Efficient Classification of Data Using Decision Tree

Bhaskar N. Patel, Satish G. Prajapati and Dr. Kamaljit I. Lakhtaria

Abstract--- *Data classification means categorization of data into different category according to rules. The aim of this proposed research is to extract a kind of "structure" from a sample of objects. To rephrase it better to learn a concise representation of these data. Present research performed over the classification algorithm learns from the training set and builds a model and that model is used to classify new objects. This paper discusses one of the most widely used supervised classification techniques is the decision tree. And perform own Decision Tree evaluate strength of own classification with Performance analysis and Results analysis.*

Keywords--- *Data Mining, Decision Tree, K-Means Algorithm*

I. Introduction

DATA MINING is the extraction of implicit, previously unknown and rotationally useful information from data. Also it is extraction of large database into useful data or information and that information is called knowledge. Data mining is always inserted in techniques for finding and describing structural patterns in data as a tool for helping that data and make prediction.

Data mining consists of five major elements. First, extract, transform, and load transaction data onto the data warehouse system. Second, store and manage the data in a multidimensional database system. Third, provide data access to business analysts and IT professionals. Fourth, analyze the data by application software. Fifth, present the data in a useful format, such as a graph or table. Many data mining techniques are closely related to some of machine learning. Others are related to techniques that have been developed in statistics, sometimes called exploratory data analysis.

We survey many techniques related to data mining and data classification techniques. We select clustering algorithm k-means to improve the training phase of Classification. Learning classification methods in data mining can be classified into three basic types: Supervised, unsupervised and reinforced.

1.1 Supervised Learning

It is error based learning. In this, every input pattern that is used to train the machine is associated with an output pattern, which is the target or the desired pattern [1]. A teacher is assumed to be present during the learning process, when a comparison is made between the computed output and the correct expected output to determine the error [2]. The error can then be used to change parameters, which result in an improvement in performance.

1.2 Unsupervised Learning

In this learning method, the target output is not presented to the machine. It is as if there is no teacher to present the desired patterns and hence, the system learns of its own by discovering and adapting to structural features in the input [3].

1.3 Reinforced Learning

It is output based learning. In this method, a teacher though available, does not present the expected answer but only indicates if the computed output is correct or incorrect [4]. The information provided helps the machine in its learning process.

II. Available Classification Techniques

There are so many techniques are available for data classification. For this research we had consider only four well known techniques:

a. *Neural Networks* include the most popular architecture: a single hidden layer preceptor with optional short cut connections. We restrict the so-defined model class (which also includes the linear model) to architectures with at least one hidden unit and no short cut connections to prevent a "fall-back" to LDA [5].

b. *Support Vector Machines (SVM)* are used for support vector classification with linear kernel and non-linear kernel functions.

c. *KNN (K- Means)* classifier uses the training data to assign new features to the class determined by the nearest data point. It uses the Euclidean distance measure (not suitable for categorical predictors) to it.

d. *Decision Tree* try to find an optimal partitioning of the space of possible observations, mainly by the means of subsequent recursive splits.

2.1 Summary of Classification Techniques

Among all above four techniques we summarized here with some most effective factors and based on that we have finally detail study of two techniques name KNN and Decision

Bhaskar N. Patel, Professor, Department of Computer & IT , B.S.Patel. Polytechnic, Kherva, India.

Satish G. Prajapati, Lecturer, Department of Computer, B.S.Patel Polytechnic, Kherva, India.

Dr. Kamaljit I. Lakhtaria, B.S.Patel Polytechnic, Kherva, India. E-mail: kamaljit.ilakhtaria@gmail.com

Tree.

Table 1: Comparison of Various Classification Techniques

Facto Affecting	Decision trees	Neural Network	KNN	SVM
Accuracy in general	**	***	**	****
Speed of learning with respect to number of attributes and the number of instances	***	*	****	*
Speed of classification	****	****	*	****
Tolerance to missing values	***	*	*	**
lerance to irrelevant Attributes	***	*	**	****
Dealing withdanger of overfitting	**	*	***	**
Explanation ability/ transparency of knowledge/ classifications	****	*	**	
Model parameter handling	***	*	***	

2.2 Algorithm Selection

Once preliminary testing is judged to be satisfactory, the classifier (mapping from unlabeled instances to classes) is available for routine use. The classifier's evaluation is most often based on prediction accuracy (the percentage of correct prediction divided by the total number of predictions).

