High Performance Random Forests

Hyontai Sug

***Abstract*—** Because of the property of data fragmentation in decision trees that makes them more dependent on available data sets, random forests were invented to cope with the data fragmentation. Random forests generates their own decision trees with two parameters, the number of attributes that can be the candidates of split in each subtree and the number of trees in the forests. There are known recommended default parameters. But, the default parameters may not work well always. This paper investigates what kind of property of data sets may not be good for the default parameters, especially the first parameter of the number of attributes that can be the candidates of the split, and propose a guide line.

***Keywords*—**Random forests, decision trees, data preprocessing, random sampling.

I. Introduction

GENERATING high performance classifiers is a major research interest in the task of data mining or machine learning. Among the related research, decision tree-based methods and neural network-based methods can be some two representative results of the research. There are many examples that use the methods [1, 2, 3, 4, 5].

A good point of decision tree-based methods is relatively short time to train so that scalability is good [6, 7]. On the other hand, even though neural network-based methods have a good performance, the methods need very long computing time to train so that scalability of the neural network based methods is limited [8, 9]. On the other hand, the weak point of decision trees in accuracy is innate. Because decision tree algorithms fragment data sets during training, even if the size of available data set is large, training data are drained quickly due to the data fragmentation [10].

Random forests [11] were invented to overcome the weakness of decision trees by resorting to many randomly generated decision trees in the forests. Random forests are known to be robust for missing and erroneous data to some degree, and known that the accuracy of random forests is comparable to that of SVM [12]. The accuracy of SVM is known that it is one of the most accurate classifiers. Moreover, we may say that random forests are more useful than SVM in most cases, because random forests can treat multiple classes naturally, while the original SVM can treat only binary class problems, if we do not add any other functions.

But, the accuracy of random forests for a data set can vary depending on the number of trees in the forests and the number of attributes to pick randomly that determines the subtree of each branch in the tree. So, in this paper we want to find some guideline based on the characteristics of data sets to train. In section two we will discuss about related work, and in section three we will discuss about insights for better random forests based on experiments performed for random forests with a variety of data sets, and in section four some conclusions will be given.

This work was supported by Dongseo University's Research Year of 2014.

H. Sug is with Dongseo University, Busan, 617-716 Korea (corresponding author to provide phone: 51-320-1733; fax: 51-327-8955; e-mail: sht@gdsu.dongseo.ac.kr).

II. Related Work

Because decision tree algorithms decisively divide a training data set based on the branching criteria during training, the performance of decision trees is more heavily dependent on the training data sets than other data mining algorithms like neural networks that do not decisively divide a training data set [13]. So, we can infer that the performance of decision trees is dependent on the composition of data in the training data sets.

As a way to mitigate the decision trees' dependency in performance to a specific training data set, many decision trees can be trained based on the same data sets, but the training data sets come from sampling with replacement and vote for classification. The method is called BAGGING [14]. BAGGING stands for Bootstrap AGGregatING. Random forests are based on BAGGING, and additionally, the trees are built based on some random selection of attributes and with no pruning.

Because tree building process is independent from random sampling in random forests, some researchers use entropy-based measure [15, 16], while the original algorithm uses a purity-based measure [14, 17] in building the trees for their random forests. It has been known that tree generation of random forests is relatively faster than other data mining algorithms like neural networks [18].

Some other researchers tried to find better knowledge models based on randomness in attribute selection and training data sets. Random subspace method [19] tries to select the subset of attributes randomly and applies aggregating to find better classification models. On the other hand, SubBag method [20] tries BAGGING and random subspace method together. It was also combined with decision tree algorithm based on C4.5 and rule generator based on RIPPER(Repeated Incremental Pruning to Produce Error Reduction) by Cohen [21].

Because each decision tree in the random forests is independent, Boström tried concurrent training of each decision tree in the random forests using a concurrent programming language [15]. The random forests were made based on the default number of attributes to pick randomly and have massive number of trees up to 100,000 trees in forests.

III. INSIGHTS FOR BETTER RANDOM FORESTS

A. The Issues of Random Forests

Because decision tree algorithms split a given data set, the data set is fragmented so that data for training is consumed rapidly. This data fragmentation phenomenon makes the decision trees more dependent on the available data sets than other data mining or machine learning algorithms. In order to mitigate this property of decision trees, random forests could be used. There are many decision trees in the random forests, and each training data set for each tree in the forests is made based on sampling with replacement called bootstrap method [23]. Statistically, 63.2% of instances are sampled in average from the original training data set in the bootstrap method. In other words, 36.8% of the instances in the training set are not selected in average. So, bootstrap method is a kind of oversampling method, and the unselected instances may be used for testing.

