Improvisation of Random Forest Classifier by Feature Selection

Chetna Sharma [1], Aman Kumar Sharma [2]
[1] *Department of Computer Science, Himachal Pradesh University, Shimla, India*
[2] *Department of Computer Science, Himachal Pradesh University, Shimla, India (E-mail:sharmachetna688@gmail.com)*

Abstract— Random forest has found its wide spread use in various applications the acceptability of random forest can be primarily attributed to its capacity of handling non-linear classification task efficiently. According to, classification performance of random forest improves with the number of trees. But experimental evidences suggest that adding trees beyond certain pre-determined limit may not significantly improve the classification performance of random forest. The proposed method uses feature selection methods. Method of Feature Selection is based on the important and unimportant features which yields a way of reducing calculation time, improving classification accuracy data in machine learning. Improved Random Forest classifier is proposed which performs classification with minimum number of trees. This algorithm meets with a reduced but important set of features. It is to be proved that further reduction of features improve accuracy of the random forest.

Keywords— *Random Forest, Feature selection, Feature ranking, Bagging, Boosting, Classification accuracy, Precision, Recall.*

I. INTRODUCTION

In Machine Learning, classification is a supervised learning approach in which the program learns from the data inputs given to it and then that program classify new observations by applying the supervised learning approach. Decision Tree is a practical and popular approach in machine learning domain for solving classification problems [16]. Random forest is one of the popular decision tree classifiers within supervised learning. The general method for random forest was first suggested by Ho in 1995. Random forest is ensemble of pruned binary decision tree, unlike others it generates numerous trees which creates forest like classification [1]. Ensemble learning method of the random forest is a promising technique in terms of accuracy [3]. Random forest is one of best techniques used for the classification of unbalanced data in machine learning and data mining for data analysis and data extraction [5]. Random forest has found its wide spread use in various applications [2]. Random forest is known for taking care of data imbalances in different classes, especially for large datasets [4]. Due to its parallel architecture, random forest classifier is faster compared to other classifiers like ID3, C4.5, and CART [15]. A large number of variants of random forest can be found in literature [5].

According to classification, performance of random forest improves with increase in the number of trees [14]. But experimental evidences suggest that adding trees beyond certain pre-determined limit may not significantly improve the classification performance of random forest [6]. Further, it has been shown that random forest may perform biased feature selection for individual trees [7]. As a result, an unimportant feature may be favored in a noisy feature set. Consequently, classification accuracy may degrade [8]. So, an increased proportion of important features (i.e. removal of unimportant features) may have significant impact on the classification performance of random forest. A number of feature selection strategies can be found in the literature [9]. But features of initial forest leads to the performance criteria.

The remaining paper is divided in three sections. In Section 2, the methods of feature selection are discussed. Section 3 illustrates briefly the proposed Improved Random Forest. Section 4 discusses test investigation and examinations of random forest and Improved Random Forest. Section 5 highlights conclusions and future scope.

II. PROPOSED MECHANISM OF FEATURE SELECTION

In machine learning applications, feature selection is a substantial preprocessing step to find the features of smallest subset that ultimately increases the performance of the model. Other benefits of applying feature selection include the ability to build simpler and faster models using only a subset of all features, by focusing on a selected subset of features [9]. Sampling of forest is done with the help of bagging and boosting where bagging is used to reduce the deviation of tree by creating several subsets of data from training sample which will be chosen randomly. The main goal of bagging is to solve the accuracy of prediction [15]. Boosting converts the weak learners to strong learners. Boosting is used to generate a group of predictors. The main goal of boosting is to solve net errors from the prior trees [24].

Feature selection techniques can be divided into three categories, namely feature ranking, finding important and unimportant features, and finding number of trees to be added depending on how they interact with the classifier. Feature selection methods

directly operate on the dataset, and provide a feature weighting leading to ranking as output [10]. These methods have the advantage of being fast and independent of the classification model, but at the cost of inferior results. In the next few subsections feature selection methods are discussed with three major steps: feature ranking, finding important and unimportant features and finding the number of trees to be added.

A. Feature Ranking

In feature ranking, feature vector is an n-dimensional vector of numerical features that represent some object**.** First the calculation of weight is carried out for different features. Then rank features weight wise. The features with weight below the threshold are subsequently removed. A feature with value higher than weight is taken as important feature for classification. Based on the global weights, ranking of the features is done to find the important and unimportant features [21].

