Volume 5, Issue 1, January 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Design and Implementation of ANFIS Model for Software Development Time Estimation Using Ensemble Learning

Abhishek Kumar*
M.Tech Student, Department of CSE
Maharana Pratap College of Technology,
Gwalior, M.P., India

Unmukh Datta
H.O.D Department of Computer Science Engg.
Maharana Pratap College of Technology,
Gwalior, M.P., India

Abstract— To develop a project successfully, it is important for any organization that the project should be completed within budget, on time and the project should have requisite quality. This paper presents an Adaptive Neuro-Fuzzy Approach for Software Development Time Estimation using Ensemble Learning. The main idea behind ensemble learning is to combine multiple models into one. At every step, the ensemble fits a new learner to the difference between the observed response and the aggregated prediction of all learners grown previously. The ensemble fits to minimize mean-squared error .As in ANFIS; we are providing only one model, it was not possible to combine an arbitrary number of models into one.

Keywords— Adaptive Neuro Fuzzy Inference System (ANFIS), Neural Network, Fuzzy Logic, Prediction, MRE, MMRE, BRE, Development Time (DT), Membership Function (MF)

I. INTRODUCTION

To develop a project successfully, it is necessary for any organization that the project should be completed within budget, on time and the project should have required quality. In order to produce a successful project, cost estimation is essential in managing software projects because of the uncertainty and diversity nature inherent in project development. Estimation is the intelligent anticipation of quantum of the work that needs to be performed and the resources required to perform the work in a defined environment using specified methods [1]. Software cost can be defined as cost incurred in various resources to develop a software project. The most valuable resource to develop software is man power. Software estimation gives the approximate calculation of software size, software development cost and effort, and development schedule for a specified software project[2]. Software development effort can be characterize as the required human resources necessary for developing the software project of an estimated size[3]. It is measured in "person-month" or "person-hours". Software development effort estimates are the basis for project bidding, budgeting and planning. It is all about the future prediction of a work so that managers can make decision to how long and how many resources are required to complete the project. The use of inaccurate estimation makes the manager's decision as a recipe of disaster and looses the control and execute plan in a wrong direction.

II. RELATED WORK

Lopez Martin et al. [4] proposed a fuzzy logic model for development time estimation. Ting su et al. [5] described an enhanced fuzzy logic model for the estimation of software development effort which had the similar capabilities as the previous fuzzy logic model in addition to enhancements in empirical accuracy in terms of MMRE. Abbas Heiat [7] used artificial neural network techniques like RBF (Radial Basis Function) and MLP (Multi Layer Perceptron) for software development effort. The main goal of this paper is to evaluate software development time using an adaptive Neuro fuzzy approach

A. Software Effort Drivers

A Software project can be defined with a number of attributes such as size and complexity. The effort estimation of the software project depends on some of those attributes. The cost drivers are multiplicative factors that determine the effort required to complete your software project. Different models define its own cost drivers such as Boehm's COCOMO defines many cost drivers based on product, computer, personnel, and project attributes and Albrecht's Function Point defines five main components system for function point analysis [8]. The main cost driver used for a number of estimation model is the size of the project and usually given as Kilo Lines of delivered Source Code (KSLOC). In addition, the other cost drivers are used to define the complexity of the software project, given by the Effort Adjustment Factor (EAF).

