NOVEL APPROACH SOFTWARE EFFORT ESTIMATION BY HYBIRDIZATION OF RANDOMFOREST AND BOOSTING APPROACHES

Er. Nisha Kumari[1], Er. Poonam chaudhary [2]
[1, 2] *Computer science and engineering, SIRDA Group of Institutions*

Abstract- In the area 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. 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, In software project estimation reduce feature and improve classification is big challenge which done by hybridization of random forest with bagging and boosting approach

Keywords- *software, effort, bagging, boosting*

I. INTRODUCTION

In the development of any software its estimation plays an important role presenting a more challenging task. The process based on accurate form of estimation helps in determining the success of a project on overall basis. An effective way of project management and planning is very difficult to obtain without proper guidance.

1.1 Software Project Estimation

If the estimation of the project is not proper then the development of the software also not in proper way and organized [1] [11]. Even when all the factors related to the software development are considered during development process but still projects are not estimated accurately. In this estimation process time of improvement is not calculated. When project is underestimated the effects like under scoping and understaffing affects the project most and project does not meet the deadlines and it loses its credibility [2] [3]. To overcome the issues of overestimation and underestimation software project estimation approach is used. If the number of resources is more than required resources it enhances the cost of the project and this condition arise the demand of software project estimation.

In small project it is not difficult to estimate the project and mainly estimated by expert judgment approach but in the embedded and large scale projects accuracy and precision of result matters most and they need effective estimation approach [5] [9]. The estimation process with good reliability is an issue that was faced in the projects. In the software estimation process these are the basic steps that are considered:-

- Estimation of project Size: This factor related to the size of th project and measured in the term of function point and line of codes. The UCP (Use case point) and Story points are another method which also helps to estimate the project size.
- Effort estimation: Effort estimation for the project based on the manpower and their working hours in the terms of person per month and person hours.
- Scheduling estimation: To decide the total time for project development.
- Cost estimation to decide the overall budget.

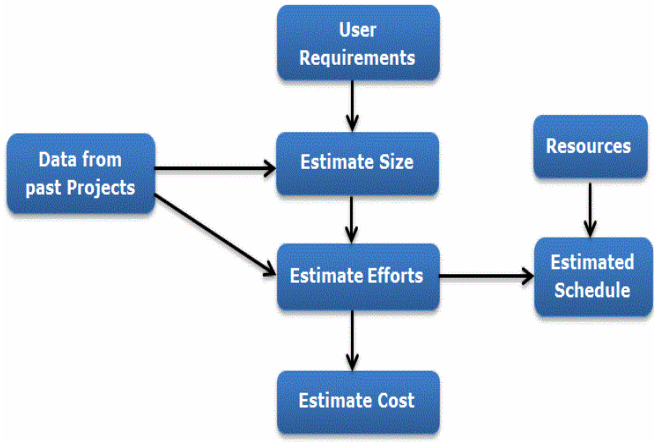

Figure 1: Software Project Estimation

1.2 Estimating Size

Effective size estimation is the first step towards an effective product. During the phase of requirement gathering and analysis project size also estimate according to the formal description with client. The cost estimation of the project also depends on the requirement specification and proposal request [8]. The size estimation also depends on the SRS and its details and re-estimation of the size can also be changed according to this in later phases of life cycle.

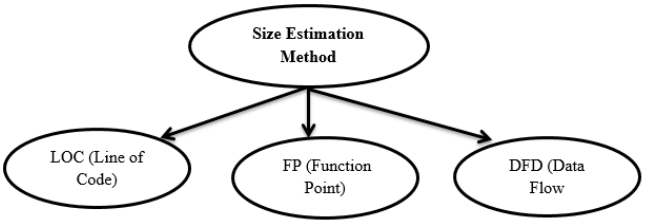

Figure 2: Size Estimation Methods

Following are the two methods that are used for the product size estimation.

1. Size by Analogy: This method of estimation based on the existing projects and the size estimation. The size of the new project is estimated accordingly because the existing project is similar to new project. This helps to estimate the total cost of project similar to previous one. BY using the analogy approach only experienced estimator can estimates the better size estimate. This approach work effectively only when we have accurate dimensions of the previous project.