After the sampling decision trees are generated based on some conventional decision tree generation algorithms without pruning step. But, while conventional decision tree generation algorithms consider all yet unselected attributes to choose the best split in the root of each subtree, decision tree algorithms in random forests consider arbitrary number of attributes randomly, where the arbitrary number is based on user's choice, then the best split among the selected attributes is chosen like conventional decision tree algorithms.

Breiman, who is the inventor of random forests, used the number, the first integer less than $\log_2 A + 1$ where A is the number of attributes of the target data set [11]. Later, he recommended square root of A also as the number, and also half and double of the number if time permit [23, 24]. Another issue is the number of decision trees in the forests. According to Breiman tens to hundreds of decision trees are enough, because he concluded that thousands of trees do not give better performance than the smaller number of trees in the forests. More recently, Boström [15] empirically showed in his experiments that better accuracy could be found in average, when the number of tree was 1000. He generated random forests of 100 trees, 1,000 trees, 10,000 trees, and 100,000 trees for 34 different data sets.

B. Data Sets for Experiment

There are two parameters when we generate random forests; the number of trees in random forests and the number of randomly chosen attributes to generate each tree. We can also understand that the second parameter is used to mitigate the degree of dependency on the splitting criterion of decision tree algorithm of the random forests by adjusting the number of attributes to consider for the best split. Different splitting criteria have different effect for the performance of decision trees. For example, C4.5 [16] that is one of the most popular decision tree algorithms uses the criterion of gain ratio which is based on entropy, while CART [17] that is also one of the most popular decision tree algorithms uses the criterion of Gini index which checks the purity of child nodes, and as a result, the two algorithms have different performance for the same data sets. If the number is equal to the whole number of attributes, the tree generation process becomes identical to that of conventional decision tree algorithms without pruning. As the number becomes smaller, the degree of dependency on splitting criterion becomes lighter. If the number is one, it just selects an attribute randomly for split. We name the parameter of the degree of randomness in building decision trees as drT.

Let's consider the first parameter of the number of trees in the forests. It may compensate the data fragmentation problem somewhat, because each tree will be made from the samples that are based on sampling with replacement. But, it has some limited effect only, because no new instances that may improve the performance of the decision tree can be made. Repeatedly sampled instances in minor classes have more effect in the tree, because of the rates of increase in the number of instances in the minor classes are larger than those of major classes. Minor classes are the classes that have smaller number of instances of the classes, and have more conflicting class values in their instances, while major classes have opposite property.

Among the two parameters we can conclude that the second parameter, drT, has more effect in the performance of decision tree, in case that we have insufficient number of instances or insufficient information for accurate classification.

We want to see what kinds of data sets need more degree of randomness in building decision trees, so we consider in the experiment the cases of drT being one and also the default value, which is the first integer less than $\log_2 A + 1$. For the number of trees in forests we consider the cases of the number of trees in random forests being 100 and 1000 only. 100 trees are chosen because of Breiman's guideline of hundreds of trees in the forests [11]. The parameter of 1000 trees in the forests is chosen, because it produced the best accuracy in average in the experiments conducted by Boström [15]. Experiments were run using 33 different data sets of classification task in UCI machine learning repository [25]. Table 1 shows the property of the data sets.

Table 1. The property of data sets

Data set	No. of attributes	No. of instances	No. of classes
Abalone	9	4177	29
Anneal	39	898	6
Arrhythmia	280	452	16
Audiology	71	226	24
Bridges	13	108	7
Bupa	7	345	2
Credit screening (Japanese)	16	690	2
Cylinder bands	40	540	2
Db world e-mails (bodies)	4703	64	2
Dermatology	35	366	6
Glass	11	214	7
Hayes roth	6	132	3
Hepatitis	20	155	2

Iris	5	150	3
Lenses	5	24	3
Lung cancer	57	32	3
Post operative	9	90	3
Sonar	61	208	2
Stalog_vehicle	19	752	4
Spect	23	267	2
Teaching assistant evaluation	6	151	3
Vertebral column 2C	7	310	2
Vowel	14	990	11
Yeast	9	1484	10
Zoo	18	101	7
Car	7	1728	4
King rook vs king pawn	37	3196	2
Internet ads	1559	3279	2
Magic4	11	19020	2
Nursery original	9	12960	5
Ozone 8 hour	73	2534	2
Stalog landsat satellite	37	6435	7
Stalog shuttle	10	58000	7

