Random Forests and Decision Trees

Jehad Ali[1], Rehanullah Khan[2], Nasir Ahmad[3], Imran Maqsood[4]

[1] **Computer Systems Engineering, UET Peshawar, Pakistan**

[2] **Sarhad University of Science and Information Technology, Peshawar, Pakistan**

[3] **Computer Systems Engineering, UET Peshawar, Pakistan**

[4] **Computer Software Engineering, UET Mardan, Pakistan**

Abstract

In this paper, we have compared the classification results of two models i.e. Random Forest and the J48 for classifying twenty versatile datasets. We took 20 data sets available from UCI repository [1] containing instances varying from 148 to 20000. We compared the classification results obtained from methods i.e. Random Forest and Decision Tree (J48). The classification parameters consist of correctly classified instances, incorrectly classified instances, F-Measure, Precision, Accuracy and Recall. We discussed the pros and cons of using these models for large and small data sets. The classification results show that Random Forest gives better results for the same number of attributes and large data sets i.e. with greater number of instances, while J48 is handy with small data sets (less number of instances). The results from breast cancer data set depicts that when the number of instances increased from 286 to 699, the percentage of correctly classified instances increased from 69.23% to 96.13% for Random Forest i.e. for dataset with same number of attributes but having more instances, the Random Forest accuracy increased.

Keywords: *Random Forests, Decision Trees, J48.*

1. Introduction

The application of the Decision Tree algorithm [2] can be observed in various fields. Text classification and text extraction, comparing data statistically etc. are the fields where they are used. Besides this in libraries books can be classified into different categories on the basis of its type with the implementation of Decision Tree algorithm. In hospitals it can be used for diagnosis of diseases i.e. brain tumor, Cancer, heart problems, Hepatitis etc. Companies, hospitals, Schools, colleges and universities use it for maintaining their records. Similarly, In Stock market, it can be used for statistics.

Decision Tree algorithms are effective [3] in that they provide human-readable rules of classification. Beside this it has some drawbacks, one of which is the sorting of all numerical attributes when the tree decides to split a node. Such split on sorting all numerical attributes becomes costly i.e. efficiency or running time and memory size, especially if Decision Trees are set on data the size of which is large i.e. it has more number of instances.
In 2001, Breiman [4] presented the idea of Random Forests which perform well as compared with other classifiers including Support Vector Machines, Neural Networks and Discriminant Analysis, and overcomes the over fitting problem.

Those methods such as Bagging or Random subspaces [5,6] which are made from ensemble of various classifiers and those which use randomization for producing diversity have proven to be very efficient. In order to introduce diversity and to build classifiers different from each other, they use randomization in the induction process. Random Forests have gained a substantial interest in machine learning because of its efficient discriminative classification [7, 8].

In computer vision community, Random Forests were introduced by Lepetit et. al. [9, 10]. His work in this field provided a foundation for papers such as class recognition [11, 12], bi-layer video segmentation [13], image classification [14] and person identification [15], which use Random Forests. A wide range of visual cues are also enabled naturally by the Random Forest including color, shape, texture and depth. Random Forests are considered general purpose vision tools and considered as efficient.

Random Forest as defined in [4] is a generic principle of classifier combination that uses L tree-structured base classifiers $\{h(X,\Theta_n), N=1,2,3,\ldots L\}$, where X denotes the input data and $\{\Theta_n\}$ is a family of identical and dependent distributed random vectors. Every Decision Tree is made by randomly selecting the data from the available data. For example a Random Forest for each Decision Tree (as in Random Subspaces) can be built by randomly sampling a feature subset, and/or by the random sampling of a training data subset for each Decision Tree (the concept of Bagging).
In a Random Forest, the features are randomly selected in each decision split. The correlation between trees is reduces by randomly selecting the features which improves

the prediction power and results in higher efficiency. As such the advantages of Random Forest are [16]:

- Overcoming the problem of over fitting
- In training data, they are less sensitive to outlier data
- Parameters can be set easily and therefore, eliminates the need for pruning the trees
- variable importance and accuracy is generated automatically

Random Forest not only keeps the benefits achieved by the Decision Trees but through the use of bagging on samples, its voting scheme [17] through which decision is made and a random subsets of variables, it most of the time achieves better results than Decision Trees.

