2019 IEEE World Congress on Services (SERVICES)

# Applicability of Neural Network based Models 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

***Abstract*—Effort Estimation is a very challenging task in the software development life cycle. Inaccurate estimations may cause the client dissatisfaction and thereby, decrease the quality of the product. Considering the problem of software cost and effort prediction, 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 literatures have the instances where machine learning techniques such as Linear Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbor (KNN) 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. Most of the research is based on the dataset of any single organization. Consequently, the results obtained through these models cannot be generalized. So, the main objectives of this research are: i) to use different data preparation techniques such as selection, cleaning, and transformation to improve the quality of data set given to the model ii) to use other machine learning models such as Multi-Layer Perceptron Neural Network (MLPNN), Probabilistic Neural Network (PNN), and Recurrent Neural Network (RNN) to increase the performance of software effort estimation process iii) to use different optimization techniques to tune the parameters of machine learning models iv) to use ensemble methods to improve the accuracy of software effort estimation process. In this study, first, we found out the most influential attributes in the Desharnais data set, then, MLPNN has been applied on reduced data set with to improve the accuracy of software effort estimation. Then, the performance of the MLPNN model is compared with LR, SVM and KNN models in the literature to find the best model fitting this dataset. Results obtained from the study demonstrate that some of the variables are more important in comparison to others for effort estimation. Also among the various models used in this study, the best-obtained $R^2$ value is 79 % for the MLPNN model.**

***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 projects 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 [5]. 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 [6]. In software projects, the cost and effort estimates are done with the help of historical data, which is very useful during the starting phases of development [10]. In the past, different approaches such as expert judgment method, analogy based estimation method, and algorithmic models have been proposed to estimate the effort required to develop any software. So, the main objective here is to estimate the effort required to develop software either with the help of algorithmic models or non algorithmic models [8].

In literature, machine learning techniques have been used for software effort estimation but no single technique is applicable in all the situations. So, the principal objectives of this research work are :

1) To use different data preparation techniques such as selection, cleaning, and transformation to improve the quality of data set given to the model
2) To use other machine learning models such as MLPNN, PNN, and RNN to increase the performance of software effort estimation process
3) To use different optimization techniques to tune the parameters of machine learning models
4) To use ensemble methods to improve the accuracy of software effort estimation process.

In this study, we concentrated on making a prediction model for software effort estimation utilizing the Desharnais dataset. In addition, we investigate the significance of several variables present in the dataset for the formation of this model, as well as how these variables correspond with one another. The correlation coefficient helps to measure the propensity of two variables to change their values depends upon there relationship. The value of the Pearson correlation coefficient lies between -1 and 1. An outcome of - 1 implies that there is a negative relationship between the two factors, while an outcome of 1 implies that there is a connection between the two factors. Results somewhere in the range of 0.5 and 1.0 demonstrate high relationship.

## II. Related Work

In this section, we have discussed the studies and details related to software effort estimation.

Ayyldz et al. [6], performed a study on Desharnais dataset in order to find which attributes are influencing the performance of software effort estimation process. They have used the

978-1-7281-3851-0/19/$31.00 ©2019 IEEE
DOI 10.1109/SERVICES.2019.00094

Pearson correlation coefficient to find the relationship between the attributes and software effort. They have also assessed the applicability of regression models for estimating software effort. They have found that the value of squared error is 59.44% and the most influential variable for the effort estimation process is PointsNonAdjust.

Kocaguneli et al. [4], have developed a software estimation model for a multinational bank of the Turkish Subsidiary. They have used historical data from the bank and publicly available data from other sources to construct a machine learning based effort estimation model. On the basis of results obtained through statistical tests, they have found out that the performance of SVR is better among the six algorithms used in this study. Also, the performance increases when the model is used with clustering mechanism.

Kitchenham et al. [7], also uses different models to estimate the effort required to develop any software and then compared the accuracy of proposed models by taking into account the actual values from the industrial dataset and predicted values obtained from the models. They have used data from a single company to perform their experiment. The data set contains more than 145 projects, implemented in Microsoft Access. However, they have used projects in the range of 1994 to 1998 in this study. They have found that 63% of the estimates fall in the range of 25% of actual values and the value of average absolute error is 0.26. The estimates from these models are somewhat better than the estimates of regression models, which are based on adjusted function points. However, the function point models were worked upon the homogeneous part of a complete dataset. The estimates obtained from this study cannot be generalized, because they are using the dataset of a single company.

