ISSN XXXX XXXX © 2019 IJESC

Research Article Volume 9 Issue No.3

Software Project Effort Estimation

Booja. D[1], Dharani. K[2], Kowsalya. S[3], Varshini. B. V[4]
Department of Computer Science and Engineering
R.M.D. Engineering College, Chennai, India

Abstract:
Effort estimation is a very important activity for planning and scheduling of software project life cycle in order to deliver the product on time and within budget. Machine learning techniques are proving very useful to accurately predict software values. Machine leaning technique consistently predicting accurate results because of its learning natures from previously completed projects. In this paper we discussed about estimating the software using back propagation ANN.

Keywords: Effort Estimation; machine learning; BNN.

I. INTRODUCTION

Software effort estimation is a critical part of software development life cycle model of a project. It is the process of delivering software of established quality and functionality within budget and schedule. It is expressed in terms of person hours or money require to develop the software project. software effort estimation can be considered as a super domain of many software activities, which includes the prediction of not only software development effort but also software requirement, testing maintenance effort etc. Different software development life cycle models require different amount of effort in each phase for completion of software. Software engineers are poor estimators because 13%-15% of all software projects fails to achieve their target due to poor planning and overruns of schedules, which are noticeable in software industry these days. Due to low success rate in estimation there is an increasing pressure on software project teams to abandon the expert-based methods for estimating and planning and to replace them with more objective approaches, such as measurement and analysis-based methods. In this paper we have discussed how the machine learning techniques used for software effort estimation.

II. RELATED WORK

S. Emre Alptekin PRN with each project, correct effort estimation has become a vital drawback for the companies, particularly for project managers. Since Nineteen Seventies completely different ways and models are developed for estimating software system projects' efforts. the primary milestone model was COCOMO, that may be a constructive method planned within the late Nineteen Seventies. many various models followed, the foremost standard and usable models being Function purpose and Use Case purpose. when 2000s, thanks to advances in technology, Artificial Neural Networks has gained in importance particularly among the matter domains that take pleasure in knowledge analysis and self-learning. Software development effort estimation conjointly share similar characteristics as there's generally recent projects' knowledge on hand that ought to facilitate foresee new projects' efforts. Therefore, during this paper we have a tendency to build a software system estimation model by victimization neural network methodology. The options for the network were chosen as a results of Associate in Nursing extensive survey. Dharmesh Santana, Mahesh Bundele, Poonam Rijwani, et al the software package effort estimation ways square measure chiefly categorized into algorithmic and non-algorithmic ways. The algorithmic ways square measure chiefly COCOMO [1], operate Points [2] and SLIM [3]. algorithmic ways are better-known as constant ways as they predict software package development effort employing a fastened mathematical formula that's parameterized from historical information. However, estimates at the preliminary stages of the project square measure tough to get because the first supply to estimate the hassle comes from the SRS document. Also, they need issue in modelling the inherent advanced relationships [4]. the restrictions of algorithmic ways induce to appear towards non-algorithmic methods that square measure soft computing based mostly. These ways have ability to be told from previous information and square measure able to model complex relationship between the dependent (effort) and freelance variables. Ch. Satyananda Reddy, KVSVN Raju is concerned with constructing value estimation model supported artificial neural networks, particularly multi-layer feed forward neural networks. The network model is meant consequently to accommodate the COCOMO model and its parameters, and back propagation learning algorithmic program is employed to coach the network by iteratively process a collection of coaching samples and examination the network's prediction with the actual. The distinction within the estimation is propagated to the input for adjusting the coefficients. This method consists of repeatedly feeding input and output knowledge from empirical observations, propagating the error values, and adjusting the association weights till the error values fall below a user-specified tolerance level. G. E. Wittig, et al. [1] used a dataset of fifteen industrial systems, and used feed-forward back-propagation multilayer neural network for his or her experiment. ANN employed in this paper square measure with numbers of hidden layers' variable from 1-6, however found the simplest performance for under one hidden layer with sigmoid perform. it's been determined that for smaller system the error was 1 Chronicles and for larger systems error was fourteen.2% of the particular effort. Jaswinder Kaur, et al. [2] enforced a back-propagation ANN of 2-2-1 architecture on National Aeronautics and Space Administration dataset include eighteen comes. Input was KDLOC and development methodology and energy was the output. He got result MMRE as eleven.78. several researchers used their completely different ANN and different dataset, to predict the trouble additional correctly. F. Barceló's Toronto, et al. [4], additionally used COCOMO-81 dataset, with just one input, i.e. TOTKDSI (thousands of

delivered supply instructions). All the input file was normalized to [0, 1] range. Here a feed-forward multilayer back propagation ANN was used with the 1-9-4-1 design