The Random Forest is appropriate for high dimensional data modeling because it can handle missing values and can handle continuous, categorical and binary data. The bootstrapping and ensemble scheme makes Random Forest strong enough to overcome the problems of over fitting and hence there is no need to prune the trees. Besides high prediction accuracy, Random Forest is efficient, interpretable and non-parametric for various types of datasets [18]. The model interpretability and prediction accuracy provided by Random Forest is very unique among popular machine learning methods. Accurate predictions and better generalizations are achieved due to utilization of ensemble strategies and random sampling.

Bagging scheme provides generalization property which improves with the decrease of variance and improves the over-all generalization error. As such, the decrease in bias [19] is achieved by the boosting method. Random Forest three main features that gained focus [17] are:

- Accurate predictions results for a variety of applications
- Through model training, the importance of each feature can be measured
- Trained model can measure the pair-wise proximity between the samples

In this article, we concentrate on the classification performance of the Decision Tree (J48) and the Random Forest for large and small datasets. The objective of this comparison is creating a base-line, which will be useful for the classification scenarios. It will also help in the selection of appropriate model.

The rest of the paper is organized as follows: Section 2 describes Decision Tree related classification algorithms including the Random Forest. Experimental setup and the datasets used are described in Section 3. Section 4 presents results and conclusion.

2. Classification Methods

2.1 Decision Trees

Decision Trees embody a supervised classification approach [20]. The idea came from the ordinary tree structure which is made-up of a root and nodes (the positions where places branches divides), branches and leaves. In a similar manner, a Decision Tree is constructed from nodes which represent circles and the branches are represented by the segments that connect the nodes. A Decision Tree starts from the root, moves downward and generally are drawn from left to right. The node from where the tree starts is called a root node. The node where the chain ends is known as the "leaf" node. Two or more branches can be extended from each internal node i.e. a node that is not leaf node. A node represents a certain characteristic while the branches represent a range of values. These ranges of values act as a partition points for the set of values of the given characteristic. Figure 1 describes the structure of a tree.

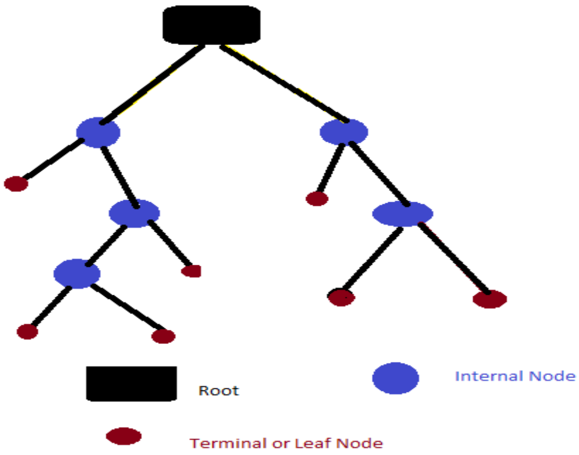


Figure 1: Tree Structure

The grouping of data in the Decision Tree is based on the values of attributes of the given data. A Decision Tree is made from the pre-classified data. The division into classes is decided upon the features that best divides the data. The data items are split according to the values of these features. This process is applied to each split subset of the data items recursively. The process terminates as for as all the data items in current subset belong to the same class.

We use the J48 implementation of the Decision Trees of WEKA (open source software). In WEKA, we can analyze data and besides this, it also includes implementation for regression, data pre-processing, clustering, classification and visualization through various algorithms. More than sixty algorithms are available in WEKA. Following is an overview of few of the Decision Tree based algorithms.

