Heuristic Based Improvements for Effective Random Forest Classifier

Vrushali Kulkarni Dr Pradeep Sinha Aashu Singh Farah Shaikh Mehul Mittal

I. Abstract

Random Forest is an ensemble supervised machine learning technique. Based on bagging and random feature selection, number of decision trees (base classifiers) is generated and majority voting is taken for classification. In this paper, we are presenting some heuristic based improvements towards effective learning of Random Forest classifier. These efforts include disjoint partitions of datasets for learning of base trees, reducing depth of base trees by avoiding repetitive selecton of attributes, and selecting smaller subsets of attributes for split at each node. The results of our work are encouraging and there is future research scope in this direction.

Keywords: Data Mining, Classification, Ensemble, Random Forest

II. Introduction

Random Forest is an ensemble supervised machine learning algorithm which is comparable with bagging and boosting. Machine learning techniques have applications in the domain of Data Mining. Classification and Prediction are commonly used tasks in Data Mining where a huge amount of past data is analyzed to predict future trends or values. In this process, a number of input variables named as predictors are used to predict the output variable which commonly known as target. In case where target variable is nominal, the process is known as Classification and where target variable is numeric, it is known as Regression. Random Forest is an ensemble in which base classifiers are Decision Trees. As it is proved theoretically and empirically [3], an ensemble always gives better accuracy than the individual base classifier. Bagging and Boosting are fundamental ensemble techniques. Bagging works on the principle of bootstrap samples, while boosting works by assigning votes to input samples on the basis of their accurate prediction. Bagged ensembles can be built in parallel while boosting is a sequential process. Random Forest is based on the concept of Bagging plus random selection of features. As ensemble consists of multiple classifiers, the time required to learn using ensemble is more as compared to a single classifier. In this paper we are presenting some heuristic based methods to improve learning of Random Forest classifier. The experiments made are use of disjoint partitions of datasets to generate base decision trees, controlling depth of individual decision tree, and reducing the size of attribute set for selection of split at each node. The empirical results are encouraging so that we can continue our work further in this direction.

The paper is organized in following way: Section III gives Theoretical Foundation and Literature review where Random Forest classifier is presented and discussed in detail along with the literature survey. Section IV presents Method and Algorithms for the experiments performed. Section V presents Results and Discussions and section VI gives Conclusion and future work.

PhD student, COEP, Pune, India (e-mail: kulkarnivy@rediffmail.com)
Senior Director, HPC, CDAC, Pune, India
UG students, MIT, Pune, India

III. Theoretical Foundation and Literature Review

A. Random Forest

Definition: A Random Forest is a classifier consisting of a collection of tree-structured classifiers $h(x, \Theta_k)$ $k=1, 2,\ldots$ where the Θ_k are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x. The specialty of this combination is that each decision tree is built from a random vector of parameters [1]. Brieman[2] showed that bagging is effective on unstable learning algorithms like decision tree. Random Forest generates multiple Decision Trees. In this process, the randomization is present in two ways: 1) Generating bootstrap samples from original dataset as it is done in bagging. 2) At each node, a subset of attributes is selected randomly from which an attribute for best split is to be decided. As per Breiman, this algorithm is referred to as Forest RI [1]. Each decision tree in Random Forest is generated in following way: 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 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 the M and the best split on these m is used to split the node. The value of m is held constant during forest growing. Each tree is grown to the largest extent possible. There is no pruning. To classify a new record from an input vector, the input vector is run down each of the trees in the forest. Each tree gives a classification and each tree votes for the class. The forest chooses the classification having maximum votes (over all the trees in the forest). The design parameters for Random Forest are Number of features to be selected randomly for growing each tree (Mtry or K), Number of trees to be generated (Ntree), and Number of samples in the leaf node to be taken as stopping criterion for individual tree (Nodesize). As per Breiman, Mtry or K is the only adjustable parameter to which Random Forest design is sensitive. The

Generalization error of Random Forest is given as,

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

The margin function is given as,

$$mg\ (X,Y) = avk\ I(h_k\ (X) = Y) - max_j \neq_Y\ avk\ I(h_k\ (X) = j)$$

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. Here X is the predictor vector and Y is the classification. Margin is directly proportional to confidence in the classification. Strength of Random Forest is given in terms of the expected value of margin function as,

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

The generalization error of ensemble classifier is bounded above by a function of mean correlation between base classifiers and their average strength. If ρ is mean value of correlation, an upper bound for generalization error is given by,

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

B. Random Forest - Literature Review

Meta Random Forest [9] are based on the concept of using random forest themselves as base classifiers for making ensembles, and the performance of this model is tested and compared with the existing Random Forest algorithm. Meta Random Forests are generated by both bagging and boosting approaches. Comparative study of both these techniques and original Random Forest technique has shown that Bagged Random Forest gives the best results among the three techniques.

