International Journal of Engineering Research and General Science Volume 3, Issue 3, May-June, 2015
ISSN 2091-2730

Efficient Decision Tree Algorithm Using J48 and Reduced Error Pruning

Prerna Kapoor[1], Reena Rani[2]

[1]M.Tech Student, [2]Assistant Professor, Computer Science & Engineering Department

JMIT, Radaur/Kurukshetra University, India

[1]prernakapoor31@gmail.com
[2]reena@jmit.ac.in
9416296392

Abstract— Decision trees are few of the most extensively researched domains in Knowledge Discovery. Irrespective of such advantages as the ability to explain the choice procedure and low computational costs, decision trees also usually produce relatively great outcomes in assessment with other machine finding out formulas. Although the best decision tree induction algorithms, such as J48, had been developed some time ago, they continue to be regularly used for solving everyday classification tasks. In this work we aim to improve the predictive performance of these algorithms by mitigating three of their major disadvantages by Pruning Trees. Pruning decreases the complexity in the final classifier, and therefore improves predictive accuracy from the decrease of over fitting. This paper introduces a new decision tree algorithm based on J48 and reduced error pruning. Tree obtained is fast decision tree learning and will be based on the information gain or reducing the variance.

Keywords— Machine learning, Data mining, Decision trees, C4.5, J48, Tree Pruning, Reduced error pruning.

INTRODUCTION

Decision trees are utilized to delineate decision-making process. It is a classifier embodied by a flowchart like tree construction that has been extensively utilized to embody association models, due to its graspable nature that hold to mind the human reasoning. They are utilized to categorize instances by sorting them down the tree from origin to a little leaf node that runs the association of the instance. Every single node specifies an examination of the instance and every single division corresponds to one of the probable benefits for this attribute. Decision tree [1] builds classification or regression models in the form of tree structure. It divides a dataset into tinier and tinier subsets as at the alike period an associated decision tree is incrementally developed. The final consequence is a tree alongside decision nodes and leaf nodes. Decision node has two or extra divisions and leaf node embodies an association or decision. The top most decision node in a tree that corresponds to the best predictor shouted origin node.

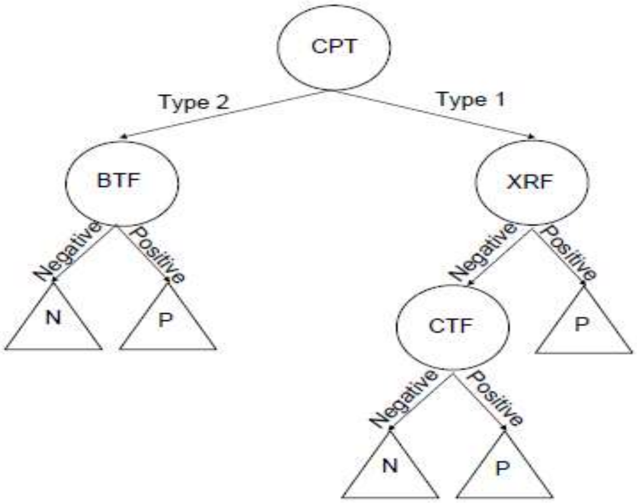

Figure 1. Decision Tree For Medical Applications

The above figure is an example of decision trees which illustrates the diagnosis process of patients that suffer from a certain respiratory problem. The decision tree employs the following attributes: CT finding (CTF); X-ray finding (XRF); chest pain type (CPT); and blood test finding (BTF). The physician will order an X-ray, if chest pain type is “1”. However, if chest pain type is “2”, then the physician will order a blood test. Thus medical tests are performed just when needed and the total cost of medical tests is reduced.

Decision Trees are one of the most extensively analyzed areas in machine learning. Aside from such gains as the skill to clarify the decision procedure and low computational prices, decision trees additionally normally produce moderately good aftermath in analogy alongside supplementary contraption discovering algorithms. Decision trees are best suited for the complications alongside pursuing characteristics:

- Instances are represented by attribute value pair. Example, for attribute "temperature", the value will be "hot or cold".
- The target function has discrete output variable.
- Disjunctive explanations may be necessary.
- The training data may contain errors.
- Training data may cover missing attribute values.