2.1.1 REPTree

In REPTree, decision/regression tree is constructed with information gain as the splitting criterion and reduced error pruning is used to prune it. It sorts values only for numeric attributes once. The method of fractional instances is used to handle missing values with C4.5. REP Tree is a fast Decision Tree learner.

2.1.2 Random Tree

A random tree is a tree constructed randomly from a set of possible trees having *K* random features at each node. “At random” in this context means that in the set of trees each tree has an equal chance of being sampled. Or we can say that trees have a “uniform” distribution. Random trees can be generated efficiently and the combination of large sets of random trees generally leads to accurate models. There has been an extensive research in the recent years over Random trees in the field of machine Learning.

2.1.3 J.48

Ross Quinlan [21] developed C4.5 algorithm which is used to generate a Decision Tree. Decision Trees are produced from the J48 i.e. Open Source Java implementation of C4.5 release in WEKA data mining tool [22]. This is a standard Decision Tree algorithm. One of the classification algorithms in data mining is Decision Tree Induction. The Classification algorithm [23] is inductively learned to construct a model from the pre-classified data set. Each data item is defined by values of the characteristics or features. Classification may be viewed as mapping from a set of features to a particular class.

2.2 Random Forests

Random Forest developed by Leo Breiman [4] is a group of un-pruned classification or regression trees made from the random selection of samples of the training data. Random features are selected in the induction process. Prediction is made by aggregating (majority vote for classification or averaging for regression) the predictions of the ensemble. Each tree is grown as described in [24]:

- By Sampling *N* randomly, If the number of cases in the training set is *N* but with replacement, from the original data. This sample will be used as the training set for growing the tree.
- For *M* number of input variables, the variable *m* is selected such that *m*<<*M* is specified at each node, *m* variables are selected at random out of the *M* and the best split on these *m* is used for splitting the node. During the forest growing, the value of *m* is held constant.
- Each tree is grown to the largest possible extent. No pruning is used.

Random Forest generally exhibits a significant performance improvement as compared to single tree classifier such as C4.5. The generalization error rate that it yields compares favorably to Adaboost, however it is more robust to noise.

3. Experimental Analysis

In this section, we concentrate on the classification performance of the Decision Tree (J48) and the Random Forest for large and small datasets. The objective of this comparison is creating a base-line, which will be useful for the classification scenarios. It will also help in the selection of appropriate model.

3.1 Data Sets

For classification problems, we took these datasets from the UCI Machine Learning repository [1]. In breast cancer data, some attributes are linear and few are nominal. The detailed description, attributes, source of each dataset can be found from UCI repository. Table 1 shows the names of the dataset, the number of instances and number of attributes for the twenty datasets we used for our analysis and comparison. As an visual information, the Figures 2, 3, 4 shows the distribution of data variables in the corresponding three sampled data sets. Figure 2 shows the Dataset Lymphography. Its total number of instances are 148, the total attributes are 19 and having four classes. Figure 3 shows the Dataset Sonar with 208 instances and 61 attributes and bi-classes data. Figure 4 shows the Dataset Heart-h. It has 14 attributes and 294 instances and binary class data.

Table 1: Datasets used and their details

Serial Number	Dataset name	number of instances	number of attributes
1	Lymph	148	19
2	Autos	205	26
3	Sonar	208	61
4	Heart-h	270	14
5	Breast cancer	286	10
6	Heart-c	303	14
7	Ionosphere	351	35
8	colic	368	23
9	Colic.org	368	28
10	Primary tumor	399	18
11	Balance Scale	625	25
12	Soyben	683	36
13	Credit a	690	16
14	Breast W	699	10
15	Vehicle	846	19
16	vowel	990	14
17	Credit g	1000	21
18	Segment	2310	20
19	Waveform	5000	41
20	Letter	20,000	17