Experiments are done with Random Forests of Probability Estimation Trees (PETs) [10]; the result shows that learning Random Forests of PETs using relative class frequency significantly outperforms learning Random Forest of classification trees.

In original Random Forest, Gini Index is used for attribute split by Brieman. Gini Index particularly is not able to detect strong conditional dependencies among attributes . For deciding the splits, Gini index evaluates each attribute separately; assuming conditional independence of attributes, and hence does not give good results with data involving more dependencies of the attributes. ReliefF measure for attribute split gives better results in this case. Robnik and Sikonja [11] experimented with Random Forest using five different attribute measures; each fifth of the trees in the forest is generated using different split measure (Gini index, Gain ratio, MDL, Myopic ReliefF, or ReliefF). This helped in decreasing correlation between the trees while retaining their strengths. The performance increase observed was not much significant.

Experiments are done using different voting schemes instead of majority voting. Dynamic Integration [12] demonstrated that performance of random forest is improved in some domains by replacing majority voting with Dynamic Integration, which is based on local prediction performances of base decision trees.

Robnik and Sikonja [11] used Weighted Voting with Random Forest. They found out average margin of the trees on instances most similar to the new instance (to be classified). Trees with negative margin are discarded, and votes of remaining trees are weighted as per margin. The results show that weighted voting with Random Forest is clearly beneficial though classification takes more time and O (n.K) additional space is needed, where K is number of trees.

Simon Bernard, Laurent Heutte, and Sebastien Adam proposed a new Random Forest algorithm called Forest RK [13] in which K, the number of features, is randomly selected at each node during tree induction process. In this work it is stated that K is not a hyper-parameter as it is not playing a crucial role in generating accurate Random Forest classifier. They used McNemars statistical test of significance to compare predictions generated by original Random Forest due to Breiman, and Forest RK. They claimed that two algorithms are statistically equivalent.

There are specific methods suggested to find a sub-forest that can achieve prediction accuracy of a large random forest [4][5][6], i.e. pruning of random forest. These sub-forests are generally found by following Overproduce and Choose strategy. Here first the forest is grown to a fixed large number of trees, and then one by one each tree is considered for including or removing from the forest by exhaustive enumeration.

Machine Learning algorithms are frequently applied in data mining applications. Many of the tasks in this domain concern high-dimensional data. Consequently, these tasks are often complex and computationally expensive. A GPU-based implementation of Random Forest algorithm is developed, which is based on Compute Unified Device Architecture (CUDA) [14]. The algorithm is experimentally evaluated on NVIDIA GT 220 graphics card with 48 CUDA cores and 1 GB of memory. Both training phase and classification phase are parallelized in CUDA implementation. Performance is compared with two state-of-the-art implementations of Random Forest; one sequential i.e. LibRF and one parallel i.e. FastRF in Weka.

Online Random forest algorithm [15] generates on-line decision trees based on concepts from on-line bagging and extremely randomized forests. It also uses Temporal Weighting scheme to discard non performing trees based on their out-of-bag error performance. The algorithm is ported onto NVIDIA GPU which has shown ten times speed up.

Incremental Extremely Random forest algorithm [16] is specially designed for small data streams. The algorithm works on the basis of expanding the leaf nodes without reconstructing the whole trees. This approach avoids use of Hoeffding bounds which need large number of samples. Random Forest has shown good results for imbalanced data [18], and problems of large P small n paradigm [19]. Fuzzy Random Forest [20] and Random Forest using semi supervised learning approach [17] are also being targeted by the researchers.

