ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

Effective Learning and Classification using Random Forest Algorithm

[1]Vrushali Y Kulkarni, [2]Pradeep K Sinha

***Abstract*— Random Forest is a supervised machine learning algorithm. In Data Mining domain, machine learning algorithms are extensively used to analyze data, and generate predictions based on this data. Being an ensemble algorithm, Random Forest generates multiple decision trees as base classifiers and applies majority voting to combine the outcomes of the base trees. Strength of individual decision trees and correlation among the base trees are key issues which decide generalization error of Random Forest classifiers. Based on accuracy measure, Random Forest classifiers are at par with existing ensemble techniques like bagging and boosting.**

In this research work an attempt is made to improve performance of Random Forest classifiers in terms of accuracy, and time required for learning and classification. To achieve this, five new approaches are proposed. The empirical analysis and outcomes of experiments carried out in this research work lead to effective learning and classification using Random Forest algorithm.

***Index Terms*— Random Forest, Ensemble classifier, Decision Tree, Disjoint Partitioning, Pruning, Optimal subset, Parallel algorithm.**

I. INTRODUCTION

Random Forest (RF) is an ensemble, supervised machine learning algorithm. Machine learning techniques are applied in the domain of Data Mining [9]. Random Forest [Breiman 2001] uses decision tree as base classifier. Random Forest generates multiple decision trees. The randomization is present in two ways: first random sampling of data for bootstrap samples, and second random selection of input attributes for generating individual base decision trees. Strength of individual decision tree and correlation among base trees are key issues which decide generalization error of Random Forest classifier [6]. Based on accuracy measure, Random Forest classifier is at par with existing ensemble techniques like bagging [5] and boosting [17]. As per Breiman, Random Forest runs efficiently on large databases, it can handle thousands of input variables without variable deletion, it gives estimates of important variables, it generates an internal unbiased estimate of generalization error as forest growing progresses, it has effective method for estimating missing data and maintains accuracy when a large proportion of data are missing, and it has methods for balancing class error in class population unbalanced data sets [6]. The inherent parallel nature of Random Forest has led to its parallel implementations using multithreading, multi-core, and parallel architectures. Random Forest is used in many recent classification and prediction applications [11] due to above mentioned features.

It has been proved theoretically and empirically that the ensemble always gives better accuracy than an individual classifier [10]. The fundamental of ensemble design is creating diversity among the base classifiers [10] [13]. In [1] [2] [4] it is confirmed empirically that, the generation of Random Forest should be done in such a way that the trees will be diverse as well as they retain their strength. We have performed some experiments to achieve this and have come up with improvements in Random Forest so as to achieve effective learning and classification using this algorithm.

In this paper, we present our research work which attempts to improve performance of Random Forest classifier in terms of accuracy, and time required for learning and classification. To achieve this five new approaches are proposed. They are based on disjoint partitions of training datasets, use of different attribute evaluation / split measures to induce base decision trees of Random Forest, application of weighted voting instead of majority voting, use of diversity among bootstrap datasets to generate maximum diverse classifiers, and application of dynamic programming approach to find optimal subset of Random Forest. Taking into consideration the inherent parallel nature of Random Forest, a new approach is proposed for parallel implementation of Random Forest. Figure1 summarizes our research work presented in this paper. The experiments carried out in this research work lead to effective learning and classification using Random Forest algorithm.

This paper is organized in the following way: section 2 explains in brief the working of Random Forest classifier. Section 3 describes proposed new approaches for Random Forest along with results. Section 4 gives concluding Remarks.

II. RANDOM FOREST

A Random Forest is a classifier consisting of a collection of tree-structured classifiers $\{h(x, \Theta_k)\ k=1, 2, \ldots.\}$, where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x [6].

Random Forest generates an ensemble of decision trees. To generate each single tree in Random Forest, Breiman followed following steps: If the number of records in the training set is N, then N records are sampled at random but with replacement, from the original data; this is bootstrap sample. This sample will be the training set for growing the

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

tree. If there are M input variables, a number m << M is selected such that at each node, m variables are selected at Random out of M and the best split on these m attributes is used to split the node. The value of m is held constant during forest growing. Each tree is grown to the largest extent po

