, resources and time required for a project. Accomplishment of software enhancement depends on cost estimation. There are various budget assessment techniques to compute cost of the development and Function point analysis (FPA) is the technique of calculating the dimension of software. The benefit is that it can avoid source code error when selecting dissimilar programming languages. Shivani Sharma, Aman Kaushik, AbhishekTomar proposed a model for computing budget of project based on Top down method. The whole process will be done by Ant colony optimization algorithm. To compare and evaluate the outcomes of the proposed algorithm with K Modes algorithm and RF model and it has been noticed that when we have compared with K modes and RF model then proposed work gives better results.
- Software development effort estimation is treated a fundamental task for software development life cycle as well as for managing project cost, time and quality. In recent years, software effort estimation has received a considerable amount of concentration from researchers and became a challenge for software industry. Nazeeh Ghatasheh, HossamFaris, Ibrahim Aljarah, Rizik M. H. Al-Sayyed, Firefly Algorithm is proposed as a meta-heuristic optimization technique for optimizing the parameters of three COCOMO models. These models include the basic COCOMO model and other two models proposed in the literature as extensions of the fundamental COCOMO model. The developed estimation models are evaluated using diverse evaluation metrics. Experimental outcome show high accuracy and significant error minimization of Firefly Algorithm over other meta-heuristic optimization algorithms including Genetic Algorithms and Particle Swarm Optimization.
- In most of the Software Cost Estimation (SCE) techniques, algorithmic models such as COCOMO algorithm models have been used. COCOMO model is not able to estimating the close approximations to the genuine cost, because it runs in the type of linear. So, the models should be adapted that concurrently with the number of Lines of Code (LOC) has the ability to estimation in a decent and accurate fashion for effort factors. Meta-heuristic algorithms can be a good model for Software Cost Estimation due to the ability of local and global search. MajidAhadi, Ahmad Jafarian introduced a hybrid of Particle Swarm Optimization (PSO) and Differential Evolution (DE) algorithms for the SCE. Experiment results on NASA60 software dataset shows the rate of Mean Magnitude of Relative Error (MMRE) error on hybrid model, in assessment with COCOMO model is reduced to about 9.55%.
- Software cost estimation is one of the majority demanding tasks in software engineering. Over the past years the estimators have used parametric cost assessment models to establish software cost, however the challenges to accurate cost estimation keep evolving with the advancing technology. Gaurav Kumar, Pradeep Kumar Bhatia uses Back-Propagation neural networks for software cost estimation. Neural Network has been proposed that takes KLOC of the project as input, uses COCOMO model parameters and gives gust as output. Artificial Neural Network represents a complex set of relationship between the effort and the cost drivers and is a potential device for estimation.
- The process of estimating time and cost necessary for developing software is called software cost estimation. It is one of the steps to be carried out in project planning. Early software estimation models are based on regression assessment or mathematical derivations. Today's models are based on simulation, neural network, genetic algorithm, soft computing, fuzzy logic modeling etc. Ravishankar. S P. Latha utilized an adaptive fuzzy logic model to improve the truth of software time and cost estimation. Using advantages of fuzzy set and fuzzy logic can produce accurate software attributes which result in accurate software estimates. 63 Historic projects of NASA dataset having COCOMO design is used in the assessment of the proposed Fuzzy Logic COCOMO II. Eight membership functions available in fuzzy logic are used and a comparison is prepared to find out which membership function yields enhanced result in terms of Mean Magnitude of Relative Error (MMRE) and PRED (25%).

- One of the most significant effective factors the software companies face is the Software Cost Estimation (SCE) in software improvement process. SCE is one of the subjects which have been measured in late decades in many researches. The genuine estimation in software progress needs effort and cost factors which are done by use of the algorithmic and Artificial Intelligence models. Boehm used the COCOMO model which is an algorithmic model in 1981 for SCE. The little accuracy and non-reliable structures of the algorithmic models led to high risks of software projects. So, it is desired to estimate the cost of the project yearly and compare it to the other techniques. Isa Maleki, Ali Ghaffari, and Mohammad Masdari, the Meta-Heuristic algorithms have been developed well recently in software fields and SCE. Meta-heuristic, Genetic Algorithms (GA) and Ant Colony Optimization (ACO) resolve the optimization problems are very competent in optimizing the algorithmic models and the effectual factors in cost estimation.