2. Algorithmic approach: To count the product features: The algorithmic approach for size estimation is Function Point which converts the tally into size estimation. This approach based on the classes, modules, function, and methods in the product features.

1.3 Estimating Effort

Effort estimation process starts after the estimation of size of the project. This estimation performed after the complete requirements are defined and size mentioned. The software development process includes the design, develop, and testing of modules and each modules required separate effort to complete it [7] [10]. The coding or development part of software development process takes not more effort than other phases. The writing, documentation, implementation of prototype, and review of document takes more effort.

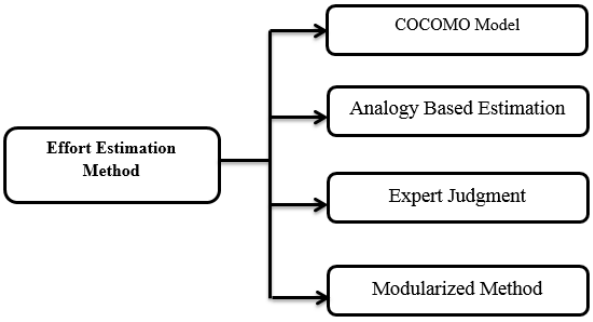

Figure 3: Effort estimation Method

Following are the two methods for estimating the effort from the size.

1. The existing data of the organization itself is helpful to estimate the project size and costs with respective to each other.
   - Documentation of actual results by using existing projects.
   - There should be minimum one project in the past which has similar size which helps to determine the estimation of side and then effort.
   - The development life cycle of the existing project helps to estimate the development time for new project.

2. When no similar type of project is available then most accepted and appreciated project. This situation occurs only when no similar project developed earlier. The most commonly used method for effort estimation is COCOCMO and Putnam Methodology. These methods help to converts the size estimation into effort estimation [13]. These models are less effective than the historical project estimation method and their accuracy varies according to the project domain and application areas.

1.4 Estimating Schedule

Schedule of the project describes the working period to complete the assigned task [2] [4]. The schedule estimation done on the basis of total effort calculated for the project. The schedule of the project includes the type of work, starting and ending time. The data gathered from this step used to decide the schedule of the project. In this work is also broken into modules according to the skill of the persons and timelines for each module is decided.

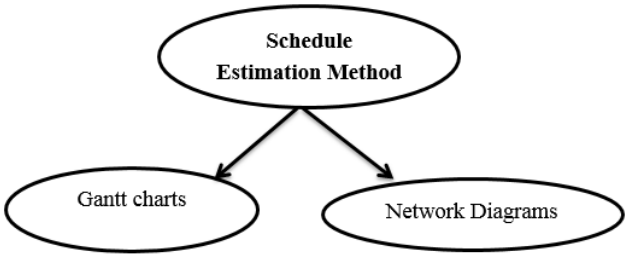

Figure 4: Schedule estimation Method

1.5 Estimating Cost

Cost estimation is process of deciding the budget for the project according to its size and modules. During the cost estimation many factors are considered and the main factors are man power, software on rentals, hardware, office rentals, and telecommunication. The cost estimation depends on the size of the project also because if the project size is large it consumes many resources and manpower and respectively cost is also increased [8] [11]. If the size of project is small it need less resources and less time to complete and its cost is also low. Project cost can be obtained by multiplying the cost of man power per month with estimated effort. After the schedule estimation is completed it is easy to compute the rate per hour for the resources of the project.