IV. Methods and Algorithms

A. Datasets

The aim of this research work is to make some improvements in Random Forest so that the time taken to learn the forest can be reduced. All experiments are carried out on datasets from UCI Machine Learning repository. Each dataset

TABLE I
DATASETS USED FOR EXPERIMENTATION

Dataset	Instances	Attributes	Classes	Imbalanced ?	Attribute Type	Missing Valuese
Hypothyroid	3772	30	4	yes	numeric/nominal	nil
Ionosphere	351	35	24	no	numeric	nil
kr-vs-kp	3196	37	2	no	nominal	nil
sick	3772	37	2	yes	numeric/nominal	20 % for some attributes
sonar	208	61	2	no	numeric	nil
soybean	683	36	19	no	nomial	18 % for some attributes
vehicle	846	19	4	no	numeric	nil
anneal	898	39	5	yes	numeric/nominal	nil
vote	435	17	2	no	nomial	10 % for some attributes
audiology	226	70	24	no	nomial	98% for one attributes
vowel	990	14	11	no	numeric/nominal	nil
waveform	5000	41	3	no	numeric	nil
breast cancer	286	10	2	no	nomial	nil
letter	20000	17	26	no	numeric	nil
mushroom	8124	23?	2	no	nomial	nil
credit-g	1000	21	2	no	numeric/nominal	nil
segment	2310	20	7	no	numeric	nil
splice	3190	62	3	no	nomial	nil
car	1728	7	4	no	nomial	nil
onher	2536	73	2	yes	numeric	12 % for some attributes
spambase	4601	58	2	no	numeric	nil
musk2	6598	169	2	no	numeric/nominal	nil

is divided into training set (2/3rd) and testing set (1/3rd). The accuracy is noted down with varying number of trees. Results are compared with original Random Forest from weka. Also the source code of weka is modified for our experiment. The random seed selection is set to 1 with weka tool so that the results of random forest for different runs, but with same parameters are same and there is no need of averaging. The details of all datasets used are listed in the table 1.

B. Experiment 1

Disjoint partitions of dataset to build base decision tree in Random Forest

The first experiment done is to generate diverse base decision trees. Brieman suggested that to yield less generalization error and hence to get more accuracy, the base trees are to be more diverse, i.e. they should predict differently. For this purpose, we are generating disjoint partitions of original dataset, 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. Though each tree is getting less number of samples here, the sample set for learning any two trees is entirely different and hence the trees are less correlated. We call this new algorithm as Disjoint Partitioning Random Forest (DPRF), but not immediately publishing it as we are still doing some improvements in it those we have mentioned in future work.

Algorithm: Experimental Protocol 1
Let N $\rightarrow$ size of dataset D
n $\rightarrow$ number of trees in Random Forest
t = N / n is size of each partition
for i = 1 to n do
Randomly Sample t instances from D without replacement to generate partition Pi
Discard these t instances from D
end for

Generate Random Forest of trees T1, T2,..., Tn using sample partitions P1, P2, ..., Pn respectively

The results of above algorithm are compared with original Random Forest and are presented in table 2.

C. Experiment 2

Controlling depth of base decision trees by avoiding repetitive selection of attributes In original Random Forest by Brieman, for base decision trees, at each node $\sqrt{m}$ attributes out of total m attributes are selected and the best split among them is decided by using Gini index. This process gets repeated at every node and hence attributes have no control over depth of decision tree. The depth of decision tree is governed by a parameter nodesize . The node is treated as leaf node if it has nodesize instances, and a default value for nodesize is considered as 5. Brieman has stated that this type of tree creation reduces biasing. We have experimented to control this depth of individual base tree through attribute selection process. Once an attribute is selected at a node, then it is discarded from the entire set of attributes. Hence there is no repetitive selection of attributes. This process stops the tree creation when every attribute is considered once. This leads to base trees of reduced depth.

Algorithm: Experimental Protocol 2
Let a_1,..., a_m is set of m attributes
For each tree creation
repeat
Let k = $\sqrt{m}$
Randomly select k attributes out of m

Decide best split out of k attributes as a_k
Split the node on a_k
Remove a_k from attribute list
until end of tree creation

The accuracy of original RF and experiment 2 are recorded for different values of number of trees and compared in table 2.

D. Experiment 3

Heuristic approach to select subset of attributes at each node to generate base decision trees

As per Brieman, Random Forest gives good accuracy if the base decision trees are less correlated. Also Brieman has proved empirically in his paper[1] that increasing number of attributes (for deciding best split) at each node does not increase strength, the strength remains almost constant after a value of 4; but it increases correlation. Hence we are trying to select less correlated features by taking smaller subsets of attributes . A heuristic for this is to have different subsets of attributes for selection at each node. To achieve a balance between strength and correlation, at each node creation, we have randomly taken subset of total m attributes as (2/3*m) where m is total number of attributes. Then we selected $\sqrt{m}$ attributes from this subset, as it is done in original Random Forest. In this way, we are selecting attributes at each node from different subsets and there is a chance that m attributes at each node will be different though they are not disjoint. This leads to more diverse tree creation in Random Forest which can improve accuracy. Table 3 shows results of this experiment.