C. The Classification of Data Sets

We know that not only decision tree algorithms have the tendency of preferring major classes and fragment data sets in the subtrees, but also the many decision trees in random forests can have similar tendency that is dependent upon the number of attributes to pick randomly. But this tendency can be mitigated for large data sets, because we can confront with the data fragmentation problem less. On the other hand, the data fragmentation problem could become worse, if we have some unbalanced class value distribution in the data sets of small size. So, we want to divide the target data sets into three cases based on the number of instances per class and class value distribution:

- small and unbalanced,
- small and balanced, and
- large based on experiment

So, random forests were generated based on the following combination of the parameters:

- {drT=1, the number of trees in random forests=100},
- {drT=1, the number of trees in random forests=1000},
- {drT=default value, the number of trees in random forests=100},
- {drT=default value, the number of trees in random forests=1000}

In the experiment 10-fold cross validation was used. So, a data set is divided into ten equal subsets and each subset was used for test, while nine other subsets were used for training. Random forests in weka were used for the experiment. Weka is a comprehensive data mining package written in java [26].

In order to divide the data sets based on the degree of balance in class value distribution, purity was checked based on Gini index [12].

The purity of data sets can be calculated by Gini index;

,

$$G = 1 - \sum f_i^2, \; i = 1, \ldots, m, \tag{1}$$

where f_i is the fraction of instances having class value i and m is the total number of class values. We can use equation (1) as a basis to determine whether a class distribution is balanced or not. If each f_i has the same value, the data set has balanced distribution in class values. For example, if we have a data set of perfectly balanced distribution with 2 classes, each f_i is 0.5, so $G = 1 - (0.5^2 + 0.5^2) = 0.5$. This value changes depending on the composition of class values. For example, if each f_1 is 0.25 and f_2 is 0.75, then G = 0.375.

Let G_b and G_d are the Gini indexes for the data set of perfectly balanced class distribution and the data set's own class distribution respectively. We may define the class distribution of data set is relatively unbalanced, if

$$\log_{10} G_b - \log_{10} G_d > \alpha \tag{2}$$

The α value should be set carefully to reflect the property of data sets. Table 2 shows some possible values of equation (2) as the ratio of class 1 and class 2 changes for the data set having 2 classes.

Table 2. The sample values of equation (2) as the ratio of class 1 and class 2 change

Ratio of class 1	Ratio of class 2	Value of equation (2)
0.3	0.7	0.75721
0.4	0.6	0.017729
0.425	0.575	0.009883
0.45	0.55	0.004365
0.475	0.525	0.001087

If we look at table 2, the third row has our approximation on whether a data set is some balanced class value distribution or not, So α = 0.01 was used as the criterion of being balanced or not.

After generating random forests for the whole data sets, another number that classifies the data sets into two groups, small and large, was set 400. So the whole data sets were classified into four categories as in table 3. Note that for large data sets, being balanced or unbalanced is not important.

Table 3. The class value distribution and size of data sets

Data set	**No. of Instances/ class**	**Size**	**Class value distribution**

Abalone	144	Small	Unbalanced
Anneal	199.7		Unbalanced
Arrhythmia	28.3		Unbalanced
Audiology	9.4		Unbalanced
Bridges	13.5		Unbalanced
Bupa	172.5		Unbalanced
Credit screening (Japanese)	340		Balanced
Cylinder bands	270		Unbalanced
Db world e-mails (bodies)	32		Balanced
Dermatology	61		Unbalanced
Glass	30.6		Unbalanced
Hayes roth	44		Unbalanced
Hepatitis	77.5		Unbalanced
Iris	50		Balanced
Lenses	8		Unbalanced
Lung cancer	10.7		Balanced
Post operative	30		Unbalanced
Sonar	104		Balanced
Stalog_vehicle	188		Balanced
Spect	133.5		Unbalanced
Teaching assistant evaluation	50.3		Balanced
Vertebral column 2C	155		Unbalanced
Vowel	90		Balanced
Yeast	148.4		Unbalanced
Zoo	14.4		Unbalanced
Car	432	Large	Unbalanced
King rook vs king pawn	1598		Balanced
Internet ads	1639.5		Unbalanced
Magic4	9510		Unbalanced
Nursery original	2592		Unbalanced
Ozone_8 hour	1267		Unbalanced
Statlog_landsat satellite	1072.5		Unbalanced
Statlog_shuttle	8285.7		Unbalanced