III. SOFTWARE COST ESTIMATION PROCESS

Cost estimation is a prediction method to get close result of necessary cost. It includes the process of considering the necessary cost, experiences, time constraints, risks, schedules, resources and other elements related to the expansion of a project. Hence, cost estimation is vital in managing a project mainly to the project manager when proposing budget for assured project. In software development a broadly used term is “software project estimation” where its task is to calculate the estimation procedure. Cost estimation is the determination of abundance and predicting or forecasting within a clear scope of the costs requisite to construct and equip a facility to constructer goods or to furnish a service. These costs are included assessments and a valuation of risks and uncertainties. A cost estimation process considers and determines utilized experience by professional, calculating and forecasting the future cost of resources and schedule for any project development. It provides input to unique baselines and changes the baselines against cost comparisons right through a project. It is performed at a sure point based on the obtainable information at a certain time. Normally, it includes cost estimation details, a cost estimation review, basis of estimation which describes the project information, estimation methodologies, type of cost estimation together with risk, cost driven, cost adjustment and so on.

In software life cycle, there are many decision situations involving limited resources in which software engineering techniques offer useful assistance. To provide a feel for the environment of these economic decision issues, the following example is given below for the major phases in the software life cycle.

- **Feasibility Phase:** How much should one spend in information system analyses (user questionnaires interviews, current-system analysis, workload characterizations, simulations, and scenarios, prototypes) in order to obtain convergence on an appropriate definition and concept of operation for the system to be implemented?
- **Plans and Requirements Phase:** How rigorously should necessities to be specified? How much should be invested in requirements validation activities (automated completeness, consistency, traceability checks, analytic models, simulations, prototypes) before proceeding to design and develop a software system?
- **Product Design Phase:** Should developers arrange software to create possible to use a complex piece of existing software which usually but not totally meets requirements?
- **Programming Phase:** Given a selection between three data storage and recovery schemes which are primarily execution time-efficient, storage-efficient, and easy-to-modify, respectively; which of these should be implemented?
- **Integration and Test Phase:** How much testing and formal verification should be performed on a product before releasing it to users?
- **Maintenance Phase:** Given an extensive list of suggested product improvements, which ones should be implemented first?
- **Phase-out:** Given an aging, hard-to-modify software product, should it be replaced with a new product, should it be restructured, or should it be left alone?

The actual cost estimation process involves seven steps:

- Set up cost-estimating objectives.
- Generate a project plan for required data and assets.
- Pin down software necessities.
- Work out as much detail about the software system as realistic.
- Use several independent cost assessment techniques to capitalize on their combined strengths
- Compare different estimates and iterate the estimation process.
- Once the project has started, monitor its actual cost and progress, and feedback results to project management

For an effective management accurate assessment of various measures is a must. With accurate estimation managers can manage and manage the project more efficiently and effectively. Project assessment may involve the following:

- **Software size estimation:** Software size may be estimated either in terms of KLOC (Kilo Line of Code) or by calculating number of function points in the software.
- **Effort estimation:** The managers estimate efforts in terms of personnel requirement and man-hour required to construct the software.
- **Time estimation:** Once size and efforts are estimated, the time required to construct the software can be estimated. The sum of time required to complete all tasks in hours or days is the total time invested to complete the project.

- **Cost estimation:** This might be considered as the most difficult of all because it depends on more elements than any of the previous ones. For estimating project cost, it is required to consider - Size of software, Software quality, Hardware, Additional software or tools, licenses etc., Skilled personnel with task-specific skills, Travel involved, Communication, Training and support.