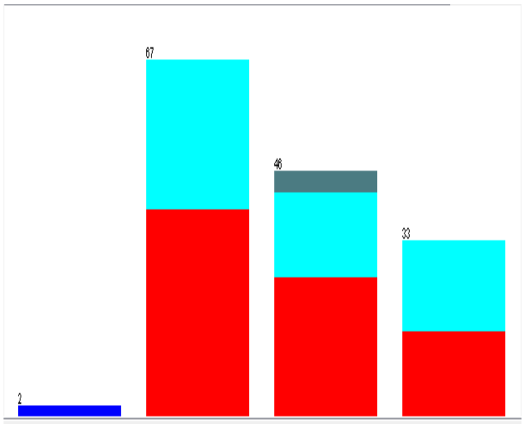


Figure 2: Dataset Lymphography

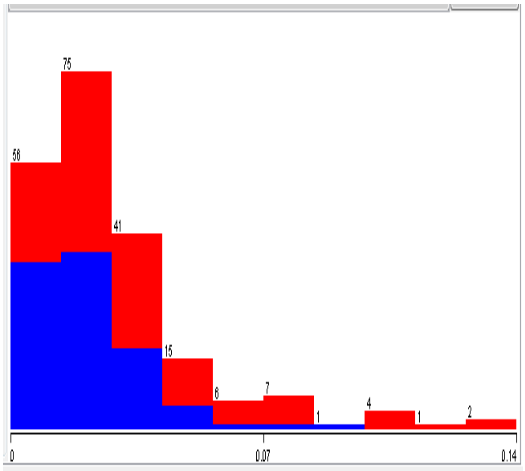


Figure 3: Dataset Sonar

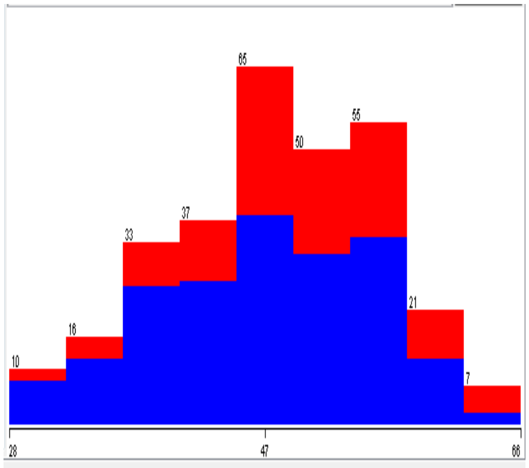
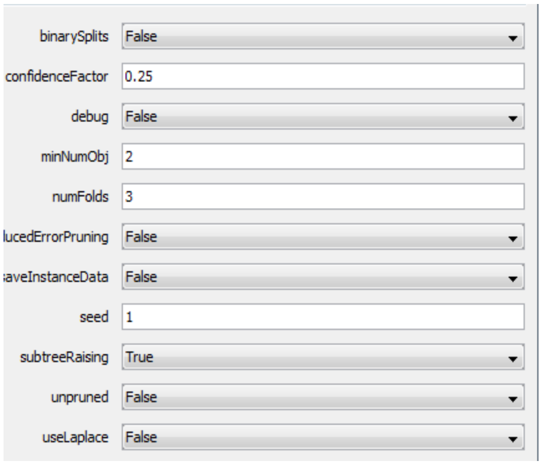


