ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
**Indian Journal of Science and Technology,** *Vol 8(9), 868–874, May 2015*
DOI: 10.17485/ijst/2015/v8i9/53064

# A Comparative Study on MMDBM Classifier Incorporating Various Sorting Procedure

**P. Ganesan[1], S. Sivakumar[1*] and S. Sundar[2]**

[1]Department of Mathematics, College of Engineering Guindy, Anna University, Chennai - 600 025, Tamil Nadu, India; ganesan@annauniv.edu, sivaiit79@gmail.com
[2]Department of Mathematics, Indian Institute of Technology Chennai, Chennai - 600 036, Tamil Nadu, India; slnt@iitm.ac.in

## Abstract

Classification is one of the most important methods in data mining. These methods are used to extract meaningful information from large database which can be effectively used for predicting unknown class. The classifier based on decision tree is called Mixed Mode Data Base Miner (MMDBM) which is tested with different sorting techniques (merge sort, quick sort, radix sort) to compare the processing time to SLIQ classifier. In this paper, we carried out a comparative study on MMDBM classifier incorporating various sorting procedure by using Blood Pressure (BP) database, and finally the proposed method MMDBM classifier is one of the best classifiers among SLIQ supervised learning method. This proposed method achieved less processing time and higher rate of accuracy.

**Keywords:** Classification, Data Mining, Java Sorting.

## 1. Introduction

Data mining is used to extract hidden information from large Datasets. The data mining tools are used to predict the future trends and behaviours. They are also used as automated decision support systems. Data mining is also used to uncover hidden patterns in large databases[16]. This paper addresses the well-known classification task of data mining[4]. In this task, the discovered knowledge is often expressed as a set of rules of the form: IF <conditions> THEN <prediction (class)>[10]. Classification has been successfully applied using several approaches. On the one hand, there are approaches such as Artificial Neural Networks (ANN)[11] and other had Support Vector Machines (SVM)[8], decision tree[2,12], and instance-based learning methods. An important tool that is frequently used in Data mining is Classification[1,3,17]. Classification is the process of dividing a dataset into mutually exclusive groups, a class based on appropriate characteristics or attributes. It is also used to analyze the input data and provide an accurate description (model) for each of the classes. An input that emerges from classification can be represented in the form of a tree. The tree, thus obtained, is used for classification of any new data whose class is not known. The tree transformed a set of rules, which can e easily be codified as SQL statements and incorporated into any decision making system. Moreover the applicability of SQL makes impacts cross-platform functionality for the system. Furthermore this idea can be extended with the aid of JDBC and information stored in different Database formats can be effectively utilized. It is known that the accuracy can be improved by using large databases for classification[6,7]. The important characteristics of these algorithms are accuracy of prediction and scalability. Many algorithms suffer from the problem that they do not scale easily, i.e. the algorithm usually has an upper limit to the size of the data, called the training set consisting of records with certain attributes and a classifying attribute that indicates the class

*Author for correspondence

to which the record belongs, is used for classification. This is mostly due to the fact that the algorithms load the entire database onto the memory and can thus handle only restricted quantities of data. In this paper, Object oriented design of MMDBM has been implemented in Java and the code samples has been provided. Three different case studies have been taken into consideration and tested for accuracy with the provided code and different types of sorting algorithm are compared with an existing classification technique called MMDBM[16].

This paper has different section which is as follows. Section 2 formally states the related works by the paper. Section 3 deal with MMDBM algorithm. Section 4 precisely describes the case study of medical database for BP. Finally, Section 5 deals with conclusion, results and discussion.

## 2. Related Works

The research work is mainly based on three algorithms, SPRINT planned classification algorithm called SPRINT that eliminates all memory limitations that restrict decision-tree algorithm, which proves that planned algorithm is fast and scalable[16].

SLIQ is a decision tree classifier that can handle both numerical and categorical attributes. It builds compact and accurate trees. It uses a novel pre-sorting technique in the tree-growth phase to reduce the cost of evaluating numeric attributes. This sorting procedure is integrated with a breadth first tree growing strategy to enable classification of disk-resident datasets. SLIQ uses a fast sub setting algorithm for determining splits for categorical attributes[9].