II. RELATED WORK

Jodpimai et al. [1] proposed the data mining approach for re-estimation of software efforts. Statistical approach used for the preprocessing of prior phase and selection of input features for learning approach. This model work on four phases that are transformation of data, outlier detection, feature selection, and learning. The result evaluation of the proposed approach done by comparing it with proportion based method and it gives more effective results. Bilgaiyan et al. [2] proposed the genetic algorithm for cost estimation and this method is used to construct the dilation-erosion perceptron to overcome the drawbacks of morphological operators. The performance analysis is done by estimating the 5 different SDCE problem and three metrics. Silhavy et al. [3] worked on the use case point's estimation by using the subset selection techniques and predict the accuracy of the regression model. Different methods like k-mean, spectral clustering and Gaussian model used for selection of subset. The performance evaluation of the approach done by using two different data sets. The proposed clustering method reduces the prediction error of the regression approach. Benala et al. [4] presented an approach for effort estimation by using the concept of analogy based estimation. The work is based on the differential evolution algorithm and used to optimize the weight of features of similarity functions. The simulation of the proposed work done on promise repository and check the effectiveness of proposed DABE model. This model performs better than PSO, G.A, and neural network. Wu, Dengsheng et al. [5] presented the particle swarm optimization algorithm for weight optimization in software effort estimation. The proposed model is used to predict the effort of the project in advance which helps to manage the activities in advance.

In this work PSO is combined with the case based reasoning system to provide the optimal weights in CBR system. The implementation of the model was done on two datasets that are Maxwell and Desharnis. The result evaluation is based on the two measures that are MMRE and MdMMRE and show that how PSO based model provides effective effort estimation. Rao, Ch Prasada, et al. [6] presented the concept of machine learning for effort estimation based on the story points. The effort estimation is based on the functional point, object points and use case points. This approach is applicable on the agile methodology project which increases the chances of the success. The proposed model estimate the effort for the project developed by using agile methodology and machine learning optimize the results for better prediction effort.Liu, Qin et al. [7] proposed an approach for feature selection in software effort estimation by using the qlocalized neighborhood mutual information. The experiment performed on six different dataset and results were compared with randomized baseline approach. The result verification is done by using cross validation method and gives effective results with better improvement. Idri, Ali, et al. [8] worked on finding the solution of missing data in software estimation process. The proposed work based on the support vector regression to handle the fuzzy and classical analogy. The model result compared with existing KNN method and SVR. The accuracy of the proposed algorithm for effort estimation enhanced in support vector regression. Dragicevic et al. [9] proposed the Bayesian method for the effort estimation of software development. This model is simple and small and it can be used from the initial stage of the software development. This model is able to estimate the parameters automatically and learned them from the dataset. The data collected from the single company a precision of the model calculated by using different metrics. The statistical results show good prediction accuracy. Moosavi, et al. [10] presented a model which is a combination of bird optimization algorithm and adaptive neuro-fuzzy inference system. Optimization algorithm used to adjust the variables. This model is based on the optimized ANFIS which produced the effective accuracy to estimate the effort on wide range of projects. The test function in this model includes the unimodal and multimodal function. The results evaluation of the proposed work is based on the three models which improves the performance of the model. Dhaka, V. S., et al. [11] proposed the fuzzy inference system for the effort estimation. This work considered as the complexity in use cases are high and it takes more time to develop, test and implement. The proposed method provides the reliable results on the use case points and it is produced from actual business process. Azzeh, et al. [12] proposed model is designed for the classification and prediction stages by using the concept of radial basis neural network and support vector machine. The industrial projects and student projects are used for the construction of observations. This model produced better accuracy from the UCP prediction model. The proposed model gives better accuracy on all datasets by using the environmental factors of UCP to classify and estimate the productivity. Sarro, Federica, et al. [13] introduced the multi-objective effort estimation model which combines the Confidence interval analysis and mean absolute error. The proposed work done by using the PROMISE repository dataset. The statistical analysis of the work shows that this method is significant and gives better accuracy. This model also reduced the uncertainty of the estimation.

III. THE PROPOSED METHOD

3.1 Proposed Methodology

Steps of 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.
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.