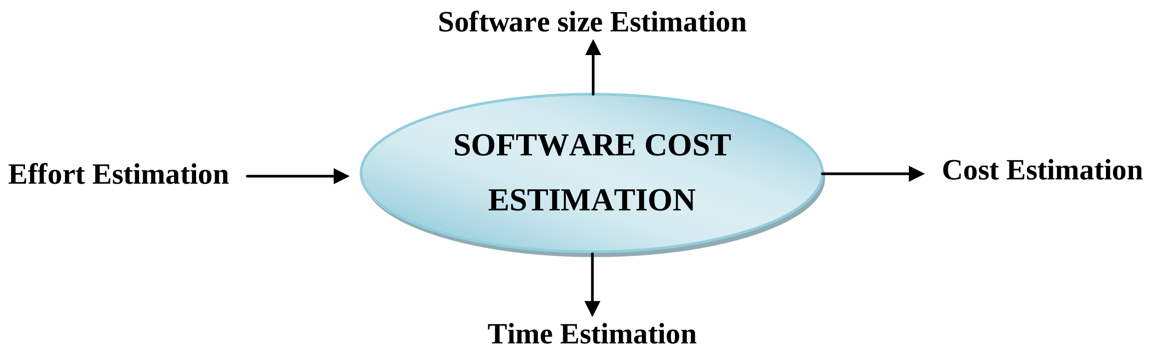

Fig 3.1 Software Cost Estimation Process

IV. PROPOSED WORK

The Bayesian optimization construction has two key factors. The first factor is a probabilistic model, which consists of a prior distribution that confiscation our beliefs about the performance of the unknown objective function and a scrutiny model that describes the data generation mechanism. The second factor is a loss function that describes how optimal sequences of queries are; in practice, these loss functions often take the form of regret, either simple or cumulative. Ideally, the predictable loss is then minimized to select an optimal sequence of queries. After observing the result of each query of the objective, the prior is updated to produce a more informative posterior distribution over the space of objective functions. The proposed algorithm is called the Bayesian optimization algorithm (BOA). The permutation of prior information and the set of promising solutions is used to estimate the distribution. Prior information about the structure of a problem as well as the information represented by the set of high-quality solutions can be incorporated into the algorithm. The ratio between the prior information and the information acquired during the run used to generate new solutions can be controlled. The pseudo-code of the BOA follows:

(1) Set i ← 0 randomly generate initial population F (0)
(2) Select a set of promising strings G (i) from F (i)
(3) construct the network A using a chosen metric and constraints
(4) generate a set of new strings H (t) according to the joint distribution encoded by A
(5) Create a new population F (t+1) by replacing some strings from F (t) with H (t) set i ← i + 1
(6) If the termination criteria are not met, go to (2)

Bayes methods can be trained very efficiently and Bayes' theorem is used for calculating conditional probabilities. The conditional Probability that an event 'γ' will occur, given that 'δ' has occurred is denoted by the symbol P (A/B) and is defined as by

$$P(\gamma/\delta) = \frac{p(\gamma \cap \delta)}{p(\delta)} \; ; P(\delta) > 0 \quad (1)$$

The Probability of the instantaneous occurrence of two events 'C' and 'D' is equal to the product of probability of 'C' and the conditional Probability of 'D' on the assumption that 'γ' occurred.

i.e. $P(C \cap D) = P(C).P(C/D) = P(C).P(C/D)$ (2)

From (1) & (2) $P(C/D) = P(C).P(C/D)/P(D)$. (3)

If $(C \cap D)$ and $(C \cap D')$ are mutually exclusive events then by axiomatic definition

$P(C) = P(D).P(C/D) + P(D').P(C/D')$ (4)

From (3) & (4) $$P(C/D) = \frac{P(C).P(\gamma/D)}{P(D).P(C/D) + P(D').P(C/D')} \quad (5)$$

If D_1, D_2, D_3..................... D_n are ' n' mutually exclusive events of which one of the event occur then,

$P(C) = \sum_{i=1}^{i=n} P(Di).P(C/Di)$

From (3) $$P(C/D) = \frac{P(C).P(D/C)}{\sum_{i=1}^{i=n} P(Di).P(\frac{C}{Di})} \quad (6)$$

Bayes' theorem provides a way of calculating the posterior probability, $P(C/D)$ from P(D), P(D /CD) and P(C).

The main advantage of Bayes optimization is that it only craves a small amount of training data to estimate the parameters necessary for classification. It also show high accuracy and speed when applied to large databases. The step-by-step procedures used for Software cost estimation based on Bayesian optimization algorithm are:

Algorithm: Pseudo code of Bayesian optimization

Task: Software cost estimation based on Bayesian optimization algorithm
Parameter: COCOMO-II data set and Bayesian classifier.

Methodology:
1: i = 1
2: while samples available do

3: Update the expressions for the sufficient statistics of the posterior distribution using the available data D_1:i.
4: Find x_i+1 by minimizing the expected deviation from the maximum. The expectation is taken with respect
To the posterior distribution P(fjD1:n).
5: Calculate the objective function, g(x_i+1)
6: Enhance the data G_1: i+1 = g (G1: i; (y_i+1; g (y_i+1)) h
7: i = i + 1
8: end while

Output: Statistical results show that our method could significantly improve accuracy, error minimization and has potential to become an
Effective method for software cost estimation.

