International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

Cost Estimation of Software by ANFIS based Artificial Intelligence Approach

Himanshu Kumar Shukla
Computer Science & Engg. Dept
BBDITM, Lucknow (U.P.)
himanshu0590@gmail.com

Surya Vikram Singh
Computer Science & Engg. Dept
BBDITM, Lucknow (U.P.)
m.tech.surya@gmail.com

Rudrendra Bahadur Singh
Computer Science & Engg. Dept
BBDITM, Lucknow (U.P.)
rudra.rathor20@gmail.com

***Abstract*—Estimating the work-effort and the schedule required to develop and/or maintain a software system is one of the most critical activities in managing software projects. Software cost estimation is a challenging task. Estimation by correlation is one of the suitable techniques in software cost estimation field. However, the method used for the estimation of software effort by analogy is not able to handle the categorical data in an explicit and accurate manner. This paper aims to make use of ANFIS, an AI technique to improve the precision of software cost estimation. The data used is the DESHARNAIS dataset from PROMISE Software Engineering Repository. The proposed model performance has been analyzed in terms of MAE, Correlation Coefficient and RMSE. It was seen that the proposed model performed better than the Regression model, having RMSE value of 780.97 against 3007.05 of regression model.**

***Keywords*—Software cost estimation, RMSE, ANFIS, MAE**

1. INTRODUCTION

Software Cost Estimation is process of predicting the effort required to develop a software method [1]. The main category of models are algorithmic and non-algorithmic having its own merits and demerits. Optimum choice is a key factor of accuracy. The estimates depend on effort, project duration, cost etc. Cost estimation will remain complex problem and researches should indulge to approach new technique for this task. Models developed on Artificial Intelligence techniques should be used for more accurate estimation. Thus software cost estimation or software effort estimation is the process of predicting the effort required to develop a software system [1]. Software engineering cost models and estimation techniques are used for a number of purposes including; budget, trade offs and risk analysis, project plans and power, and analysis of software development. The accuracy of the software project cost estimation has a direct and significant impact on the quality of the firm's software investment decisions.

Due to the inadequacy and inefficiency of the mathematical models to explain the nonlinear properties existing between the input and output parameters of software cost estimation datasets led to the development of intelligent modeling techniques. There are a number of contending software cost estimation methods available for software developers to predict effort and test effort required for software development, from the instinctive opinion of the expert methods to the more complex algorithmic modeling methods and the methods based on anology[2]. In estimating software projects, it is important to balance the relationships between effort, program and class. It is widely accepted that just estimation of one of these aspects without considering the others will result in impractical estimations. Models based on classical estimates are well-known based on linear or non-linear regression analysis, which includes fixed input factors and permanent outputs. The project size determining the scope is modeled as a main input of such models [2]. One representative of such models is COCOMO. The most critical problem in such an approach is the wealth of data that is needed to get regression parameters, which is often impossible to get in desired quantities. It actually limits their use for project estimation. In recent years, a flexible and competitive method, Bayesian Belief Network (BBN), has been proposed for estimating software projects[2]. Taking due consideration, ANFIS [3] a state of the art artificial intelligent method, has the option to improve the prediction of software cost estimation

The rest of the paper is organized as follows: Section II describes the Literature Review. Section III deals with data used, IV the methodology part of work done, followed by results and discussions in section V. In the end , based on the discussion various conclusions are drawn and the future scope for the present work is discussed.

