2012 IEEE International Conference on Control System, Computing and Engineering, 23 - 25 Nov. 2012, Penang, Malaysia

A Comparative Study of Reduced Error Pruning Method in Decision Tree Algorithms

W. Nor Haizan W. Mohamed, Mohd Najib Mohd Salleh, Abdul Halim Omar
Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia
Batu Pahat, Malaysia
gi110016@siswa.uthm.edu.my, najib@uthm.edu.my, gi110017@siswa.uthm.edu.my

***Abstract*—Decision tree is one of the most popular and efficient technique in data mining. This technique has been established and well-explored by many researchers. However, some decision tree algorithms may produce a large structure of tree size and it is difficult to understand. Furthermore, misclassification of data often occurs in learning process. Therefore, a decision tree algorithm that can produce a simple tree structure with high accuracy in term of classification rate is a need to work with huge volume of data. Pruning methods have been introduced to reduce the complexity of tree structure without decrease the accuracy of classification. One of pruning methods is the Reduced Error Pruning (REP). To better understand pruning methods, an experiment was conducted using Weka application to compare the performance in term of complexity of tree structure and accuracy of classification for J48, REPTree, PART, JRip, and Ridor algorithms using seven standard datasets from UCI machine learning repository. In data modeling, J48 and REPTree generate tree structure as an output while PART, Ridor and JRip generate rules. In additional J48, REPTree and PART using REP method for pruning while Ridor and JRip using improvement of REP method, namely IREP and RIPPER methods. The experiment result shown performance of J48 and REPTree are competitive in producing better result. Between J48 and REPTree, average differences performance of accuracy of classification is 7.1006% and 6.2857% for complexity of tree structure. For classification rules algorithms, Ridor is the best algorithms compare to PART and JRip due to highest percentage of accuracy of classification in five dataset from seven datasets. An algorithm that produces high accuracy with simple tree structure or simple rules can be awarded as the best algorithm in decision tree.**

***Keywords*-Decision tree, rules, and Reduced Error Pruning.**

I. INTRODUCTION

In data mining, a decision tree is a predictive model [1] which can be used to represent both classifiers and regression models [2]. Decision trees are categorized as a supervised method that trying to find the relationship between input attributes and target attributes which represent the relationship in structure as a model [2]. The model constructed by using input attributes to predict target attribute values [3] where an input pattern is classified into one of several classes based on their attributes values.

Classification of data is one of the important tasks in data mining [4, 5, 6]. Decision tree is one of the most widely used technique [7, 8, 9, 18, 20], and is very popular among researchers [1, 9, 10] because of their simplicity [11], intelligibility [12, 13, 18], ease of implementation [8, 18], decision tree construction faster and produce better accuracy [13] then other classification algorithms. A decision tree is a flow chart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node represents a class [13, 14].

Decision tree can be represented by two types of structures; usually it is represented in tree structure (hierarchical structure) and rules (if-then statement). If decision tree is complicated, tree structure and rules might be wasted [20]. For a complex tree, pruning procedures must be developed to facilitate the interpretation. According to Drazin [15], pruning is a methods that reduces the size of the tree by removing parts of the tree that not meaningful to avoid unnecessary complexity and to avoid over-fitting of the dataset. Pruning decision trees is a fundamental step in optimizing the computational efficiency as well as classification accuracy of such a model. The complexity of tree is clearly controlled by the pruning method used in [2].

There are two standard classes of methods proposed for pruning namely pre-pruning (forward pruning) and post-pruning (backward pruning). Mahmood mention in [16], pre-pruning works with stop growing the tree earlier based on some stopping criteria, before it classifies the training set perfectly. In pre-pruning, the advantage is not generating full tree and the disadvantage is horizon effect phenomenon [17]. Post-pruning has two phases, growing phase and pruning phase. First of all, it allows over-fitting the data, and then post prunes the grown tree. In practice post-pruning methods has a better performance than pre-pruning [16]. Reduced Error Pruning (REP) is a post-pruning method decision tree [19]. REP finds the smallest version of the most accurate subtree with respect to the pruning set [21].