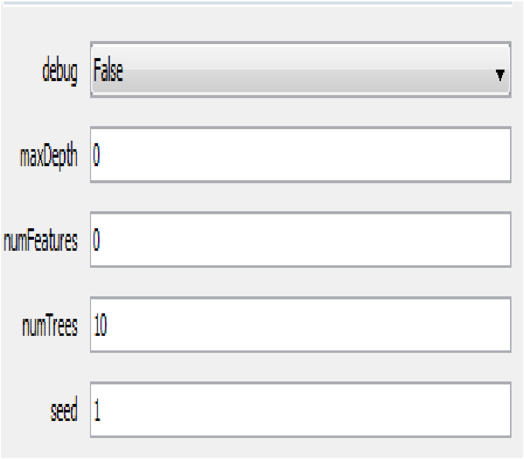
Figure 4: Dataset Heart-h

Figure 5 and Figure 6 shows the different parameter settings and the variables used for the J48 and the Random Forest. **Binary splits**: show the use of binary splits when building the trees. **Confidence factor**: shows the pruning of the trees smaller values show more pruning. **Debug**: if this is set to true additional information are displayed on the console. **Seed**: used for randomizing the data when reduced error pruning is used. **Unprunned:** shows whether pruning is used or not. **MinNumObj**: Shows the minimum number of instances per leaf. **Save Instance Data**: whether to save data for visualization. **numFolds**: shows the amount of data used for pruning. **Reduced error pruning**: whether reduced error pruning is used or not instead of C.4.5. **Sub-tree Raising**: Used for Sub-tree rising when we pruning is used. **Use Laplace**: whether counts at leafs are smoothed based on Laplace. **MaxDepth**: shows the maximum depth of the trees, 0 is used for unlimited. **numFeatures**: The number of attributes used while random selection. **numTrees**: the number of trees to be generated. **Seed**: The random number that will be used as seed value.



binarySplits False
confidenceFactor 0.25
debug False
minNumObj 2
numFolds 3
lucedErrorPruning False
aveInstanceData False
seed 1
subtreeRaising True
unpruned False
useLaplace False

Figure 5: Parameters settings for the J48



debug False
maxDepth 0
numFeatures 0
numTrees 10
seed 1

Figure 6: Parameters settings for the Random Forest

4. Results and Discussion

We compared the classification results of the J48 and the Random Forest. To avoid over fitting problem, we obtained the accuracy using 10-fold cross validation which uses 9/10 of data as for training the algorithm and the remaining for testing purpose and repeats the process 10 times.

Table 2: Comparison of the Random Forest and the J48 classification results for the 20 datasets.

Serial NO	Data Set	No. of instances	No. of attributes	Random Forest		J-48 Results	
				Correctly classified instances	Incorrectly classified instances	Correctly classified instances	Incorrectly classified instances
1	Lymph	148	19	81.08%	18.91%	77.02%	22.97%
2	Autos	205	26	83.41%	16.58%	80.95%	18.04%
3	Sonar	208	61	80.77%	19.23%	71.15%	28.84%
4	Heart-h	270	14	77.89%	22.10%	80.95%	19.04%
5	Breast cancer	286	10	69.23%	30.76%	75.52%	24.47%
6	Heart-c	303	14	81.51%	18.48%	77.56%	22.44%
7	Ionosphere	351	35	92.88%	7.12%	91.45%	8.54%
8	colic	368	23	86.14%	13.85%	85.32%	14.67%
9	Colic.org	368	28	68.47%	31.52%	66.30%	33.69%
10	Primary tumor	399	18	42.48%	57.52%	39.82%	60.17%
11	Balance Scale	625	25	80.48%	19.52%	76.64%	23.36%
12	Soyben	683	36	91.65%	8.34%	91.50%	8.49%
13	Credit a	690	16	85.07%	14.92%	86.09%	13.91%
14	Breast W	699	10	96.13%	3.68%	94.56%	5.43%
15	Vehicle	846	19	77.06%	22.93%	72.45%	27.54%
16	vowel	990	14	96.06%	3.03%	81.51%	18.48%
17	Credit g	1000	21	72.50%	27.50%	70.50%	29.50%
18	**Segment**	**2310**	**20**	**97.66%**	**2.33%**	**96.92%**	**3.07%**
19	Waveform	5000	41	81.94%	18.06%	75.30%	24.70%
20	Letter	20,000	17	94.71%	5.29%	87.98%	12.02%

Table 2 shows the correctly classified instances and incorrectly classified instances for the Random Forest and J48 classifiers, the name of the corresponding dataset, number of instances and number of attributes are shown in columns 2, 3 and 4 respectively.