A new decision tree classifier for handling numerical and categorical attributes in large datasets called Mixed Mode Database Miner (MMDBM) has been developed. The classifier method used can handle large databases with large set of data or large number of attributes. A new efficient index for qualifying split points is presented. This is partly divided into two sections, first one predictive classifier gives a detailed description of our algorithm and another one object oriented design, gives the object oriented implementation of our algorithm and description of the front-end developed for the algorithm[15]. In their paper, the authors compare their algorithm with well-known SLIQ, SPRINT and MMDBM algorithm. We describe that decision tree classifier SLIQ, SPRINT and MMDBM have achieved good accuracy, compactness and efficiency for very large data sets[5,7].

### 2.1 Decision Tree Algorithm

A decision tree is one well-known classification technique. Decision tree is a tree consisting of a root node, child nodes and edges. Each internal node is a test node that indicates the attribute the edges indicate the possible values taken on by that attribute. Each non-leaf node consists of a splitting point and the main task for building the decision tree is to identify the test attribute for each splitting points[9,13,14].

```
MakeTree (Training Data T)
        Partition (T);
BuildTree (Data set S)
        if (all records in S are in same class)
        return;
                for each attribute A
        Use best split found to partition S1 into S2;
                Partition (S1);
                Partition (S2);
```

## 3. MMDBM Algorithm

**Input:** A is the attribute containing n attributes A = $\{a_1,a_2,.....,a_n\}$ from data base

**Output:** Distribution of the node count and construction of the decision tree.

1. Start sorting (Merge sort or Quick sort or Radix sort) of the all attributes.
2. Get the midpoint value of each and every attribute.
3. Scan the attributes of all the records.
4. For every attribute. Place all nodes into a class histogram.
5. Start evaluation of the splits for each attribute A do scan the attribute list of A = $\{A_1,A_2,.....,A_n\}$ for each attribute name and data value v in the attribute list do split point at each and every node based on mid-point value. If A is a numeric attribute then Compute splitting index ($A \leq v$) up to terminal node l.
6. Apply the split condition producing a new node
7. Finally count the class value or traversal node which already exist and then update the appropriate class count value.

### 3.1 Algorithm Implementation

Once the pre-processing is complete the implementation of the object oriented programming for fast classifier mining algorithm commences. After getting mid-point value dynamically start the classification based on the predicated rules used in medical datasets. The mid-point value is compared to each record from database and a travel path is created. The condition is true splitting the node go to left side node up to n number of records and condition is false splitting node go to right side node up to n number of records and else condition is to count the missing values and the histogram is calculated. As explained in the algorithm of fast classifier mining algorithm, the data that satisfy the condition and the data that does not satisfy the condition and the missing data have been recorded. Based on that, a histogram is developed for each distribution of the node. The decision tree created should be a binary tree from distribution of the node count values; the number of nodes in the tree is 2n-1, where n is the number of attributes. In this case, the number of attribute is 7(Sex, Age, Weight, Sports, Sleep, Drink, BP), so the number of nodes is 128-1 = 127.

## 3.2 Performing Splits

Once the pre-processing is complete the implementation of the algorithm starts. For each attribute, the corresponding attribute array is loaded and the data is passed to the node pointed to by the leaf in the corresponding class list entry.

### *3.2.1 Splits Point for Numerical Attribute*

```
mergeSort(arr);
for (int i = 0; i < arr.length; i++)
{if (i = count || i = nextVal)
{age = (int) arr [i];
result = result+age;}}
midAge = result/2;
return midAge;}
mergeSort(arr);
for (int i = 0; i < arr.length; i++)
{if (i = count || i = nextVal)
{weight = (int) arr[i];
result = result+wegiht;}
midWeight = result/2;}
return midWeight;
```

After getting mid-point value and classifying all the attributes of all the records from connected datasets. We classify the node using If <Condition> Then (class value) rule. Finally distribution of the node count values are evaluated based on rules and histogram of the classified nodes (Figure 2.) construct the decision tree from distribution of the travelled path (Figure 3).