3.2 Proposed methodology: Flowchart

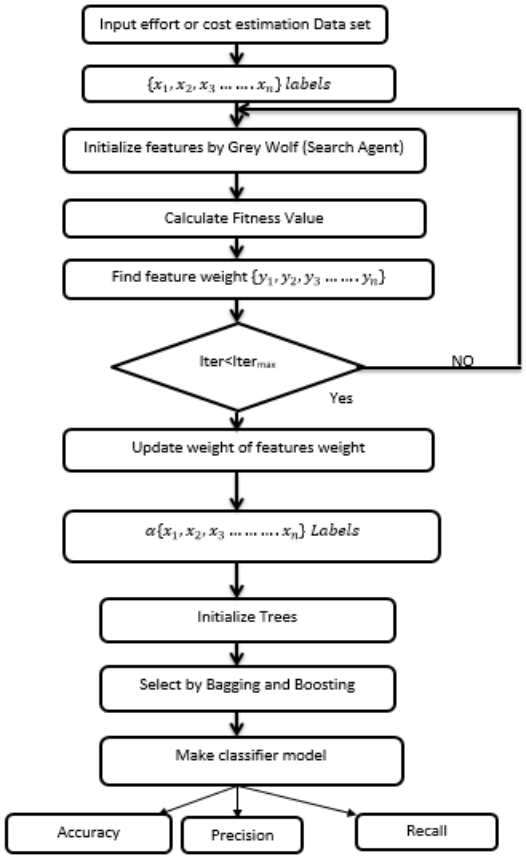

Figure 5: Flow Chart

3.3 Proposed Algorithm

1. Grey Wolf Optimization Algorithm (GWO): Grey Wolf optimization algorithm is a bio-inspired algorithm which is based on the leadership and hunting behavior of the wolves in the pack. The grey wolves prefer to live in the pack which is a group of approximate 5-12 wolves. In the pack each member has social dominant and consisting according to four different levels.

- The wolves on the first level are called alpha wolves (α) and they are leaders in the hierarchy.

- Second level wolves are called beta (β). These wolves are called subordinates and advisors of alpha nodes. The beta wolf council helps in decision making.
- The wolves of the third level are called Delta wolves (δ) and called scouts. Scout wolves at this level are responsible for monitoring boundaries and territory.
- The last and fourth level of the hierarchy are called Omega (ω). They are also called scapegoats and they must submit to all the other dominant wolves. These wolves follow the other three wolves.

2. 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.

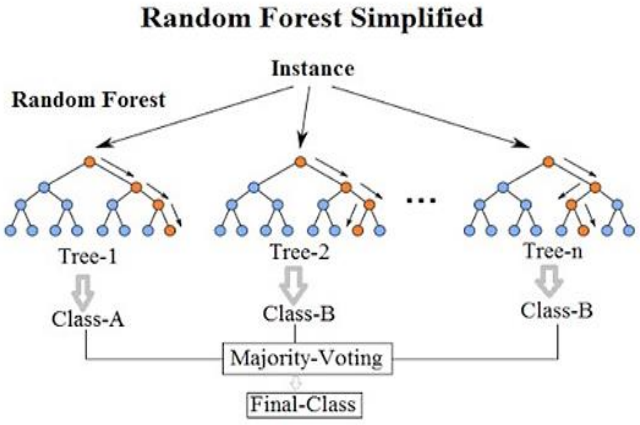

Figure 6: Random forest

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.

IV. RESULT ANALYSIS

This section describes the result and discussion in the graphical form. The result of different classifiers used for the comparison and discussed for evaluation. The results evaluation based on the precision, recall and accuracy of the classifiers.

4.1 Results of classification

Table.1 Result of Classification

Classification	Accuracy	Precision	Recall
Random forest + Boost	62	52	69
Random forest + Boost+ GWO	71	93	94
Random forest +Bagging+ GWO	69	68	58
Random forest + Bagging	35	92	97