There are at least three techniques which are used to calculate a classifier's accuracy. One technique is to split the training set by using two-thirds for training and the other third for estimating performance. In another technique, known as cross-validation, the training set is divided into mutually exclusive and equal-sized subsets and for each subset the classifier is trained on the union of all the other subsets. The average of the error rate of each subset is therefore an estimate of the error rate of the classifier. Leave-one-out validation is a special case of cross validation. All test subsets consist of a single instance. This type of validation is, of course, more expensive computationally, but useful when the most accurate estimate of a classifier's error rates required.

Training a standard decision tree leads to a quadratic optimization problem with bound constraints and one linear equality constraints. Training support vector machines involves a huge optimization problem and many specially designed algorithms have been proposed. We used an algorithm called "Decision Tree Induction" that accelerates the training process by exploiting the distributional properties of the training data, that is, the natural clustering of the training data and the overall layout of these clusters relative to the decision boundary of support vector machines.

2.2.1. Sub Process

A fast training algorithm called DTInduction whose idea is to speed up the training process by reducing the number of training data. This is accomplished by partitioning the training data into pair-wise disjoint clusters, each of which consists of either only support vectors or only non-support vectors, and replacing the cluster containing only non-support vectors by a representative. In order to identify the cluster that contains only non-support vectors, the training data is first partitioned into several pair-wise disjoint clusters and an initial support vector machine is trained using the representatives of these clusters [6].

Based on this initial decision tree, we can judge whether a cluster contains only nonsupport vectors or not. For the cluster that contains both support vectors and non-support vectors, based on the decision boundary of the initial decision tree, we can split it into two subclusters such that, approximately, one contains only non-support vectors and the other contains only support vectors. This process is then repeated if one of the subclusters contains both support vectors and non-support vectors. The training time of this strategy scales with the square of the number of support vectors and, as shown by experiments, an approximate solution can be found even faster.

III. KNN (K-Means Algorithm)

K-means is the simplest and most popular classical classification and clustering method that is easy to implement. The method called k-means since each of the K clusters is represented by the mean of the objects within it. It is also called the centroid method since at each step the centroid point of each cluster is assumed to be known and each of the remaining points are allocated to the cluster whose centroid is closest to it. Once this allocation is completed, the centroids of the clusters are recomputed using simple means and the process of allocating points to each cluster is repeated until there is no change in the clusters [7].

The K-means algorithm proceeds as follows. First, it randomly selects k of the objects, each of which initially represents a center. For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster. It then computes the new mean for each cluster. This process iterates until the criterion function converges.

The K-Means algorithm is well known for its efficiency in clustering large data sets. K-means clustering is a method of cluster analysis which aims to partition *n* samples of dataset into *k* clusters in which each sample belongs to the cluster with the nearest mean. Given a set of sample (x1, x2, …, x*n*), where each sample is a *d*-dimensional real vector, then *k*-means clustering aims to partition the *n* samples into *k* sets ($k < n$) S={*S*1, *S*2, …, *Sk*} so as to minimize the within-cluster sum of squares S:

$$avg\,\min\sum_{i-1}^{k}=\sum\|\,Xi-\bar{X}\,\|$$

The most common algorithm uses an iterative refinement technique. The basic step of k-means clustering is simple. K means algorithm will do the three steps below until convergence Iterate until *stable* (= no object move group):

1. Determine the centroid coordinate
2. Determine the distance of each object to the centroids
3. Group the object based on minimum distance

Given an initial set of *k* means m$1^{(1)}$,…,m$k^{(1)}$,
Which may be specified randomly or by some Heuristic, the algorithm proceeds as follows.

Input:

k: the number of clusters

D: a data set containing n objects

Output:

A set of clusters

The algorithm is guaranteed to have converged when the assignments no longer change. Although the K-means method is most widely known and used [8], there are a number of issues related to the method as given below.

- The K-means method needs to compute Euclidean distances and means of the attribute values of objects within a cluster. The classical algorithm is suitable for continuous data.
- The K-means method implicitly assumes spherical probability distributions.
- The results of the K-means method depend strongly on the initial guesses.
- The K-means method can be sensitive to outliers.
- The K-means method does not consider the size of the clusters. Some clusters may be large and some very small.
- The K-means method does not deal with overlapping clusters.