TREE PRUNING

When a decision tree is crafted, countless of the divisions imitate anomalies in the training data due to noise or outliers. Tree pruning [2] methods report this setback of above fitting the data. Such methods normally use statistical measures to remove the least reliable divisions there are two common approaches to tree pruning: pre-pruning and post-pruning. Key motivation of pruning is "trading accuracy for simplicity". There are assorted methods for pruning decision trees. Most of them present top down or bottom up traversal of the nodes. A node is pruned if the procedure improves precise conditions. . Pruning is a technique in device reading that reduces the dimensions of decision trees by detaching parts of the tree that provide little control to categorize instances.

A. Cost-Complexity Pruning

Cost intricacy pruning (also renowned as weakest link pruning or error intricacy pruning) takings in two stages. In the early period, sequences of trees are crafted on the training datasets, whereas the early tree beforehand pruning is the root tree. In the subsequent period, one of these trees is selected as the pruned tree, established on its generality of error estimation.

B. Pessimistic Pruning

Pessimistic pruning avoids the need of pruning set or cross validation and it uses the pessimistic statistical association test in its place. The basic idea is that the error ratio estimate during the training set is not consistent sufficiently. Instead a more practical measure known as "continuity correction" for binomial allocation should be used.

C. Reduced-Error Pruning

As traversing above the inner nodes from the bottom to the top of a tree, the REP procedure Checks for every single internal node, whether substituting it alongside the most recapped class that does not cut the accuracy of trees. In this case, the node is pruned. The procedure endures till each more pruning would cut the accuracy. In order to guesstimate the accuracy Quinlan provides to use a pruning set. It can be shown that this procedure ends alongside the smallest accurate sub- tree alongside respect to a given pruning set.

BENEFITS AND LIMITATION OF DECISION TREE

There are assorted benefits of decision trees are described as:

- Decision trees can generate understandable rules.
- Decision tree furnish a clear indication of that fields are most vital for forecast or classification.
- It performs association lacking far calculation.
- It can work on constant and categorical variables.

Some of the limitations of decision trees are given below:

- It is computationally luxurious as, at every single node every single candidate dividing earth have to be sorted beforehand its best tear can be found.
- Pruning algorithms can additionally be expensive as countless candidate sub trees have to be industrialized and compared.

- Entropy measurement is far larger.
- The decision tree algorithm produces a colossal size tree, that reduces the understandability.
- Reduced presentation after the Training Set is small. Tiny example sizes pose a main examination to decision trees, in particular because the number of obtainable training instances drops exponentially as the tree splits out
- Rigid decision criteria as decision at every single solitary node of the tree is rigid in the sense that merely one node can be selected.
- Deep decision trees encompassing of countless levels might additionally encompass countless at- tributes that leads to outlier attribute values.

RELATED WORK

Gilad Katz et al., 2014 [3] In this paper Decision trees have three main disadvantages: reduced performance when the training set is small; rigid decision criteria; and the fact that a single “uncharacteristic" attribute might “derail" the classification process. In this paper they present Conf D Tree (Confidence-Based Decision Tree) | a post-processing method that enables decision trees to better classify outlier instances. The experimental study indicates that the proposed post-processing method consistently and significantly improves the predictive performance of decision trees, particularly for small, imbalanced or multi-class datasets in which an average improvement of 5%»9% in the AUC performance is reported. In this paper, They presented and evaluated two variations of the same method for enhancing decision trees both for nominal and continuous attributes. The method, ConfDtree, can be used to deal with three important problems that affect decision trees: reduced performance when operating on small training sets; the rigidness of the classification process; and outlier attribute values that interfere with the correct classification of an instance. The method's ability to integrate with every type of decision tree algorithm is important. This makes it possible to select the most suitable algorithm for a specific dataset and still retain the benefits gained from using confidence intervals. The proposed algorithm has two drawbacks:

1) It slightly increases the computational cost of classifying a new instance; and
2) It reduces the comprehensibility of the model. In particular some instances are affected by their method and eventually assigned a different class distribution.

Leszek Rutkowski et al., 2013 [4] In this paper it is shown that the Hoeffding’s inequality is not appropriate to resolve the underlying problem. They clarify two theorems giving the McDiarmid’s attached for both the data gain, utilized in ID3 algorithm, and for Gini index, utilized in CART algorithm. The outcome of the paper promise that a decision tree discovering arrangement, requested to data streams and established on the McDiarmid’s attached, has the property that its output is nearly identical to that of a standard learner.