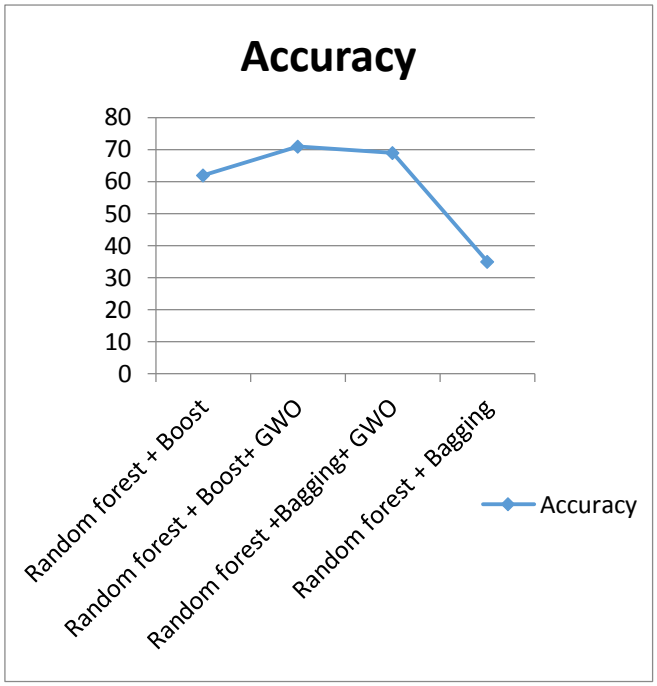

Figure 7: Accuracy of classifiers

Figure 7 depicts the accuracy of the Random forest + Boost, Random forest + Boost+ GWO, Random forest +Bagging+ GWO and Random forest + Bagging classifiers. The highest accuracy 93 % in graph shown by Random forest + Boost+ GWO and minimum by Random forest + Bagging classifier that is 52%.

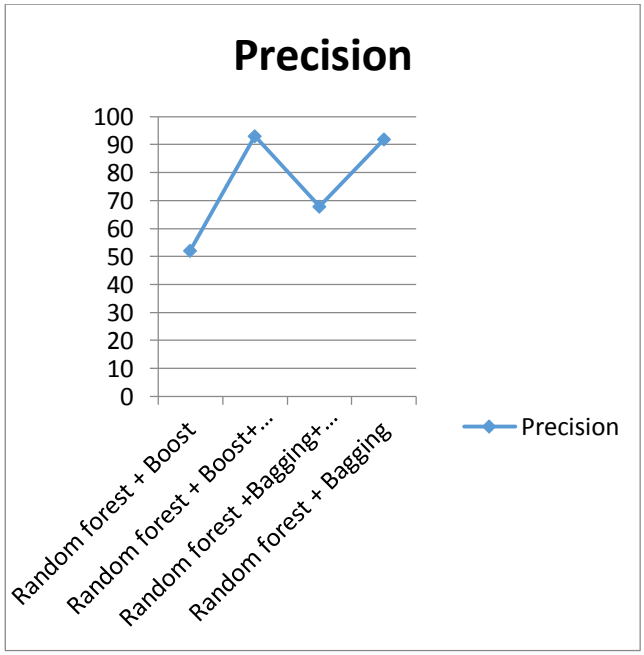

Figure 8 Precision of classifiers

Figure 8 depicts the precision of the Random forest + Boost, Random forest + Boost+ GWO, Random forest +Bagging+ GWO and Random forest + Bagging classifiers. The high precision 94 % in graph shown by Random forest + Boost+ GWO, Random forest + Bagging classifier and minimum by Random forest + Boost classifier that is 52%.