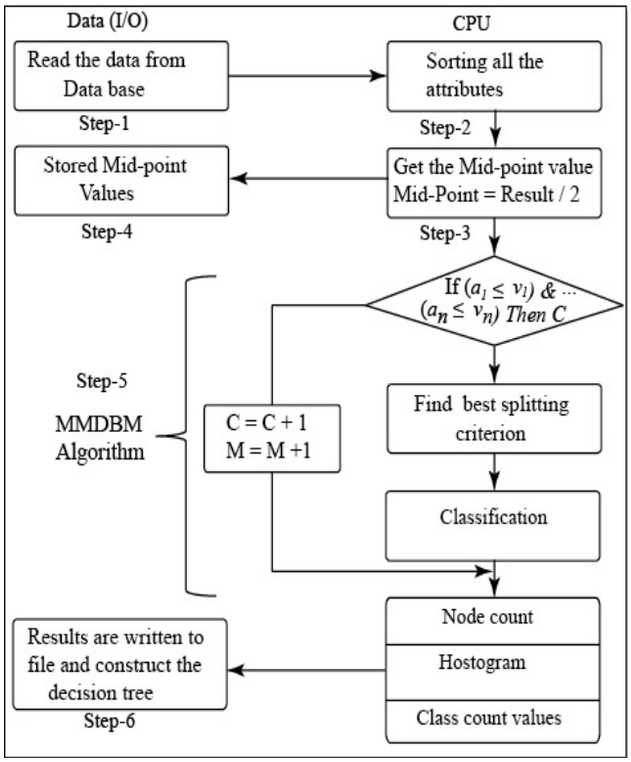


**Figure 1.** Design for MMDBM Algorithm.

## 3.3 Proposed Method

This algorithm is divided into six step process. Read the data from connected database (step 1). All the numeric attributes are sorted by ascending order and sorted the data an array (step 2). All the sorted attributes get the mid-point and the values are stored in an array (step 3 and 4). Mid-point value checks the condition in first attribute value, if condition is true go to the left side another node and, again the midpoint value checks the condition to second attribute up to n number of attributes which is called as distribution (step 5) using MMDBM algorithm. Finally, it counts the class value and calculates the histogram of every distribution. Final result are written to the file and is constructed the decision tree for every distribution (step 6).