Algorithm: Experimental Protocol 3
m $\rightarrow$ Total number of attributes of dataset D
A= (a_1, a_2,...,a_m) is set of attributes for dataset D
n$\rightarrow$ total number of trees in Random Forest
p = 2/3*m
for i= 1 to n
generate base decision tree T_i with following steps
for each node in base tree T_i
A1 = (a_1, a_2,...,a_p) , attributes randomly selected from attribute set A
k = $\sqrt{m}$
Randomly select k attributes from A_1
Decide best split on these attributes
Generate decision tree
end for
end for

V. Results and Discussions

Forexperiment 1, our basic thinking was that it should give good results for large datasets as disjoint partitioning of dataset in large datasets can provide sufficient number of samples for each tree. The empirical results are showing that experiment 1 is not giving good results for small datasets (i.e. number of samples less that 1000). It is also not giving good results for dataset of large size but with more number of classes (letter dataset). But it is giving good results for datasets of moderate size and less number of classes. The bar-graph in figure 1 shows that out of all datasets we have tested, 75% of times experiment 1 results are either same as original Random Forest or better. At primary level, our conclusion is that experiment 1 results are not getting affected by the nature of dataset, i.e. whether it is balanced or imbalanced; but we need to do more experimentation on this. We think experiment 1 as a good achievement as with this experiment, the learning time for Random Forest is reduced. Our future work to continue in this direction is to increase size of each disjoint partition by randomly sampling instances with replacement inside the partitions (similar to bootstrap); this will help in increasing the strength of individual tree. Also we will testexperiment 1 with large datasets and less number of classes. The aim of experiment 2 was to limit the depth of each tree by avoiding repetitive selection of attributes in generating base decision trees. The results are presented in table 3.These empirical results show that this experiment is not giving good results. The reason analyzed is that at the deeper levels of tree creation, limited attributes are available. The attributes available at those nodes may not be generating pure partitions and hence reducing the overall classification capability of the decision tree. The results for experiment 3 are presented in table 4. As per our expectation, it is giving good results for datasets where number of attributes is moderate (i.e. in the range 10-50). With datasets having large number of attributes, the results are not good as selecting 2/3*m gives a subset of large size which reduces strength and increases correlation (as per Breiman[1]). The future work in this direction is to test subsets of (1/2*m) or (1/3*m) for large datasets. The results of this experiment are not getting affected by number of classes. The bar-graphs in figure 2 and 3 shows that for 77% out of total readings, our results are either same or better than that of original Random Forest. Overall these three heuristic experiments for effective learning using Random Forest gave very good insight for in depth study of Random Forest classifier and are encouraging for continuing our future work in this direction

VI. Conclusion

The heuristic based approaches presented in this paper are carried out with a goal of achieving effective learning using Random Forest classifier. With all the experiments, we are trying to achieve learning of base decision tree with either less number of instances or less number of attributes. This will help in learning of individual tree and in turn the entire forest in lesser time. The encouraging results of two of our experiments are leading us to do further work in this direction. The future work is to improve the concept of disjoint partitioning by random sampling with replacement within the partition, Selecting smaller subsets for datasets with large attributes, and improvements related to datasets with multiple classes.