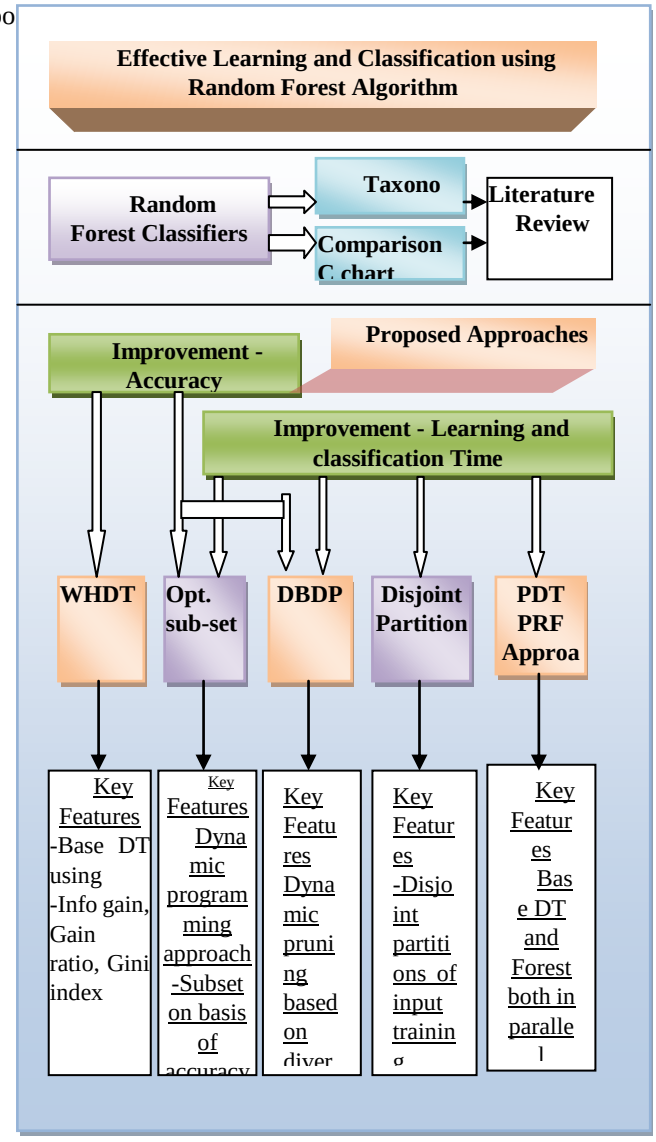

Fig 1. Overview of research work

In this way, multiple trees are induced in the forest; the number of trees is pre-decided by the parameter Ntree. The number of variables (m) selected at each node is also referred to as m_{try} or k in the literature. The depth of the tree can be controlled by a parameter node size (i.e. number of instances in the leaf node) which is usually set to one.

Once the forest is trained or built as explained above, to classify a new instance, it is run across all the trees grown in the forest. Each tree gives classification for the new instance which is recorded as a vote. The votes from all trees are combined and the class for which maximum votes are counted (majority voting) is declared as classification of the new instance.

This process is referred to as Forest RI in the literature [6]. Here onwards, Random Forest means the forest of decision trees generated using Forest RI process.

In the forest building process, when bootstrap sample set is drawn by sampling with replacement for each tree, about 1/3rd of original instances are left out. This set of instances is called OOB (Out-of-bag) data. Each tree has its own OOB data set which is used for error estimation of individual tree in the forest, called as OOB error estimation.

The Generalization error of Random Forest is given as,

$$PE^{*} = P_{x,y}(mg(X,Y)) < 0$$

The margin function is given as,

$$mg(X,Y) = av_k I(h_k(X) = Y) - max_{j \neq Y} av_k I(h_k(X) = j)$$

The margin function measures the extent to which the average number of votes at (X, Y) for the right class exceeds the average vote for any other class. Strength of Random Forest is given in terms of the expected value of margin function as,

$$S = E_{X,Y}(mg(X, Y))$$

If ρ is mean value of correlation between base trees, an upper bound for generalization error is given by,

$$PE^{*} \leq \rho (1 - s^2) / s^2$$

Hence, to yield better accuracy in Random Forest, the base decision trees are to be diverse and accurate.

III. METHODS AND RESULTS

The aim behind this research work is to improve performance of Random Forest classifier in terms of accuracy and time for learning and classification. Theoretical study and analysis as well as empirical analysis and experimentation are carried out to reach the goals mentioned earlier in section 3. Brief description along with few experimental results is presented in this section. Benchmarking datasets are used from UCI machine learning repository [22]. WEKA [20] [24] machine learning library is used for experimentation. All the experiments are carried out with 10 -fold cross-validation.