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| N1 | sex=M | Node goto N2 | else N3 | N23 | sleep<=6 | Node goto N46 | else N47 | N45 | drink<=4 | Node goto N90 | else N91 |
| N2 | age<=35 | Node goto N4 | else N5 | N24 | sleep<=6 | Node goto N48 | else N49 | N46 | drink<=4 | Node goto N92 | else N93 |
| N3 | age<=35 | Node goto N6 | else N7 | N25 | sleep<=6 | Node goto N50 | else N51 | N47 | drink<=4 | Node goto N94 | else N95 |
| N4 | weight<=48 | Node gpto N8 | else N9 | N26 | sleep<=6 | Node goto N52 | else N53 | N48 | drink<=4 | Node goto N96 | else N97 |
| N5 | weight<=48 | Node goto N10 | else N11 | N27 | sleep<=6 | Node goto N54 | else N55 | N49 | drink<=4 | Node goto N98 | else N99 |
| N6 | weight<=48 | Node goto N12 | else N13 | N28 | sleep<=6 | Node goto N56 | else N57 | N50 | drink<=4 | Node goto N100 | else N101 |
| N7 | weight<=48 | Node goto N14 | else N15 | N29 | sleep<=6 | Node goto N58 | else N59 | N51 | drink<=4 | Node goto N102 | else N103 |
| N8 | sport<=4 | Node goto N16 | else N17 | N30 | sleep<=6 | Node goto N60 | else N61 | N52 | drink<=4 | Node goto N104 | else N105 |
| N9 | sport<=4 | Node goto N18 | else N19 | N31 | sleep<=6 | Node goto N62 | else N63 | N53 | drink<=4 | Node goto N106 | else N107 |
| N10 | sport<=4 | Node goto N20 | else N21 | N32 | drink<=4 | Node goto N64 | else N65 | N54 | drink<=4 | Node goto N108 | else N109 |
| N11 | sport<=4 | Node goto N22 | else N23 | N33 | drink<=4 | Node goto N66 | else N67 | N55 | drink<=4 | Node goto N110 | else N111 |
| N12 | sport<=4 | Node goto N24 | else N25 | N34 | drink<=4 | Node goto N68 | else N69 | N56 | drink<=4 | Node goto N112 | else N113 |
| N13 | sport<=4 | Node goto N26 | else N27 | N35 | drink<=4 | Node goto N70 | else N71 | N57 | drink<=4 | Node goto N114 | else N115 |
| N14 | sport<=4 | Node goto N28 | else N29 | N36 | drink<=4 | Node goto N72 | else N73 | N58 | drink<=4 | Node goto N116 | else N117 |
| N15 | sport<=4 | Node goto N30 | else N31 | N37 | drink<=4 | Node goto N74 | else N75 | N59 | drink<=4 | Node goto N118 | else N119 |
| N16 | sleep<=6 | Node goto N32 | else N33 | N38 | drink<=4 | Node goto N76 | else N77 | N60 | drink<=4 | Node goto N120 | else N121 |
| N17 | sleep<=6 | Node goto N34 | else N35 | N39 | drink<=4 | Node goto N78 | else N79 | N61 | drink<=4 | Node goto N122 | else N123 |
| N18 | sleep<=6 | Node goto N36 | else N37 | N40 | drink<=4 | Node goto N80 | else N81 | N62 | drink<=4 | Node goto N124 | else N125 |
| N19 | sleep<=6 | Node goto N38 | else N39 | N41 | drink<=4 | Node goto N82 | else N83 | N63 | drink<=4 | Node goto N126 | else N127 |
| N20 | sleep<=6 | Node goto N40 | else N41 | N42 | drink<=4 | Node goto N84 | else N85 | N64 | Terminated 100% with H | | |
| N21 | sleep<=6 | Node goto N42 | else N43 | N43 | drink<=4 | Node goto N86 | else N87 | N65 | Terminated 100% with L | | |
| N22 | sleep<=6 | Node goto N44 | else N45 | N44 | drink<=4 | Node goto N88 | else N89 | N66 | Terminated 100% with H | | |

**Figure 2.** Predicted Rules for Medical database in BP.

## 4. Case Study

To implement and test the above algorithm, three areas where data mining is used has been taken into consideration. Real data has been recorded and given as an input to the algorithm to check its effectiveness and accuracy. The case study is from the medical data base where the algorithm is given the task to predict risk of the person for having high BP based on seven different attributes.

### 4.1 Medical database for Blood Pressure (BP)

The history contains data from surveys conducted among patients. The database contains records of the following Attributes. 1. Sex: Categorical; M/F], 2. Age: representing the age of a person; Numeric (years), 3.Weight: The weight of a person; Numeric Kilo grams), 4. Sports: The extent of exercise a person; Numeric; 1-10), 5. Sleep: The number of hours a person sleeps on an average; Numeric (0, 24). 6. Drink: The extent of drinking of a person; Numeric, (1–5). 7. BP: Categorical [HP, LP, NP], this is class value. We have classified all records in connected datasets and got 24 distribution of the travelled path from 30,000 records and counted the number of patients with low BP, high BP and normal BP (Figure 2). Each distribution have been generated by multiple IF <condition> Then rule[5,10]. This rule was generated dynamically based on predicted rules of the class count values and travelled path is given below.

To the given algorithm, a large data set was given as input to test the accuracy of the prediction. Figure 3 shows the decision tree of the medical database. Number of data ranging from 10000 to 30000 is supplied to the program as input and classification is done. It is observed that with the increase in the number of records, the prediction of accuracy is improved and the processing time for classification is shown in the given table. Hence it is proved that quick sort is used in and it is MMDBM algorithm is superior to other algorithm (Figure 4).

The comparison of processing time between the supervised learning method SLIQ classifier and the proposed MMDBM classifier on blood pressure data set with 100000 records is shown in Figure 5.