Pospieszny et al. [8], developed an ensemble model to estimate the effort and duration required to develop a software project. They have used Version R12 of ISBSG dataset which consists of 6006 projects from various companies of 25 countries. They choose Support Vector Machines (SVM), Multi-layer Perceptron Artificial Neural Network (MLP-ANN) and Generalized Linear Models (GLM) to build there model. The results obtained were merged by ensemble averaging, to reduce bias and variance. In the case of individual models, the performance of SVM is better in comparison to MLP-ANN and GLM. The accuracy of ensemble models is less than that of SVM, but the ensemble models can prevent overfitting and underfitting and they can also handle the noise and outliers present in the new input data.

Huang et al. [9], investigated three-stage data preprocessing (DP) techniques for analogy based software effort estimation (ASEE). the three stages of data preprocessing for ASEE are missing data imputation, data normalization, and feature selection. They have considered ISBSG R10 dataset, which consists of 4106 projects for their experiment. They have found that the mean mode imputation is a safer option than kNN imputation when there are several missing values in the dataset but appropriate use of kNN imputation will lead to better results. They have found that the Z-score normalization is better than [0,1] or [1,1].

Existing literatures have the instances where machine learning techniques such as LR, SVM, and KNN 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 our study, MLPNN model is used for effort estimation and the performance of the MLPNN model is compared with LR, SVM and KNN models in the literature to find the best model fitting this dataset.

## III. Analysis Procedure and Methods

Desharnais 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, and then the seven machine learning algorithms have been applied to find the relationship between the most correlated element by Pearson correlation and the effort variable. Then, the prediction performance is analyzed on the basis of error values.

### *A. Dataset Used*

Desharnais [8] dataset consists of 81 projects and each project has 12 variables: Project id, Team Experience, Manager Experience, Year End, Length, Effort, Transaction, Entities, Point Adj, Adjustment, Point Non-Adjust, and Language. In this dataset, basically, nine attributes are numerical, out of which eight attributes are independent namely, Team Experience, Manager Experience, Year End, Length, Transaction, Entities, Point Adj, and Point Non-Adjust. The target attribute is effort.

### *B. Attribute Selection*

As we have already mentioned, the correlation coefficient helps to measure the propensity of two variables to change their values depends upon there relationship. The main objective of this study is to analyze which attributes are strongly related and which influence the estimation process. As mentioned in [8] and [5], a set of characteristics or metrics can be used in order to identify the correlation.

In this study, we have used the Pearson correlation coefficient to show the linear relationship between two variables [13]. Pearson correlation coefficient for different attributes and effort in Desharnais data set is shown in Figure 1. From Figure 1, we can see that the Length, Transactions, Entities, Point Adj, and Non-Point Adjust are the variables whose value is greater than 0.5.

As we have already mentioned, if the value of correlation coefficient lies between 0.5 to 1 then there is a high correlation among those attributes. Thus, we can say that the above-mentioned attributes are the most relevant attributes for model construction.

### *C. Model Construction*

In this study, four machine learning algorithms: LR, SVM, KNN, and MLP have been used. When using predictive models for estimation, it is also required to have historical data with