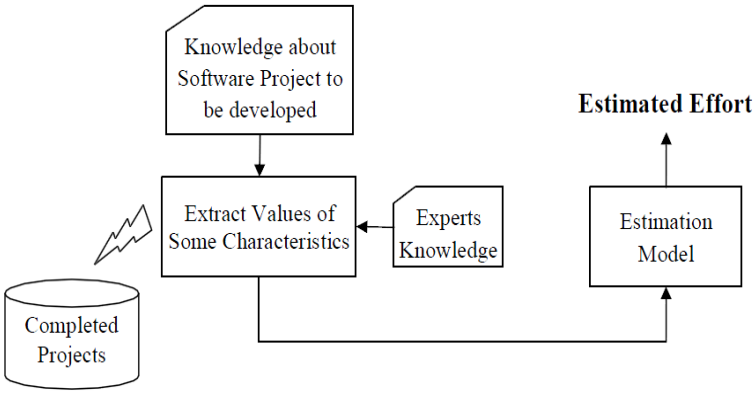

Figure 1 Software effort estimation process

B. Software Effort Estimation Model

Accurate effort estimation is the state of art of software engineering, effort estimation of software is the preliminary phase between the client and the business enterprise. The relationship between the client and the business enterprise begins with the estimation of the software [9]. Over the last three decades, a growing trend has been observed in using variety of software effort estimation models in diversified software development processes. Along with this tremendous growth, it is also realized the essentiality of all these models in estimating the software development costs and preparing the schedules more quickly and easily in the anticipated environments [10]. Although a great amount of research time and money have been devoted to improving accuracy of the various estimation models, due to the inherent uncertainty in software development projects as like complex and dynamic interaction factors, intrinsic software complexity, pressure on standardization and lack of software data, it is unrealistic to expect very accurate effort estimation of software development processes [11].

C. Needs of Estimation in Early Stages

1. Estimates are carried out at various stages of a software project. Software estimation is assuming more importance as a natural consequence of increased outsourcing of software development work. In an outsourcing scenario, software estimation is needed for the valuation of suppliers' proposals. Potential contractors considering a bid would need to scrutinize the system specification and produce estimates on which to base proposals. In the case of in-house development, estimation is necessary to evaluate competing demands for the software and to apportion the available resources to the highest priority work. Therefore, the outcome of software cost estimation can be listed as:
2. Budget for outsourced assignment is ready.
3. Evaluation of different vendors' proposal is done based on which an agreement with the selected vender on the price to be paid is contracted.
4. A budget is allocated in the case of in-house development.

III. METHODOLOGY

A. Ensemble Learning

The main idea behind ensemble learning is to combine multiple models into one. At every step, the ensemble fits a new learner to the difference between the observed response and the aggregated prediction of all learners grown previously. The ensemble fits to minimize mean-squared error.As in ANFIS we are providing a only one model, it was not possible to combine an arbitrary number of models into one. In this work we are using two general ensemble learning method:

(a) The Bagging nodes first train a set of models (each only on a subset of the data) afterwards the test data is predicted with each of the models and finally the voting node detects the majority class.

(b) The Boosting nodes (Learner and Predictor) apply the LS Boost algorithm to the data set together with the chosen model.

The fig.2 shows the proposed model for modeling the Bagging and LS Boost predictor

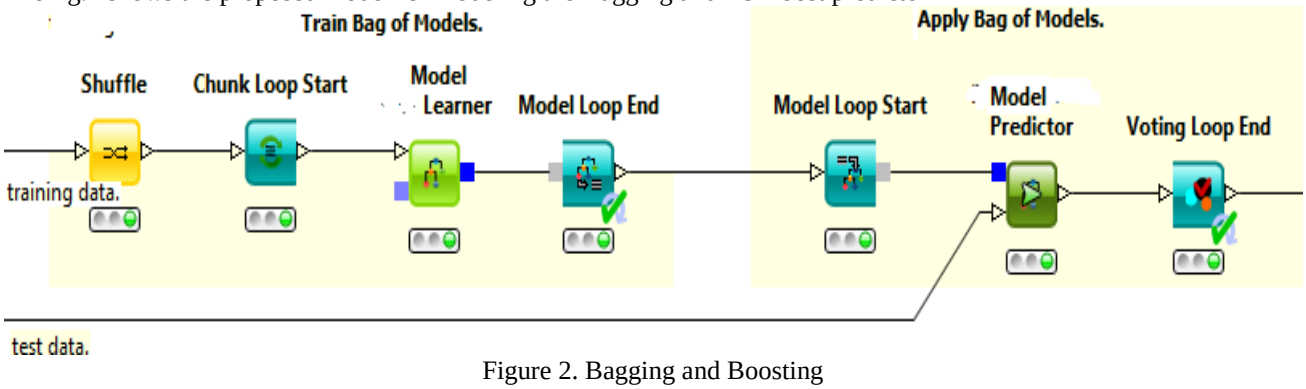

Figure 2. Bagging and Boosting

B. ANFIS Framework

Adaptive Neuro Fuzzy Inference System (ANFIS) as it is a combination of Fuzzy Logic and Neural Network so ANFIS takes advantages from fuzzy logic and neural network. This ANFIS constructs a Fuzzy inference system by using given training data set whose membership function parameters are adjusted by back propagation algorithm or in combination with least square type of method. Fig.3 shows a high level diagram of the proposed ANFIS. Inputs and their membership functions appear to the left of the ANFIS structural characteristics, while outputs and their membership functions appear on the right.