The classification results show that the Random Forest gives better results for the same number of attributes and large datasets i.e. with greater number of instances while J48 is handy with small datasets i.e. less number of instances.

The results from breast cancer data set depicts that when the number of instances increased from 286 to 699, the correctly classified instances increased from 69.23% to 96.13% for the Random Forest.

Table 3: Comparison of Random Forest and the J48 in terms of Precision, Recall and F-measure

Dataset	No. of Instances	No. of attributes	Random forest			J-48		
			Precision	Recall	F-measure	Precision	Recall	F-measure
Breast cancer	286	10	0.667	0.692	0.674	0.752	0.755	0.713
Breast W	699	10	0.962	0.961	0.961	0.946	0.946	0.946
Credit a	690	16	0.851	0.851	0.851	0.861	0.861	0.861
Credit g	1000	21	0.705	0.725	0.707	0.687	0.705	0.692
colic	368	23	0.854	0.853	0.85	0.86	0.861	0.861
Colic.org	368	28	0.662	0.685	0.63	0.44	0.663	0.529
Heart-h	270	14	0.775	0.779	0.774	0.807	0.81	0.806
Heart-c	303	14	0.819	0.815	0.813	0.776	0.776	0.774
vowel	990	14	0.961	0.961	0.961	0.816	0.815	0.815
Ionosphere	351	35	0.929	0.929	0.929	0.915	0.915	0.913
Soyben	683	36	0.926	0.917	0.918	0.917	0.915	0.913
Vehicle	846	19	0.764	0.771	0.767	0.722	0.725	0.722
Sonar	208	61	0.813	0.808	0.808	0.713	0.712	0.712
Autos	205	26	0.836	0.834	0.834	0.833	0.822	0.82
Balance Scale	625	25	0.817	0.805	0.81	0.732	0.766	0.749
Lymph	148	19	0.804	0.811	0.8	0.776	0.77	0.772
Segment	**2310**	**20**	**0.977**	**0.977**	**0.977**	**0.969**	**0.969**	**0.969**
Primary tumor	399	18	0.394	0.425	0.406	0.333	0.398	0.704
Waveform	5000	41	0.82	0.819	0.82	0.753	0.753	0.753
Letter	20,000	17	0.948	0.947	0.947	0.881	0.88	0.88

Table 3 shows the Precision, Recall and the F-measure for the Random Forest and J48 for the 20 datasets. From the table, it can be seen that for the same data set with greater number of instances, i.e. when the number of instances increased from 286 to 699 while keeping the attributes constant the precision increases from 0.667 to 0.962, F-measure from 0.674 to 0.961 and Recall from 0.692 to 0.961 for the Random Forest classifier. Similarly for the J48, the precision increased from 0.752 to 0.946, F-measure from 0.713 to 0.946 and recall from 0.755 to 0.946. The highest precision value obtained is for the Random Forest i.e. 0.977 for the Segment Dataset which shows the highly accurate model of the Random Forest ensemble method.

From the results, it can be concluded that the Random Forest achieves increased classification performance and yields results that are accurate and precise in the cases of large number of instances. These scenarios also cover the missing values problem in the datasets and thus besides accuracy, it also overcomes the over-fitting problem generated due to missing values in the datasets. Therefore, for the classification problems, if one has to choose a classifier among the tree based classifiers set, we recommend to use the Random Forest with confidence for variety of classification problems.

References

[1] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://archive.ics.uci.edu/ml/

[2] T.M. Mitchell, Machine Learning. McGraw-Hill, 1997.

[3] Yael Ben-Haim, “A Streaming Parallel Decision Tree Algorithm” , Elad Tom-Tov , 2010

[4] Breiman, L., Random Forests, Machine Learning 45(1), 5-32, 2001.