V. RESULTS AND DISCUSSIONS

In this paper, the implementation of Bayesian multi-class algorithm for software cost estimation is done using COCOMO-II dataset. A generalization of the R^2 evaluates the regular Rsquare for continuous responses. RMSE, Gives the root mean square error. When the response is nominal or ordinal, the differences are between 1 and p. Mean Abs Dev, The average of the absolute values of the differences between the response and the predicted response. SSE, Gives the error sums of squares. Available only when the response is continuous. Sum Freq, Gives the number of observations that are used. If we precise a Freq variable in the neural launch window, Sum Freq gives the sum of the frequency column. A model report is created for every neural network model is shown in table5.1. Measures of fit appear for the training and validation sets and table5.2 shows sample COCOMO-II dataset.

Table5.1: A model report with training and validation sets

Bayesian Optimization Algorithm			
S.No	**Parameters**	**Training Phase**	**Validation Phase**
1	RMSE	0.16126	0.20164
2	SSE	1.09221	0.85382
3	MAD	0.12284	0.15098
4	SF	42	21
5	R2	0.23560	0.02431

Table5.2: A Sample COCOMO-II dataset

rely	data	cplx	time	stor	virt	turn	acap	aexp	pcap	vexp	lexp	modp	tool	sced	loc	actual
0.88	1.16	0.7	1	1.06	1.15	1.07	1.19	1.13	1.17	1.1	1	1.24	1.1	1.04	113	2040
0.88	1.16	0.85	1	1.06	1	1.07	1	0.91	1	0.9	0.95	1.1	1	1	293	1600
1	1.16	0.85	1	1	0.87	0.94	0.86	0.82	0.86	0.9	0.95	0.91	0.91	1	132	243
0.75	1.16	0.7	1	1	0.87	1	1.19	0.91	1.42	1	0.95	1.24	1	1.04	60	240
0.88	0.94	1	1	1	0.87	1	1	1	0.86	0.9	0.95	1.24	1	1	16	33
0.75	1	0.85	1	1.21	1	1	1.46	1	1.42	0.9	0.95	1.24	1.1	1	4	43
0.75	1	1	1	1	0.87	0.87	1	1	1	0.9	0.95	0.91	0.91	1	6.9	8
1.15	0.94	1.3	1.66	1.56	1.3	1	0.71	0.91	1	1.21	1.14	1.1	1.1	1.08	22	1075
1.15	0.94	1.3	1.3	1.21	1.15	1	0.86	1	0.86	1.1	1.07	0.91	1	1	30	423
1.4	0.94	1.3	1.11	1.56	1	1.07	0.86	0.82	0.86	0.9	1	1	1	1	29	321
1.4	0.94	1.3	1.11	1.56	1	1.07	0.86	0.82	0.86	0.9	1	1	1	1	32	218
1.15	0.94	1.3	1.11	1.06	1	1	0.86	0.82	0.86	1	0.95	0.91	1	1.08	37	201
1.15	0.94	1.3	1.11	1.06	1.15	1	0.71	1	0.7	1.1	1	0.82	1	1	25	79
1.15	0.94	1.65	1.3	1.56	1.15	1	0.86	1	0.7	1.1	1.07	1.1	1.24	1.23	3	60
1.4	0.94	1.3	1.3	1.06	1.15	0.87	0.86	1.13	0.86	1.21	1.14	0.91	1	1.23	3.9	61
1.4	1	1.3	1.3	1.56	1	0.87	0.86	1	0.86	1	1	1	1	1	6.1	40
1.4	1	1.3	1.3	1.56	1	0.87	0.86	0.82	0.86	1	1	1	1	1	3.6	9
1.15	1.16	1.15	1.3	1.21	1	1.07	0.86	1	1	1	1	1.24	1.1	1.08	320	11400
1.15	1.08	1	1.11	1.21	0.87	0.94	0.71	0.91	1	1	1	0.91	0.91	1	1150	6600

The following 5.1 shows Plots the actual versus the predicted response, Available only for continuous responses. The predicted value for the observations in each leaf is the average response. The plot is divided into three sections, corresponding to the three leafs. These predicted values are shown on the plot with black lines. The points are put into random horizontal positions in each section. The vertical position is based on the response produces a plot of actual values by predicted values. This is for continuous responses only.