A. Disjoint Partitioning Approach

In this approach, disjoint sets or partitions of the original training dataset are used to induce individual decision trees. i.e. for each tree we are selecting fixed number of samples from original dataset without replacement. The size of each partition is same and is decided by the number of trees in Random Forest. This ensures diversity among individual trees. Another measure taken to increase diversity is use of less correlated attributes. A heuristic for this is to have different subsets of attributes for best split selection at each node. To achieve a balance between strength and correlation, at each node creation, we have randomly taken subsets of total m attributes as (2/3*m) and (1/3*m). Then we selected √m attributes from this subset, as it is done in original Random Forest. The values 1/3 and 2/3 are based on heuristics. Here as each individual tree is trained with less number of samples,

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

the learning of each tree and hence learning of forest is more efficient. The results are presented in Table 1. Original Random Forest and Disjoint Partitioning approach are compared on the basis of time for learning, and accuracy. We have tested this approach on many datasets and found out that the approach works well on datasets which are highly imbalanced in nature (especially datasets from medical diagnosis field). To support our observation, we have generated two synthetic datasets of imbalanced nature using Agrawal generator from weka tool. For every dataset, we have varied number of trees in Random Forest from 2 to 10. This is done because for datasets of moderate size, generating more number of disjoint partitions from dataset will affect learning negatively. Also learning of Random Forest will be efficient if it contains less number of trees. Along with this, time taken to build Random Forest is also recorded in each case. To ensure that accuracy achieved with Disjoint partitioning approach is comparable with Random Forest, the original Random Forest is run by varying number of trees from 2 to 100. If maximum accuracy is not obtained within first 2 to 10 trees, then the maximum accuracy value between 11 to 100 trees, number of trees for getting maximum accuracy, and time taken is recorded. Readings are taken for Disjoint Partitioning with attribute subset as (1/3*m) where m is total number of attributes – this is called as DP (1/3) and Disjoint Partitioning with attribute subset as (2/3*m) – called as DP (2/3).

Readings for time taken to learn the forest are in seconds. With weka tool, we were able to record time up to millisecond level. Hence the time values below milliseconds are recorded as 0. Table 1 gives readings for Maximum % Accuracy and Learning Time for the datasets under experimentation. It also contains the number of trees for which the maximum accuracy is achieved. The experimental readings prove that with Disjoint Partitioning approach, Random Forest learns in less time and at least with the same or increased accuracy as that of the original Random Forest.

B. Weighted Hybrid decision tree model (WHDT)

In the process of decision tree induction, an attribute evaluation or split measure is used to decide best split at each node of the decision tree. Earlier, R Sikonja [15] has done experiments with Random Forest by using different split measures. Each measure has its own pros and cons. E.g. Information gain is biased towards multi-valued attributes; Gain ratio reduces biasing, etc. Hence no split measure is the best and the choice may depend on the nature of dataset at hand. With our empirical analysis [12], it is found that there is not much variation in accuracy achieved for Random Forest classifier with different split measures. By considering these two points, a hybrid approach is proposed for decision tree induction. In this approach, at each node split, one of the three chosen split measures (Information gain, Gain ratio and Gini index) is selected randomly. Due to hybrid approach, there is possibility of achieving balance with pros and cons of various split measures, and another reason is with Hybrid approach there is possibility of generating diverse decision trees.

The hybrid decision tree model is augmented with weighted voting. In [15] and [19], it is verified that weighted voting yields better results with Random Forest. With our approach, weight of individual decision tree is computed based on OOB error [6] of individual tree. If it is less than average OOB error of the entire forest, then the tree is assigned higher weight, and vice a versa. Here OOB error is considered as measure of strength of individual tree.

OOB error for the forest is computed by taking average of OOB errors of individual trees. OOB error for an individual tree is calculated on unseen data (which is not selected as a part of bootstrap sample for the tree under consideration). Table 2 presents comparative results for % accuracy for original Random Forest, Random forest with hybrid decision tree, and Random Forest with hybrid decision tree and weighted voting.

C. Optimal Subset of Random Forest

Based on literature survey, we found that finding an optimal subset of Random Forest classifier is still an open research problem [11]. We found that the problem of selection of optimal subset of Random Forest follows the dynamic programming paradigm. Applying this approach to various UCI data-sets, corresponding subsets are obtained and studied.