978-1-4673-3141-8©2012 IEEE

II. Decision Tree Algorithms

Nowadays there are many available tools in data mining, which allow execution of several task in data mining such as data preprocessing, classification, regression, clustering, association rules, features selection and visualisation. All the above mention tasks are closed under different algorithms and are available an application or a tool. In this research we choose WEKA (The Waikato Environment for Knowledge Analysis) for running several algorithms in decision tree. Each algorithm was explained in subsection from A to E.

A. J48

J48 is an implementation of C4.5 algorithm [22]. C4.5 was a version earlier algorithm developed by J. Ross Quinlan. There two methods in pruning support by J48 first are known as subtree replacement, it work by replacing nodes in decision tree with leaf. Basically by reduce the number of test with certain path. It works with the process of starting from leaves that overall formed tree and do a backward toward the root. The second type implemented in J48 is subtree raising by moved nodes upwards toward the root of tree and also replacing other nodes on the same way.

According to Zhao and Zhang [23], C4.5 algorithm produce decision tree classification for a given dataset by recursive division of the data and the decision tree is grown using Depth-first strategy. On data testing this algorithm will emphasized on splitting dataset and by selecting a test that will give best result in information gain. In discrete attributes as well, these algorithms consider a test with a result of many as the number of different values and test binary attribute for each attribute will continue to grow in different values each attribute will be considered.

Furthermore Zhao and Zhang said [23], In order to gather the entropy gain of all these binary tests efficiently, the training data set belonging to the node in consideration is sorted for the values of the continuous attribute and the entropy gains of the binary cut based on each distinct values are calculated in one scan of the sorted data. This process is repeated for each continuous attributes.

B. REPTree

Basically Reduced Error Pruning Tree ("REPT") is a fast decision tree learning and it builds a decision tree based on the information gain or reducing the variance. The basic of pruning of this algorithm is it used REP with back over fitting. It kindly sorts values for numerical attribute once and it handling the missing values with embedded method by C4.5 in fractional instances. In this algorithm we can see it used the method from C4.5 and the basic REP also count in it process.

C. PART

PART algorithm [24] is a relatively simple algorithm who does not execute global optimization to generate accurate rules, but it is practiced separately and-conquer strategy, for example it builds a rule, removes the instances it covers, and continues to create a recursive rule for instances rest until there is no longer the instances is left.

Furthermore Eibe and Witten [24] said, the algorithm producing sets of rules called 'decision lists' which are ordered set of rules. A new data is compared to each rule in the list in turn, and the item is assigned the category of the first matching rule (a default is applied if no rule successfully matches). PART builds a partial C4.5 decision tree in every iterative and makes the "best" leaf into a rule. The algorithm is a combination of C4.5 and RIPPER rule learning.

D. Ridor

Brian R. Gaines and Paul Compton [25] has develop Ridor or RIpple-DOwn Rule learner. This algorithm generate default rule first and after that it generate the exceptions for default rule along with the least error rate. Then it generates the "best" exceptions for each exception and iterates until pure. Thus it performs a tree-like expansion of exceptions. The exceptions are a set of rules that predict classes other than the default. IREP is used to generate the exceptions.

E. JRip

In 1995 JRip was implemented by Cohen, W. W, in this algorithm were implemented a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER). By the way, Cohen implementing RIPPER [26] in order to increase the accuracy of rules by replacing or revising individual rules. Reduce Error Pruning was used where it isolate some data for training and decided when stop from adding more condition to a rule. By using the heuristic based on minimum description length as stopping criterion. Post-processing steps followed in the induction rule revising the regulations in the estimates obtained by global pruning strategy and it improves the accuracy.

III. Datasets

To review the performance of the five decision tree algorithms (J48, REPTree, PART, Ridor and JRip), seven datasets were used as shown in Table I. Dataset is required to represent the problem. Therefore, the problem of classification of real-world datasets has been selected and the data were taken from the University of California Irvine Machine Learning Repository (UCIMLR) [27].