[5] "Bagging predictors," Machine Learning, vol. 24, no. 2, pp. 123-140, 1996.

[6] T. Ho, "The random subspace method for constructing decision forests," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 832-844, 1998.

[7] Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. Neural Computation 9(7), 1545–1588 (1997)

[8] Breiman, L.: Random Forests. ML Journal 45(1), 5–32 (2001)

[9] Lepetit, V., Fua, P.: Keypoint recognition using randomized trees. IEEE Trans. Pattern Anal. Mach. Intell. 28(9), 1465–1479 (2006)

[10] Ozuysal, M., Fua, P., Lepetit, V.: Fast keypoint recognition in ten lines of code. In: IEEE CVPR (2007)

[11] Winn, J., Criminisi, A.: Object class recognition at a glance. In: IEEE CVPR, video track (2006)

[12] Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: IEEE CVPR, Anchorage (2008)

[13] Yin, P., Criminisi, A., Winn, J.M., Essa, I.A.: Tree-based classifiers for bilayer video segmentation. In: CVPR (2007)

[14] Bosh, A., Zisserman, A., Munoz, X.: Image classification using Random Forests and ferns. In: IEEE ICCV (2007)

[15] Apostolof, N., Zisserman, A.: Who are you? - real-time person identification. In: BMVC (2007).

[16] Introduction to Decision Trees and Random Forests, Ned Horning; American Museum of Natural History’s

[17] Breiman, L.: Random Forests. Machine. Learning. 45, 5–32 (2001). DOI 10.1023/A:1010933404324

[18] Yanjun Qi., “Random Forest for Bioinformatics”. www.cs.cmu.edu/~qyj/papersA08/11-rfbook.pdf

[19] Yang, P., Hwa Yang, Y., Zhou, B., Zomaya, Y., et al.: “A review of ensemble methods in bioinformatics”. Current Bioinformatics 5(4), 296–308 (2010)

[20] “Comparison of Decision Tree methods for finding active objects” Yongheng Zhao and Yanxia Zhang, National Astronomical Observatories, CAS, 20A Datun Road, Chaoyang District, Bejing 100012 China

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

[22] http://en.wikipedia.org/wiki/C4.5_algorithm

[23] Report from Pike research, http://www.pikeresearch.com/research/smartgrid- data-analytics

[24]http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#prox Symposium, volume 1, July, 2005.

Jehad Ali is pursuing his M.Sc Computer Systems Engineering from University of Engineering and Technology, Peshawar, Pakistan. He did his B.Sc. Computer Systems Engineering from the same university. He is working as a Computer Engineer in Ghulam Ishaq Khan Institute (GIKI) of Engineering Sciences and Technology, Topi, Pakistan. His research interest’s areas are image processing, computer vision, machine learning, Computer Networks and pattern recognition.

Rehanullah Khan graduated from the University of Engineering and Technology Peshawar, with a B.Sc degree (Computer Engineering) in 2004 and M.Sc (Information Systems) in 2006. He obtained PhD degree (Computer Engineering) in 2011 from Vienna University of Technology, Austria. He is currently an Associate Professor at the Sarhad University of Science and Technology, Peshawar. His research interests include color interpretation, segmentation and object recognition.

Nasir Ahmad graduated from University of Engineering and Technology Peshawar with a B.Sc Electrical Engineering degree. He obtained his PhD degree from UK in 2011. He is a faculty member of Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan. His Research Areas include Pattern Recognition, Computer vision and Digital Signal Processing.

Imran Maqsood graduated from the University of Engineering and Technology Peshawar, with a B.Sc degree (Computer Engineering) in 2004 and M.Sc in 2006. He is pursuing his PhD degree. He is currently an Assistant Professor at the Department of Computer Software Engineering, UET Mardan Campus, Peshawar, Pakistan.