## 5. Conclusion

Classification has always been one of the major areas in data mining. A new classifier called Mixed Mode Database Miner (MMDBM) is programmed in Java with three different sorting algorithms and is tested using datasets. The algorithm can handle large datasets with large number of attributes. The algorithm gives excellent scalability of medical databases that are taken for analysis and experimenting. A case study is taken into consideration

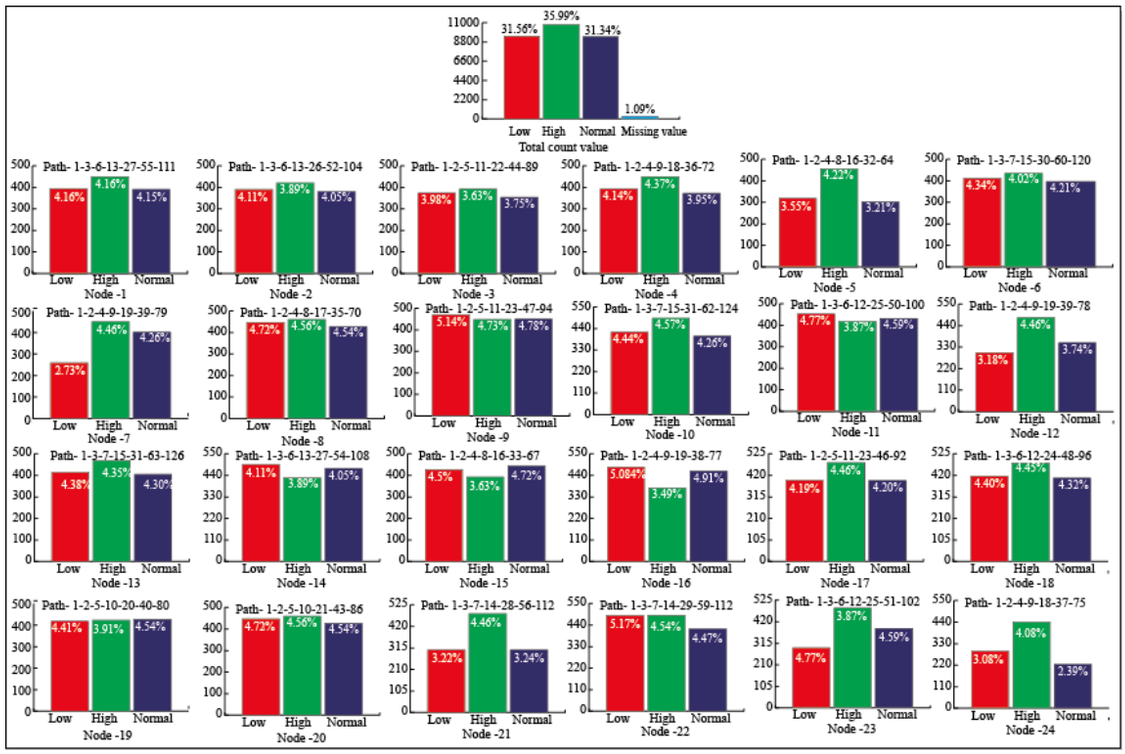


