Analyzing Effect of Ensemble Models on Multi-Layer Perceptron Network for Software Effort Estimation

Suyash Shukla
Dept. of Computer Science and Engg.
Indian Institute of Technology
Roorkee, India
suyashshukla2811@gmail.com

Sandeep Kumar
Dept. of Computer Science and Engg.
Indian Institute of Technology
Roorkee, India
sgargfec@iitr.ac.in

Pravas Ranjan Bal
Dept. of Computer Science and Engg.
Indian Institute of Technology
Roorkee, India
pravasranjan90@gmail.com

***Abstract*—Effort Estimation is a very challenging task in the software development life cycle. Inaccurate estimations may cause client dissatisfaction and thereby, decrease the quality of the product. Considering the problem of software cost and effort estimation, it is conceivable to call attention to that the estimation procedure considers the qualities present in the data set, as well as the aspects of the environment in which the model is embedded. Existing literature have the instances where machine learning techniques have been used to estimate the effort required to develop any software. Yet it is quite uncertain for any particular model to perform well with all the data sets. In this paper, Multi-Layer Perceptron (MLPNN) and its ensembles are explored in order to improve the performance of software effort estimation process. Firstly, MLPNN, Ridge-MLPNN, Lasso-MLPNN, Bagging-MLPNN, and AdaBoost-MLPNN models are developed and, then, the performance of these models are compared on the basis of R^2 score to find the best model fitting this dataset. Results obtained from the study demonstrate that the R^2 score of AdaBoost-MLPNN is 82.213%, which is highest among all the models.**

***Index Terms*—Machine Learning, Software Metrics, Predictive Model, Effort Estimation**

I. Introduction

The most challenging task in software effort estimation is to manage the complexity in software development. Thus, the managers should consider the following things before making investments: advantages of that project, the expenses caused by that project, the dangers and the future open doors that the projects will make [1]. The estimation of the effort required to develop software is one of the most established and most significant issues in the administration of software projects [2]. In the past, different algorithmic and nonalgorithmic techiques have been used for software effort estimation.

In literature, machine learning techniques have been used for software effort estimation. Ayyldz et al. [3], performed a study on Desharnais dataset in order to find which attributes are influencing the performance of software effort estimation process and also assessed the applicability of regression models for estimating software effort. Kocaguneli et al. [4], have developed a software estimation model for a multinational bank of the Turkish Subsidiary. They have implemented six different models for effort estimation and concluded that the SVM model outperforms every other model. Kitchenham et al. [5], Pospieszny et al. [6], and Huang et al. [7] also implemented various machine learning models and there ensembles for software effort estimation.

From the literature, we have found that Neural Networks and its ensemble models have not been explored for software effort estimation until now. In this manner, for estimating the software effort accurately, we mean to address two research questions:

RQ1: How the ensembles of the MLPNN model are performing for software effort estimation?

RQ2: How much improvement the ensemble models are generating over MLPNN for software effort estimation?

So, in this paper, the performance of MLPNN and its ensemble models have been evaluated on the basis of R^2 score to find the best model fitting this dataset.

II. Analysis Procedure and Methods

Desharnais [8] dataset from the PROMISE software engineering repository has been used to perform this study. This dataset contains 81 software projects from a Canadian software company. Firstly, the correlation of each variable in the dataset is analyzed with the effort variable by Pearson correlation coefficient (PCC). If the value of PCC lies between 0.5 to 1 then there is a high correlation among those attributes.

In Desharnais dataset, each project has 12 attributes. After PCC analysis, we have found that the Length, Transactions, Entities, Point Adj, and Non-Point Adjust are the most influential attributes. After finding most influential attributes, MLPNN, Ridge-MLPNN, Lasso-MLPNN, Bagging-MLPNN, and AdaBoost-MLPNN models have been applied to find the best model fitting this dataset. Fig. 1 shows the methodology used for the software effort estimation process.

III. Results and Observations

In the section, we have discussed the results obtained through this study on Desharnais data set. The performance of MLPNN, Ridge-MLPNN, Lasso-MLPNN, Bagging-MLPNN,

978-1-7281-3851-0/19/$31.00 ©2019 IEEE
DOI 10.1109/SERVICES.2019.00116

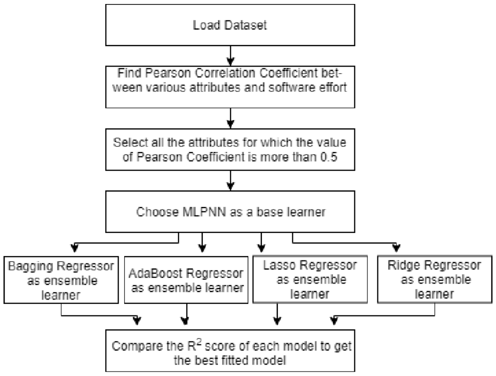

Fig. 1: Methodology used for performing study

and AdaBoost-MLPNN have been evaluated as shown in Table 1. The illustration of each correlated feature based upon the values after experimenting for 100 times with different hyper parameters is shown in Fig. 2.