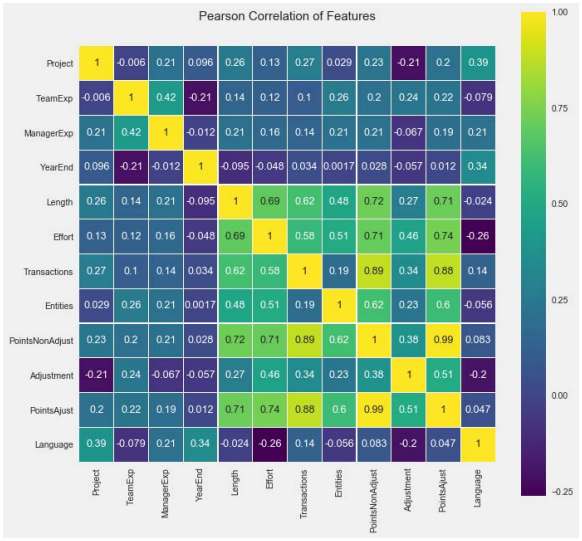


Fig. 1: Pearson correlation for Desharnais data set

the known results to validate the performance of the model [14]. The training of these models have been done by 67% of the instances from Desharnais dataset and the remaining 33% instances have been used to test the performance of these models. Python language with libraries such as Numpy, Pandas, Scikit-learn, Seaborn and Matplotlib has been used to implement these models.

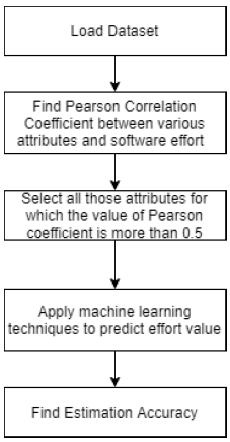


Fig. 2: Methodology used for effort estimation

In regression analysis, we found the relationship of a variable with one or more variables [11]. In linear regression method, the training of the model is done to generate a value of the target variable. Here, the Support Vector Machine is also used to solve this regression problem. In this, we maintain all the major features that characterize the maximum margin algorithm [9]. Similarly, the K nearest neighbor algorithm is also used in this study. In the KNN algorithm, the information from historical projects is used to find the value of the target variable on the basis of similarity.

Lastly, MLPNN is also used in this study. MLPNN is a feed-forward neural network, which consists of several layers such as input layer, hidden layer, and output layer. It uses backpropagation algorithm for the training of the models. MLPNN can be used for both classification as well regression problems. Figure 2, is showing the methodology used for the software effort estimation process.

## IV. Results

In the section, we have discussed the results obtained through this study on Desharnais data set. The performance of LR, SVM, KNN, and MLPNN has been evaluated and it is observed that the MLPNN method is better than the other three models used in this study. The comparison of above-

| Algorithmic Models | $R^2$ Score |
|---|---|
| Linear Regression Model | 0.76801 |
| K-Nearest Neighbour | 0.73799 |
| Support Vector Machine | 0.73591 |
| Multi-Layer Perceptron | 0.79380 |

TABLE I: Performance evaluation of LR, SVM, KNN and MLPNN

mentioned models has been done on the basis coefficient of determination ($R^2$) value. The $R^2$ value find the change in the value of one variable by changing another variable. The value of the coefficient of determination lies between 0 and 1. The value 0 indicates that no relationship exists between two variables, while the value 1 indicates that the two variables are strongly related.

The illustration of each correlated feature is shown in Figure 3. By looking at the Figure 3, it is visible that the performance of the MLPNN model is better than the other models with 79% successful estimates and there exists a difference of 6-7% with other models.

## V. Work Plan

Results obtained from the study suggest, the performance of the MLPNN model is better than the other models with 79% successful estimates and there exists a difference of 6-7% with other models. Motivated from these results, we are planning to use other neural network models, there variants and ensemble of these models for software effort estimation.

Figure 4, is showing the work plan to fulfill our objectives. In this, first the raw data is collected from various sources, then, we perform missing data analysis and feature extraction. After feature extraction, dimensionality reduction has to be performed in order to reduce the dimensions of the data set. Then, we are planning to apply various individual machine learning algorithms as well as ensemble models to improve the performance of software effort estimation. Lastly, the parameters of different ML algorithms have to be tune in order to improve accuracy.