Datasets used in the classification problem are Breast Tissue, Iris, Vertebral Colum2c, Vertebral Column3c, Ecoli, Balance Scale, and Wall. Breast tissue has 106 instances, 9 attributes and 6 classes and it was published in 2010. For the second dataset chosen is Iris dataset it come with 150 instances, 4 attributes, 3 classes and published in 1988. This Iris dataset has been produced to manage of classification 3 types of Iris plant. The third dataset is Vertebral Column2c is a kind of orthopedic patient dataset and it has 310 instances, 6 attributes, 2 classes and it has been published in

978-1-4673-3141-8©2012 IEEE

2011. Furthermore, Vertebral Column3c is the original dataset, it has three classes and has been merged into two classes to become Vertebral Column2c. Ecoli also selected and it has 336 instances, attributes and classes share the same value is 8 and has been published in 1996. Balance scale is a dataset about measurement it came with 625 instances, 4 attributes, 3 classes and published during 1994. For the last dataset is Wall which is have the most highest instances and the value is 5456 instances among dataset chosen followed by 24 attributes and 4 classes and the year published is 2010.

TABLE I. DATASETS FROM UCIMLR

Dataset	Instance	Attribute	Class	Year
Breast Tissue	106	9	6	2010
Iris	150	4	3	1988
Vertebral Column2c	310	6	2	2011
Vertebral Column3c	310	6	3	2011
Ecoli	336	8	8	1996
Balance Scale	625	4	3	1994
Wall	5456	24	4	2010

IV. EXPERIMENT, RESULT AND DISCUSSION

In this section, we conducted an experiment using Weka application. Weka is a comprehensive suite of Java class libraries that perform many advanced machine learning and data mining algorithms [29]. We analyze and compare the performance of decision tree algorithms namely J48, REPTree, PART, Ridor and JRip. All datasets using standard default ten folds cross validation. Data were randomly divided into ten parts where classes are represented in approximately the same proportion as in the full dataset. Each held the next and learning scheme trained nine-tenths of the residue, then the error rate is calculated on the holdout set. Thus, the learning procedure performed ten times on different training set. Finally, an average of ten error estimates to produce estimates of the overall error.

J48, REPTree and PART are using REP method for pruning while Ridor and JRip using IREP and RIPPER method. IREP and RIPPER is an enhancement of REP method. We focus on the accuracy of classification and complexity size of tree. Main learning methods in Weka is a classifier, and they induce a set of rules or decision trees that model data [28]. Tree complexity obviously controlled by stopping criteria used and the method of pruning works [2]. Typically, the complexity of the tree is measured by the following metrics:

- total number of nodes (tree size)
- total leaf
- tree depth
- the number of attributes used

Seven datasets are used to compare the performance of decision tree algorithms to classify objects or instances. Data can be model in tree structure and rules. For J48 and REPTree model data in tree structure while PART, Ridor and JRip model data in rules.

Table II shows the performance of J48 and REPTree using seven datasets, namely Breast Tissue, Iris, Vertebral Column2c, Vertebral Column3c, Balance Scale and Wall. For Breast Tissue dataset, accuracy of J48 correctly classify instances 66.0377% while REPTree 72.6415%. It shows that REPTree classify instances more accurate than J48. In term of complexity, J48 produce seventeen total numbers of nodes and nine numbers of leaves while REPTree produce eleven total numbers of nodes and six numbers of leaves. It shows REPTree produce small size of tree than J48. So, for Breast Tissue dataset, REPTree is better than J48 in term of accuracy of classification and complexity size of tree. For Iris dataset, J48 classify instances 0.6667% more accurate than REPTree but REPTree produce simpler tree structure than J48. Figure 1 shows tree structures for Iris dataset using different algorithms J48 and REPTree. These algorithms using same method for pruning tree that is REP but J48 allow attributes to be repeated in the process while REPTree allow only one repeat.