Dynamic Programming [7] is an algorithm design method that can be used when the solution to a problem can be viewed as the result of sequence of decisions. It solves problems by combining solutions to sub-problems. It is applicable when the sub-problems are not independent, that is, when sub-problems share sub-sub-problems. A Dynamic Programming algorithm solves every sub-sub-problem just once and then saves its answer in a table, thereby avoiding the work of re-computing the answer every time the sub-problem is encountered.

Experiments have been performed in Java using Weka machine learning library. Since the time-complexity is O (2^N), the number of trees of Random Forest (i.e. N) is taken to be 15 to keep processing time within reasonable limits. For each dataset, $(2^{15} - 2)$ subsets of Random Forest are generated and 10-fold cross-validation is performed on each of these subsets. The resulting accuracy A(S) is compared with 10-fold cross-validated accuracy of original Random Forest A(RF). The subset is stored if accuracy of subset is greater than original Random Forest or if its size is less than that of Random Forest.

Mathematically, it can be expressed as:

$$((A(S) > A(RF)) \;||\; ((A(S) == A(RF) \;\&\&\; \text{Size of } S < \text{Size of RF})))$$

Figure 2 shows plots of subset accuracy vs. subset size. Only subsets having accuracy greater than or equal to original Random Forest have been plotted. In order that accuracy plots

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

of two subsets of same size do not overlap each other, it has been distributed randomly in the range between current subset size and next higher one. E.g. Accuracy measure corresponding to subset of size S is plotted in the range [S, S+1).

Table 1. Comparative results of Random Forest and Disjoint partitioning approach

Dataset	% Accuracy			Number of Trees			Learning Time (in seconds)		
	RF	DP(1/3)	DP(2/3)	RF	DP(1/3)	DP(2/3)	RF	DP(1/3)	DP(2/3)
Breast Cancer	70.97	**72.02**	69.23	61	2	12	1.03	**0**	0.016
Haberman	66.99	**71.24**	**71.24**	2	3	3	0	**0**	**0**
POP data	66.33	67.77	**68.88**	2	4	11	0.02	0	**0.015**
Onehr	97.14	**97.23**	97.2	4	17	7	0.56	**0.062**	0.078
Synthetic1	98.4	98.4	**98.8**	2	2	2	0.02	0	**0.015**
Synthetic2	**98.7**	98.34	98.48	30	5	5	**0.84**	0.031	0.032

In table, 0 indicates time recorded is less than milli-seconds
DP(1/3) – Disjoint Partitioning approach with (1/3*m) attribute subset
DP(2/3) – Disjoint Partitioning approach with (2/3*m) attribute subset

Table 2. Comparison of % accuracy values

Dataset	%Accuracy original RF	%Accuracy Hybrid DT	%Accuracy Hybrid DT with weighted voting
breast_cancer	65.38	70.98	72.02
diabetes	75.78	76.43	76.30
ecoli	84.52	85.42	86.01
glass	78.50	79.44	80.37
vowel	98.59	98.69	98.59
segment	97.92	98.10	98.10
sonar	84.62	85.10	86.06
vehicle	74.94	76.60	76.36
musk1	95.80	97.27	97.06
car	93.98	94.68	94.44
anneal	99.55	99.78	99.78
dermatology	94.54	96.45	96.72
credit-a	85.22	87.10	87.25
flags	59.28	61.86	63.40

It can be observed that there are multiple subsets of different sizes having accuracy higher than the original Random Forest of size 15 and that many of these have same accuracy. The size of these subsets varies from one dataset to another. The optimal subset among these is the one with the highest accuracy.

A. Diversity Based Dynamic Pruning (DBDP)

Pruning of Random Forest is eliminating some of the base decision trees from Random Forest to reduce its size without affecting its accuracy. Most of the earlier work [1][14][21] in this direction is based on "overproduce and choose" strategy [16] and is static in nature. In [18] attempt is made for dynamic pruning, but generation of unwanted trees is still there. In [3], the approach is truly dynamic, but it eliminates inherent parallelism in Random Forest. We propose a new approach which is dynamic in nature and preserves the inherent parallelism in Random Forest. The key idea with our approach is to generate only those trees which are having maximum diversity i.e. they are less correlated with each other. This is to be in line with the principle that the generalization error of Random Forest reduces with increase in diversity among the base decision trees.