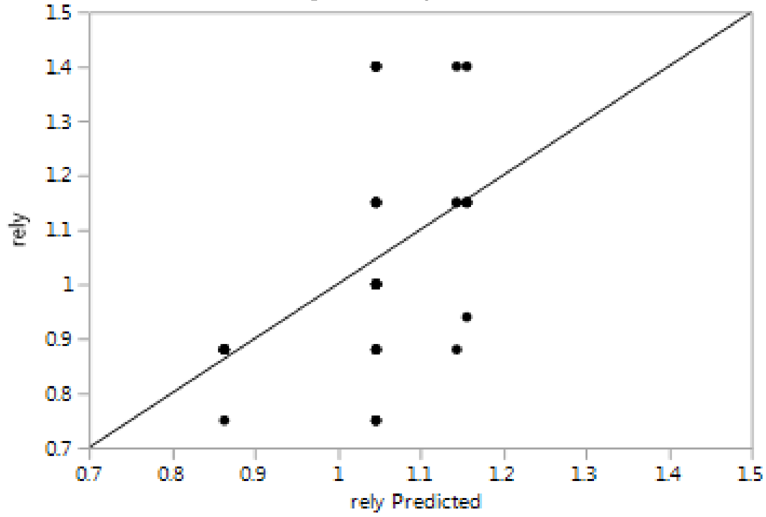

Fig 5.1(a) actual versus the predicted response

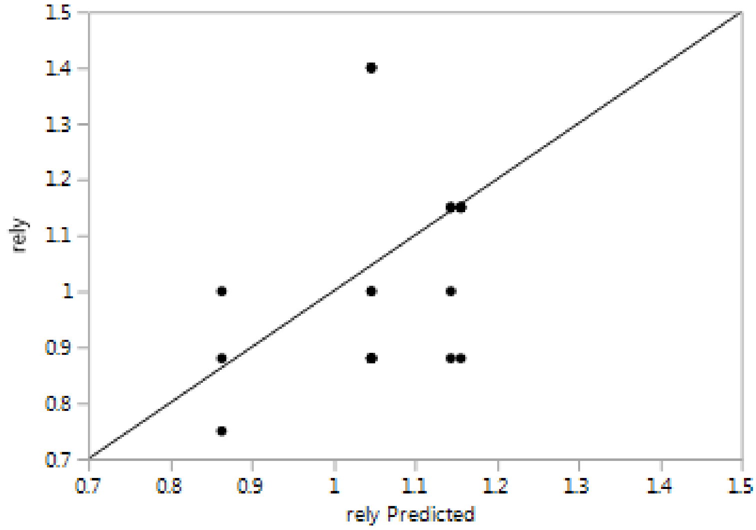

Fig 5.1(b) actual versus the predicted response

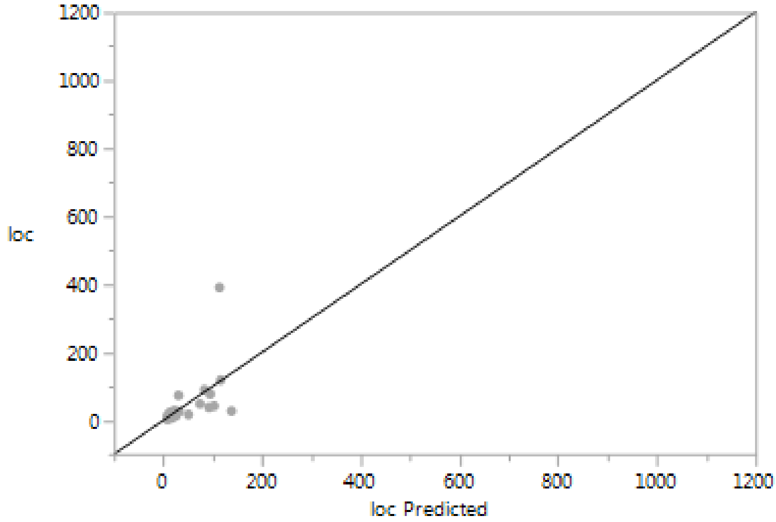

Fig 5.1(b) actual versus the predicted response

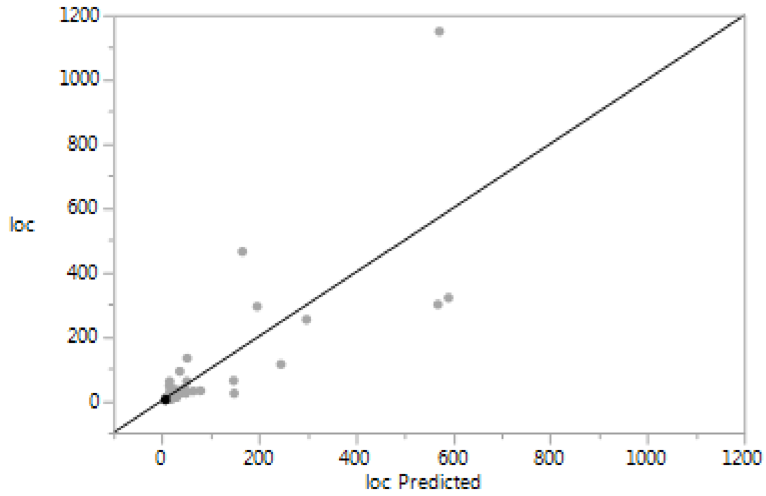

Fig 5.1(b) actual versus the predicted response

The following figure 5.2 shows Plots the residuals versus the predicted responses. Available only for continuous responses. Training

