Novel Approach on Feature Weighting and Grey Wolf Optimization for Effort Estimation

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

Abstract- In this thesis feature set is increased by adding Line of code (LOC). By using these features effort information is improved and help in software development. In second part features are selected by using Grey wolf optimization (GWO) algorithm. In both works random forest and sampling with boosting and bagging method is used which improves the random forest training model. In this work Random forest with GWO and Random forest without GWO is compared with parameter Accuracy, Precision and Recall.

I. INTRODUCTION

Software project estimation is necessary to handle underestimates and overestimates in terms of cost, effort etc. [19]. If a project is given more number of resources than it actually requires the resources will be utilized by it but the cost of the product increases, due to this reason deployment of estimation techniques is essential.

Software cost estimation model is a backhanded measure, which is utilized by software personnel to predict the cost of a project. The development of software product shifts depending upon the earth in which it is being developed. For projects with familiar environment it is anything but difficult to predict the cost of the project [24].

For the organization to develop a cost estimation model the following things are required.

- List important or critical cost drivers.
- Prepare a scaling model for every cost driver.
- Find projects with similar environments.
- Compare the project with previous familiar projects.
- Evaluate the project whether it is feasible inside the budget constraints.
- Incorporate the critical features in an iterative manner.

Cost drivers are those critical features which affect the project. The cost drivers may vary the cost of building a project. The most important cost driver is size of the project. Size of the project is measured in Kilo lines of code (KLOC). Function points are the empirical measurement to measure size of the project.

II. LITERATURE REVIEW

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 [1].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 [2].

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 [3].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 [5].

Chen, Xiang, et al.proposed multi-objective optimization which is based on supervised method MULTI to build JIT-SDP model. In this method identification of buggy changes and efforts are minimized by using another object. Logistic regression is used to build the model and generate non-dominated solutions. Coefficient vector is also denoted by each solution. Performance evaluation is done by using cross validation method, time-wise validation and cross-project validation [6]. Chen, Jianfeng, et al.sampling method is used

for baseline optimizer which solves the search based problem of software engineering. In this paper SWAY method is proposed to solve the problem and provides better solutions. 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 [7].

Anish Mittal et al.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**[8]**.Sangwanet 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 [9]

S. M. et al. presented a meta-heuristic optimization algorithm for predicting effort estimation for developing project. The paper presents an approach to optimize the Constructive Cost Estimation Model (COCOMO) using Meta-heuristic Harmony Search Algorithm. The NASA dataset was used to evaluate the model. The Proposed model optimized the Mean Magnitude Relative Error (MMRE) to nearly 21% [10].

III. PROPOSED METHODOLOGY

This section of the paper describes the proposed methodology of the work in detail by using flow chart. The proposed works is done by applying the Grey Wolf Optimization and then apply Random Forest on the optimized output given by the GWO. The bagging and boosting process is applied on the trees and then classifier model is used for analysis.

ALGORITHM USED

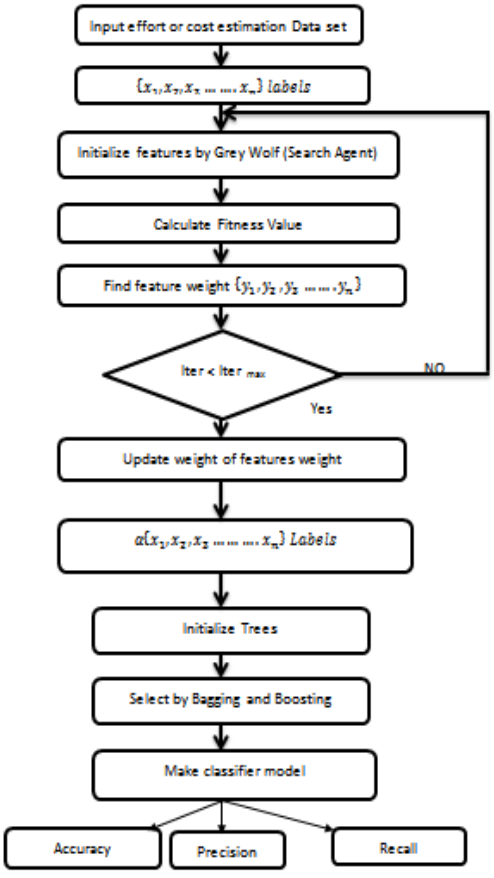
A) Grey Wolf Optimization: It is a population based meta-heuristics algorithm which is used to simulate the leadership hierarchy and hunting mechanism of grey wolves in nature. The main concept of grey wolf optimization algorithm is to simulate the behavior of grey wolf which are living in a pack. We use to assume fitness solution as alpha, beta and delta. Alpha is known as the level leaders and is responsible for decision making in the pack. The wolf pack persistence is based on the decision of alpha. Beta is known as the second level subordinate wolves. The beta operation is for help in making the decision for alpha or other activities. Delta is known as the third level subordinate wolves. This category member consists of elders, scouts, hunters, caretakers, and sentinels. For region boundary observation and in any danger case, scouts are liable for the warning. The protection and pack's safety guarantee is given by sentinels. The expertise wolves are the elders, denoted as alpha or beta. Alphas and betas are helped by hunters while prey hunting and caring for the ill, weak, and wounded wolves by caretakers and providing food for a pack. Omega is the lowest level. All dominant wolves with omega wolves have to comply.