B. Finding Important and Unimportant Features The main purpose of feature selection is to find out the important and unimportant features from the feature vector (weight). It is not known which and how many features are important. So a novel strategy to find the important features is carried out. Initially, from the ranked list mark some features as 'important' based on a feature weight. Note that, once a feature is marked to be important at a construction pass, it will remain important till the end and will not be removed in the subsequent passes. Thus the probability of discarding an important feature is reduced [25]. After getting certain important and unimportant features, formulate a theoretical bound of maximum number of trees to be added to the forest at that construction pass.

C. Finding Number of trees to be Added

To find the number of trees to be added, first define two quantities that controls the classification performance of random forest. These two quantities are strength and correlation. Classification accuracy is defined based on strength and correlation [25]. Find the number of trees to be added using the formulation of classification accuracy.

- **Probability of Good Split:** Probability of good split is the probability that a node is split by an important feature. A good split creates child nodes with more homogeneity compared to the parent node. A good split in node is possible only if at least one important feature is present in corresponding. There is possibility that some feature, present in the bag of unimportant features might turn out to be important in the subsequent construction passes. Hence, if the features selected from only the important features bag, it will lead to greedy selection and which will miss some potential important features [22]. Hence, choose the features from both the bags of important and unimportant features.
- **Strength:** The strength of a forest is dependent on the minimum classification accuracy of individual trees. Hence, define the strength of the forest as the probability that all the nodes in at least one tree has good splits.
- **Correlation:** After probability and strength of forest, it is a measure of similarity between the trees. For random forest, correlation between trees is dependent on the features used at different nodes of those trees.

III. THE IMPROVED RANDOM FOREST

Improved Random Forest (IRF) is introduced which takes care of feature selection and sampling by finding optimal number of trees simultaneously. IRF starts with a forest of less trees. The initial forest finds a small number of important features. Then at each construction pass, update the list of important and unimportant features through following four steps. First calculate the weights of different features and sort features rank wise. Then calculate a threshold weight. The features with weight below the threshold are subsequently removed.

Next, from the collection of remaining features, mark some features as 'important' based on a novel criterion. The remaining features are marked as 'unimportant'. Once a feature is marked to be important at a construction pass, it will remain important till the end and it will not be removed in the subsequent passes. After getting certain important and unimportant features, formulate a theoretical bound of maximum number of trees to be added to the forest at that construction pass.

Show that if trees are added satisfying the bound, the classification accuracy of the forest certainly improves. The construction passes are continued until a novel termination criterion is reached.

As a result, probability of discarding an important feature is reduced. Now select bagging (bag of important and unimportant features) and boosting (set of predictors to remove error rate) for selecting effective trees for the forest. Thus the proposed forest provides optimal classification accuracy with precision and recall in terms of the number of trees and in terms of feature reduction. Notably, the count of trees in our method is not pre-determined like for specific datasets. So IRF has low data dependence. IRF is fast and hence useful for industrial applications.

A. Algorithm

Step1: Procedure IRF
Step2: Initialize Random forest.
Step3: Grow Random forest with dataset random trees and feature vector.
Step4: Calculate, assign weight and rank the features.
Step5: Sort features rank wise.
Step6: Initialize n = 0, where n is the number of construction pass.
Step7: End procedure

Step8: While number of unimportant features at the n^{th} construction pass is >= the number of features one node is selected for node splitting.
Do
Step9: Compute mean and standard deviation of feature weights in bag of important features at the n^{th} construction pass.
Step10: Find features to be removed at the n^{th} construction pass.
Step11: From the bag of unimportant features, find the set of features with maximum threshold weight.
Step12: Find Feature vector at the (n +1) construction pass.
Step13: Find bag of important features at the (n+1) construction pass.
Step14: Find Number of trees at (n+1) construction pass.
Step15: Select boosting algorithm for the precision and recall.
Step16: Make Classification Model.
Step17: End while

IV. EXPERIMENTAL ANALYSIS

In this section the Random Forest classifiers and Improvised Random Forest are compared based on their Accuracy, Precision, Recall, Specificity, FNR, FPR and FDR.
The imitations were regulated using a dataset namely EEG Eye State Dataset taken from the UCI Repository: www.archive.ics.uci.edu/ml/datasets.html. And EEG Eye State dataset consists of EEG values and these values indicate the eye state. EEG dataset includes 14980 instances. The tool used for simulation is Scikit-learn. Scikit-learn is a free and open source library software in Python used for machine learning. It is a simple and efficient tool for data science and pattern recognition. It is based upon numpy, scipy and matplotlib. It is downloaded from the repository https://github.com/scikit-learn/scikit-learn. This library incorporates a well ordered tools for classification, regression, clustering, and dimensionality reduction in machine learning. It is notified that scikit-learn is used to create models.