D. The Results of Experiments

According to the definition, the results were categorized into the followings:

- The random forests that have better accuracy when drT is one, while the size of data set per class is relatively small and class distribution is relatively unbalanced (table 4-1 and table 4-2).
- The random forests that have better accuracy when drT is the default value, while the size of data set per class is relatively small and class distribution is relatively unbalanced (table 5-1 and table 5-2).
- The random forests that have better accuracy when drT is the default value, while the size of data set per class is relatively small and class distribution is relatively balanced (table 6-1 and table 6-2).
- The random forests that have better accuracy when drT is the default value, while the size of data set per class is relative large regardless of class distribution (table 7-1 and table 7-2).

The following table 4-1 and table 4-2 show the accuracy of random forests that have better accuracy when drT is one, while the size of data set per class is relatively small and class distribution is relatively unbalanced.

Table 4-1. The random forests that have better accuracy when drT is one, while the size of data set per class is relatively small and class distribution is relatively unbalanced

Data set	No. of instances per class	Number of attributes to pick randomly = 1	
		100 trees	1000 trees
Abalone	144	24.2897	24.4434
Anneal	199.7	95.7684	95.7684
Audiology	9.4	81.4159	78.7611
Bridges	13.5	65.0943	66.9811
Bupa	172.5	74.7826	76.2319
Cylinder bands	270	84.4444	84.8148
Dermatology	61	96.4881	97.2678
Hepatitis	77.5	84.5161	84.5161
Lenses	8	12.5	12.5
Post operative	30	65.5556	65.5556
Spect	133.5	82.397	82.397
Yeast	148.4	61.9272	62.6685
Zoo	14.4	96.0396	97.0927

Table 4-2. The random forests that have better accuracy when drT is one, while the size of data set per class is relatively small and class distribution is relatively unbalanced

Data set	No. of instances per class	Number of attributes to pick randomly = default value	
		100 trees	1000 trees
Abalone	144	23.9885	23.7252
Anneal	199.7	95.1002	95.1002
Audiology	9.4	78.7611	77.4336
Bridges	13.5	63.2073	63.2075
Bupa	172.5	71.0145	72.1739
Cylinder bands	270	82.5926	82.963
Dermatology	61	95.6284	96.1749
Hepatitis	77.5	81.9355	83.2258
Lenses	8	8.3333	12.5
Post operative	30	62.2222	62.2222
Spect	133.5	81.2734	81.6479
Yeast	148.4	61.5903	61.3881
Zoo	14.4	94.0494	94.0594

On the contrary to table 4-1 and table 4-2, even though the size of data set per class is relatively small and class distribution

is relatively unbalanced, some data sets have better accuracy when drT is default value as in table 5-1 and table 5-2. So, table 5-1 and table 5-2 show some exceptional cases compared to table 4-1 and table 4-2. Comparing table 4-1 and table 4-2, and table 5-1 and table 5-2, the ratio is 13:4, or 76%:24% in percentages.

Table 5-1. The random forests that have better accuracy when drT is default value, while the size of data set per class is relatively small and class distribution is relatively unbalanced

Data set	No. of instances per class	Number of attributes to pick randomly = 1	
		100 trees	1000 trees
Arrhythnria	28.3	62.1681	60.3982
glass	30.6	92.9907	93.4579
Hayes roth	44	79.5455	82.5758
Vertebral column 2C	155	82.5806	82.2581

Table 5-2. The random forests that have better accuracy when drT is default value, while the size of data set per class is relatively small and class distribution is relatively unbalanced

Data set	No. of instances per class	Number of attributes to pick randomly = default value	
		100 trees	1000 trees
Arrhythnria	28.3	68.5841	69.0265
glass	30.6	98.5981	99.0654
Hayes roth	44	82.5758	83.3333
Vertebral column 2C	155	83.2258	83.2258

Even though the size of data set per class is relatively small, if the class distribution is relatively balanced, then the default value in drT generates better random forests. Table 6-1 and table 6-2 show the result of the experiment, and no exception was found like the cases of table 4-1, table 4-2, table 5-1, and table 5-2. So, we can say that we may resort to the default parameter value on the number of attributes to pick randomly, if the size of data set per class is relatively small and class distribution is relatively balanced.