How can we make the k-means algorithm more scalable? A recent approach to scaling the k-means algorithm is based on the idea of identifying three kinds of regions in data. Regions that are compressible, regions that can be maintained in the main memory, and regions that are discard able. An object is discarding if its membership in a cluster is ascertained. An object is compressible if it is not discarding but belongs to a tight sub cluster. A data structure known as a clustering feature is used to summarize objects that have been discarded or compressed. If an object is neither discard able nor compressible, then it should be retained in main memory [9, 10].

To achieve scalability, the iterative clustering algorithm only includes the clustering features of the compressible objects and the objects that must be retained in main memory, thereby turning a secondary memory based algorithm into a main memory based algorithm [11]. An alternative approach to scaling the k-means algorithm explores the micro clustering idea, which first groups nearby objects into micro clusters and then performs k-means clustering on the micro clusters [12].

i. Advantages

- The K-means is an iterative improvement of greedy method.
- Ability to deal with noisy and missing data.
- Ability to deal with large problems.
- Ability to deal with a variety of attribute types and magnitudes.
- This method is easy to implement.

ii. Disadvantages

- This method does not explicitly assume any probability distributions for the attribute values.
- It is not time-efficient and does not scale well.
- It does not handle outliers properly.
- K-means method is not suitable for discovering clusters with nonconvex shapes.
- It is sensitive to noise and outlier data points.

IV. Decision Tree

It is flow-chart like tree structure, where each internal node denotes a test on an attribute, each branch denotes an outcome of test, and each leaf node holds a class label. The topmost node in a tree is the root node [14]. Given a tuple, *X*, for which the associated class label is unknown, the attribute values of the tuple are tested against decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple.

Decision tree is useful because construction of decision tree classifiers does not require any domain knowledge. It can handle hidimensional data. The learning and classification steps of decision tree induction are simple and fast. Their representation of acquired knowledge in tree form is easy to assimilate by users. Decision tree classifiers have good accuracy [15].

4.1 Mining Classification Rules

Every data classification project is different but the projects have some common features. Data classification requires some rules [16]. This classification rules are given below

- The data must be available
- The data must be relevant, adequate, and clean
- There must be a well-defined problem
- The problem should not be solvable by means of ordinary query
- The result must be actionable

4.2 Proposed Decision Tree Algorithm

The decision tree algorithm is a top-down induction algorithm. The aim of this algorithm is to build a tree that has leaves that are homogeneous as possible. The major step of this algorithm is to continue to divide leaves that are not homogeneous into leaves that are as homogeneous as possible. Steps of this algorithm are given below.

Input:

- Data partition, D, which is a set of training tuples and their associated class labels.
- Attribute_list, the set of candidate attributes.
- Attribute_selection_method, a procedure to determine the splitting criterion that “best” partitions the data tuples into individual classes.

Output: A decision tree.

4.3 Decision Tree Rules

There are a number of advantages in converting a decision tree to rules. Decision tree make it easier to make pruning decisions. Since it is easier to see the context of each rule. Also, converting to rules removes the distinction between attribute tests that occur near the root of the tree and those that occur near the leaves. These rules are easier to read and to understand for people. The basic rules for decision tree are as below.

- Each path from the root to the leaf of the decision tree therefore consists of attribute tests, finally reaching a leaf that describes the class.
- If-then rules may be derived based on the various paths from the root to the leaf nodes.
- Rules can often be combined to produce a smaller set of rules. For example:
- If result = “distinction %” then credit rating = excellent
- If stream = “arts” and result = “70 %” then credit rating = average.
- Once all the rules have been generated, it may be possible to simplify the rules.
- Rules with only one antecedent (e.g. if result = “distinction”) can not be further simplified. So we only consider those with two or more antecedents.
- Eliminate unnecessary rule antecedents that have no effect on the conclusion reached by the rule.
- In some cases, a number of rules that lead to the same class may be combined.

4.4 *Generation of Standard Decision Tree*

For generating decision tree, first we require data table that is given in table-2 as following.

As, shown in given table-2, we see that there are four attributes (e.g. outlook, temperature, humidity, wind) to decide that tennis should be played or not. For result (play tennis), there are two classes such as Yes of No.

These attributes may be increased or decreased. But if the numbers of attributes are more than data classification can be done with more accuracy. The decision tree for above data can be generated as shown in figure-1.