The criterion used for comparison:

- **Accuracy** is the parameter used for testing the samples which are correctly classified. As accuracy is used for comparing different approaches, considering the experimental results.

- **Precision** is the depiction of errors. It is also called as positive predicted value. Precision tells how many selected items are relevant.

- **Recall** is also called as the sensitivity or true positive rate. Recall highlights the number of relevant items selected.

- **Specificity** is also called selectivity or true negative rate as it measures the proportion of negatives that were correctly classified as negative.

- **False Negative Rate (FNR)** is also called miss rate as it measures the proportion of positives that were incorrectly classified as negative.

- **False Positive Rate (FPR)** is also called fall out as it measures the proportion of negatives cases that were incorrectly classified as positive.

- **False Discovery Rate (FDR)** is a method of speculating the rate of errors in testing.

A. Analysis

The tool was applied on the dataset using Random Forest and as well as proposed Improved Random Forest. The results of the experiment are presented in Table 1.

TABLE 1. COMPARISON OF RANDOM FOREST AND IMPROVED RANDOM FOREST ON EEG EYE STATE DATASET.

Parameters	Random Forest	Improved Random Forest
Recall	63.2	81.2
Specificity	78.12	94.5
Precision	70.12	92.2
FNR	35.23	18.2
FPR	21	5
FDR	27.79	7
Accuracy	72.5	98.9

Table 1 indicates the results of IRF and Random forest for the EEG Eye State Dataset in which IRF have improved upon Random forest in terms of accuracy 98.9%, recall 81.2%, specificity 94.5%, precision 92.2%, with low FNR, FPR and FDR as compared to Random Forest. So IRF have gone beyond the Random Forest in terms of classification.

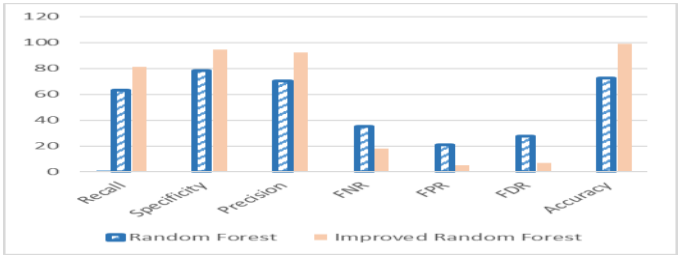

Fig. 1 Comparative performance of Random Forest with Improved Random Forest

The Fig. 1 shows graphical representation of experimental results. It can be identified from the graph that IRF provides better classification accuracy, recall, precision, specificity, FDR, FNR and FPR than generalized Random forest.

B. Advantages Compared to Conventional Random Forest IRF is a modification of conventional random forest. It has already been observed that given the same number of trees, IRF outperforms RF. Next it is investigated that by adding trees in RF can lead to results comparable to IRF. For each data, find the numbers of trees in RF that produce the lowest average error by using the algorithm. Even with much larger number of trees, RF does not beat the proposed IRF. Thus, IRF method beats RF with less computational burden. Therefore IRF has low data dependence. IRF is fast and hence useful for industrial applications.

V. CONCLUSIONS AND FUTURE SCOPE

IRF is a quick and precise solution for automatic classification by improvising random forest classifier. The proposed classifier not only removes excessive features, but also dynamically changes the size of the forest (number of trees) to produce optimal performance in terms of classification accuracy. The proposed method out-performs in EEG Eye State dataset. IRF classifier has also proven to be useful in classification problems of features extraction from EEG values to predict EEG Eye State. The proposed classifier has the potential to be applied in industrial applications. In future, random forest guided autoencoder will be explored for feature encoding.

REFERENCES

[1] Miroslav Kubat, "An Introduction to Machine Learning", 2nd edition, Springer International Publishing AG, 2017.