2. LITERATURE SURVEY

Various techniques, such as linear regression, discriminate analysis, decision trees, neural networks etc. have been evolved and applied to predict cost in software. **N. Veeranjaneyulu et. al.** proposed a classical method**,** focussed around relationship thoughts, fuzzy logic and quantifying by phonetics, which can be viably utilized when the software activities are portrayed by all out as well as numerical information. The preference of this system is that it can deal with the imprecision and the instability distinctively while portraying the software project. **Lalit V. Patil, et. al.** proposed a hybrid approach, which consists of Functional Link Artificial Neural Network (FLANN) and COCOMO-II with algorithm training. FLANN lessens the computational complexity in multilayer neural network. It has no hidden layer, and it has fast learning ability. **Jyoti G. Borade** focussed on the existing estimation of software methods. Also, dealt with backdrop information on software project models and software metrics to be used for effort and cost estimation. No model can estimate the cost of software with high degree of accuracy. Estimation is complex action that requires knowledge of a number of key attributes.

K. Subba Rao, et. al. showed that accurate and reliable estimation of cost is very important for software project development. An enhanced model of Holdback through neural network was developed. **Vahid Khatibi et. al.** studied and methodically illustrated the present estimation techniques. Since software project managers are used to select the best estimation method based on the conditions and status of the project, describing and comprising of estimation techniques can be useful for decreasing of the project failures. **Swati Waghmode et. al.** in their model proposed a value shrinking technique based on multilayer feed-forward neural network, and auto-associative clustering. **Isa Maleki et. al.,** investigated SCE using the FL and the capabilities of COCOMO model. **Soumyabrata Mukherjee et. al.** in their work summarized the different models and metrics. Also provided an overview of software cost estimation tools which is essential for estimating.

3. Dataset Used

The data that is going to be used is the DESHARNAIS dataset available from PROMISE Software Engineering Repository made publicly available in order to encourage repeated, verified, refuted, and/or improved predictive models of software engineering. The main source of the dataset is J. M. Desharnais [7], whereas M. J. Shepperd [6] gives a basic overview of the dataset. It has 81 instances and 12 attributes, of which 4 projects are incomplete, so only 77 instances has been used. Further, for model development out of 12 attributes, only 9 have been used for model development. The Desharnais dataset features are given in Table 1 below.

Table 1. Desharnais Project Features

Variable	**Description**	**Type**	**Model Parameter**
ExpTeam	Experience of Team	Continuous	Input
ExpProj Man	Experience of Project Manager	Continuous	Input
Transac	Transactions	Continuous	Input
Raw FP	Raw Function points	Continuous	Input
Adj. Factor	Technology Adjustment Factor	Continuous	Input
Adj. FP	Adjustment Function Points	Continuous	Input
Dev Env	Development Environment	Categorical	Input
Entities	Number of Entities	Continuous	Input
Effort	**Actual Effort**	**Continuous**	**Output**

4. ANFIS Model Development

4.1 Parameter Selection

ANFIS [11],[14] is a judicious integration of FIS and ANN, capable of learning, high-level thinking and reasoning and it combines the benefits of these two techniques into a single capsule. Identification of the rule base is the key of a FIS. The problems are (1) there are no standard methods for transforming human knowledge or experience into rule base; and (2) it is required to further tune the MFs to minimise the output error and to maximise the performance. Now to generate a FIS using ANFIS, it is significant to choose proper parameter, inclusive of the number of membership functions (MFs) for each individual antecedent variables. It is also important to select proper parameters for learning and refining process, including the initial step size (ss). In the present work the commonly used rule extraction method applied for FIS identification and refinement is subtractive clustering. MATLAB Fuzzy Logic Toolbox [4] has been used to simulate the ANFIS.

Here the initial parameters of the ANFIS are identified using the subtractive clustering method [8]. However, the parameters of the subtractive clustering algorithm still need to be specified. The clustering radius is very important parameter in the subtractive clustering algorithm and is optimally determined through a trial and error procedure. By varying the clustering radius r_a between 1 to 0.1 with 0.01 step size, the most appropriate parameters are got by reducing the root mean squared error obtained on a representative validation set. Clustering radius r_b is selected as 1.5 r_a. For other parameters default values are used in the subtractive clustering algorithm.