## VI. Conclusion

In this work, four different machine learning techniques have been compared for the estimation of effort required to

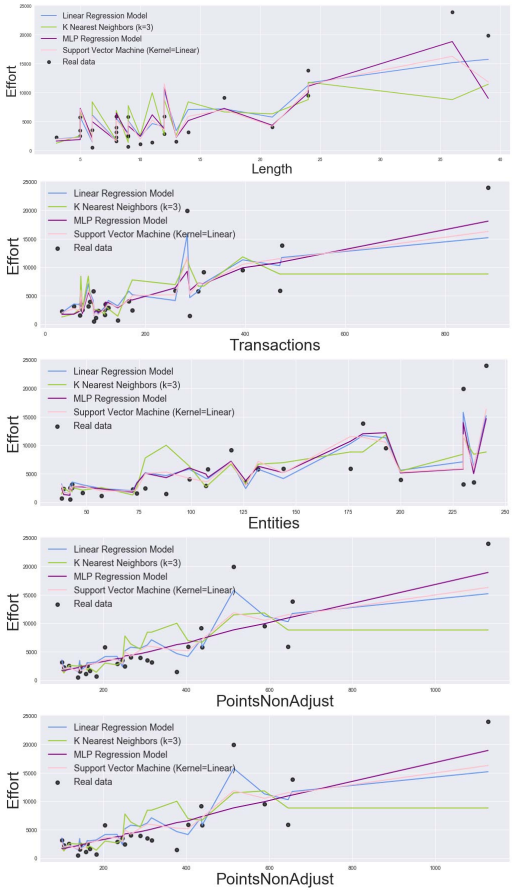


Fig. 3: Comparison of errors of KNN, SVM, LR and MLPNN techniques

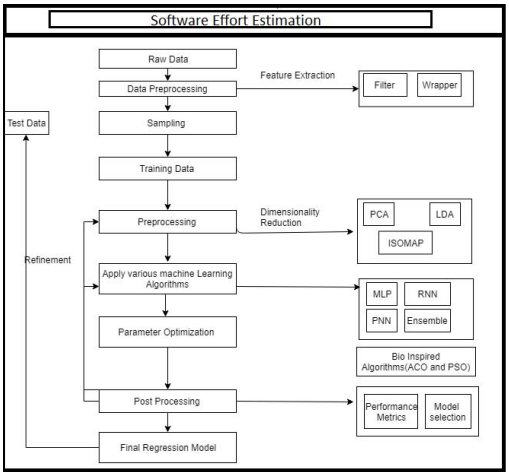


Fig. 4: Work Plan for software effort estimation

develop any software. The data from previously completed projects, more specifically Desharnais data set is been used to perform this experiment.

The proposed methodology used in this paper consists of four machine learning models namely, LR, SVM, KNN, and MLPNN. Thus, the results of these models are compared in order to find which model is more feasible for the estimation of software effort.

In addition to these models, we have used the Pearson correlation coefficient to find out the relationship between different attributes and software effort in Desharnais dataset and we have found that the Length, Transactions, Entities, Point Adj, and Non-Point Adjust are the most influential attributes in comparison to other attributes for effort estimation.

The results obtained through our study show that the best $R^2$ value is 0.79380 for MLPNN model, which means 79% of the estimates are successful and there exists only 6-7% difference among the models.

## ACKNOWLEDGMENT

Thanks to Indian Institute of Technology Roorkee for institute assistantship and the funding under Visvesvaraya PhD Scheme of MeitY, Government of India being implemented by Digital India Corporation.

## REFERENCES

[1] A. Trendowicz and R. Jeffery, Software Project Effort Estimation, Foundations and Best Practice Guidelines for Success, Constructive Cost Model-COCOMO pags, 2014, pp. 277–293.

[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] R. Malhotra, Empirical Research in Software Engineering: Concepts, Analysis, and Applications. CRC Press, 2015.

[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] S. Boslaugh and P. A. Watters, Statistics in a nutshell: a desktop quickreference, OReilly Media Inc.: Sebastopol, CA, 2008.

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

[7] 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.

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

[9] 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.

[10] J. L. Rodgers and W. A. Nicewander, Thirteen ways to lookat the correlation coefficient, The American Statistician, 1988, vol. 42, 1, pp. 59–66.

[11] Jun Ye, Correlation coefficient of dual hesitant fuzzy sets and itsapplication to multiple attribute decision making, Applied Mathematical Modelling, 2014, vol. 38, 2, pp. 659–666.

[12] B. Sigweni, M. Shepperd, and T. Turchi, Realistic assessment of software effort estimation models, In Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, 2016, pp. 41–48.

[13] D. C Montgomery and G. C. Runger, Applied statistics and probability for engineers, John Wiley and Sons, 2010.

[14] 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.

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