[2] "Introduction to boosting algorithms" from https://www.analytics vidhya.com/blog/2015/11/quick-introduction-boosting-algorithmsmachine-learning/accessed on 17/06/2018 at 1300 hrs.

[3] "Random Forest" from https://en.wikipedia.org/wiki/Random_forest accessed on 28/07/2018 at 2600 hrs.

[4] "Bagging anrandom forest ensemble algorithms for machine learning" from https://algorithm–machine-learning–mastery.com/bagging-and–random–forest–ensemble–algorithms–for machine -learning/accessed on 03/08/2018 at 2100 hrs.

[5] "Random—Forest--Algorithms" from https://www.Datasciencecentral .com/profiles/blogs/random-forests-algorithm accessed on 09/08/2018 at 1200 hrs.

[6] Lomax. S, "A survey of cost-sensitive decision tree induction algorithms", ACM Computing Surveys, pp. 34-44, 2013.

[7] C. Luo, Z. Wang, S. Wang, J. Zhang, and J. Yu, "Locating facial landmarks using probabilistic random forest," Signal Processing Letters, IEEE, vol. 22, no. 12, pp. 2324–2328, Dec 2015.

[8] A. Criminisi and J. Shotton, "Decision forests for computer vision and medical image analysis", Springer Science & Business Media, 2013.

[9] Janecek, "On the Relationship between Feature Selection and Classification Accuracy", JMLR: Workshop and Conference Proceedings, vol. 4, pp. 90-105, 2008.

[10] Girish ChandraShekar, "A Survey on Feature Selection Methods", ELSEVIER, Computer and Electrical Engineering, pp. 16-24, 2014.

[11] T. M. Khoshgoftaar, M. Golawala, and J. V. Hulse, "An empirical study of learning from imbalanced data using random forest," in ICTAI 2007, vol. 2. IEEE, pp. 310–317, 2007.

[12] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory under sampling for class-imbalance learning," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 39, no. 2, pp. 539–550, 2009.

[13] T. G. Dietterich, "Ensemble methods in machine learning," in multiple classifier systems. Springer, pp. 1–15, 2000.

[14] N. Quadrianto and Z. Ghahramani, "A very simple safe-bayesian random forest," PAMI, IEEE Trans. on, vol. 37, no. 6, pp. 1297–1303, June 2015.

[15] A. Paul, A. Dey, D. P. Mukherjee, J. Sivaswamy, and V. Tourani, "Regenerative random forest with automatic feature selection to detect mitosis in histopathological breast cancer images," in MICCAI 2015. Springer, pp. 94–102, 2015.

[16] L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5– 32, 2001.

[17] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How many trees in a random forest?" in MLDM Springer, pp. 154–168, 2012.

[18] P. Latinne, O. Debeir, and C. Decaestecker, "Limiting the number of trees in random forests," in Multiple Classifier Systems. Springer, pp. 178–187, 2001.

[19] A. Cuzzocrea, S. L. Francis, and M. M. Gaber, "An information theoretic approach for setting the optimal number of decision trees in random forests," in Systems, Man, and Cybernetics (SMC), IEEE International Conference on. IEEE, pp. 1013–1019, 2013.

[20] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," BMC bioinformatics, vol. 8, no. 1, p. 25, 2007.

[21] I. Guyon, S. R. Gunn, A. Ben-Hur, and G. Dror, "Result analysis of the nips 2003 feature selection challenge." in NIPS, vol. 4, pp. 545–552, 2004.

[22] T. Bylander, "Estimating generalization error on two-class datasets using out-of-bag estimates," Machine Learning, vol. 48, no. 1-3, pp. 287–297, 2002.

[23] H. Ishwaran, "The effect of splitting on random forests," Machine Learning, vol. 99, no. 1, pp. 75–118, 2014.

[24] Y. Freund, R. E. Schapire, "Experiments with a new boosting algorithm," in ICML, vol. 96, pp. 148–156, 1996.

[25] Eugene Tuv, "Feature Selection with Ensembles, Artificial Variables, and Redundancy Elimination", Journal of Machine Learning Research, vol. 10, pp. 1341-1366, 2009.

[26] Chetna Sharma, Aman Kumar Sharma, "An Elusive Study of Decision Tree Classifiers in Machine Learning", International Journal for Research in Technological Studies, vol. 5, Issue 6, May 2018.