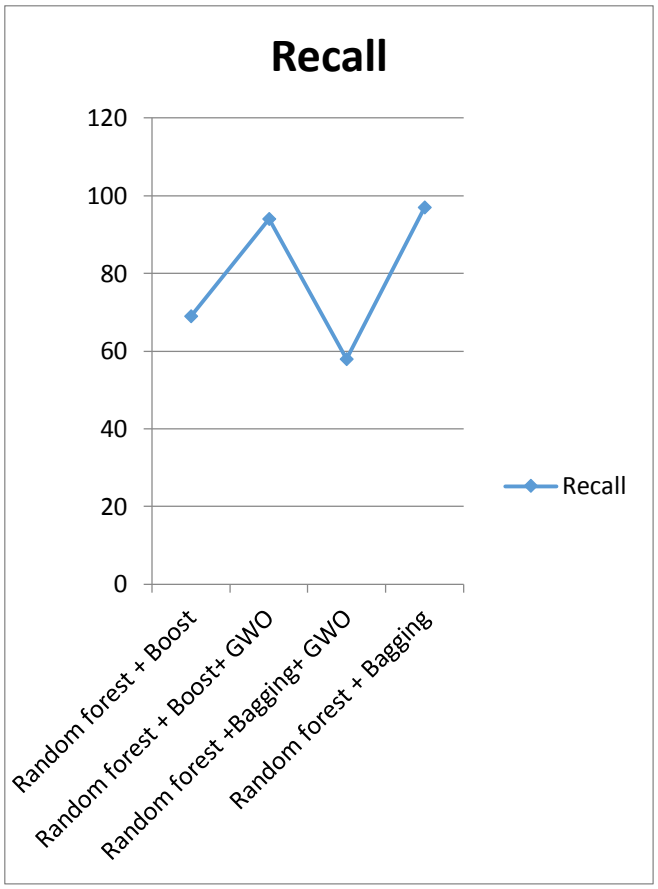

Figure 9: Recall of classifiers

Figure 9 depicts the recall of the Random forest + Boost, Random forest + Boost+ GWO, Random forest +Bagging+ GWO and Random forest + Bagging classifiers. The high recall 97 % in graph shown by Random forest + Boost+ GWO, Random forest + Bagging classifier and minimum by Random forest + Bagging+ GWO classifier that is 58%.

Figure 10 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. The red blue curve in the graph represents the accuracy of the different classifiers, Red curve in the graph represents the precision, and green curve represents the recall of the classifier.

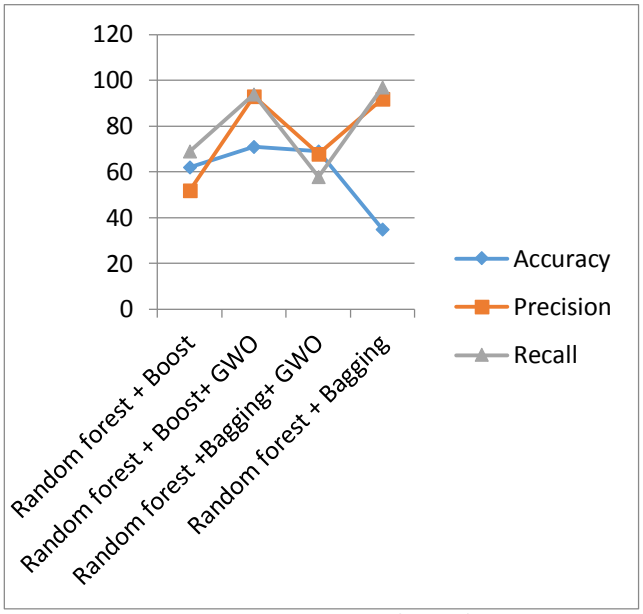

Figure 10 Comparison of classifiers

4.2 Random Forest Regression

Table.2 Random Forest Regression

Random Forest Regression	Accuracy
RF+ GWO	79
RF	52.10

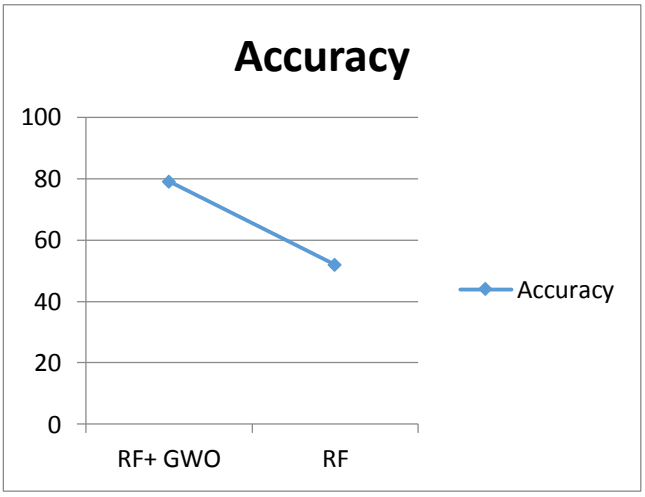

Figure 11Accuracy of the classifier

In figure 11 accuracy comparison is shown with Random forest and Random forest with GWO. The x axis of graph represents the classifiers and y axis of graph represents the random values of accuracy. The accuracy of the Random forest with GWO is better than random forest.