B) Random Forest

Random forest is a learning method for classification, regression and generating the multitude of decision trees. It generates the multitude at the time of training and output of the class. It provides the high accuracy and learning is very fast in it. It works very effectively on the large size database. It easily handles the large size input variables without variable deletion.

Methodology

1. Input the effort or cost estimation Data set.
2. Initialize the features by Grey wolf search agent.
3. Calculate the fitness value.
4. Find the features weight$\{y_1, y_2, y_3 \dots \dots . y_n\}$.
5. Check the Iter < Iter $_{Max}$ if yes go to next step otherwise go to step 4.
6. Update the weight of the features.
7. Initialize the tree after labeling.
8. Select by Bagging and Boosting and make the model for the classification.
9. Analysis the accuracy, precision and recall.



IV. RESULTS AND DISCUSSION

The results of the proposed work are explained in this section. This section also discussed the comparison of the proposed work with the existing approaches for evaluation. The parameters used in this work are precision, recall and accuracy.

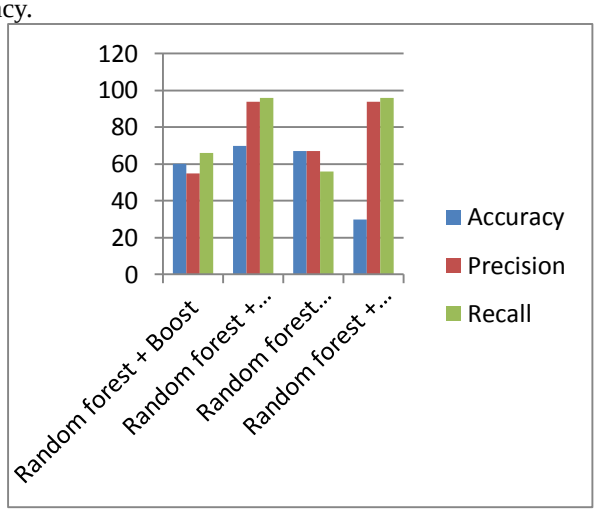


Fig.1: Comparison graphs of classifiers

Figure.1 depicts the comparison of the Random forest + Boost, Random forest + Boost+ GWO, Random forest +Bagging+ GWO and Random forest + Bagging classifiers. The effective result shown by Random forest + Boost+ GWO classifier.