Gaussian membership functions are used for each fuzzy set in the fuzzy system. The number of membership functions and fuzzy rules required for a particular ANFIS is determined through the subtractive clustering algorithm. Gaussian membership function parameters are optimally determined using the hybrid learning algorithm. ANFIS training is done for 500 epochs.

Gaussian membership function has been used as the input membership function and linear membership function for the output function. Here separate set of input and output data has been used as input arguments. In MATLAB *genfis2* generates a Sugeno-type FIS structure using subtractive clustering. Since there is only one output, *genfis2* has been used to generate an initial FIS for ANFIS training. *genfis2* achieves this by extract a set of rules that models the data performance. The rule extraction method initially uses the *subclust* function to determine the number of rules and antecedent membership functions and then uses linear least squares estimation to determine each rule's consequent equations. This function generates a FIS system that contains a set of fuzzy rules to cover the feature space.

Table 1 shows the input and output model parameters used for model development. The parameters used in the model for training ANFIS are given in Table 2 and the rule extraction method used are given in Table 3. Table 4 summerizes the results of types and values of model parameters used for training ANFIS.

Table 2. Parameters used in all the models for training ANFIS

Rule extraction method used	Subtractive clustering
Input MF type	Gaussian membership (‘gaussmf’)
Input partitioning	variable
Output MF Type	Linear
Number of output MFs	one
Training algorithm	Hybrid learning
Training epoch number	500
Initial step size	0.01

Table 3. Rule extraction method used for training ANFIS

Rule Extraction Method	**Type**
And method	‘prod’
Or method	‘probor’
Defuzzy method	‘wtever’
Implication method	‘prod’
Aggregation method	‘max’

Table 4. Values of parameters used for training ANFIS

No. of nodes	569
No. of linear parameters	279
No. of non-linear parameters	496
Total no. of parameters	775

No. of training data pairs	60
No. of testing data pairs	17
No. of fuzzy rules	31

5. Results and Discussions

ANFIS model having five input variables are trained and tested by ANFIS method and their performances compared and evaluated based on training and testing data. The best fit model structure is determined according to criteria of performance evaluation. The performances of the ANFIS model are shown in Fig. 1&2 and their RMSE values both for training and testing data are 0.6243 and 2737.42 respectively (Table 5 below).

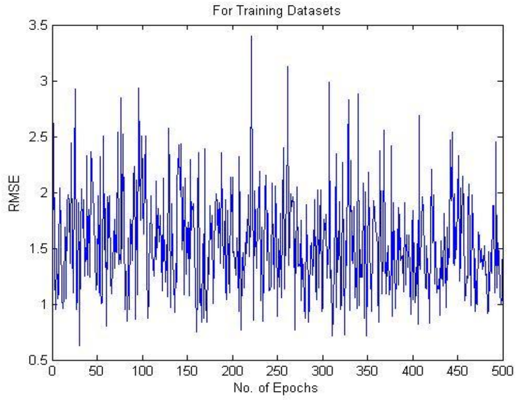

Fig. 1. RMSE Plot of Training Datasets

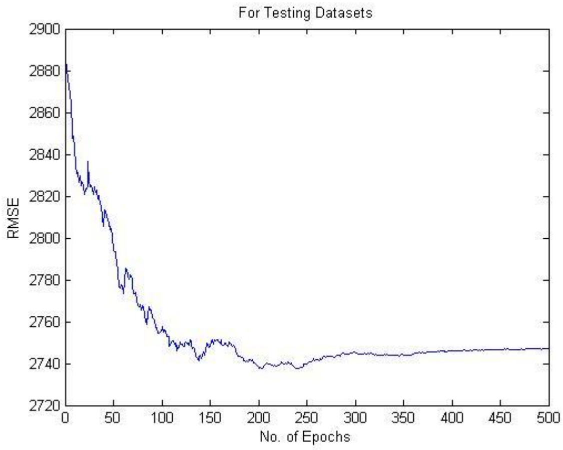

Fig. 2. RMSE Plot of Testing Datasets

Table 5: RMSE Values for Datasets

	Training Datasets	Testing Datasets	Total Datasets
RMSE	0.6243	2737.42	780.9711

A graphical plot of both observed and predicted (ANFIS_Output) Software Effort Estimation values for training and testing data are given in Fig. 3 and 4 below.