TABLE II. PERFORMANCE OF J48 AND REPTree (TREE)

Dataset	Algorithm	Correctly Classified (%)	Size of the Tree	No. of Leaves
Breast Tissue	J48	66.0377	17	9
	REPTree	72.6415	11	6
Iris	J48	94.6667	9	5
	REPTree	94.0000	5	3
Vertebral Column2c	J48	80.3226	7	4
	REPTree	80.3226	11	6
Vertebral Column3c	J48	80.3226	9	5
	REPTree	80.6452	17	9
Ecoli	J48	82.1429	11	6
	REPTree	42.5595	1	1
Balance Scale	J48	78.8800	47	24
	REPTree	76.4800	57	29
Wall	J48	99.4501	35	18
	REPTree	99.5784	33	17

In term of accuracy for Vertebral Column2c dataset, J48 and REPTree produce same performance 80.3226% but in term of complexity, J48 produce simpler tree structure than REPTree. For Vertebral Column3c dataset, REPTree classify instances 0.3226% more accurate than J48 while in term of complexity, size of tree by J48 is better than REPTree. For Ecoli dataset, J48 classify instances better than REPTree while size of tree by REPTree better than J48 because it produces smaller tree structure. For Balance Scale dataset, J48 is better than REPTree in term of accuracy of

978-1-4673-3141-8©2012 IEEE

classification and size of tree. And last for Wall dataset, REPTree is better than J48 because it produces high accuracy of classification and produces smaller tree structure. Average different performance of accuracy of classification between J48 and REPTree is 7.1006% while 6.2857% for complexity of tree structure.

TABLE III. SCORE OF ACCURACY (J48 AND REPTree)

Accuracy of classification								
Algorithm	**Score**							**Total win cases**
J48	0	1	0	0	1	1	0	3
REPTree	1	0	0	1	0	0	1	3

TABLE IV. SCORE OF COMPLEXITY (J48 AND REPTree)

Complexity of tree structure								
Algorithm	**Score**							**Total win cases**
J48	0	0	1	1	0	1	0	3
REPTree	1	1	0	0	1	0	1	4

Zero score represents a good algorithm and one score represents a better algorithm. From the comparison of performance in Table III, we can see J48 and REPTree have the same performance in term of accuracy of classification. Both achieved three scores from seven cases. For third case of accuracy both scored same result as zero, J48 and REPTree produce same percentage of accuracy of classification 80.3226% as we mentioned in Table II, so we represent this case with zero score due to it cannot give any effect in the comparison. From Table IV, REPTree is a better than J48 due to won four scores from seven cases were tested. It shows REPTree produce more simple tree structure than J48.

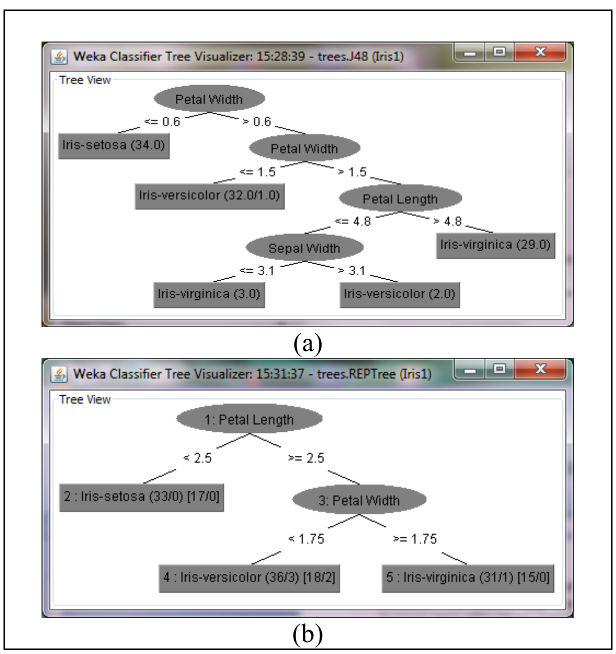

Figure 1. Tree structure for Iris using (a) J48 and (b) REPTree

Performance of J48 and REPTree in Table II shows the relationship between accuracy of classification, size of tree and number of leaves with different algorithms but same dataset. It shows the method in algorithms were fluent the accuracy of classification and complexity of tree structure. Each algorithm using reduced error pruning method but it has different parameters.

