Review on Software Effort estimation by machine learning Approaches

Er Meenakshi[1],Er. Sumeet kaur Sehra[2]
Lecturer, GTBGARH, CSE, Punjab
Assistant Professor, GNE, CSE, Punjab

Abstract - In the field of software development, software project estimation is the most challenging task. If there is no proper and reliable estimation provided in the software development, there will be no proper arrangement as well as control of the project. Software project estimation is necessary to handle underestimates and overestimates in terms of cost, effort etc. Small projects are not difficult to estimate and accuracy can be improved by traditional approach of Expert judgment. As the measure of project size increases i.e. for embedded and large-scale projects, precision and accuracy become important concern.

I. INTRODUCTON

Software project estimation is the most challenging task if there is no proper and reliable estimation provided in the software development, there will be no proper arrangement as well as control of the project. Even when all the important factors are taken into consideration during the software development process still projects are not accurately estimated. It doesn't utilize estimates for improving the development of software. When a project is underestimated the effects such as under-staffing, under-scoping the quality assurance effort and missing the deadlines resulting in loss of credibility are seen

For the purpose of project estimation, these four steps are considered [20]:

1) Development product's size estimation: There are Lines of Code (LOC) as well as Function Points (FP) which help in this estimation. However, various other methods are also used for measuring the estimation such as Use case points (UCP), Story points etc. There are certain merits as well as demerits of this estimation.
2) Effort estimation in terms of person-month or person-hours.
3) Scheduling estimation for months of a calendar.
4) Project cost estimation in dollars or any other local currency.

- **Estimating size**

The first step towards achieving an effective project estimate is to provide an accurate size estimate of the software. Along with the formal descriptions of the requirements for project , scope of size estimation might start.

- **Estimating effort**

After receiving the size estimation of a product, it becomes very easy to estimate the effort of it. When the software development lifecycle of a project is defined only then the conversion from software size to total project effort estimation is possible.

- **Estimating schedule**

The third important step in software development project is the determination of project schedule from the effort estimate. The number of people working on a project, the type of work they will do, the starting time and ending time of the project are the factors that are to be involved here.

- **Estimating Cost**

Total cost of the project can be estimated through various components such as labor, hardware and software or rentals, travel for meeting or testing related issues, telecommunications, instructional classes, office area, etc. All of the enlisted factors affect the estimation of effort in one way or the other.

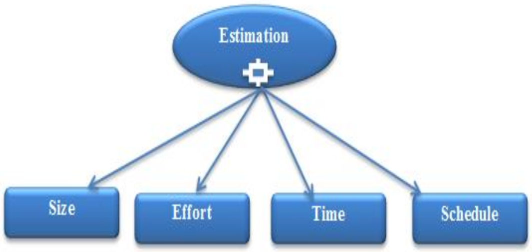

Figure 1.1 Software estimation Modules

II. RELATED STUDY

This section presents the study related to the proposed work done by the

A) Fuzzy Logic Based Estimation

Anish Mittal et al. has proposed an Enhanced Fuzzy system for enhancing estimations of COCOMO model by incorporating Triangular Fuzzy function. The results were evaluated for a firm dataset and were promising **[1]**.

Sangwan et al**.** has presented an analysis of various machine learning techniques used in software effort estimation. The machine learning techniques employed so far are based on Artificial Neural Network, Fuzzy Logic, Analogy Based Estimations, Genetic Algorithm and other techniques. The paper highlights relevance of each techniques depending upon its own nature and environment in which it employed [2]

B) **Artificial Neural Networks (ANN) Based Estimation**

Ali Bou Nassif et al. presented Log Linear Regression (LLP) and Multi-Layer Perceptron Models (MLP) for effort estimation of software based on Use case points. The result of the two models shows their performance was better for small projects than for large sized projects [3].

Ali Bou Nassif et al. proposed a Treeboost Model based on stochastic gradient boosting which employed Use case points (UCP). Software size in the form of UCP's, complexity and productivity were inputted to the Treeboost model and the results were evaluated against the Multiple Linear Regression model. The Treeboost Model results were promising. The performance metrics used are Mean magnitude relative error (MMRE), Prediction (PRED), Mean Square Error (MSE) and Median of Magnitude of Relative Error (MdMRE) [4]. Shashank Mouli Satapathy et al. proposed a machine learning techniques based model of Stochastic Gradient Boosting (SGB) to improve the software effort estimates using class point approach. The effort parameters are improved by using SGB model resulting in improved prediction accuracy. A comparative analysis of SGB along with Multi-Layer Perceptron and the Radial Basis Function Network has been presented [5].