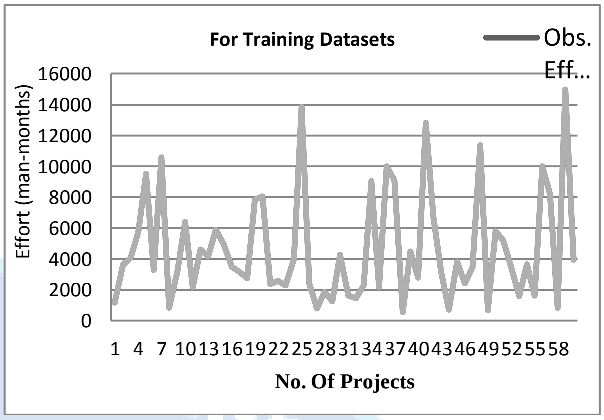

Fig. 3. Comparative plot of Predicted vs. Observed effort values

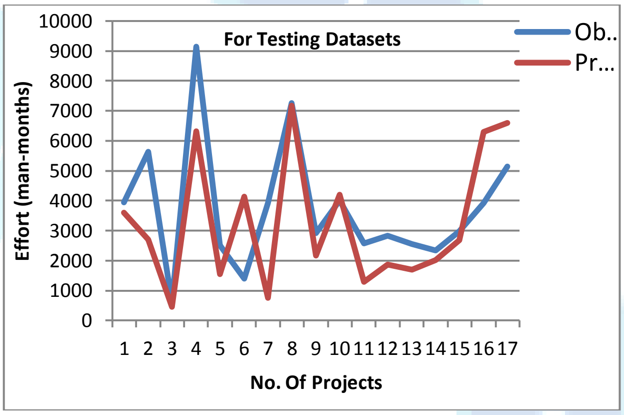

Fig. 4. Comparative Plot of Predicted vs. Observed effort values

Further in order to judge the ability and efficiency of the model for cost estimation values MRE has been used. MRE is an indication of the average deviation of the predicted values from the corresponding measured data and can provide information on long term performance of the models; the lower MRE the better is the long term model prediction. A positive MRE value indicates the amount of overestimation in the predicated effort estimation and vice versa. The MRE of training and testing data sets for Software effort estimation are shown in fig. 5 and 6 below.