Table V shows performance of PART, Ridor and JRip algorithm classify instances of seven datasets. These decision tree algorithms produce rules. Small value of number of rules indicate these algorithms produce simple rules and easy to understand. For Breast Tissue, accuracy of classification for Ridor 66.0377%, it classifies instances more accurate than PART and JRip while Ridor and JRip produce more simple of rules. Comparison between these three algorithms for Breast Tissue, Ridor is the best performance in term of accuracy of classification and complexity. For accuracy of Iris, PART, Ridor and JRip produce same percentage, 94% but for complexity, number of rules Ridor is the best because generate 3 rules while PART 5 rules and JRip 4 rules. Figure 2 shows rules of three algorithms PART, Ridor and JRip for Iris.

TABLE V. PERFORMANCE OF PART, RIDOR AND JRIP (RULES)

Dataset	**Algorithm**	**Correctly Classified (%)**	**No. of Rules**
Breast Tissue	PART	65.0943	8
	Ridor	66.0377	7
	JRip	61.3208	7
Iris	PART	94.0000	5
	Ridor	94.0000	3
	JRip	94.0000	4
Vertebral Column2c	PART	82.9032	6
	Ridor	83.8710	4
	JRip	82.2581	3
Vertebral Column3c	PART	79.6774	5
	Ridor	78.0645	8
	JRip	82.5806	4
Ecoli	PART	82.1429	12
	Ridor	80.9524	34
	JRip	81.2500	8
Balance Scale	PART	79.2000	14
	Ridor	80.3200	12
	JRip	79.8400	14
Wall	PART	99.2302	10
	Ridor	99.6884	26
	JRip	98.8270	14

For Vertebral Column2c, Ridor produce highest percentage for accuracy than PART and JRip, different percentage between Ridor and PART 0.9678% and different percentage between Ridor and JRip 1.6129% but JRip produce more simple number of rules. For Vertebral Column3c, JRip is the best compare PART and Ridor in term of accuracy of classification and complexity.

For Ecoli, PART produces highest accuracy than Ridor and JRip but JRip produce more simple rules compare to PART 12 rules and Ridor 34 rules. For Balance Scale, 80.32% accuracy by Ridor is the best compare to PART and JRip while Ridor also produce more simple rules than PART and JRip. For Wall, in term of accuracy 99.6884% is the highest percentage by Ridor but PART in term of complexity is the best. Average differences percentage of accuracy of classification between Ridor and PART 0.8989% while Ridor and JRip 1.7835%. Average differences percentage of complexity between JRip and PART 2% while JRip and Ridor 6.5714%. Performance of PART, Ridor and JRip algorithms in Table V shows the algorithms works influenced to the accuracy of classification and number of rules. Each algorithm generates different performance even used the same dataset.

```
=== Classifier model (full training set) ===

PART decision list
------------------

Petal Width <= 0.6: Iris-setosa (33.0)

Petal Length <= 4.7: Iris-versicolor (32.0/1.0)

Petal Width > 1.7: Iris-virginica (29.0)

Petal Length <= 5: Iris-versicolor (3.0/1.0)

: Iris-virginica (3.0)

Number of Rules  :      5
```

(a)

```
=== Classifier model (full training set) ===

RIpple DOwn Rule Learner(Ridor) rules
--------------------------------------

Class = Iris-setosa  (150.0/100.0)
  Except (Petal Length > 2.45)
  => Class = Iris-versicolor (67.0/0.0) [33.0/0.0]
    Except (Petal Width > 1.75) and (Petal Length > 4.85)
    => Class = Iris-virginica (29.0/0.0)[14.0/0.0

Total number of rules (incl. the default rule): 3
```

(b)

```
=== Classifier model (full training set) ===

JRIP rules:
===========

(Petal Length <= 1.9) => Class=Iris-setosa (50.0/0.0)
(Petal Width >= 1.7) => Class=Iris-virginica (48.0/2.0)
(Petal Length >= 5) => Class=Iris-virginica (5.0/1.0)
 => Class=Iris-versicolor (47.0/0.0)

Number of Rules : 4
```