5.3 Result screenshots

5.3.1 Random Forest

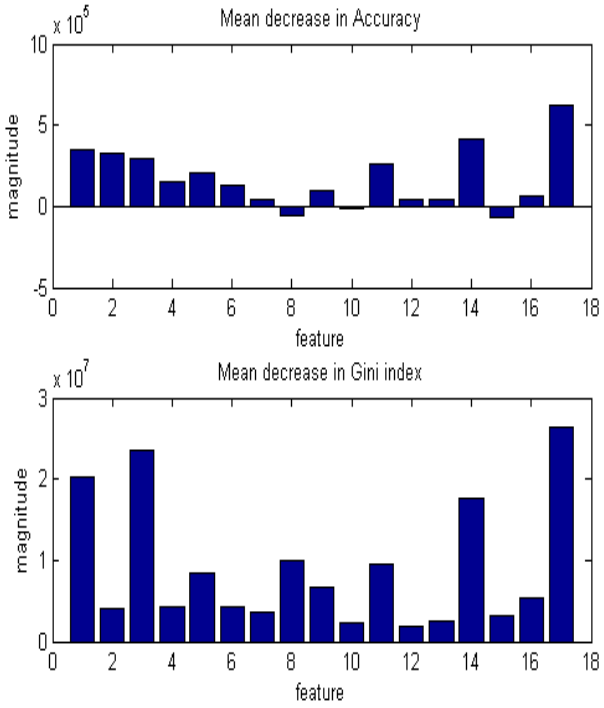

Figure 12: Mean Decreases in Accuracy

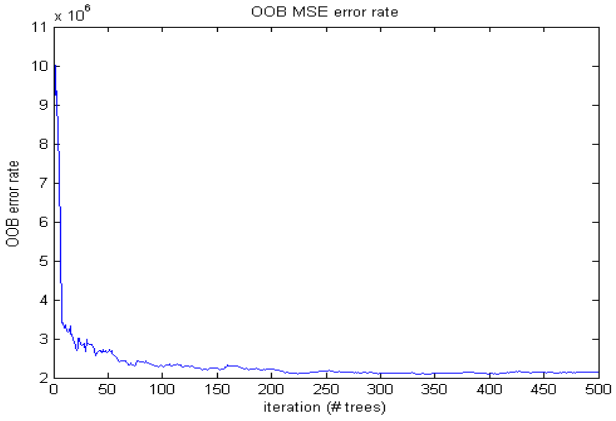

Figure 13 OOB MSE error rate

5.3.2 Random forest+ GWO

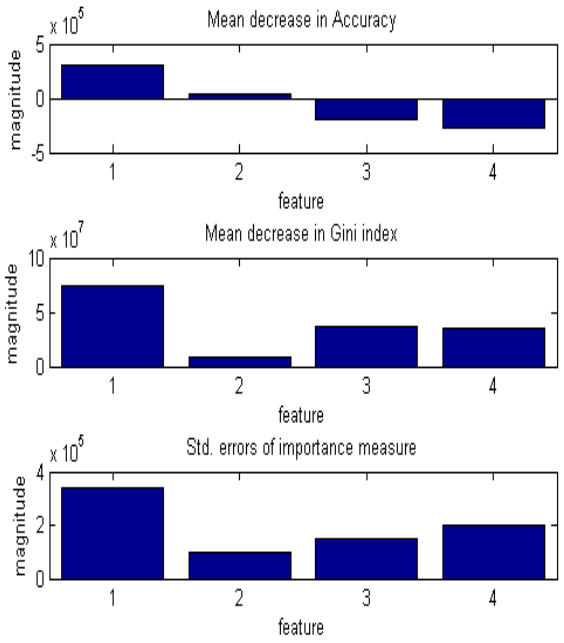

Figure 14: Mean Decreases in Accuracy

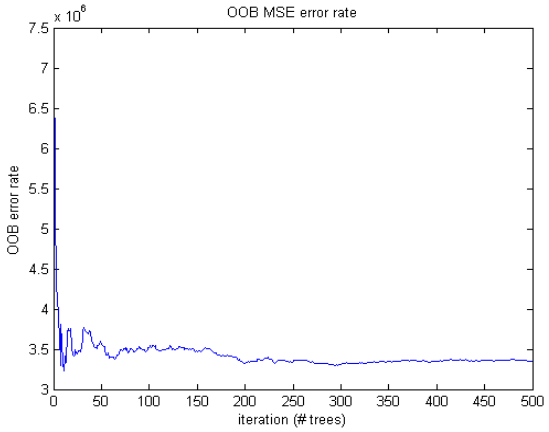

Figure 15: OOB MSE error rate

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. 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.

- In random forest and random forest Grey wolf optimization algorithm, the accuracy is predicted with and without feature selection.

- In random forest and random forest Grey wolf optimization algorithm shows the accuracy without and with feature selection with GWO only features, therefore reduce the high dimension space and get effective accuracy.
- In accuracy of the classifier, the 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 comparative analysis of boosting and bagging method is shown. In this experiment it is clear that boosting with GWO significant improves accuracy, precision and recall.

By the result analysis it is concluded that the feature selection is effective process for improving accuracy and GWO algorithm is more effective because it optimize local and global convex optimizer.

V. REFERENCES

[1] Jodpimai, Pichai, Peraphon Sophatsathit, and Chidchanok Lursinsap. "Re-estimating software effort using prior phase efforts and data mining techniques." *Innovations in Systems and Software Engineering* (2018): 1-20.

[2] Bilgaiyan, Saurabh, et al. "Chaos-based Modified Morphological Genetic Algorithm for Software Development Cost Estimation." *Progress in Computing, Analytics and Networking*. Springer, Singapore, 2018. 31-40.

[3] Silhavy, Radek, Petr Silhavy, and Zdenka Prokopová. "Evaluating subset selection methods for use case points estimation." *Information and Software Technology* 97 (2018): 1-9.