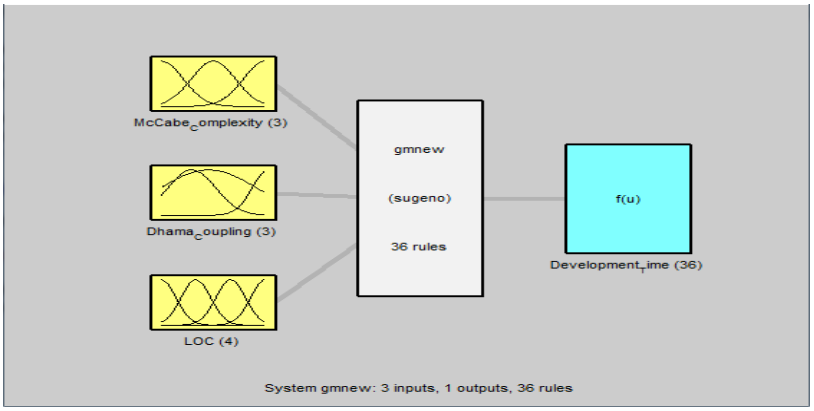

Figure 3 shows a high level diagram of the proposed ANFIS

After designing the different model the final step is to compare the result. Fig.4 show the final methodology

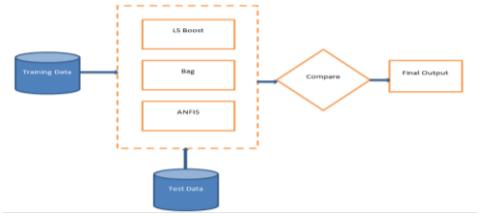

Figure 4 Final methodology

C. Algorithm

The algorithm for the proposed work is as follow:

Step 1: Determine the inputs of the model.

Step 3: Divide the data into two sets, one for estimating the models, called the Train data set, and the other one for evaluating the validity of the estimated model, called the test data set. Usually the train data set contains 70% to 90% of all data and the remaining data are used for the test data set.

Step 4: Generate an ANFIS model by assigning the no. of membership functions and the type of membership function.

Step 5: Train this network by taking the value of epochs 100 and save this Fuzzy Inference System file (.fis).

Step 6: Generate the Bagging and Boosting model by:

i. Setting the Number of Ensemble Members
ii. Preparing the Weak Learners
iii. **Fit the ensemble**s (Bagging and Boosting)
iv. Save the fitted ensembles

Step 7: Evaluate the value of Development Time with these saved models i.e ANFIS, Bagging and Boosting file and the testing data set.

Step 8: Evaluate and Compare the Value of MMRE and PRED.

IV. RESULTS & DISCUSSION

Our experiment consists in estimating the software development effort by using the neural networks approach on the Lopez-Martin et.al. Data set. They used the sets of system development projects, where the Development Time (DT), Dhama Coupling (DC), McCabe Complexity (MC) and the Lines of Code (LOC) metrics were registered for 41 modules. Since all the programs were written in Matlab, the module categories mostly belong to procedures and functions. The development time of each of the forty-one modules were registered including five phases: requirements understanding, algorithm design, coding, compiling and testing [6, 7]. Table I shows the dataset used for carrying out experimentation. All these models were trained with first 31inputs from the standard dataset and later 10 inputs from the same dataset were used to test the models. The effectiveness of neural network model over the regression analysis model has already been proved with the same dataset[8].The Development Time (DT') were obtained for each of the trained neural network models and finally a comparative analysis was carried out based on the standard assessment criterions like MRE, MMRE, BRE and Pred[25]. The output responses of the above mentioned models in table 1. On the basis of the results obtained the performance is compared on various evaluation criteria's. The simulation was done using the fuzzy logic toolbox, neural network toolbox and ANFIS toolbox in MATLAB.

