SOFTWARE DEVELOPMENT EFFORT DURATION AND COST ESTIMATION USING LINEAR REGRESSION AND K-NEAREST NEIGHBORS MACHINE LEARNING ALGORITHMS

Bhaskar Marapelli
Research Scholar
Shri JJT University
Rajasthan
bhaskarmarapelli@gmail.com

Dr. Prasadu Peddi
Professor
Shri JJT University
Rajasthan

ABSTRACT

Effort estimation is a crucial step that leads to Duration estimation and cost estimation in software development. Estimations done in the initial stage of projects are based on requirements that may lead to success or failure of the project. Accurate estimations lead to success and inaccurate estimates lead to failure. There is no one particular method which cloud do accurate estimations. In this work, we propose Machine learning techniques linear regression and K-nearest Neighbors to predict Software Effort estimation using COCOMO81, COCOMONasa, and COCOMONasa2 datasets. The results obtained from these two methods have been compared. The 80% data in data sets used for training and remaining used as the test set. The correlation coefficient, Mean squared error (MSE) and Mean magnitude relative error (MMRE) are used as performance metrics. The experimental results show that these models forecast the software effort accurately.

Keywords: *Machine Learning, Linear Regression, K-Nearest Neighbors, COCOMO.*

INTRODUCTION

Nowadays the software development system is becoming complicated. The usage of software arises in most companies. Depending on the organization's size and accompanied tasks, each activity under a software project development should be updated regularly. They must deal in giving high-quality software with a low-cost budget. Therefore, more intelligent approaches are needed to solve the challenging problems in this domain. A software development project is one of the processes in the planning of software project management. It needs to be monitored by the manager to ensure a high quality software can be produced at a low cost within a specified time and budget. Software effort estimation (SEE) assumes an essential part of the improvement of software development. Lately, the product has turned into the costliest part of the software development efforts.

The critical aspect of cost in software advancement is the human-effort, and most cost estimation techniques concentrate on this aspect and give estimates in regard to the individual. During the early phase of software development project, there is a lot of work to be done. Software effort estimation is one of the steps in software development project which targets on the production of quality software, which can be delivered on time and within budget, and satisfying its

requirements. It is also known as a feature of software engineering monetary on how to oversee restricted assets to meet the objectives and goals of the schedule, budget, and scope.

The growing concern by most developers is the complexity of estimation made in an early phase of the development process. Sometimes, the development of software product resulted differently due to uncertainties. It increases with the size of software project estimation mistakes that could cost a lot in terms of resource allocated to the project. Because software development problems have many dimensions, we need to investigate the use of several techniques to optimise these challenging issues, not only focusing on the software effort engineering approach, but also to include the incorporation of other methods that can contribute to the enhancement in effort estimation accuracy.

LITERATURE REVIEW

Ahmed BaniMustafa (2015) Nowadays the significant trend of the effort estimation is in demand. It needs more data to be collected and the stakeholders require an effective and efficient software for processing, which makes the hardware and software cost development becomes steeply increasing. This scenario is true especially in the area of large industry, as the size of a software project is becoming more complex and bigger, the complexity of estimation is continuously increased. Effort estimation is part of the software engineering economic study on how to manage limited resources in a way a project could meet its target goal in a specified schedule, budget and scope. It is necessary to develop or adopt a useful software development process in executing a software development project by acting as a key constraint to the project. The accuracy of estimation is the main critical evaluation for every study. Recently, the machine learning techniques are becoming widely used in many effort estimation problems but there are limitations in some of the models and the variation research is still not enough. This paper presents an overview of the effort estimation using machine learning techniques and will be useful for researchers to provide future direction in the field of machine learning adoption in software effort estimation.