[4] Benala, Tirimula Rao, and Rajib Mall. "DABE: Differential evolution in analogy-based software development effort estimation." *Swarm and Evolutionary Computation* 38 (2018): 158-172.

[5] Wu, Dengsheng, Jianping Li, and Chunbing Bao. "Case-based reasoning with optimized weight derived by particle swarm optimization for software effort estimation." *Soft Computing*22.16 (2018): 5299-5310.

[6] Rao, Ch Prasada, et al. "An Agile Effort Estimation Based on Story Points Using Machine Learning Techniques." *Proceedings of the Second International Conference on Computational Intelligence and Informatics*. Springer, Singapore, 2018.

[7] Liu, Qin, Jiakai Xiao, and Hongming Zhu. "Feature selection for software effort estimation with localized neighborhood mutual information." *Cluster Computing* (2018): 1-9.

[8] Idri, Ali, Ibtissam Abnane, and Alain Abran. "Support vector regression-based imputation in analogy-based software development effort estimation." *Journal of Software: Evolution and Process* (2018): e2114.

[9] 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.

[10] Moosavi, Seyyed Hamid Samareh, and Vahid Khatibi Bardsiri. "Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation." *Engineering Applications of Artificial Intelligence* 60 (2017): 1-15.

[11] Dhaka, V. S., et al. "Software Project Estimation Using Fuzzy Inference System." *Proceedings of International Conference on ICT for Sustainable Development*. Springer, Singapore, 2016.

[12] Azzeh, Mohammad, and Ali Bou Nassif. "A hybrid model for estimating software project effort from Use Case Points." *Applied Soft Computing* 49 (2016): 981-989.

[13] Sarro, Federica, Alessio Petrozziello, and Mark Harman. "Multi-objective software effort estimation." *Software Engineering (ICSE),2016 IEEE/ACM 38th International Conference on*. IEEE, 2016.