TABLE II
RESULTS OF EXPERIMENT 1

Graphs are on next page

Dataset	Acuracy (50)		Accuracy(100)		Accuracy(150)		Accuracy(200)		Accuracy(250)		Accuracy(300)	
	RF	Exp1	RF	Exp1	RF	Exp1	RF	Exp1	RF	Exp1	RF	Exp1
Anneal	78.59	77.92	73.57	61.87	77.25	81.93						
Car	78.12	78.12	70.48	68.92	70.48	67.53	65.27	65.10	69.44	69.44	70.48	55.03
Credit-g	70.27	69.96	71.77	71.77	26.72	26.72	63.96	58.55				
Hypothyroid	93.39	93.39	93.79	94.03	92.28	92.28	92.60	92.60	92.28	92.28	92.12	89.26
Kr vs kp	83.28	81.12	74.08	76.33	59.15	62.44	70.42	72.86	60.84	62.15	63.47	64.97
Letter	60.95	39.90	53.09	42.90	39.90	37.11	31.92	28.44	31.08	29.05	24.93	26.94
Mushroom	98.04	98.44	93.46	93.83	90.32	91.32	93.79	94.53	88.99	88.99	96.23	97.48
Musk2	85.58	86.08	85.17	83.71	86.03	86.53	87.13	87.94	85.49	85.35	86.22	86.22
Onher	97.27	97.15	96.80	96.68	96.44	96.44	96.80	96.80	96.80	96.80	96.80	96.80
Segment	80.64	84.28	62.98	60.90	47.14	47.66	43.11	48.96	38.57	39.48	26.75	27.79
Sick	95.14	94.82	92.99	92.99	94.03	94.43	94.03	94.03	93.63	92.20	93.71	93.71
Soybean	33.03	31.71	25.11	28.19	17.18	16.29						
Spambase	89.62	88.84	86.10	86.75	86.49	85.32	79.71	80.30	78.08	78.60	45.46	51.33
Splice	64.06	63.59	54.93	64.34	51.92	52.21	50.70	51.45	53.52	53.15	38.09	43.08
Vehicle	49.29	48.58	34.04	37.94	34.04	28.01						
Vowel	20.30	16.96	17.87	16.66	18.18	20.30	10.6	8.78				

TABLE III
RESULTS OF EXPERIMENT 2

Dataset	Acuracy (50)		Accuracy(100)		Accuracy(150)		Accuracy(200)		Accuracy(250)		Accuracy(300)	
	RF	Exp2	RF	Exp2	RF	Exp2	RF	Exp2	RF	Exp2	RF	Exp2
Car	94.09	73.26	94.09	71.52	92.53	70.65	94.44	72.39	93.92	70.65	91.31	68.05
Hypothyroid	99.20	93.79	99.52	93.95	99.28	93.15	99.44	93.87	99.44	94.51	99.20	94.51
Kr vs kp	98.59	90.23	99.24	93.70	99.06	94.17	99.43	90.89	98.77	93.89	98.49	94.92
Letter	95.76	55.17	95.87	55.14	95.79	56.28	96.00	53.81	96.02	53.60	95.99	54.89
Mushroom	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Musk2	98.81	97.81	99.72	98.49	98.63	98.04	99.04	98.45	99.27	98.22	99.40	98.22
Onher	96.68	96.80	97.15	97.15	97.51	97.51	96.56	96.56	95.97	95.97	97.15	97.15
Segment	97.40	93.89	97.01	78.70	97.40	92.33	97.53	93.11	98.05	92.85	98.18	93.37
Soybean	92.51	85.90	92.07	88.10	93.83	88.54	94.27	94.71	92.07	87.22	92.07	89.42
Spambase	95.17	93.28	95.10	92.49	96.60	94.19	95.69	92.04	96.08	93.41	96.34	94.06
Vehicle	72.69	67.73	69.85	64.18	75.88	69.85	78.36	71.27	72.69	70.56	72.69	67.37
Audiology	70.66	56.0	77.33	65.33	77.33	62.66	74.66	72.0	74.66	57.33	68.0	64.0
Breast cancer	73.68	71.57	73.68	77.89	71.57	66.31	72.63	67.36	74.73	67.36	75.78	77.89
Ionosphere	90.59	92.30	94.87	94.87	92.30	91.45	96.58	96.58	93.16	94.01	94.87	94.87
Sonar	76.81	78.26	79.71	76.81	85.5	84.05	85.50	84.05	78.26	81.15	81.15	84.05
Vote	95.86	93.10	93.79	92.41	96.55	95.86	95.86	95.86	96.55	94.48	97.24	94.48