(c)

Figure 2. Rules for Iris using (a) PART, (b) Ridor and (c) JRip

TABLE VI. SCORE OF ACCURACY (PART, RIDOR AND JRIP)

Accuracy of classification								
Algorithm	**Score**							**Total win cases**
PART	2	**1**	2	2	**1**	3	2	2
Ridor	**1**	**1**	**1**	3	3	**1**	**1**	5
JRip	3	**1**	3	**1**	2	2	3	2

TABLE VII. SCORE OF COMPLEXITY (PART, RIDOR AND JRIP)

Complexity of rules								
Algorithm	**Score**							**Total win cases**
PART	2	3	3	2	2	2	**1**	1
Ridor	**1**	**1**	2	3	3	**1**	3	3
JRip	**1**	2	**1**	**1**	**1**	2	2	4

The scale to measure the performance of algorithms is from one to three where represents the best algorithm, better algorithm and good algorithm. Table VI shows Ridor is the best algorithm for accuracy of classification because the total win cases is five cases from seven cases while Table VII shows JRip is the best algorithm for complexity of tree structure due to win four cases from seven cases.

V. CONCLUSION

The aim of this study is to analyze performance of REP method in decision tree algorithms. We found that how pruning perform well and can influence the accuracy of classification and complexity of tree structure. An interesting question for future research is to proof that datasets also fluent the performance of algorithms. Overall, J48 and REPTree produce high accuracy of classification and simple tree structure. For algorithms based on rules, Ridor outperform best performance because in seven datasets it produces five times highest accuracy of classification while JRip gives best performance in term of complexity of tree structure.

The main thing to be considered while choosing classification algorithms is about high accuracy of classification, in order to describe that we need some acknowledgement about the circumstances of misclassification of instances that will significantly affected the quality of the algorithms. Furthermore about the complexity, this is the key thing of appropriateness to enhance the speed of algorithms and it involve the structure of tree either it simple or very complex what we need is the simple tree with high accuracy. If the algorithm produce simple tree and low accuracy it mean that algorithm compute high misclassification during instances learning process. To archive the best algorithm with simple tree and high accuracy we considered about pruning method and how it work inside the algorithm to make sure algorithm generate simple tree with good result and also avoid high percentage of misclassifications.

ACKNOWLEDGMENT

This research is supported by Graduate Research Incentive Grant (GIPS) Vote 0968 from Universiti Tun Hussein Onn Malaysia. And not to forget special thankful to the commentary and suggestion by Dr. Hj. Sazali Khalid and Saima Lashari which is improved this paper.

REFERENCES

[1] A. M. Mahmood, N. Satuluri, and M. R. Kuppa, "An Overview of Recent and Traditional Decision Tree Classifiers in Machine Learning.", International Journal of Research and Reviews in Ad Hoc Networks, Vol. 1, No.1, 2011.

[2] L. Rokach, and O. Maimon, "Data Mining With Decision Trees: Theory and Applications", Series in Machine Perception and Artificial Intelligence, World Scientific Publishing, Singapore, 2008.

[3] R. J. Roiger, and M. W. Geatz, "Data Mining A Tutorial-Based Primer", Addison Wesley, United State of America, 2003.

[4] B. Chandra, S. Mazumdar, V. Arena, and N. Parimi, "Elegant Decision Tree Algorithm for Classification in Data Mining," International Conference on Web Information Systems Engineering (Workshops), 2002.

[5] B. Chandra, and P. P. Varghese, "On Improving Efficiency of SLIQ Decision Tree Algorithm," International Joint Conference on Neural Networks, Orlando, Florida, USA, 2007.

[6] L. E. Raileanu, and K. Stoffel, "Theoretical Comparison Between The Gini Index and Information Gain Criteria," Annals of Mathematics and Artificial Intelligence, pages 77-93, 2004.

[7] L. Rokach, and O. Maimon, "Top-Down Induction of Decision Trees Classifiers – A Survey", IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, Vol. 35, No. 4, 2005.