Mohammed Abdul Khaleel et al., 2013 [5] In this paper in the last decade there has been rising custom of data excavating methods on health data for discovering functional trends or outlines that are utilized in diagnosis and decision making. The main focus of this paper is to examine data excavating methods needed for health data excavating exceptionally to notice innately recurrent illnesses such as heart ailments, lung cancer, and breast cancer and so on. They assess the data excavating methods for discovering innately recurrent outlines in words of price, presentation, speed and accuracy. They additionally difference data excavating methods alongside standard methods.

Rodrigo Coelho Barros et al., 2011 [6] Decision tree induction is one of the most employed methods to extract knowledge from data, since the representation of knowledge is very intuitive and easily understandable by humans. The most successful strategy for inducing decision trees, the greedy top-down approach, has been continuously improved by researchers over the years. This work, following recent breakthroughs in the automatic design of machine learning algorithms, proposes two different approaches for automatically generating generic decision tree induction algorithms. Both approaches are based on the evolutionary algorithms paradigm, which improves solutions based on metaphors of biological processes. They also propose guidelines to design interesting fitness functions for these evolutionary algorithms, which take into account the requirements and needs of the end-user

PROPOSED WORK

J48 [7]is an open basis implementation of C4.5 algorithm . There two methods in pruning prop by J48 main are understood as sub tree substitute, it work by exchanging nodes in decision tree alongside leaf. J48 is an unpruned decision tree. The key point in assembly of decision tree is the choice of the best attribute to tear the believed node [8].

There are three prospects for the content of the set of training examples T in the given node of decision tree:

1. T contains one or more examples, all belonging to a only class Cj. The decision tree for T is a leaf identifying class Cj.

2. T contains no samples. The decision tree is once more a leaf, but the class to be associated alongside the leaf have to be ambitious from data supplementary than T, such as the finished bulk class in T. C4.5 algorithm uses as a criterion the most frequent class at the parent of the given node

3. T contains samples that belong to a blend of classes. In this situation, the idea is to refine T into subsets of samples that are heading towards single-class groups of samples. An appropriate test is preferred, based on single attribute, that has one or more mutually exclusive outcomesO_1,O_2, …,O_n}:

- T is separated into subsets T_1, T_2,……,T_n where Ti comprises all the samples in T that have outcome O_i of the prefered test. The decision tree for T consists of a decision node identifying the test and one branch for each possible outcome.
- Test – entropy: If *S* is any set of samples, let *freq* (C_i, *S*) stand for the number of samples in S that belong to class C_i (out of *k* possible classes), and $|S|$ denotes the number of samples in the set *S*. Formerly the entropy of the set *S*:

  $$\text{Info}(S) = - \sum_{i=1}^{k}((\text{freq}(C_i, S)/ |S|) \cdot \log_2 (\text{freq}(C_i, S)/ |S|))$$
- After set T has been split in accordance with n outcomes of one attribute test X:
  $\text{Info}_x(T) = \sum_{i=1}^{n}((|Ti|/ |T|) . \text{Info}(Ti))$
  $\text{Gain}(X) = \text{Info}(T) - \text{Info}_x(T)$
- Criterion: select an attribute alongside the highest Gain value.

Proposed Algorithm:

Input: C4.5 or J48 Decision Tree **T**

```
Procedure PostPruning(Data, TreeSplits)

SplitData(TreeSplits, Data, GrowingSet, PruningSet)

Estimate = DivideAndConquer(GrowingSet)

loop

    NewEstimate = Selection(Estimate,PruningSet)

    if Accuracy(NewEstimate, PruningSet) < Accuracy(Estimate,PruningSet)

        exit loop

    Estimate = NewEstimate

return(Estimate)
```

```
Procedure DivideAndConquer(Data)
Estimate = Ø
while Positive(Data) != Ø
        Leaves = Ø
        Instance = Data
while Negative(Instance) != Ø
        Leaves = Leaves ∪Find(Leaves, Instance)
        Instance = Instance(Leaves,Instance)
  Estimate = Estimate ∪ Leaves
  Data = Data - Instance
return(Estimate)
Output: REP TREE
```

The above algorithm states that, Firstly the training data are split into two subsets: a growing set and a pruning set. We use divide and conquer approach. The resulting Estimate is then repeatedly simplified by greedily deleting literals and rules from the Estimate until any further deletion would result in a decrease of predictive accuracy as measured on the pruning set.