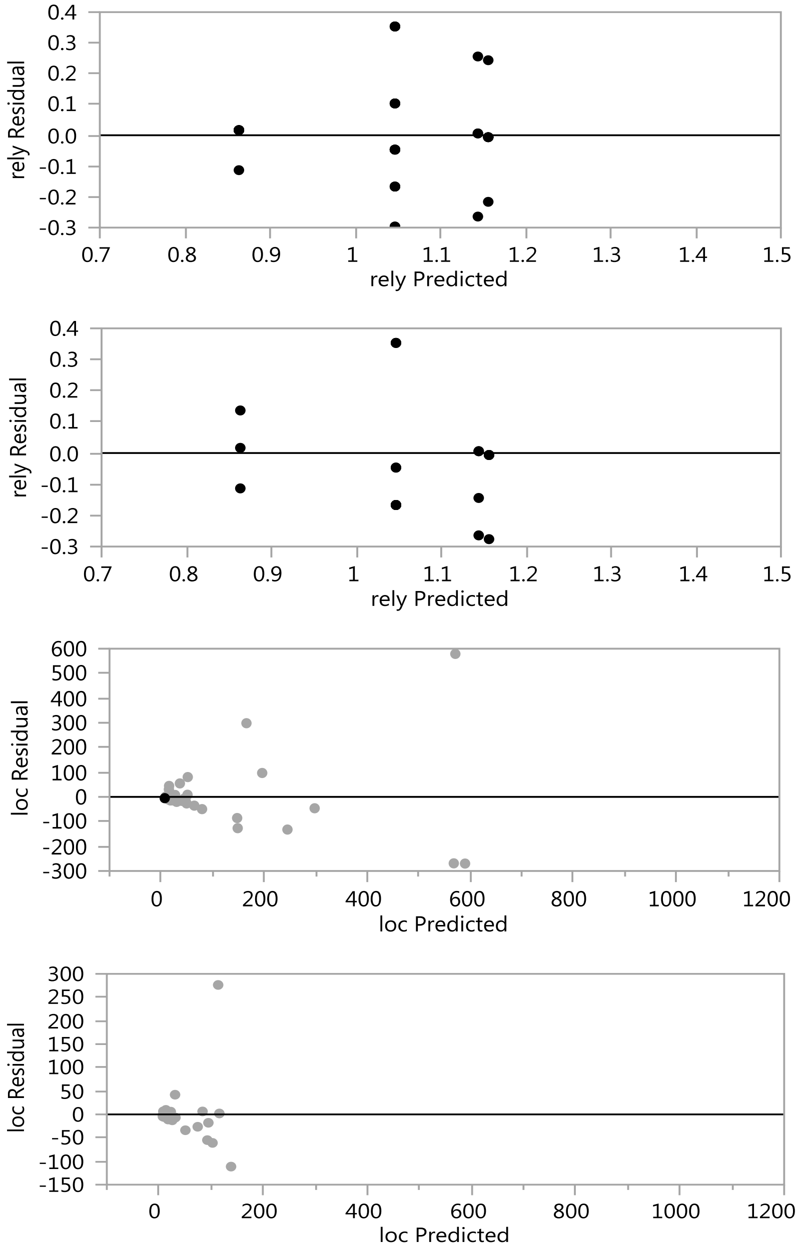

Fig 5.2 Residuals versus the predicted response.