Table 6-1. The random forests that have better accuracy when drT is default, while the size of data set per class is relatively small and class distribution is relatively balanced

Data set	No. of instances per class	Number of attributes to pick randomly = 1	
		100 trees	1000 trees
Credit screening	340	85.942	86.5217
Db world bodies	32	73.4375	78.125
Iris	50	94.6667	94.0
Lung cancer	10.7	43.75	37.5
Teaching assistant evaluation	50.3	62.9139	62.9139
Statlog vehicle	188	73.9362	74.734
Sonar	104	85.0962	86.0577
vowel	90	97.3737	98.2828

Table 6-2. The random forests that have better accuracy when drT is default, while the size of data set per class is relatively small and class distribution is relatively balanced

Data set	No. of instances per class	Number of attributes to pick randomly = default value	
		100 trees	1000 trees
Credit screening	340	86.8116	86.087
Db world bodies	32	85.9375	84.375
Iris	50	95.3333	94.6667
Lung cancer	10.7	56.25	46.875
Teaching assistant evaluation	50.3	64.9007	65.5629
Statlog vehicle	188	75.266	73.9362
Sonar	104	85.5769	86.5385
vowel	90	96.9657	97.6768

The following table 7-1 and table 7-2 show the accuracy of random forests that have better accuracy when the number of instances per class is large, while drT is the default value regardless of class distribution. So, we can say that we may resort to the default parameter value on drT, if the size of data set per class is relatively large.

Table 7-1. The random forests that have better accuracy when drT is default value, while the size of data set per class is relative large regardless of class distribution

Data set	No. of instances per class	Class distribution	Number of attributes to pick randomly = 1	
			100 trees	1000 trees
Car	432	Unbalanced	93.3449	93.8657
King rook vs king pawn	1598	Balanced	98.4043	98.592
Magic4	9510	Unbalanced	88.0074	88.1073
Nursery original	2592	Unbalanced	97.6775	97.8704
Ozone 8 hours	1267	Unbalanced	93.6859	93.6859
Internet ads	1639.5	Unbalanced	97.2858	97.3163
Statlog_landsat satellite	1072.5	Unbalanced	91.2821	91.453

Statlog shuttle	8285.7	Unbalanced	99.9655	99.969

Table 7-2. The random forests that have better accuracy when drT is default value, while the size of data set per class is relative large regardless of class distribution

Data set	No. of instances per class	Class distribution	Number of attributes to pick randomly = default value	
			100 trees	1000 trees
Car	432	Unbalanced	94.6181	94.9653
King rook vs king pawn	1598	Balanced	99.3429	99.3429
Magic4	9510	Unbalanced	88.1388	88.1651
Nursery original	2592	Unbalanced	99.159	99.3364
Ozone 8 hours	1267	Unbalanced	93.8437	93.7859
Internet ads	1639.5	Unbalanced	97.5602	97.6517
Statlog_landsat satellite	1072.5	Unbalanced	92.1368	92.1368
Statlog shuttle	8285.7	Unbalanced	99.9914	99.9931

As a conclusion of the experiments, we may want to recommend the number, one, as the value of drT, if the number of instances is relatively small and class distribution is unbalanced.

In order to confirm the above assertion further, another experiment for the same data sets in table 4-1 and table 4-2 was performed based on the parameters of the square root of A, the half and default value as the value of drT. The result is summarized in table 8-1, table 8-2, and table 8-3 for the parameters of the square root of A, the half and default value respectively.

Table 8-1. More random forests for the data sets in table 4-1 and table 4-2 when drT is the square root of A

Data set	Number of attributes to pick randomly: the square root of A	
	100 trees	1000 trees
Abalone	23.9885	23.7252
Anneal	95.1002	95.1002
Audiology	78.7611	78.7611
Bridges	63.2073	63.2075
bupa	71.0145	72.1739
Cylinder bands	82.5926	82.963
Dermatology	95.6284	96.1749
Hepatitis	82.5806	83.871
Lenses	8.3333	8.3333
Post operative	62.2222	62.2222
Spect	81.2734	81.6479
yeast	61.9272	61.9946
zoo	95.0495	95.0495