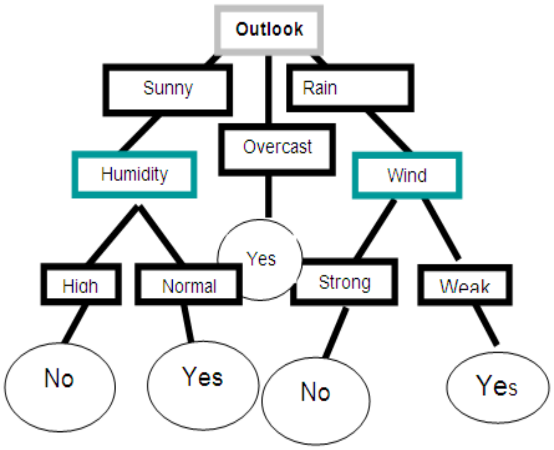

Figure 1: Decision Tree for Data Shown in Table 2

Table 2: Different Attributes

Outlook	Tempe-rature	Humidity	Wind	Play tennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

V. Results

Here I have collect the data of 10 students with different attributed like Roll no, City, qualification, 10th result,12th Result , father occupation etc. and base on that I have try to find the knowledge with help of both the above mention technique like KNN and decision tree. Finally I found with graph in terms of accuracy and complexity both decision trees are better than K-Means.

5.1 *Experimental Table of Dataset*

As shown in above table-3, there are seven attributes for selection. You can choose any number of attributes from given set. First we have to choose the best attribute. From above set, the best attribute may be student qualification. Student may be of 10th and 12th standard. Here, with one attribute, classification can be achieved with lower accuracy. So, with more and more attributes, higher accuracy can be achieved for data classification but increase in number of attributes for classification also increases complications.

Table 3: Experimental Dataset Conducted for Experiment

Reg_No	City	S_Qualification	Stream	10_Result	12_Result	Father_Occupatio	Fianancial_Status
05CP001	Ahmedabad	12th	Sci.	86 %	75 %	Service	Middle Class
05CP002	Nadiad	10th		70 %		Business	Rich
05CP003	Jamnagar	12th	Com	78 %	85 %	Service	Rich
05CP004	Palanpur	12th	Com	70 %	65 %	Worker	Poor
05CP005	Mehsana	12th	Arts	65 %	72 %	Worker	Poor
05CP006	Patan	10th		81 %		Service	Middle Class
05CP007	Anand	12th	Sci.	80 %	90 %	Worker	Poor
05CP008	Ahmedabad	12th	Sci.	80 %	61 %	Business	Rich
05CP009	Dang	10th		55 %		Politician	Rich
05CP010	Idar	12th	Com	50 %	45 %	Politician	Rich

5.2 *Experimental Results*

In this section, this project reports derived results of our reformulated decision tree with standard decision tree. Table-4 shows that if complexity increases from 0% to 80%, the results obtained with datasets so that it is possible to reduce the training set even up to 90% without significant effect upon the classification results. Also, these result shows that if number of attributes increase then both the accuracy and complexity increase.

Table 4: Comparison of Attributes with Accuracy and Complexity

No of attributes	Name of attributes	Complexity (%)	Accuracy (%)
1	Qualification	0	50
2	10th and 12th result	10	60
3	10th-12th result, city	20	65
4	3 and stream	35	70
5	4 and father_occupation	45	80
6	5 and financial status	60	85
7	All	80	95

5.3 *Performance Analysis*

In this section, it compares reformulated decision tree with standard decision tree for dataset. Our comparison is from threshold (complexity) from low to high with reference to the testing accuracy. We took the threshold 35%, 45%, 60% and 80%. The result shows that our method gives better accuracy with decision tree rather than K- Means. The dataset is of type text and number. As the threshold is increased, the time taken to train decision tree is to be decreased.

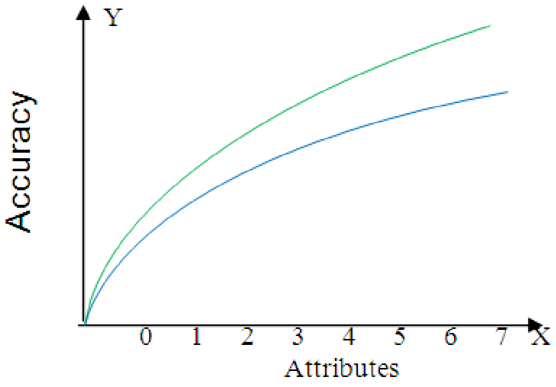

Figure 2: Comparison of Attributes and Accuracy

- Decision Tree Induction Algorithm
- K-Means Algorithm

This graph shows that if the number of attributes increases, then accuracy of classification also increases.