**Figure 3.** Class Distribution for Medical database in BP.

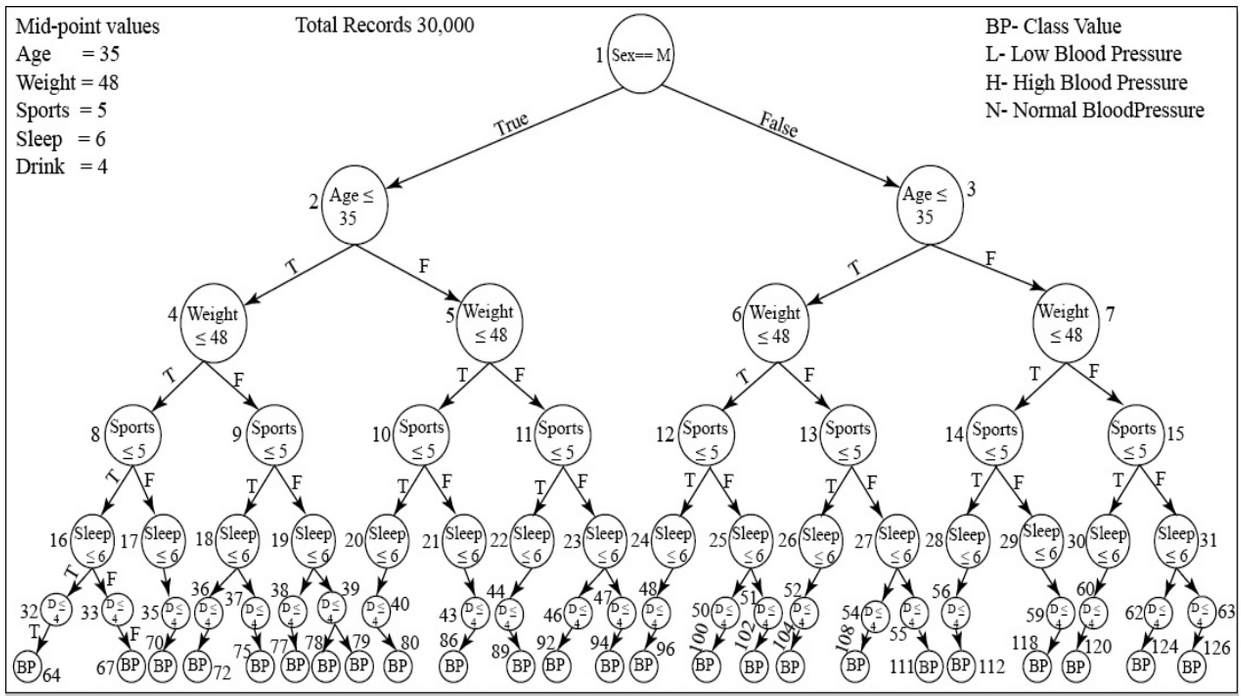


**Figure 4.** Classification tree in Medical database for BP.

**Table 1.** Processing time of BP

| Algorithm Name | Records 10000 | Records 20000 | Records 30000 | Records 40000 | Records 50000 | Records 100000 |
|---|---|---|---|---|---|---|
| SLIQ | 6.21 | 12.5 | 17.6 | 20.5 | 26.8 | 42.5 |
| MMDBM Quick sort | 0.432 | 0.769 | 1.032 | 1.076 | 1.467 | 2.106 |
| MMDBM Merge sort | 0.483 | 0.771 | 1.069 | 1.134 | 1.482 | 2.558 |
| MMDM Radix Sort | 5.992 | 7.45 | 11.487 | 12.967 | 14.783 | 14.987 |