For nominal or ordinal responses, each response level is represented by a separate row in the Prediction Profiler. The following figure 5.3 shows prediction profiler

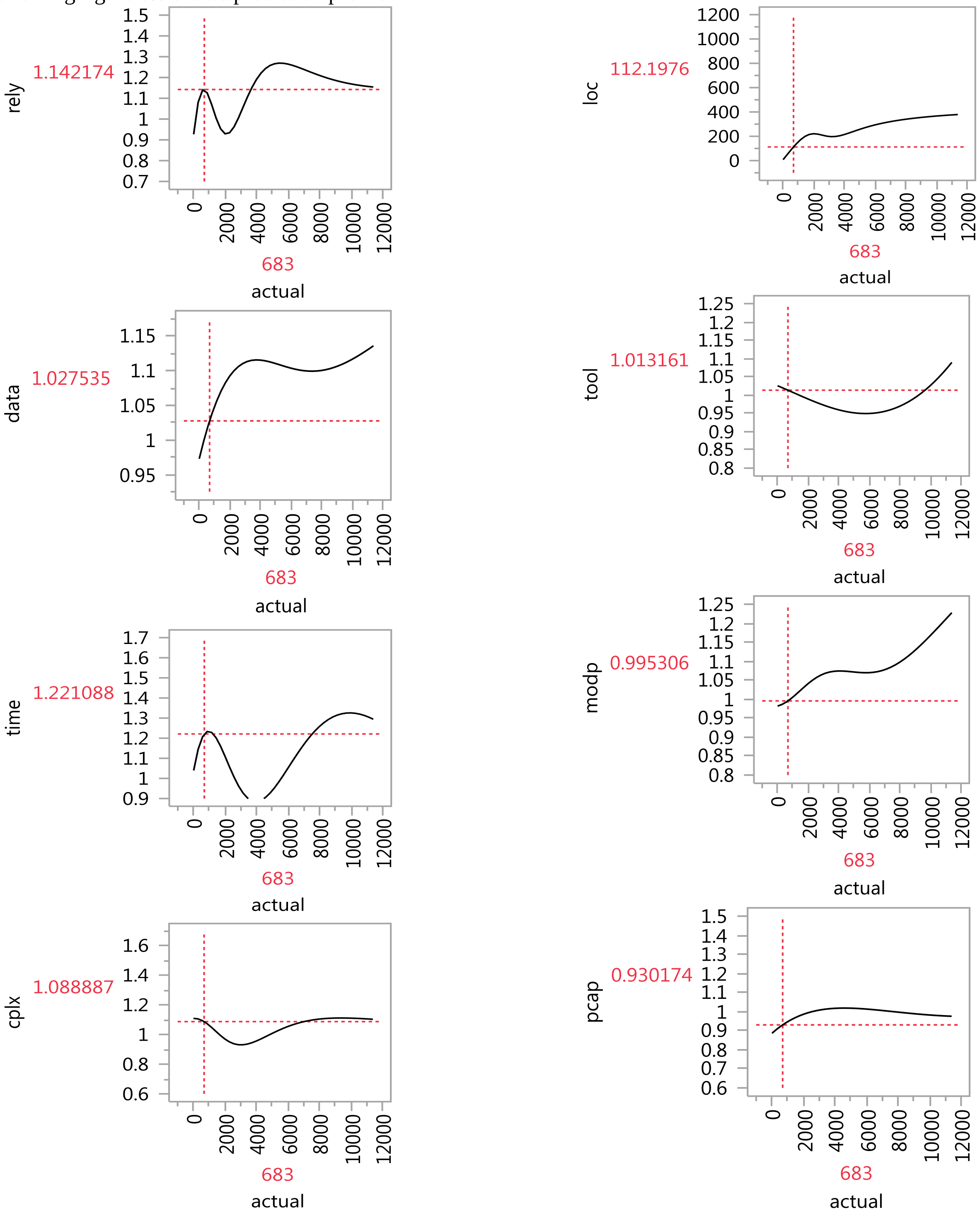

Fig 5.3 It shows Prediction Profiler

VI. CONCLUSION

The accurate and reliable cost estimation is very crucial task for software project development. The estimation in software development depends on various factors particularly managing project cost, time and quality and effort factors. Therefore, accurate assessment is a consequently factor in projects success and reducing the risks. Researchers have developed various models for assessment but there is no estimation method which can present the best estimates in all cases and each technique can be suitable in the special project. In this paper, a novel approach based Bayesian multi-class algorithm is proposed for software cost estimation. It helps project manager to provide nimble and realistic cost estimation. The proposed work is carried in two steps, in first phase known as training phase, optimizing the parameters and second step known as validating phase, the prediction process. The Parameters SSE, RME, MAD and $R^2$are calculated for COCOMO-II dataset. Statistical results show that our method could significantly improve accuracy, error minimization and has likely to become an efficient method for software cost estimation.

ACKNOWLEDGEMENT

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

REFERENCES

[1] Molokken, K. and Jorgensen, M. (2003) A Review of Software Surveys on Software Effort Estimation. 2003 International Symposium on Empirical Software Engineering, 30 September-1 October 2003, 223-230.