Table 8-2. More random forests for the data sets in table 4 when drT is the half of default value

Data set	Number of attributes to pick randomly: the half of the default value	
	100 trees	1000 trees
Abalone	23.9646	23.9885
Anneal	95.7684	95.657
Audiology	76.9912	78.7611
Bridges	64.1509	66.0377
bupa	71.8841	73.0435
Cylinder bands	84.0741	84.2593
Dermatology	96.4481	96.9945
Hepatitis	83.871	84.5161
Lenses	8.3333	8.3333
Post operative	63.3333	63.3333
Spect	82.0225	81.2734
yeast	62.4663	**63.0054**
zoo	93.0396	96.0396

Table 8-3. More random forests for the data sets in table 4 when drT is the double of default value

Data set	Number of attributes to pick randomly: the double of the default value	
	100 trees	1000 trees
Abalone	22.8872	23.5815
Anneal	94.9889	95.1002
Audiology	72.5364	73.4513
Bridges	53.7736	53.7736
bupa	68.6957	69.8551
Cylinder bands	77.963	78.8889
Dermatology	95.9016	96.1749
Hepatitis	80.6432	81.2903
Lenses	4.1667	4.1667
Post operative	60.0	61.1111
Spect	80.1498	79.4009
yeast	60.5795	60.8491
zoo	66.3366	62.3762

As we compare the result of table 4-1, table 4-2, and table 8-1, table 8-2, table 8-3, we can find only one exceptional case of which the accuracy of 63.0054 (indicated with bold numbers in table 8-2), which is the result of data set, yeast. All other cases show that drT = 1 is better.

All in all, we can assure of our conjecture that we should try the alternative parameter of one as the number of attributes to pick randomly for better random forests to maximize randomness. In other words, if the size of data set per class is relatively small and class distribution is relatively unbalanced, we had better try the option of one as number of attributes to pick randomly to find better random forests.

IV. CONCLUSIONS

Decision trees have been being favored by many data mining experts. But, because decision tree algorithms split the training data set decisively to build a tree, the reliability of each branch of the decision trees diminishes unless the terminal nodes of the decision tree have some enough number of training instances. As a result, the decision tree algorithms drain training instances rapidly, so called data fragmentation problem. As a way to mitigate the data fragmentation problem, we may resort to the ensemble of many somewhat randomly generated decision trees. Because the data set is fixed, we cannot use more data for training to generate better trees. But, if we use samples that can be created with random sampling with replacement called bootstrap method, we can have many slightly different training data sets. Random forests use the training data sets, and the trees in the random forests are generated based on some random selection of attributes for each root of in their subtrees.

Because random forests use many trees in which some trees could have higher accuracy than average, random forests are known to have higher accuracy than decision trees, and known to have accuracy that is comparable to SVM. Moreover, they can take advantage of the fast building property of decision trees, so they do not require much computing time unless the size of data set is very large, even the forests have many trees. Note that the training time of SVM is relatively longer than that of random forests.

But, random forests have different accuracy depending on the degree of adaptation of randomness in splitting branches of decision trees as well as the number of trees in the forests. So, in this paper, some extensive experiments for empirical study to select the parameter of selecting the number of attributes for each decision tree in the forests were investigated based on the property of available training data sets.

As a conclusion of the empirical study, we can say, if the size of data set per class is relatively small like being less than 400 and class distribution is relatively unbalanced like being $\log_{10}G_b - \log_{10}G_d > 0.1$, the parameter value of one that is smaller than the default value can be recommended to generate better random forests.

REFERENCES

[1] Y. Hui, Z. Longqun, and L. Xianwen, "Classification of Wetland from TM imageries based on Decision Tree", *WSEAS Transactions on Information Science and Applications*, issue 7, vol. 6, July 2009, pp. 1155-1164.

[2] C. Huang, Y. Lin, and C. Lin, "Implementation of classifiers for choosing insurance policy using decision trees: a case study," *WSEAS Transactions on Computers*, issue 10, vol. 7, October 2008, pp. 1679-1689.

[3] Y.C. Lin, "Design and Implementation of an Ontology-Based Psychiatric Disorder Detection System," *WSEAS Transactions on Information Sciences and Applications*, issue 1, vol. 7, January 2010, pp. 56-69.

[4] A. Khasman, K.I. Dimilier, "Image Compression using Neural Networks and Haar Wavelet," *WSEAS Transactions on Signal Processing,* issue 5, vol. 4, May 2008, pp. 330-339.