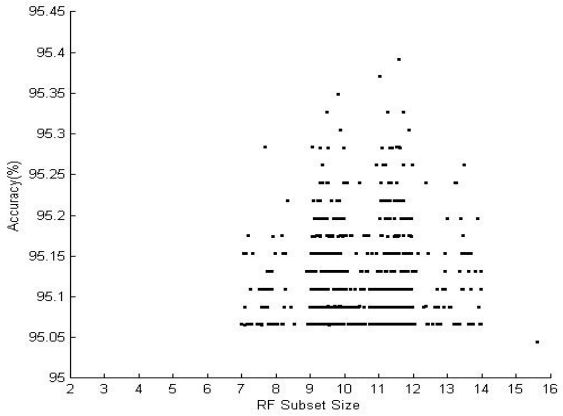

Fig 2 a. Spambase dataset

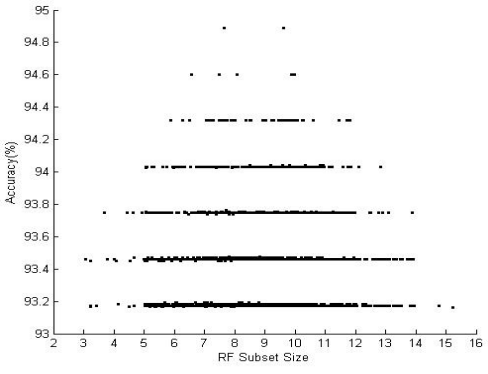

Fig 2 b. Ionosphere dataset

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

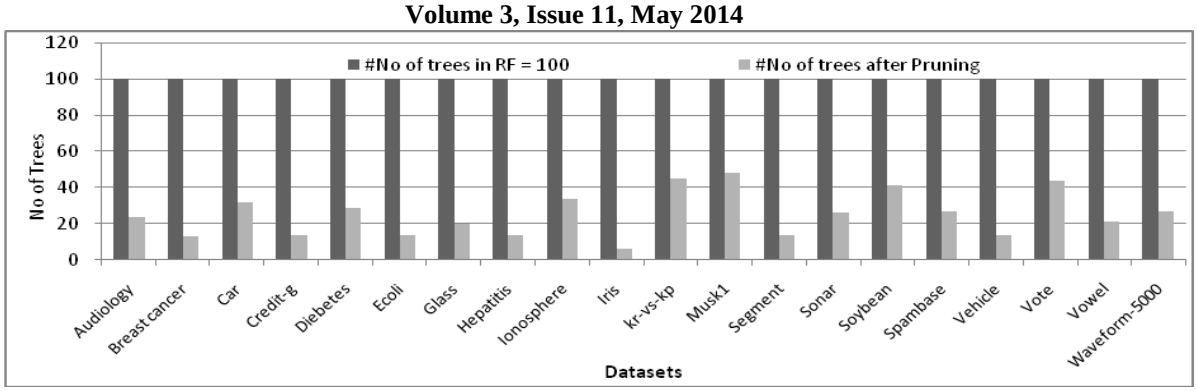

Fig 3. Bar graph showing reduction in size achieved after Pruning of Random Forest for various Datasets

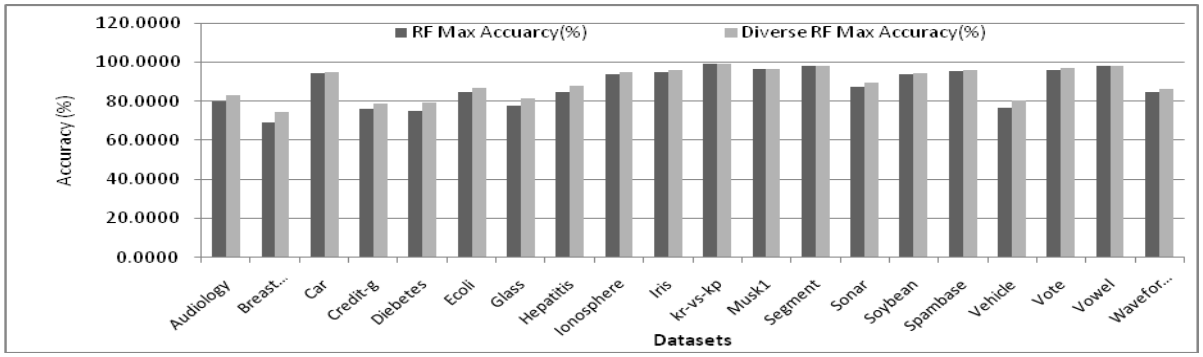

Fig 4. Comparison of Accuracy (%) of original RF and Dynamic Pruning (Diverse) RF