2 Random Forest Regressions

Random Forest Regression	Accuracy
RF+ GWO	78
RF	50.10

A. Random Forest

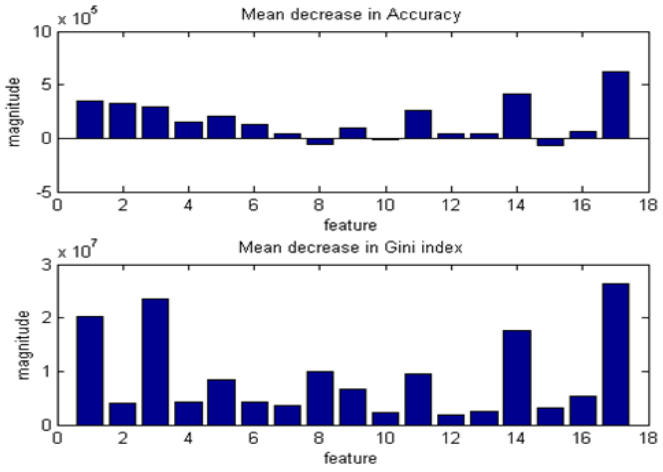


Fig.2: Mean Decreases in Accuracy

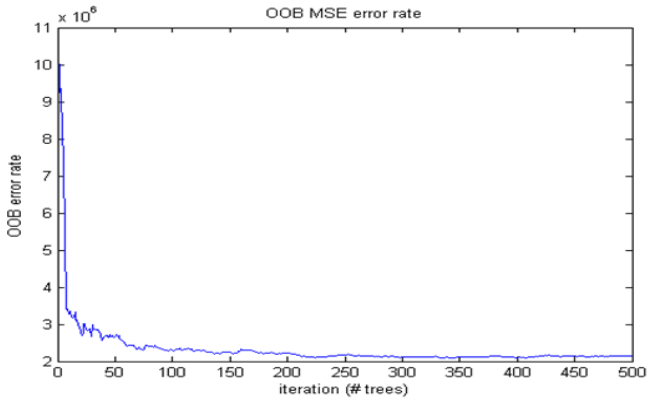


Fig.3: OOB MSE error rate

B. Random forest+ GWO

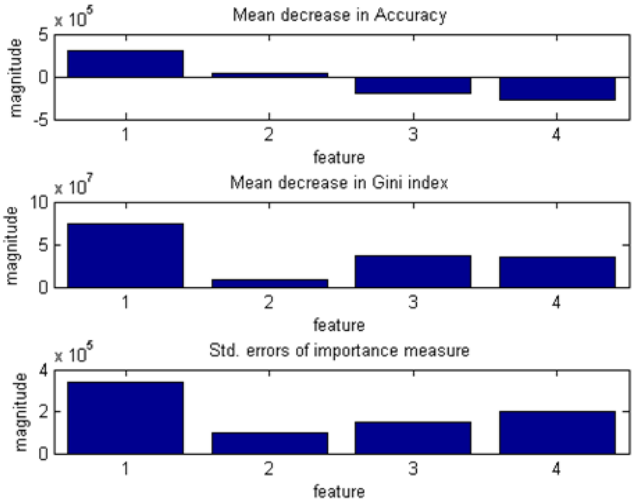


Fig.4: Mean Decreases in Accuracy

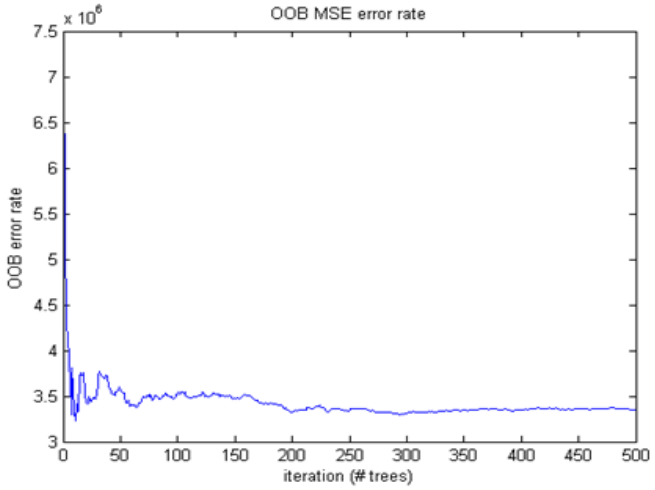


Fig.5: OOB MSE error rate

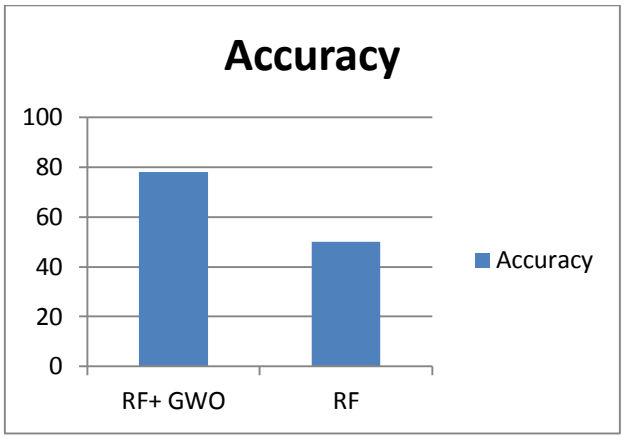


Fig.6: Accuracy of the classifier

In figure.6 accuracy comparisons is shown with random forest and Random forest with GWO. The accuracy of the Random forest with GWO is better than random forest.