Petrônio L. Braga and Adriano L. I. Oliveira, Silvio R. L. Meira (2007) In today's scenario, frequent requirement changes in software development are a notable issue in the software field. Because of the frequent changes, fulfilling the user's requirement is very difficult. As a solution to such issues, Agile Software Development (ASD) has efficiently replaced the traditional methods of software development in industries. Due to various aspects of ASD, it is extremely hard to follow, maintain and estimate the general item. Hence, in order to tackle the Effort Estimation Problem (EEP) in ASD, various types of EEP have been identified in existing methods. The Evolutionary Cost-Sensitive Deep Belief Network (ECS-DBN) model implemented in this paper for effort prediction in any agile technique. The ECS-DBN method has no impact on agility because it uses simple and small inputs. The proposed method used in planning stage of software development to support the project managers in further development of agile software. The project managers characterize the structure of the ECS-DBN, while the

parameter estimation consequently gained from a dataset. This paper used different statistics like accuracy, prediction at m level to evaluate the accuracy of the model. The ECS-DBN method achieved nearly 99% accuracy compared to the existing methods.

Vlad-Sebastian Ionescu (2017) The ECS-DBN model is relatively small and simple and all the input data are easily elicited, so that the impact on agility is minimal. The model predicts task effort, and it independently used agile methods that are suitable in the early project phase for EP. The model is validated using a database of 160 tasks from real agile projects. The prediction accuracy is measured by the percentage of correct predictions among the all predictions. The model results in very good accuracy, but having only one misclassified value. The method achieved 100% prediction in metrics of Pred. (m=25) with 25% tolerance. The MMRE values show that there are no occasional large estimation errors. All other statistical metrics used in this research support these results. In future work, the application can be extended to other deep learning methodologies with higher dimensional data for better performance.

METHODOLOGY

This paper presents the incorporation of machine learning and soft computing techniques with software effort estimation developed up to early 2007 publications. The publication search procedures are illustrated in Figure.

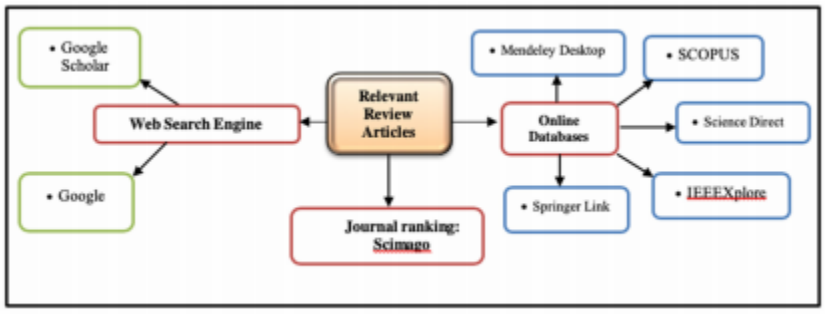

Figure Publication Search

The searches include any keywords of research articles included in title, abstract and keywords. An example of the search queries used is (“software effort estimation”) AND (“soft computing” OR “COCOMO II”). Upon completion of searching the publications, manual search was carried out to identify the redundant results. The redundancies in similar article publications were eliminated. Besides, we used the inclusion and exclusion characteristics to access the potential direction of the study as follows:

i. Inclusion: Publications with a clear focus on software effort estimation and machine learning.

ii. Exclusion: Publications of unrelated area of studies, or not peer-reviewed, for example, lecture notes and tutorials.

RESULTS

The proposed method ECS-DBN are compared with existing methods such as hybrid FFBP and ENN [18] and ensemble machine learning such as SVM, and GLM. S. Bilgaiyan, focused on two types of ANN- FFBP and ENN to solve the EEP. The FFBP-ENN method has high computation speed, fixed computation time and fault tolerance with respect to Elman network. The FFBN-ENN method didn't perform well in other datasets collected from heterogeneous SD methods, which was considered as a limitation of this method. The Table describes the comparison results of proposed with existing methods in terms of MMRE, Pred. (25) and RMSE. P. Pospieszny aimed to narrow the gap between up-to-date research results and implementations within organizations by proposing effective and practical machine learning deployment.