To achieve this, we proposed ranking of bootstrap datasets based on their diversity. As the approach is based on diversity of base classifiers, we also call it as Diversity Based Dynamic Pruning (DBDP). To generate k base decision trees, k bootstrap sample sets are generated by randomly sampling the original training dataset with replacement. The k datasets are ranked based on their diversity with each other. To achieve this, for each column / attribute in the dataset, a consolidated value (V) for that column is generated by taking Root Mean Squared (rms) value for the column. The nominal (categorical) attributes are enumerated to the numerical values so as to compute root mean squared value. Thus a vector of values (V_1, V_2... V_m) is formed for each bootstrap dataset with m columns / attributes. On these vectors Euclidian distance formula is applied to find distance between two vectors which in turn will give distance between two datasets. A Distance matrix is formed to record the distances between a pair of datasets. Based on distance matrix, datasets are ranked.

To verify diversity of the trees generated by our approach, Q statistics measure of diversity [13] is used. The number of trees in original Random Forest is set to 100. This setting is as per suggested by Breiman that fixing the number of trees to 100 is reasonable to reach the generalization error convergence for Random Forest. The accuracy of Random Forest is recorded by setting default node size and m_{try} (number of attributes at each node split) values. For diversity based dynamic pruning approach, accuracy is recorded by varying number of trees from 5 to 100 in step size of 5. The accuracy values are compared with that of original Random Forest for 100 trees. For the datasets under experimentation, DBDP approach achieves accuracy of that of the original Random Forest in less number of trees, and the reduction in size achieved is in the range of 52% to 87%. The comparison of size of original Random Forest and that of DBDP approach to achieve same or greater accuracy of that of original Random Forest is presented in figure 3. Additionally, for almost all datasets, accuracy is increased as compared to original Random Forest. This is because of the inclusion of maximum diverse trees in the forest. These comparative results are presented in the form of bar graph in figure 4.

A. Parallel Random Forest (PDTPRF)

Random Forest can be easily parallelized due to its inherent parallel nature. Being an ensemble, the parallel implementation of Random Forest yields efficient working. This has led to various parallel implementations of Random Forest [8] [23] [25]. We have studied these implementations and found that each implementation is specific to some platform and using a different programming language. Hence there is a need of presenting a generalized parallel algorithm for Random Forest. Another finding out of our survey is that all these implementations are following task parallel approach. They are generating the trees of Random Forest in parallel. We have come up with a new approach in which along with generating the trees in parallel, the individual decision tree is also parallelized.

The readings are taken for each dataset by varying the number of trees from 10 to 50 and the time required to build

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

the tree and test with 10-fold cross validation is recorded in terms of milliseconds. The speedup achieved is presented in Table 3 below.

Table 3. Speedup achieved with PDTPRF approach

Dataset				Number of trees				
Name	Instan	Attrib	Clas	10	20	30	40	50
Breast-cancer	286	9	2	2.85	2.90	3.13	3.91	4.23
Diabetes	768	8	2	1.77	2.55	2.39	2.44	3.09
Hepatitis	155	19	2	2.86	3.57	3.42	3.91	4.23
Musk1	476	168	2	1.76	1.84	2.178	2.19	2.23
Sonar	208	60	2	1.67	2.43	2.53	2.69	2.70
Vote	435	16	2	1.74	2.10	2.80	3.32	3.50
Car	1728	6	4	2.77	2.89	3.25	3.86	4.4
Ionosphere	351	34	24	1.78	2.54	2.35	2.42	2.97
Soybean	683	35	19	2.10	2.15	2.45	2.64	2.74
Vehicle	846	18	4	2.53	2.75	2.87	3.14	3.64
Vowel	990	13	11	1.76	1.84	2.18	2.19	2.24
Waveform-5	5000	40	3	1.44	1.48	1.51	1.55	1.70

IV. CONCLUSION

In this paper, we presented approaches for improving performance of Random Forest classifier in terms of accuracy, and / or time for learning and classification.