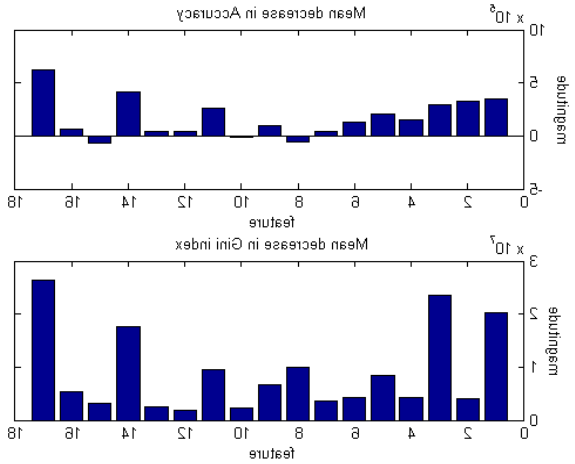


Fig.7: Mean Decreases in Accuracy

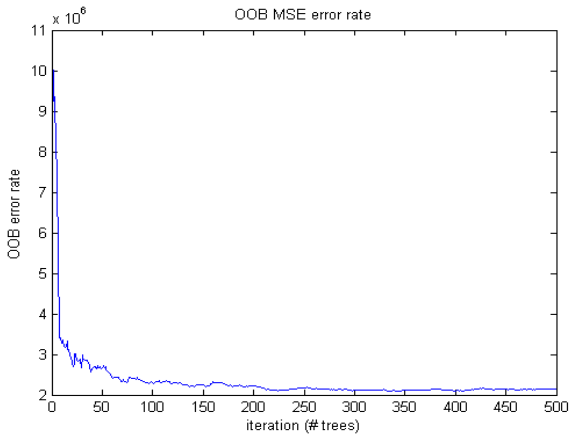


Fig.8: OOB MSE error rate

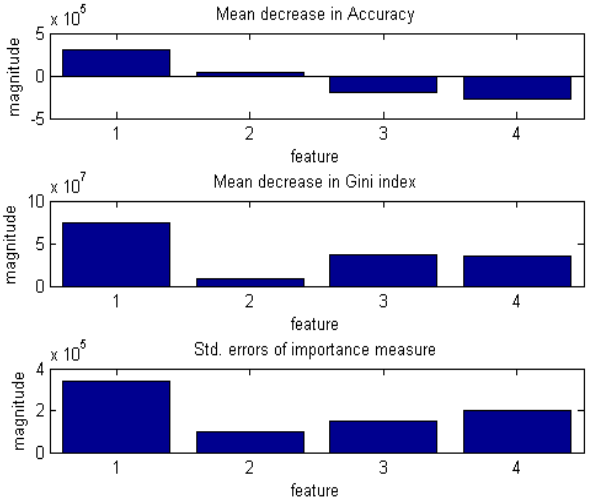


Fig.9: Mean Decreases in Accuracy

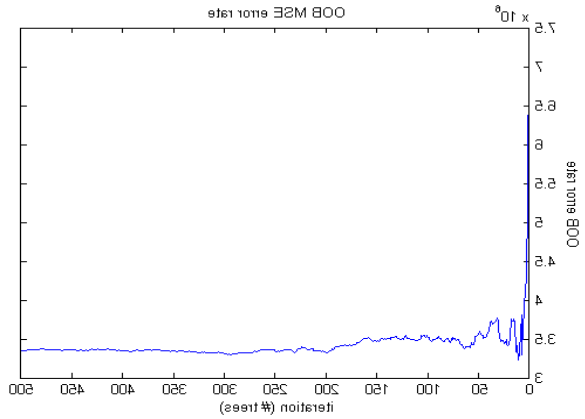


Fig.10: OOB MSE error rate

V. CONCLUSION

In this work features selection approach is done by using Grey wolf optimization algorithm. GWO algorithm is used to select the effective weighted feature. The result is shown by the analysis process.

- accuracy is predicted with and without feature selection.
- shows the accuracy without and with feature selection without 17 features and with GWO only features, therefore reduce the high dimension space and get effective accuracy.
- Random forest (RF)+ GWO accuracy shows significant high only in Random Forest Method
- In other analysis boosting method is used with RF method which improves the training process of selecting tree from a forest. In figure 5.4 comparative analysis of boosting and aging method is shown. In this experiment it is clear

that boosting with GWO significant improves accuracy, precision and recall.

VI. REFERENCES

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

[2]. Pospieszny, Przemyslaw, BeataCzarnacka-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.

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

[4]. 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.

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

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

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

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

[9]. 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.

[10]. Jafari, S. S., &Ziaaddini, F. (2016, March). Optimization of software cost estimation using harmony search algorithm. In Swarm Intelligence and Evolutionary Computation (CSIEC), 2016 1st Conference on (pp. 131-135). IEEE.