[5] L. Khin, W. Too, Y. Min, A. Arooj, E. Supriyanto, "Nuchal Translucency Marker Detection Based on Artificial Neural Network and Measurement via Bidirectional Iteration Forward Propagation," *WSEAS Transactions on Information Science and Applications*, issue 8, vol. 7, August 2010, pp. 1025-1036.

[6] X. Li, "A scalable decision tree system and its application in pattern recognition and intrusion detection," *Decision Support Systems*, vol. 41, issue 1, November 2005, pp. 112-130.

[7] J. Shafer, R. Agrawal, M. Mehta, "SPRINT: A Scalable Parallel Classifier for Data Mining," in *Proceedings of the 22nd VLDB Conference*, Mumbai(Bombay), India, 1996, pp. 544-555.

[8] Y. Shang, B.W. Wah, "Global Optimization for Neural Network Training," IEEE Computer, vol. 29, no.3, 1996, pp.45-54.

[9] G. P. Zhang, "Neural Networks for Classification: A Survey," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 30, no. 4, November 2000, pp. 451-462.

[10] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, 2001, pp. 5-32.

[11] R. Diaz-Uriarte, S.A.D. Andrés, "Gene selection and classification of microarray data using random forest," *BMC Bioinformatics*, vol. 7, 2006.

[12] L.O. Hall, X. Liu, "Why are Neural Networks Sometimes Much More Accurate Than Decision Trees: An Analysis on a Bio-Informatics Problem," in *Proceedings of IEEE International Conference on Systems, Man & Cybernetics.* 2003, pp. 2851-2856.

[13] L. Breiman, "Bagging Predictors," *Machine Learning*, vol. 24, 1996, pp. 123-140.

[14] H. Boström, "Concurrent Learning of Large-Scale Random Forests," in *Proceedings of Scandinavian Conference on Artificial Intelligence,* 2011, pp. 20-29.

[15] J.R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc., 1993

[16] L. Breiman, J. Friedman, R. Olshen, C. Stone, *Classification and Regression Trees*. Wadsworth International Group, Inc., 1984.

[17] J. Han, M. Kamber, J. Pei, *Data Mining: concepts and techniques*. Morgan Kaufmann Publishers, Inc., 2011.

[18] T.K. Ho, "The Random Subspace Method for Constructing Decision Forests," *IEEE Transactions on Pattern Analysis and machine Intelligence,* vol. 20, 1998, pp. 832-844.

[19] P. Panov, S. Džeroski, "Combining Bagging and Random Subspaces to create Better Ensembles," *Lecture notes in Computer Science*, vol. 4723 2007, pp. 118-129.

[20] W.W. Cohen, "Fast Effective Rule Induction," in *Proceedings of the 12th International Conference on Machine Learning*. 1995, pp. 115-123.

[21] B. Efron, R. Tibshirani, "Improvements on Cross-Validation: The .632 Bootstrap Method," *Journal of the American Statistical Association*, vol. 92, 1997, pp. 548-560.

[22] C. Strobl, J. Malley, G. Tutz, "An Introduction to Recursive Partitioning: Rational, Application, and Characteristics of Classification and Regression Trees, Bagging, and Random Forests," *Psychological Methods*, vol. 14, 2009, pp. 323-348.

[23] L. Breiman, A. Cutler, Random Forests. Available: http://www.stat.berkeley.edu/users/breiman/RandomForests/

[24] A. Frank, A. Asuncion, UCI *Machine Learning Repository* [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2010.

[25] I.H. Witten, E. Frank, E., M.A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. Morgan Kaufman, 2011.

Hyontai Sug received the B.S. degree in Computer Science and Statistics from Busan National University, Busan, Korea, in 1983, the M.S. degree in Computer Science from Hankuk University of Foreign Studies, Seoul, Korea, in 1986, and the Ph.D. degree in Computer and Information Science & Engineering from University of Florida, Gainesville, FL, USA in 1998. Currently he is a professor of the Division of Computer and Information

Engineering of Dongseo University, Busan, Korea since 2001. Before he join to Dongseo University, he was a full time lecturer of Pusan University of Foreign Studies, Busan, Korea from 1999 to 2001. He was also a formal researcher of Agency for Defense Development, Korea from 1986 to 1992. His areas of research interests include data mining as well as database applications.