[8] Z. Yong, C. Zhong-xian, and W. Da-gong, "Decision Tree's Pruning Algorithm Based on Deficient Data Sets," International Conference on Parallel and Distributed Computing, Application and Technologies, 2005.

[9] F. Berzal, J. C. Cubero, F. Cuenca, and M. J. Martin-Bautista, "On the Quest for Easy-To-Understand Splitting Rules," Dept. Computer Science and AI, University of Granada, Spain, 2002.

[10] J. Marques de Sa, R. Sebastiao, and J. Gama, "Tree Classifiers Based on Minimum Error Entropy Decisions," Canadian Journal on Artificial Intelligence, Machine Learning & Pattern Recognation, Vol. 2, No. 3, 2011.

[11] J. Su, and H. Zhang, "A Fast Decision Tree Learning Algorithm," America Association for Artificial Intelligence (www.aaai.org), 2006.

[12] L. A. Breslow, and D. W. Aha, "Simplifying Decision Trees: A Survey," Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, 1997.

[13] D. Lavanya, and K. U. Rani, "Performance Evaluation of Decision Tree Classifiers on Medical Datasets," International Journal of Computer Applications (0975-8887), Vol 26, No.4, 2011.

[14] J. Han, and M. Kamber, "Data Mining Concepts and Techniques". Morgan Kaufmann Publishers Inc., United States of America, 2001.

[15] S. Drazin, and M. Montag, "Decision Tree Analysis using Weka.", Machine Learning-Project II, University of Miami.

[16] A. M. Mahmood, P. Gudapati, V. G. Kavuluru, and M. R. Kuppa, "A New Pruning Approach For Better and Compact Decision Trees.", Vol.02, No.08, Pages 2551-2558, 2010.

[17] J. R. Quinlan, "C4.5:Programs for machine learning", Morgan Kaufmann Publishers Inc., California, 1993.

[18] M. N. Anyanwu, and S. G. Shiva, "Comparative Analysis of Serial Decision Tree Classification Algorithms", Vol (3) Issue (3), Pages 230-239, 2009.

[19] F. Esposito, D. Malerba, G. Semeraro, and V. Tamma, "The Effects of Pruning Methods on the Predictive Accuracy of Induced Decision Trees," Applied Stochastic Models in Business and Industry, Pages 277-299, 1999.

[20] X. Wu, and V. Kumar, "The Top Ten Algorithms in Data Mining," Data Mining and Knowledge Discovery Series, CRC Press, United States of America, 2009.

[21] J. Chen, X. Wang, and J. Zhai, "Pruning Decision Tree Using Genetic Algorithms," International Conference on Artificial Intelligence and Computational Intelligence, pp 244-248, 2009.

[22] I. H. Witten, and E. Frank, "Data Mining Practical Machine Learning Tools and Techniques," Second Edition, Morgan Kaufmann Publisher, United States of America, 2005.

[23] Y. Zhao and Y. Zhang, "Comparison of Decision Tree Methods for Finding Active Objects," National Astronomical Observatories, Advances of Space Research, 2007.

[24] E. Frank and I. H. Witten, "Generating Accurate Rule Sets Without Global Optimization," International Conference on Machine Learning, pages 144-151, 1998.

[25] B. R. Gaines and P. Compton, "Induction of Ripple-Down Rules Applied to Modeling Large Databases," J. Intell. Inf. Syst.. 5(3), pages 211-228, 1995.

[26] F. Leon, M. H. Zaharia and D. Galea, "Performance Analysis of Categorization Algorithms," International Symposium on Automatic Control and Computer Science, 2004.

[27] A. Frank, and A. Asuncion, UCI Machine Learning Repository, Retreived from http://archive.ics.uci.edu/ml.

[28] D. Jiawei, "Iterative Optimization of Rule Sets," Master Thesis, Technische Universitat Darmstadt. 2010.

[29] I. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes and S. J. Cunningham, "Weka: Practical Machine Leraning Tools and Techniques with Java Implementations," New Zealand, 1999.