III. PROPOSED SYSTEM

The proposed research methodology helps to fill the gaps in the literature and promises a better estimation in Agile based projects. It has been broadly divided in three categories viz., Data Preparation, Data Set Partitioning and Model Selection and Testing Part. The above stated categories will make use of different CASE tools and techniques to complete the process of estimation. As per the literature context it is evident that a machine learning models yields more acceptable results as compared to traditional or non-machine learning models. The steps are given below:

Data preparation
There are online data repositories like ISBSG data sets and Atalassian JIRA repositories which contains Agile and traditional projects data. As the data is heterogeneous so it is required to be filter for Agile projects as per the steps below:

a) Data Understanding: Data understanding is a crucial phase and pertains to a better data extraction from the repositories.
b) ISBSG data sets/ Atalassian JIRA repositories: The data in heterogeneous form will be extracted from the online repositories. The data from these sources are real projects data and may not be uniform. It cannot be used as such for training purposes. As data plays a significant and indispensable role with respective to estimating so it must be standardized.
c) Quantify and process missing data: The next level of standardizing data is to quantify and process any missing data.
d) Determine Agile data using discriminate Analysis: Based on the attributes and unique characteristics of Agile based projects, scrum data will be extracted from the data pool.
e) Agile development sample set: The sample set is ready.
f)Calculate correlation and coefficient matrix
g)Calculate Eigen value of correlation coefficient matrix
h) Determine the number of principle components

Data set partitioning and model selection
Standardized data from the previous step will be partitioned into training and test data sets using K-fold cross validation. Create and train the ANFIS module using SVM as a training algorithm.

a) Create a base fuzzy system
b) Get parameters of base fuzzy system
c) Carry out the optimization of new parameters using Satin Bower Bird optimization algorithm
d) Insert new parameters in the base fuzzy system
e) Classify data and inference results.
f) Estimate the effort and cost
g) Evaluate using metrics MMRE and PRED.

Testing part
After the cross validation, the testing data will then be used:
a) Create a base fuzzy system for testing data
b) Set parameter of membership function using the optimized parameters obtained in the previous step.
c) Classify data and inference results.
d) Estimate the effort and cost
e) Evaluate using metrics MMRE and PRED.

IV. CONCLUSION

Here, various research papers have been revisited, generalized and formalized and it has been found that proposed research work yields promising results for effort and cost estimation in agile software. As a future work, there are some factors like regression test efforts, software architecture erosion effort, people factors, project factors and its accelerating and decelerating factors are not exactly identified in literature and can be added to improve the results.

V. REFERENCES

[1]. Anthony Senyard et al., "Software Engineering Methods for Neural Networks," IEEE Proceedings of the Tenth Asia-Pacific Software Engineering Conference, (APSEC'03), pages 468-477, 2003.

[2]. Boehm B., Clark B., Horowitz E., Madachy R., "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0," Annals of Software Engineering, 1995.

[3]. Boehm, B.W., "Software Engineering Economics"Prentice -Hall, Englewood Cliffs, NJ, USA, 1994.

[4]. Anthony Senyard et al., "Software Engineering Methods for Neural Networks," IEEE Proceedings of the Tenth Asia-Pacific Software Engineering Conference, (APSEC'03), pages 468-477, 2003.

[5]. Boehm B., Clark B., Horowitz E., Madachy R., "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0," Annals of Software Engineering, 1995.

[6]. JaswinderKaur, Satwinder Singh, Dr. Karanjeet Singh Kahlon, PourushBassi,―Neural Network-A Novel Technique for Software Effort Estimation‖, International Journal of Computer Theory and Engineering, Vol. 2, No. 1 February, 2010, page:17-19.

[7]. Roheet Bhatnagar, Vandana Bhattacharjee and Mrinal Kanti Ghose, ―Software Development Effort Estimation – Neural Network Vs. Regression Modeling Approach‖, International Journal of Engineering Science and Technology, Vol. 2(7), 2010,page: 2950-2956.

[8]. K.K. Aggarwal, Yogesh Singh, Pravin Chandra and ManimalaPuri, ―Evaluation of various training algorithms in a neural network model for software engineering applications‖, ACM SIGSOFT Software Engineering, July 2005, Volume 30 Number 4, page: 1-4.

[9]. Mrinal Kanti Ghose, Roheet Bhatnagar and Vandana Bhattacharjee,―Comparing Some Neural Network Models for Software Development Effort Prediction‖, IEEE, 2011.

[10].Satish Kumar, ―Neural Networks: A Classroom Approach‖ , Tata McGraw-Hill, 2004.

[11]. Howard Demuth and Mark Beale, ―Neural Network Toolbox For use with MATLAB ‖,User's Guide,Version-4, Page-5.28.

[12]. N.K.Bose and P.Liang, ―Neural Network Fundamentals with Graphs, Algorithms and Applications‖, Tata McGraw Hill Edition, 1998.

[13]. B. Yegnanarayana, ―Artificial Neural Networks‖, Prentice Hall of India, 2003