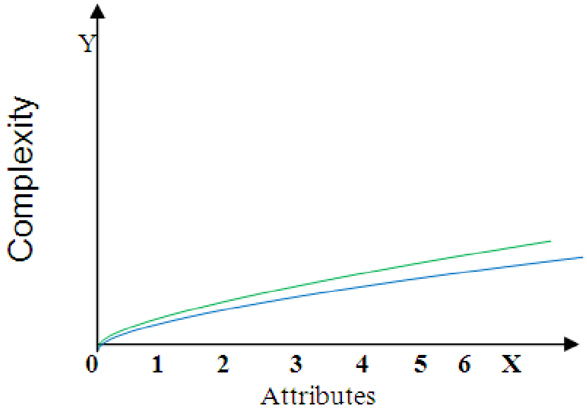

Figure 3: Comparison of Attributes and Complexity

- Decision Tree Induction Algorithm
- K-Means Algorithm

Above graph shows that with respect to attributes, complexity of classification increases. Level of prediction goes to high when complexity goes high. So from above graph we can see the complexity of decision tree is high so prediction will better.

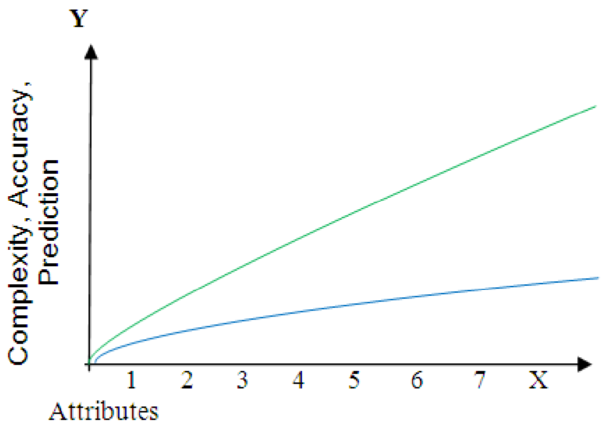

Figure 4: Comparison of Attributes with Complexity, Accuracy & Prediction

- Decision Tree Induction Algorithm
- K-Means Algorithm

As shown in the Figure 4 we can say that when number of attributes increases then complexity increases slowly, accuracy increases highly, and as a result of accuracy level of prediction goes to high. Although, it is fact that 100 % prediction is not possible with any system because ultimately the God is great.

As stated earlier that, information gain measures the information. We use the data from table to calculate information measure.

Table 5: Student Credit Rating

Stream	Qualification	Result (%)	Urban area	Rating (Intelligent, Average, Poor)
Science	12^{th}	78.50	Yes	I
Arts	12^{th}	80	No	A
Science	12^{th}	65	No	A
Commerce	12^{th}	80.78	Yes	I
	10^{th}	75	Yes	A
Science	12^{th}	45	Yes	P
Science	12^{th}	45	No	P
	10^{th}	50.59	Yes	P
Science	12^{th}	75	No	I
	10^{th}	40.30	No	P

In above table, I indicates Intelligent, A indicates Average, P indicates Poor.

There are 10 (s = 10) samples and three classes. The frequencies of these classes are:

$$I = 3, A = 3, P = 4 \qquad (5.1)$$

Information in the data due to uncertainty of outcome regarding the credit rating each student belongs to is given by

$$I = -(n/s) \log(n/s) – (p/s) \log(p/s) \qquad (5.2)$$

Transferring the values from equation 1 into equation 2, we get

$$I = -(3/10) \log(3/10) – (3/10) \log(3/10)–(4/10)\log(4/10) = 1.57$$

Let us consider using main attribute (stream) as a candidate to split sample.