Table Comparison analysis of ECS-DBN with existing methods

Techniques	MM RE	Pred. (25%)	RMS	EMSE
EL	-	- 15.3	3.91	1
ENN	14.80	94.86	- 0.05	6
FFBP	13.49	95.23	- 0.05	2
ECSDBN	12.50	100	0.19	0.04

This was achieved by smart data preparation and applying ensemble averaging of three machine learning algorithms (SVM, NN and GLM) on ISBSG dataset. The limitation was the impact of software sizing especially on EE. As an input variable, it has the most significant impact on forecasting the mentioned output parameter. The hybrid method [ENN+FFBP] achieved nearly 14.90 MMRE and 13.49 MMRE, whereas the proposed ECS-DBN achieved 12.50 MMRE by using evolutionary algorithm. The performance measure like Pred. 25% and MSE for existing methods achieved 95.23% and 0.052 in FFBN, whereas 61.96% and 0.13 error rate achieved by GLM method. But the ECS-DBN achieved 100% in Pred. 25% and achieved very low error rate as 0.043 in MSE.

CONCLUSION

We have analyzed the results of Linear regression and K-nearest neighbor's machine learning techniques in this work. We have applied these techniques on three publicly available datasets (COCOMO81, COCOMONASA, and COCOMONASA_2) for predicting software development effort. The model with the lower Root- Mean- Square Error(RMSE), Relative- Absolute- Error (RAE), Relative Root- Square- Error(RRSE), Mean- Absolute- Error (MAE), and the higher Correlation Coefficient (25) has been considered to be the best among others. From the results what we have got and presented Linear Regression model is a good estimator compared to K-nearest neighbors on the data sets COCOMO81, COCOMONASA, COCOMONASA_2 by having higher correlation coefficient value and low RMSE, RAE, RRSE, MAE

REFERENCES

1. *Ahmed BaniMustafa, Predicting Software Effort Estimation Using Machine Learning Techniques, https://www.researchgate.net /publication/331472905*

2. Petrônio L. Braga and Adriano L. I. Oliveira, Silvio R. L. Meira , Software Effort Estimation using Machine Learning Techniques with Robust Confidence Intervals , eventh International Conference on Hybrid Intelligent Systems, 2007 IEEE.

3. Vlad-Sebastian Ionescu, An approach to software development effort estimation using machine learning, 2017 IEEE

4. Miyoung Shin and Amrit L. Goel, Empirical Data Modeling in Software Engineering Using Radial Basis Functions , IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 26, NO. 6, JUNE 2000.

5. Younghee Kim, Keumsuk Lee, A Comparison of Techniques for Software Development Effort Estimating, SYSTEM INTEGRATION 2005.

6. Omar Hidmi and Betul Erdogdu Sakar, Software Development Effort Estimation Using Ensemble Machine Learning, Int'l Journal of Computing, Communications & Instrumentation Engg. (IJCCIE) Vol. 4, Issue 1 (2017) ISSN 2349-1469 EISSN 2349-1477.

7. Sonam Bhatia, Varinder Kaur Attri, Implementing Decision Tree for Software Development Effort Estimation of Software Project, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 5, May 2015, ISSN(Online): 2320-9801 , ISSN (Print): 2320-9798

8. Mohd. Sadiq, Aleem Ali, Syed Uvaid Ullah, Shadab Khan, and Qamar Alam, Prediction of Software Project Effort Using Linear Regression Model, International Journal of Information and Electronics Engineering, Vol. 3, No. 3, May 2013.

9. Sonam Bhatia, Varinder Kaur Attri, MACHINE LEARNING TECHNIQUES IN SOFTWARE EFFORT ESTIMATION USING COCOMO DATASET, IJRDO - Journal of Computer Science and Engineering, Volume-1 | Issue-6 | June,2015 | Paper-13, ISSN: 2456-1843.

10. Abdelali Zakrani, Mustapha Hain, Abdelwahed Namir , Software Development Effort Estimation Using Random Forests: An Empirical Study and Evaluation , International Journal of Intelligent Engineering and Systems, Vol.11, No.6, 2018.