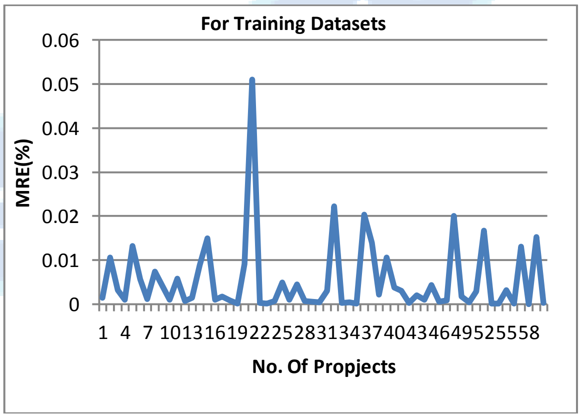

Fig. 5. Magnitude of Relative Error for Training Datasets

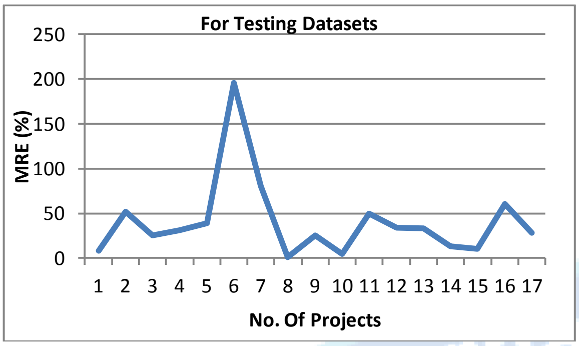

Fig. 6. Magnitude of Relative Error for Testing Datasets

6. Comparison with Regression Model

The results obtained from the present ANFIS model has been compared with those obtained from Linear Regrtession Model, available from PROMISE Software Engineering Repository using the same datasets. The comparative data analysis has been given in Table 6 and its plot in Fig. 7 below.

Table 6: Comparative chart of different Models

Performance Evaluation Criteria	Linear Regression Model	ANFIS Model
Corr. Coeff.	0.7303	0.9721
RMSE	3007.0504	780.97
MAE	2082.1182	281.433

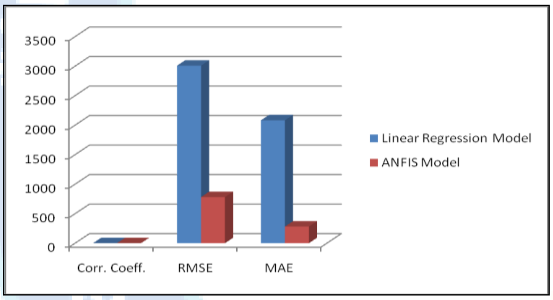

Fig. 7. Comparative Plot of Models

7. CONCLUSION

In the present chapter, applicability and capability of ANFIS techniques for software cost estimation prediction has been investigated. It is seen that ANFIS models are very rugged, characterised by speedy estimation, proficient of managing the near accurate data having noise that are typical of data used here for the present study. Due to the data being non linear, it is an efficient quantitative tool for cost estimation. The studies has been carried out using MATLAB simulation environment. The work dealt here uses the DESHARNAIS dataset available from PROMISE Software Engineering Repository.

Here the initial parameters of the ANFIS are identified using the subtractive clustering method. Gaussian membership functions are used for each fuzzy set in the fuzzy system. The number of membership functions and fuzzy rules required for a particular ANFIS is determined through the subtractive clustering algorithm. Parameters of the Gaussian membership function are optimally determined using the hybrid learning algorithm. Each ANFIS has been trained for 500 epochs.

From the analysis of the results, dealt in earlier section, it is seen that the software cost estimation prediction model developed using ANFIS technique has been able to outperform the Linear Regression Model in terms of the various performance criteria. This can be inferred from the result analysis given in Table-6 and Fig.-7. The overall RMSE value obtained from ANFIS model is 780.97.

REFERENCES

[1]. K. Subba Rao, et. al. (2013), "Software Cost Estimation in Multilayer Feed forward Network using Random Holdback Method", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 10, pp 1309-1328.

[2]. Jyoti G. Borade, (2013), " Software Project Effort and Cost Estimation Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8, pp 730-739.

[3]. JANG, J-S. R., (1993), "ANFIS-Adaptive-Network Based Fuzzy Inference System", *IEEE Transactions on Systems, Man and Cybernetics*, 23(3), pp 665-685.

[4]. http://promise.site.uottawa.ca/SERepository

[5]. "Fuzzy Logic Toolbox", MATLAB version R2011a.

[6]. M. J. Shepperd, C. Schofield and B. Kitchenham, "Estimating Software Project Effort Using Analogies", IEEE Trans. Software Eng., (1997), vol. 23, no. 12, pp. 736-743.

[7]. J.M. Desharnais, "Analyse statistique de la productivitie des projets informatique a partie de la technique des point des fonction," masters thesis, Univ. of Montreal, 1989.

[8]. CHIU, S., (1994), "Fuzzy Model Identification based on cluster estimation", *Journal of Intelligent and Fuzzy Systems*, 2 (3), pp 267–278.