Shashank Mouli Satapathy et al. proposed an approach which is based on use case points (UCP) and random forest classifier for effort estimation. In the proposed scheme, the UCP approach has estimates the UCP's considering the size, complexity, productivity and also the actual effort values of 149 projects. The random forest classifier has classified impact of each of factors. The random forest classifies features by building decision trees and it is implemented by considering various case studies. The simulation results shows that proposed approach performs well in terms of Mean magnitude relative error (MMRE) and Prediction (PRED) performance metrics [6]. Poonam Rijwani et al. presented a Multilayer Feed Forward (MLFFN) Artificial Neural Network Technique for enhanced effort estimation. COCOMO 81 dataset was employed consisting of 63 software projects. The validation method used was EpochesUntil tolerance level>.999. The results provided reduced MRE in MLFFN than in COCOMO [7].

C) OTHER TECHNIQUES USED FOR ESTIMATION

García-Floriano, Andrés, et al. has discussed the concept of support vector regression to predict software enhancement effort. This method is used for accuracy prediction of SVR. SVR uses radial basis function, linear, polynomial and sigmoid kernels. The proposed method is tested on 5 datasets which are based on development platform, data quality and levels of effort recording. The results of the paper show that kernel-SVR performs better than existing methods [8]. Pospieszny, et al. has proposed Machine learning algorithm for software project estimation and duration estimation. In this work Neural Networks, Support Vector Machine and generalized linear models are used. This method is used to enhance the project success rates and project management process. It gives good accuracy rate in prediction and suitability for deployment [9].

Arora, Shaina, et al. has used artificial neural network for the cost estimation of the software. Cost estimation of the software is required because it set the schedules and assets. The overall growth of the software is depending on the estimation of cost, time and resources so it is necessary to use good estimation method. COCOMO model is used with multi-layer feed forward neural network system. The result of the proposed model is compared with existing methods and performs better in prediction [10]. Wani, Zahid Hussain et al. has proposed Software effort estimation is a process in which it deals with the estimation of time and effort used in software development. The delivery of the software is depends on the good prediction of the efforts, resources and cost. This prediction belongs to procedure for input and neural network removes the irrelevant cost driver which leads to accurate prediction of cost. Performance evaluation is done on the basis of Relative error and median of magnitude of relative error [4]. Lélis, Cláudio et al. worked upon on estimating the effort on software maintenance. This model is based on the calculation, visualization elements and the integration with change request repositories. The experiment is based on the qualitative and quantitative data. The feasibility of the proposed method is shown clearly by statistical analysis [11].

III. LATEST WORK ON ESTIMATION

1. García-Floriano, Andrés, et al.	2018	Support vector Machine	The proposed method is tested on 5 datasets which are based on development platform, data quality and levels of effort recording. The results of the paper show that kernel-SVR performs better than existing methods.
2 Pospieszny, et al.	2018	Neural Network	This method is used to enhance the project success rates and project management process. It gives good accuracy rate in prediction and suitability for deployment.
3. Arora, Shaina, et al.	2018	COCOMO	COCOMO model is used with multi-layer feed forward neural network system. The result of the proposed model is compared with existing methods and performs better in prediction.
4. Wani, Zahid Hussain, et al.	2018	Hybrid Model	This prediction belongs to procedure for input and neural network removes the irrelevant cost driver which leads to accurate prediction of cost. Performance evaluation is done on the basis of Relative error and median of magnitude of relative error.
5 Lélis, Cláudio AS, et al.	2018	Reputation Based approach	The experiment is based on the qualitative and quantitative data. The feasibility of the proposed method is shown clearly by statistical analysis.
6 Chen, Xiang, et al.	2018	Multi objective effort	Coefficient vector is also denoted by each solution. Performance evaluation is done by using cross validation method, time-wise validation and cross-project validation.
7 Chen, Jianfeng, et al.	2018	SWAY Method	This is a competitive process and gives better results when compared with existing methods. It works very effectively on models that are very slow to execute.
8 Kumari, Sweta, et al.	2017	CUCKOO Method	This model improved the performance by using Mean Magnitude of relative error and prediction.
9 . Tanveer, Binish et al.	2017	machine learning algorithms	The machine learning algorithms have outperformed and provide better accuracy due to their learning natures.
10 Padmaja, M., et al.	2017	Grey relational algorithm	In this paper Grey relational analysis is performed for normalization and coefficient calculation then it calculates the gray grading, ranking and effort prediction.
11. Dragicevic, et al.	2017	Bayesian Network	In this work statistics is used for assess the precision of the model. This model predicts the magnitude of relative error, prediction level at m, accuracy and mean absolute error. The results of the paper show the effectiveness in accuracy.
12. Shi, Ying, et al.	2017	Metrics based approach	This experiment is performed on the safety critical software deployed in nuclear power plant. Result of the approach is calculated using software metrics at different stages of the software development cycle.