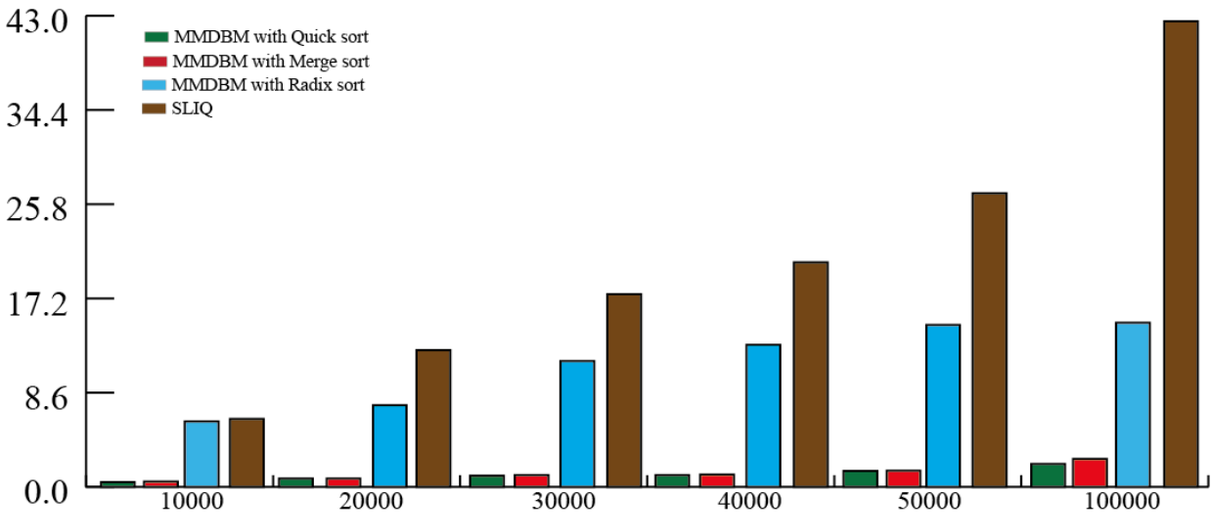


**Figure 5.** Scalability of sorting algorithm of BP database.

and tested for accuracy and the code is provided. The proposed MMDBM method using three sorting algorithms are compared to SLIQ classifier. MMDBM with quick sort algorithm provides fast and accurate results with least processing time. In future this classifier algorithm use in real time application.

## 6. References


1. Agarwal R, Srikant R. Fast Algorithm for mining association rules. Proceedings of International Conference on Very Large Data Bases. 1994 Sep. p. 487–99.
2. Ai-Hegami A. Pruning based interestingness of mined classification patterns. Int Arab J Inform Tech. 2009; 6(4):336–43.
3. Breiman L, et al. Classification and regression Trees. Belmont: Wadsworth; 1984.
4. Chandrashekar A, Vijay Kumar J. Data mining based hybrid intrusion detection system. Indian Journal of Science and Technology. 2014; 7(6):781–9.
5. Carvalhoa DR, Freitasb AA. A hybrid decision tree/genetic algorithm method for data mining. Inform Sci. 2004; 163(1–3):13–35.
6. Fayyad UM, Piatesky-Shapiro G, Smith. Advances in knowledge discovery in database. Cambridge, Massachusetts: AAAI/MIT Press; 1996.
7. Han J, Cai Y, Cercone N. Data-driven discovery of quantitative rules in relational databases. IEEE Trans Knowl Data Eng. 1993; 5(1):26–40.
8. Joachims T. Text categorization with support vector machines: learning with many relevant feature. Machine Learning: ECML'98. Berlin, Heidelberg: Springer; 1998. p. 137–42.
9. Mehta M, Agarwal R, Rissanen J. SLIQ: A fast scalable classifier for data mining. International Conference on Extending Database Technology (EDBT'96), Avignon, France; 1996. p. 18–32.

10. Omer. A rule induction algorithm for knowledge discovery and classification. Turk J Elec Eng Comput Sci. 2013; 21:1223–41.
11. Piatetsky-Shapiro G, Frawley WJ. Knowledge discovery in databases. Cambridge, Massachusetts: AAAI/MIT Press; 1991.
12. Quinlan J. C4.5: Programs for Machine Learning. San Francisco, CA, USA: The Morgan Kaufmann; 1993. p. 235–40.
13. Abraham RM, Beeda AR, Manjula R. Data mining: building social network Sayali Nishikant Chakradeo. Indian Journal of Science and Technology. 2015; 8(S2):212–6.
14. Iqbal R, Azmi Murad MA, Mustapha A, Panahy PHS, Khanahmadliravi N. An experimental study of classification algorithms for crime prediction. Indian Journal of Science and Technology. 2013; 6(3):4219–25.
15. Shafer J, Agrawal R, Metha M. SPRINT: a scalable parallel classifier for data mining. Proceedings of the 22nd VLDB Conference Mumbai, India; 1996.
16. Sundar S, Srikanth D, Shanmugam MS. A new predictive classifier for improved performance in data mining: object oriented design and implementation. Proceedings of the International Conference on Industrial Mathematics, IIT Bombay. New Delhi: Narosa; 2006. p. 491–514.
17. Weiss SM, Kulikowski CA. Computer system that learning: classification and prediction method from statistics, neural Nets. Machine Learning, and Expert System. San Francisco, CA, USA: Morgan Kaufman; 1991.