Table 1 Performance comparisons on various evaluation criteria's

Project No	Actual DT	GBell	Boosting	Bagging
31	19	18.99988	18.95345	18.95345
32	13	12.9999	12.91972	12.91972
33	12	12.00052	12.10164	12.10164
34	12	7.581409	12.23131	12.23131
35	21	15.67964	21.42979	21.42979
36	21	204.0081	26.37097	26.37097
37	19	19	19.00911	19.00911
38	18	18.00004	18.00497	18.00497
39	24	24.5	24.51646	24.51646
40	25	24.5	24.51646	24.51646
41	18	17.99998	17.98736	17.98736

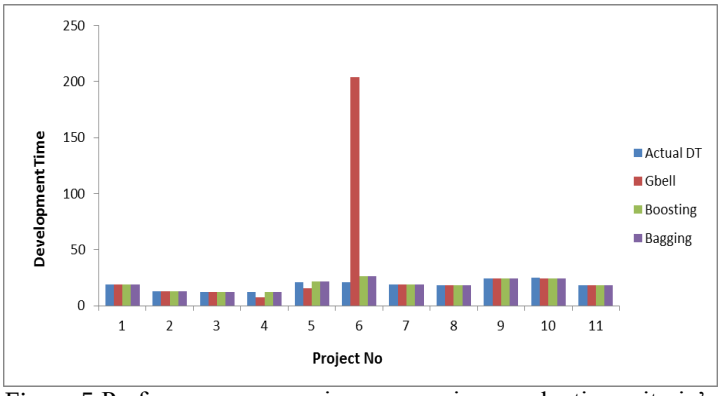

Figure 5 Performance comparisons on various evaluation criteria's

A. MMRE

The MMRE calculates the mean for the sum of the MRE of n projects. Specifically, it is used to evaluate the prediction performance of an estimation model. A model which gives lower MMRE is better than that which gives higher MMRE. Table.2 and Fig.6 shows the MMRE obtained from different models.

Table 2 Comparison of models

GBell	Boosting	Bagging
85.24667	3.226514	3.226514

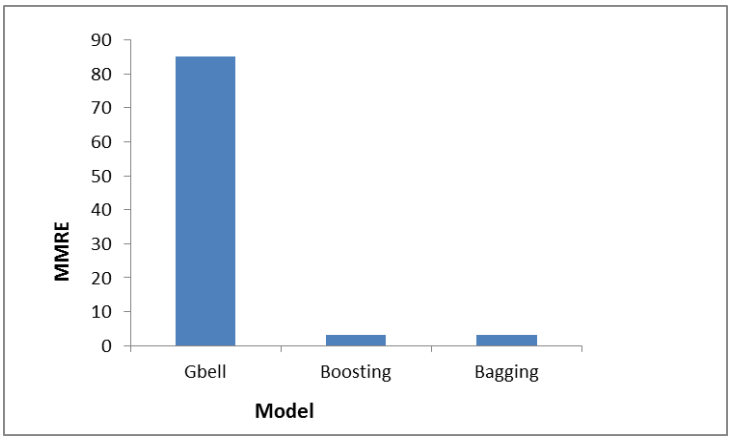

Figure 6 Comparison of MMRE

B. Prediction

Prediction (PRED) at level n is defined as the % of projects that have absolute relative error less than n .Table.3 shows the Prediction obtained from different models.

A model which gives higher PRED is better than that which gives lower PRED .And from table it is clear that Bagging and Boosting model gives better prediction then ANFIS.

Table 3 Comparison of Prediction level

GBell	Boosting	Bagging
0.727273	0.909091	0.909091

C. Balanced Relative Error (BRE)

A model which gives lower BRE is better than that which gives higher BRE.Hence from Table.4 we can observe that Bagging and Boosting model are better.

Table 4 Comparison of BRE

GBell	Boosting	Bagging
0.879828	0.032304	0.032304

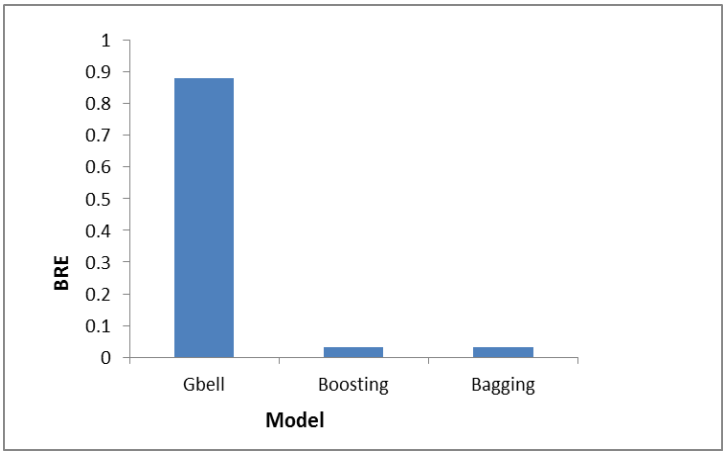

Figure 7 Comparison of BRE

V. CONCLUSION

The paper suggests a new approach for estimating of software project development timeusing ensemble Learning.. In this paper, Adaptive Neuro Fuzzy Inference model is considered and bagging and boosting are used to predict the future values. It is observed that Neuro Fuzzy model using these function gives better results than all other models. It is also observed that enseble learning techniques gives better results for all the three parameters. In order to achieve more accurate estimation, the estimated values of several other techniques and combine their results may be useful.