IV. CONCLUSION

Software effort estimation is a challenging issue in the software development process. There are various methods that are proposed by the researchers to solve this issue. In this thesis accuracy of the prediction is improved by feature selection and Machine Learning approach. Total cost of the project can be estimated through various components such as labor, hardware and software or rentals, travel for meeting or testing related issues, telecommunications, instructional classes, office area, etc. All of the enlisted factors affect the estimation of effort in one way or the other.

V. REFERENCES

[1]. Mittal, A., Parkash, K., & Mittal, H. (2010). Software cost estimation using fuzzy logic. *ACM SIGSOFT Software Engineering Notes*, *35*(1), 1-7.

[2]. Sangwan, O. P. (2017, January). Software effort estimation using machine learning techniques. In *Cloud Computing, Data Science & Engineering-Confluence, 2017 7th International Conference on* (pp. 92-98). IEEE.

[3]. Nassif, A. B., Ho, D., & Capretz, L. F. (2013). Towards an early software estimation using log-linear regression and a multilayer perceptron model. *Journal of Systems and Software*, *86*(1), 144-160.

[4]. Nassif, A. B., Capretz, L. F., Ho, D., & Azzeh, M. (2012, December). A treeboost model for software effort estimation based on use case points. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on* (Vol. 2, pp. 314-319). IEEE.

[5]. Satapathy, S. M., Acharya, B. P., & Rath, S. K. (2014). Class point approach for software effort estimation using stochastic gradient boosting technique. *ACM SIGSOFT Software Engineering Notes*, *39*(3), 1-6.

[6]. Satapathy, S. M., Acharya, B. P., & Rath, S. K. (2016). Early stage software effort estimation using random forest technique based on use case points. *IET Software*, *10*(1), 10-17.

[7]. P. Rijwani and S. Jain. “Enhanced Software effort estimation using Multi Layer Feed Forward Artificial Neural Network Technique,” Procedia Computer Science, vol 89, pp. 307-312.

[8]. García-Floriano, Andrés, et al. "Support Vector Regression for Predicting Software Enhancement Effort." *Information and Software Technology* (2018).

[9]. Pospieszny, Przemyslaw, Beata Czarnacka-Chrobot, and Andrzej Kobylinski. "An effective approach for software project effort and duration estimation with machine learning algorithms." *Journal of Systems and Software* 137 (2018): 184-196.

[10]. Arora, Shaina, and Nidhi Mishra. "Software Cost Estimation Using Artificial Neural Network." *Soft Computing: Theories and Applications*. Springer, Singapore, 2018. 51-58.

[11]. Wani, Zahid Hussain, and S. M. K. Quadri. "Software Cost Estimation Based on the Hybrid Model of Input Selection Procedure and Artificial Neural Network." *Artificial Intelligent Systems and Machine Learning* 10.1 (2018): 18-24.

[12]. Lélis, Cláudio AS, et al. "AD-Reputation: A Reputation-Based Approach to Support Effort Estimation." *Information Technology-New Generations*. Springer, Cham, 2018. 621-626.

[13]. Chen, Xiang, et al. "MULTI: Multi-objective effort-aware just-in-time software defect prediction." *Information and Software Technology* 93 (2018): 1-13.

[14]. Chen, Jianfeng, et al. " Sampling" as a Baseline Optimizer for Search-based Software Engineering." *IEEE Transactions on Software Engineering* (2018).

[15]. Kumari, Sweta, and Shashank Pushkar. "Software cost estimation using cuckoo search." *Advances in Computational Intelligence*. Springer, Singapore, 2017. 167-175.

[16]. Tanveer, Binish, Liliana Guzmán, and Ulf Martin Engel. "Effort estimation in agile software development: Case study and improvement framework." *Journal of Software: Evolution and Process* 29.11 (2017).

[17]. Padmaja, M., and Dr D. Haritha. "Software Effort Estimation using Grey Relational Analysis." *MECS in International Journal of Information Technology and Computer Science*(2017).

[18]. Dragicevic, Srdjana, Stipe Celar, and Mili Turic. "Bayesian network model for task effort estimation in agile software development." *Journal of Systems and Software* 127 (2017): 109-119.

[19]. Shi, Ying, et al. "Metric-based software reliability prediction approach and its application." *Empirical Software Engineering*22.4 (2017): 1579-1633.