TABLE IV
RESULTS OF EXPERIMENT 3

Dataset	Acuracy (50)		Accuracy(100)		Accuracy(150)		Accuracy(200)		Accuracy(250)		Accuracy(300)	
	RF	Exp3	RF	Exp3	RF	Exp3	RF	Exp3	RF	Exp3	RF	Exp3
Anneal	99.66	98.99	100.0	99.66	98.99	98.99	99.66	99.66	99.33	99.33	100.0	100.0
Car	94.27	81.77	94.27	83.50	92.53	84.37	94.09	83.68	93.22	83.33	94.27	87.15
Credit g	74.77	74.77	75.67	72.67	75.07	76.87	77.17	78.07	76.27	76.27	75.97	75.07
Hypothyroid	99.28	98.17	99.12	98.56	98.96	95.06	99.52	98.01	99.52	97.13	99.36	97.05
Kr vs kp	98.49	98.02	99.34	98.68	98.96	98.77	98.96	98.40	99.24	98.96	98.77	98.30
Letter	95.82	95.91	95.69	95.93	95.99	96.36	96.20	96.39	95.69	96.12	96.27	96.42
Mushroom	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Musk2	99.27	97.99	99.49	98.63	99.09	99.54	99.31	98.99	99.13	99.09	99.13	98.90
Onher	96.56	96.56	96.92	96.92	97.04	97.04	96.92	96.92	97.51	97.51	97.04	97.04
Segment	97.79	97.66	98.31	98.57	98.31	98.18	98.31	98.44	98.05	97.92	97.40	97.27
Sick	97.77	96.89	97.69	96.34	98.24	97.29	98.72	97.61	98.01	96.49	98.17	96.73
Soybean	91.62	92.07	90.30	90.30	92.95	94.71	93.39	94.27	90.30	89.86	91.18	92.07
Spambase	94.74	94.52	95.49	95.23	95.10	95.43	95.89	96.47	95.04	95.23	94.84	94.84
Vehicle	73.04	73.04	74.11	75.53	76.95	74.11	71.98	74.11	75.17	75.88	74.82	74.46
Vowel	96.36	94.54	96.96	96.66	97.27	96.96	93.33	94.24	98.48	98.78	96.66	96.36
Waveform	85.11	84.87	85.47	85.71	85.83	86.07	85.89	86.01	84.45	84.69	85.29	85.47
Audiology	73.33	77.33	76.0	76.0	66.66	69.33	66.66	68.0	73.33	74.66	76.0	78.66
Breast Cancer	69.47	71.57	67.36	70.52	75.78	75.78	66.31	67.36	72.63	71.57	62.10	63.15
Ionosphere	91.45	92.30	94.01	94.01	92.30	91.45	94.01	94.87	93.16	94.01	94.01	94.01
Musk1	87.34	86.07	93.03	93.67	95.56	96.83	92.40	94.30	96.20	96.20	94.30	92.40
Sonar	84.05	82.60	88.40	86.95	84.05	86.95	88.40	88.40	82.60	78.26	85.50	85.50
Vote	96.55	96.55	93.79	94.48	96.55	96.55	95.17	95.17	94.48	93.79	96.55	96.55

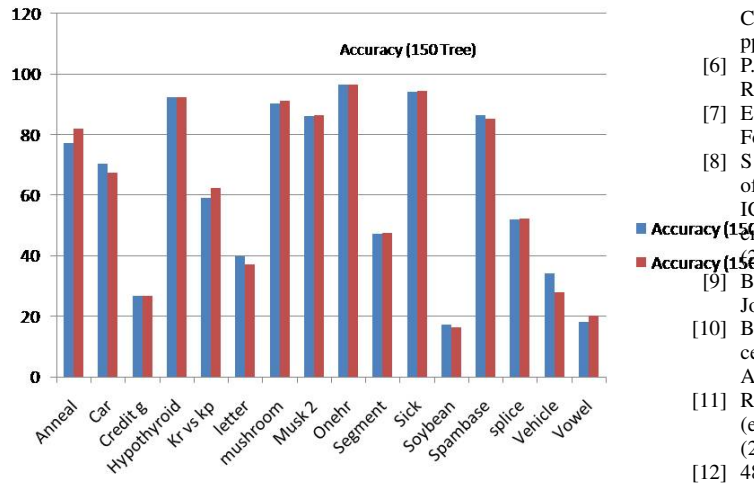

Fig. 1. Bargraph showing comparative results of RF and experiment 1

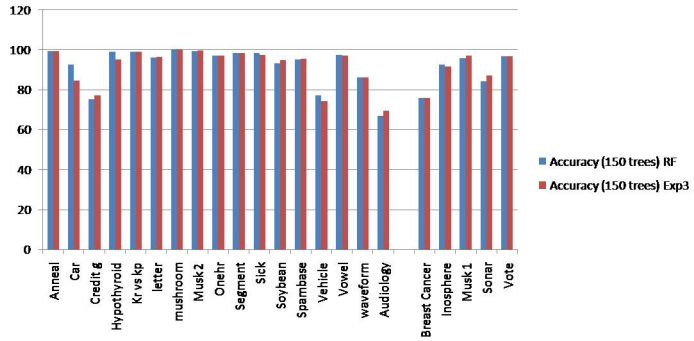

Fig. 2. Bargraph showing comparative results of RF and experiment 3 at 150 trees

REFERENCES

[1] Leo Brieman, Random Forests, Machine Learning, 45, 5-32, (2001)