REFERENCES

[1] M. Chemuturi, "Software Estimation Best Practices, Tools & Techniques: A Complete Guide for Software Projects Estimator", available at :http://books.google.co.in/books?id=IwEOB2Mfzx0C&pg=PA1 &source=gbs_toc_r &cad=4#v=onepage&q&f=false, 2009.

[2] S Basha and P. Dhavachelvan, "Analysis of Empirical Software Effort EstimationModels", International Journal of Computer Science and Information Security (IJCSIS), Vol. 7, No. 3, 2010.

[3] B. Hughes and M. Cotterell, "Software Project Management", Tata McGraw-Hill,2006.

[4] B. W. Boehm, "Software Engineering Economics", Prentice-Hall, Englewood Cliffs, NJ, USA, 1981.

[5] T. Gruschke, "Empirical Studies of Software Cost Estimation: Training of Effort Estimation Uncertainty Assessment Skills", 11th IEEE International Software Metrics Symposium, IEEE, 2005.

[6] C. C. Kung and J. Y. Su, "Affine Takagi-Sugeno fuzzy modeling algorithm by Fuzzy c-regression models clustering with a novel cluster validity criterion", IETControl Theory Appl., pp. 1255 – 1265, 2007.

[7] V. Khatibi, Dayang and N. A. Jawawi, " Software Cost Estimation Methods: AReview", Journal of Emerging Trends in Computing and Information Sciences, CIS Journal, Vol. 2, no. 1, ISSN 2079-8407, 2011.

[8] N. Sharma1, A. Bajpai and M. R. Litoriya, "A Comparison of Software Cost Estimation Methods: A Survey", The International Journal of Computer Science and Applications (TIJCSA), Vol.1, no. 3, ISSN – 2278 – 1080, May 2012.

[9] J. Keung, "Software Development Cost Estimation Using Analogy: A Review", Australian Software Engineering conference, IEEE, 2009, DOI:10.1109/ASWEC.2009.32, 1530-0803/09.

[10] T. R. Benala, S. Dehuri and R. Mall, "Computational Intelligence in Software Cost Estimation: An Emerging Paradigm", ACM SIGSOFT Software Engineering NotesPage, Vol. 37, no.3, 2012, DOI: 10.1145/180921.2180932.

[11] J.S. Pahariya, V. Ravi and M. Carr, "Software Cost Estimation using Computational Intelligence Techniques", World Congress on Nature & Biologically Inspired Computing(NaBIC 2009)978-1-4244-5612-3/09/2009 IEEE, 2009.

[12] Mrinal Kanti Ghose, Roheet Bhatnagar and Vandana Bhattacharjee. "Comparing Some Neural Network Models forSoftware Development Effort Prediction", IEEE 2011

[13] Venus Marza, Amin Seyyedi, and Luiz Fernando Capretz, "Estimating Development Time of Software Projects Using a Neuro Fuzzy Approach", World Academy of Science, Engineering and Technology 22- 2008.

[14] Vachik S. Dave Kamlesh Dutta, Neural Network based Software Effort Estimation & Evaluation criterion MMRE, International Conference on Computer & Communication Technology (ICCCT)-2011.

[15] Cuauhtémoc López Martín, Software Development Effort Estimation Using Fuzzy Logic: A Case Study, Proceedings of the Sixth Mexican International Conference on Computer Science (ENC'05), 0-7695-2454-0/05 $20.00 © IEEE 2005.

[16] Moataz A. Ahmed, Moshood Omolade Saliuand Jarallah Al Ghamdi, "Adaptive fuzzy logic-based framework for software development effort prediction", Information and Software Technology 47- 2005.

[17] C.J. Burgess, M. Lefley, Can genetic programming improve software effort estimation? A comparative evaluation, Information and Software Technology 43 (2001) 863–873.

[18] Anish Mittal,Kamal Parkash,Harish Mittal " Software Cost Estimation Using Fuzzy Logic", ACM SIGSOFT Software Engineering Notes Page 1 November 2010 Volume 35 Number 1.