VI. Conclusion

As per the implementation of this project, data classification with decision tree is easy with compared to other method. Because of previous records and pictorial view, the task of categorization of data becomes easy. Once the result obtained, it can be reused for next research. This research depicts on compares reformulated decision tree with standard decision tree for dataset. Our comparison is from threshold (complexity) from low to high with reference to the testing accuracy. With this research a set of threshold taken to show that our method gives better accuracy with decision tree rather than K- Means. The dataset is of type text and number. As the threshold is increased, the time taken to train decision tree is to be decreased. The advantage of decision tree is that it provides a theoretical framework for taking into account not only the experimental data to design an optimal classifier, but also a structural behavior for allowing better generalization capability.

References

[1] Hwanjo Yu, Jiong Yang, Jiawei Han, "Classifying Large Data Sets Using SVMs with Hierarchical Clusters", ACM SIGKDD-03, Pp. 24-27, 2003.

[2] Rich Caruana, Alexandru Niculescu-Mizil, "Data mining in metric space: an empirical analysis of supervised learning performance criteria", KDD '04 Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004

[3] Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, Timm Euler, "YALE: rapid prototyping for complex data mining tasks", KDD '06 Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.

[4] Warren Armstrong; Peter Christen; Eric McCreath; Alistair P Rendell;, "Dynamic Algorithm Selection Using Reinforcement Learning," Integrating AI and Data Mining, 2006. AIDM '06. International Workshop on , Pp.18-25, Dec. 2006

[5] Safavian, S.R.; Landgrebe, D.; , "A survey of decision tree classifier methodology," Systems, Man and Cybernetics, IEEE Transactions on , Vol. 21, No. 3, Pp.660-674, May/Jun 1991.

[6] Márcio P. Basgalupp, Rodrigo C. Barros, André C. P. L. F. de Carvalho, Alex A. Freitas, Duncan D. Ruiz, "LEGAL-tree: a lexicographic multi-objective genetic algorithm for decision tree induction", SAC '09 Proceedings of the 2009 ACM symposium on Applied Computing.

[7] Carlos Ordonez, "Clustering binary data streams with K-means", DMKD '03 Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery.

[8] Zhexue Huang, "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values", Data Mining and Knowledge Discovery, Vol. 2, No. 3, 283-304, DOI: 10.1023/A:1009769707641.

[9] A. K. Jain, M. N. Murty, P. J. Flynn, "Data clustering: a review", ACM Computing Surveys (CSUR) Surveys Homepage archive, Vol. 31, No. 3, 1999.

[10] Watanabe. N, "Fuzzy modeling by hyperbolic fuzzy k-means clustering," Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conferencr, Vol. 2, Pp.1528-1531, 2002 DOI: 10.1109/FUZZ.2002.1006733.

[11] Juanying Xie; Shuai Jiang; , "A Simple and Fast Algorithm for Global K-means Clustering," Education Technology and Computer Science (ETCS), 2010 Second International Workshop on , Vol. 2, Pp. 36-40, March 2010, DOI: 10.1109/ETCS.2010.347.

[12] Du Haizhou; Ma Chong; , "Study on Constructing Generalized Decision

Tree by Using DNA Coding Genetic Algorithm," Web Information Systems and Mining, 2009. WISM 2009. International Conference on , Pp.163-167, 7-8 Nov. 2009, DOI: 10.1109/WISM.2009.41

[13] Shekhar R. Gaddam, Vir V. Phoha, Kiran S. Balagani, "K-Means+ID3: A Novel Method for Supervised Anomaly Detection by Cascading K-Means Clustering and ID3 Decision Tree Learning Methods," Knowledge and Data Engineering, IEEE Transactions on , Vol. 19, No. 3, Pp. 345-354, March 2007.

[14] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu and Philip S. Yu, et al., "Top 10 algorithms in data mining", Knowledge and Information Systems, Vol. 14, No. 1, 1-37, DOI: 10.1007/s10115-007-0114-2.

[15] Hang Yang, Fong, S, "Optimized very fast decision tree with balanced classification accuracy and compact tree size," Data Mining and Intelligent Information Technology Applications (ICMiA), 2011 3rd International Conference on, Pp.57-64, 24-26 Oct. 2011.

[16] Guang-Hua Chen; Zheng-Qun Wang; Zhen-Zhou Yu;, "Constructing Decision Tree by Integrating Multiple Information Metrics," Chinese Conference on Pattern Recognition, 2009. CCPR 2009, Pp.1-5, 4-6 Nov. 2009 DOI: 10.1109/CCPR.2009.5344133.