[2] Leo Breiman, Bagging Predictors , Technical report No 421, (September 1994)

[3] David Opitz, Richard Maclin, Popular Ensemble Methods: An Empirical Study, Journal of Artificial Intelligence 11, 169-198, (1999)

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

[5] Simon Bernard, Laurent Heutte, and Sebastien Adam, On the Selection of Decision Trees in Random Forest, Proceedings of International Joint Cobference on Neural Networks, Atlanta, Georgia, USA, June 14-19, pp 302-307, (2009)

[6] P. Latinne, O. Debeir, C. Decastecker, Limiting the number of trees in Random Forest, MCS, UK (2001)

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

[8] S Bernard, L Heutte, and S Adam, 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)

[9] Boinee P, Angelis A and Foresti G, Meta Random Forest, International Journal of Computational Intelligence 2, (2006)

[10] Bostrom H, Estimating Class Probabilities in Random Forests, Proceeding of the International Conference on Machine Learning and Applications, 211216, (2007)

[11] Robnik M, Sikonja, Improving Random Forests, J F Boulicaut et al (eds): Machine Learning, ECML 2004 Proceedings, Springer, Berlin, (2004)

[12] 48. Tsymbal A, Pechenizkiy M and Cunningham P, Dynamic Integration with Random Forest, ECML, LNAI, 801-808, Springer-Verlag (2006)

[13] Bernard S, Heutte L and Adam S, Forest-RK : A New Random Forest Induction Method, Proceedings of 4th International Conference on Intelligent Computing: Advanced Intelligent Computing Theories and Applications with Aspects of Artificial Intelligence, Springer-Verlag, (2008)

[14] Grahn H, Lavesson N, Lapajne M and Slat D, A CUDA implementation of Random Forest Early Results, Master Thesis Software Engineering, School of Computing, Blekinge Institute of Technology, Sweden

[15] Saffari A, Leistner C, Santner J, Godec M and Bischof H, On-line Random Forests, ICCV IEEE, Conference Proceedings 1393-1400, (2009)

[16] Wang A, Wan G, Cheng Z and Li S, An Incremental Extremely Random Forest Classifier for Online Learning and Tracking, 16th IEEE International Conference on Image Processing,1449-1452, (2009)

[17] Leistner C, Saffari A, Santner J, Godec M and Bischof H, Semi-Supervised Random Forests, ICCV IEEE, Conference Proceedings, 506-513 (2009)

[18] Chain C, Liaw A and Breiman L, Using Random forest to Learn Imbalanced Data, Technical Report, Department of Statistics, U. C. Berkley (2004)

[19] Kosorok M and Ma S, Marginal Asymptotics for the Large p Small n paradigm: With Applications to Microarray Data, Ann Statist 35, 1456-1486, (2007)

[20] Bonissone P, Cadenas J, Garrido M and Diaz-Valladares R, A Fuzzy Random Forest, International Journal of Approximate Reasoning, 51, 729-747, (2010)

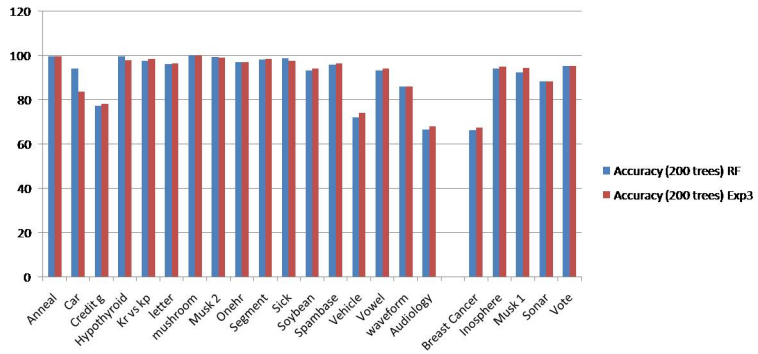

Fig. 3. Bargraph showing comparative results of RF and experiment 3 at 200 trees