Algorithmic Models	R^2 Score
MLPNN Model	0.78301
Ridge-MLPNN Ensemble Model	0.79597
Lasso-MLPNN Ensemble Model	0.78991
Bagging-MLPNN Ensemble Model	0.81096
AdaBoost-MLPNN Ensemble Model	**0.82213**

TABLE I: Performance evaluation of techniques used

Based upon the experiment results, following are the answers of Research questions raised in section 1.

RQ1: The R^2 score for AdaBoost-MLPNN model is 0.82213, which is highest among all the models, whereas the R^2 score of MLPNN is 0.7830. Bagging-MLPNN and Ridge-MLPNN are the second and third best models, respectively.

RQ2: The performance of Lasso-MLPNN and Ridge-MLPNN ensemble models are almost similar to individual MLPNN model, whereas Bagging-MLPNN and AdaBoost-MLPNN models have shown only a minor improvement in R^2 score.

Conclusion and Future Work

In this work, five different machine learning techniques have been compared for the estimation of effort required to develop any software. For the given datset, it has been observed that only a minor improvement is achieved in R^2 score by using ensembles of MLPNN. The R^2 score of AdaBoost-MLPNN is 82.213% , which is highest among all the models, whereas the R^2 score of MLPNN is 78.3%.

In the future, larger data sets can be used in order to generalize the results of this study. Other types of neural networks can also be used to create models for effort estimation, because of their adaptability and non-parametric capacity.

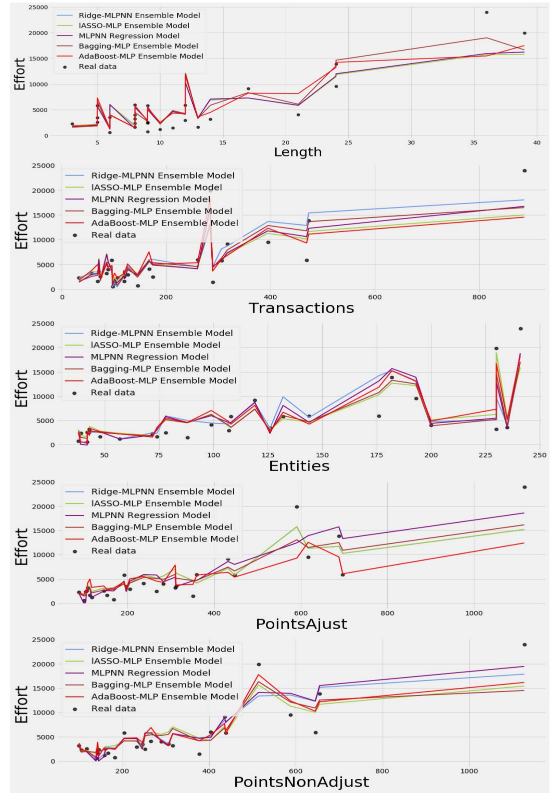

Fig. 2: Comparison of techniques based on error values

Acknowledgment

This publication partially presents the outcome of the work undertaken in the project under Visvesvaraya PhD Scheme of MeitY, Government of India, being implemented by Digital India Corporation.

References

[1] A. G. Floriano, C. L. Martin, C. Y. Marquez, and Alain Abran,*Support vector regression for predicting software enhancement effort*, Information and Software Technology, 2018, vol. 97, pp. 99–109.

[2] H. Huijgens, A. V. Deursen, L. L. Minku, and C. Lokan, *Effort and cost in software engineering: A comparison of two industrial data sets*, In Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering (EASE17), 2017, pp. 51–60.

[3] T. E. Ayyldz and H. C. Terzi, *Case study on software effort estimation*, International Journal of Information and ElectronicsEngineering, 2017, vol. 7, 3, pp. 103–107.

[4] E. Kocaguneli, A. Tosun, and A. Bener, *Ai-based modelsfor software effort estimation*, 36th EUROMICRO Conferenceon Software Engineering and Advanced Applications, 2010, pp. 323–326.

[5] B. Kitchenham, S. L. Pfleeger, B. McColl, and S. Eagan, *An empirical study of maintenance and development estimation accuracy*, Journal of systems and software, 2002, vol. 64, 1, pp. 57–77.

[6] P. Pospieszny, B. Chrobot, and A. Kobylinski, *An effective approach for software effort and duration estimation with machine learning algorithms*, Journal of Systems and Software, 2018, vol. 137, pp. 184–196.

[7] J. Huang, Y. F. Li, J. W. Keung, Y. T. Yu, and W. K. Chan, *An empirical analysis of three-stage data-preprocessing foranalogy-based software effort estimation on the isbsg data*, IEEE International Conference on Software Quality, Reliability andSecurity (QRS), 2017, pp. 442–449.

[8] M. J. Shepperd and C. Schofield, *Estimating software project effort using analogies*, IEEE Transactions on Software Engineering, 1997, vol. 23, pp. 736–743.