In case of accuracy improvement, research is done using different attribute evaluation measures and combine functions. A hybrid decision tree model along with weighted voting is suggested which improves the accuracy. Improvement in learning time mainly concerns on reducing number of base decision trees in Random Forest so that learning and in turn, classification is faster. The approaches suggested in this direction are disjoint partitions of training datasets to learn the base decision trees, and ranking of training bootstrap samples on the basis of diversity. Both these approaches are leading to efficient learning of Random Forest classifier. An attempt is made to find optimal subset of Random Forest classifier using Dynamic programming approach. Random Forest has inherent parallelism and can be easily parallelized for scalability and efficiency. A new parallel approach is proposed in which both, individual tree as well as entire forest is generated in parallel.

The new approaches presented here are leading to effective learning and classification using Random Forest algorithm.

REFERENCES

[1] Bernard S, Heutte L, Adam S, On the Selection of Decision Trees in Random Forest, Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, June 14-19,302-307, (2009).

[2] Bernard S, Heutte L, Adam S, Towards a Better Understanding of Random Forests Through the Study of Strength and Correlation, ICIC Proceedings of the Intelligent Computing 5th International Conference on Emerging Intelligent Computing Technology and Applications, (2009).

[3] Bernard S, Heutte L, Adam S, Dynamic Random forests, Pattern Recognition Letters, 33, Elsevier, 1580-1586 r (2012).

[4] Boinee P, Angelis A, Foresti G, Meta Random Forest, World Academy of Science, Engineering and Technology, Vol 18 (2008).

[5] Breiman L, Bagging Predictors, Technical report No 421, (1994).

[6] Breiman L, Random Forests, Machine Learning, 45, 5-32, (2001).

[7] Cormen, T., Leiserson, C., Rivest, R., Stein, C.: "Introduction To Algorithms" – 2nd Edition.

[8] Grahn H, Lavesson N, Lapajne M, Slat D, A CUDA implementation of Random Forest – Early Results, Master Thesis Software Engineering, School of Computing, Blekinge Institute of Technology, Sweden (2011).

[9] Han J and Kamber M, "Data Mining: Concepts and Techniques" (2nd Edition), Morgan Kaufmann Publisher, (2006).

[10] Krogh A, Vedelsby J, Neural Network Ensembles, Cross Validation, and Active Learning, Advances in Neural Information Processing Systems Vol 7, MIT Press , 231-238, (1995).

[11] Kulkarni V Y, Sinha P K, "Random Forest Classifiers: A Survey and Future research Directions", International Journal of Advanced Computing, Vol 36, Issue 1, 1144-1153.

[12] Kulkarni V Y, Sinha P K, Petare M, Analyzing Random Forest Classifier using Different Split Measures, Proceedings of International Conference on Soft Computing and Problem Solving, Jaipur, India Dec 2012 , Springer AISC series (to be published in march 2013).

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 11, May 2014

[13] Kuncheva L, Whitaker C, Measures of Diversity in Classifier Ensembles and their relationship with the Ensemble Accuracy, Machine Learning, 51, 181-207, (2003).

[14] Latinne P, Debeir O, Decastecker C, Limiting the number of trees in Random Forest, Multiple Classifier Systems, UK (2001).

[15] Robnik M, Sikonja, Improving Random Forests, ECML 2004 Proceedings, Springer, Berlin, (2004).

[16] Roli F, Giacinto G, Vernazza G, Methods for Designing Multiple Classifier Systems, Second International Workshop on Multiple Classifier Systems, Springer-Verlag, (2001).

[17] Schapire Robert E, The Boosting Approach to Machine Learning an Overview, Nonlinear Estimation and Classification, Springer, 2003.

[18] Tripoli E, Fotiadis D, Manis G, Dynamic Construction of Random Forests: Evaluation using Biomedical Engineering Problems, IEEE 2010 .

[19] Tripoli E, Fotiadis D, Manis G, Dynamic Construction of Random Forests: Evaluation using Biomedical Engineering Problems, IEEE 2010.

[20] Witten I. H., Frank E., Weka: Practical machine learning tools and techniques, Morgan Kaufmann publisher, (2005).

[21] Zhang H, Wang M, Search for the smallest Random Forest, Statistics and Its Interface Volume 2, pp 381-388, (2009)

[22] http://archive.ics.uci.edu/ml UCI machine learning repository

[23] http:// code.google.com/p/parf.

[24] http://www.cs.waikato.ac.nz/ml/weka.

[25] http://cran.r-project.org.

AUTHOR BIOGRAPHY

First Author Pursuing PhD at College of Engineering, Pune, India.

Second Author Senior Director, Corporate Strategy and R& D, CDAC, Pune, India