[2] Afzal, W. and Torkar, R. (2011) On the Application of Genetic Programming for Software Engineering Predictive Modeling: A Systematic Review. Expert Systems with Applications, 38, 11984-11997.

[3] Albrecht, A. and Gaffney, J.E. (1983) Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. IEEE Transactions on Software Engineering, SE-9, 639-648.

[4] Bailey, J.W. and Basili, V.R. (1981) A Meta-Model for Software Development Resource Expenditures. Proceedings of the 5th International Conference on Software Engineering, Piscataway, 107-116.

[5] RuchikaMalhotra, A.J. (2011) Software Effort Prediction Using Statistical and Machine Learning Methods. International Journal of Advanced Computer Science and Applications (IJACSA), 2.

[6] Wang, F.-S. and Chen, L.-H. (2013) Heuristic Optimization. In: Dubitzky, W., Wolkenhauer, O., Cho, and K.-H. And Yokota, H., Eds., Encyclopedia of Systems Biology, Springer, New York, 885-885.

[7] Uysal, M. (2010) Estimation of the Effort Component of the Software Projects Using Heuristic Algorithms. INTECH Open Access Publisher, Croatia.

[8] Maleki, I., Ghaffari, A. and Masdari, M. (2014) A New Approach for Software Cost Estimation with Hybrid Genetic Algorithm and Ant Colony Optimization. International Journal of Innovation and Applied Studies, 5, 72-81.

[9] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. IEEE International Conference on Neural Networks, 4, 1942-1948.

[10] Yang, X.-S. (2010) A New Metaheuristic Bat-Inspired Algorithm. In: Gonzlez, J., Pelta, D., Cruz, C., Terrazas, G. and Krasnogor, N., Eds., Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Vol. 284 of Studies in Computational Intelligence, 65-74. Springer, Berlin and Heidelberg.

[11] Artificial Neural Networks for Software Effort Estimation: A Review Dharmesh Santani, Mahesh Bundele, Poonam Rijwani , International Journal of Advances in Engineering Science and Technology, ISSN: 2319-1120

[12] Automation of Software Cost Estimation using Neural Network Technique, Gaurav Kumar, Pradeep Kumar Bhatia, International Journal of Computer Applications (0975 – 8887) Volume 98 – No.20, July 2014.

[13] Software Development Effort Estimation Using Fuzzy Logic: A Review Shifali, Naveen Bilandi, International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 2841-2845

[14] Software Cost Estimation using Fuzzy Logic Ravishankar. S P. Latha, International Conference on Recent Trends in Computational Methods, Communication and Controls (ICON3C 2012) Proceedings published in International Journal of Computer Applications® (IJCA).

[15] Genetic Algorithm for Optimizing Neural Network Based Software Cost Estimation Tirimula Rao Benala, Satchidananda Dehuri , Suresh Chandra Satapathy , Ch. Sudha Raghavi Computing, Volume 7076 of the series Lecture Notes in Computer Science pp 233-239

[16] A New Approach for Software Cost Estimation with Hybrid Genetic Algorithm and Ant Colony Optimization Isa Maleki , Ali Gaffer , and Mohammad Masdari, International Journal of Innovation and Applied Studies ISSN 2028-9324 Vol. 5 No. 1 Jan. 2014, pp. 72-81 © 2014.

[17] Software Cost Estimation using Hybrid Algorithm Shivani Sharma , Aman Kaushik , Abhishek Tomar International Journal of Engineering Trends and Technology (IJETT) – Volume 37 Number 2- July 2016

[18] Optimizing Software Effort Estimation Models Using Firefly Algorithm, Nazeeh Ghatasheh, Hossam Faris, Ibrahim Aljarah, Rizik M. H. Al-Sayyed, Journal of Software Engineering and Applications, 2015, 8, 133-142 Published Online March 2015 in SciRes.

[19] Novel Meta-Heuristic Algorithmic Approach for Software Cost Estimation, Ruchi Puri, Iqbaldeep kaur, International Journal of Innovations in Engineering and Technology (IJIET), Volume 5 Issue 2 April 2015, ISSN: 2319 – 1058.

[20] A New Hybrid For Software Cost Estimation Using Particle Swarm Optimization And Differential Evolution Algorithms, Majid Ahadi , Ahmad Jafarian, Informatics Engineering, An International Journal (Ieij), Vol.4, No.1, March 2016.