Estimate with the highest accuracy on the pruning set is selected. This is repeated until the accuracy of the best pruned Estimate is below that of its predecessor. The output is the reduced error pruned tree.

Results

The J48 algorithm generates a Tree with 51 Nodes using supermarket dataset shown in fig. 2. Whereas tree generated by the proposed algorithm have 7 nodes shown in fig. 3.

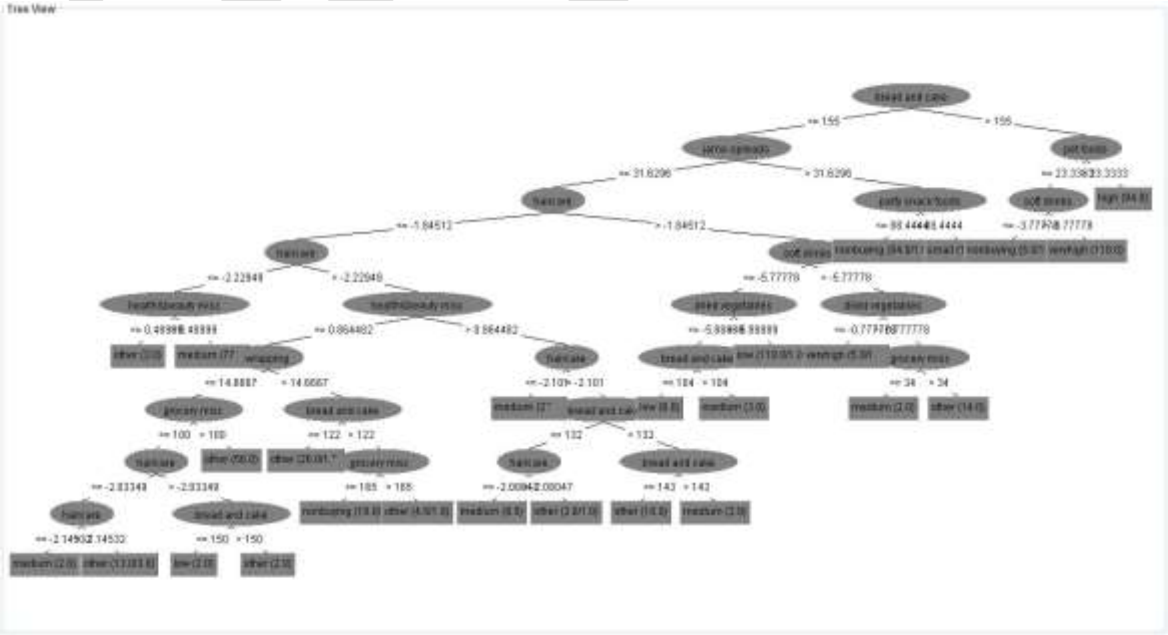

Figure 2. J48 Tree with 51 Nodes

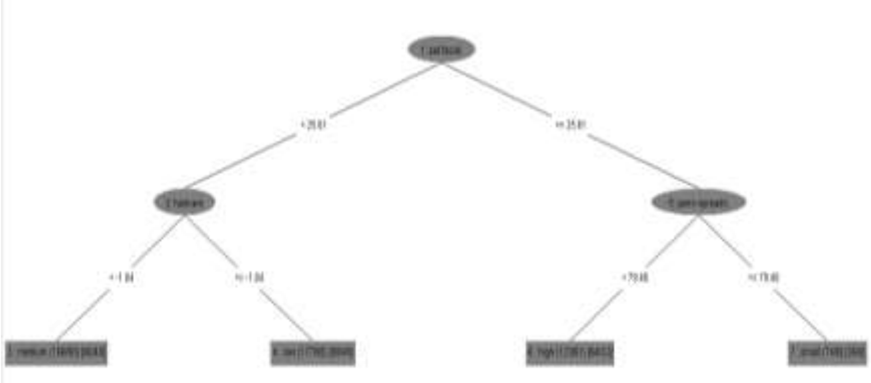

Figure 3.REPtree having tree size 7

Various performance parameters are used to compare the two algorithms as shown in table 1.

Parameter Name	REP Tree	J48
Mean absolute error	0.00583	0.01642
Root mean squared error	0.05399	0.09061
Relative absolute error	2.384	6.715
Root relative squared error	0.3496	25.91

Table 1. Parameter Analysis

These parameters can also be compared graphically as shown in the figure 4 and 5.

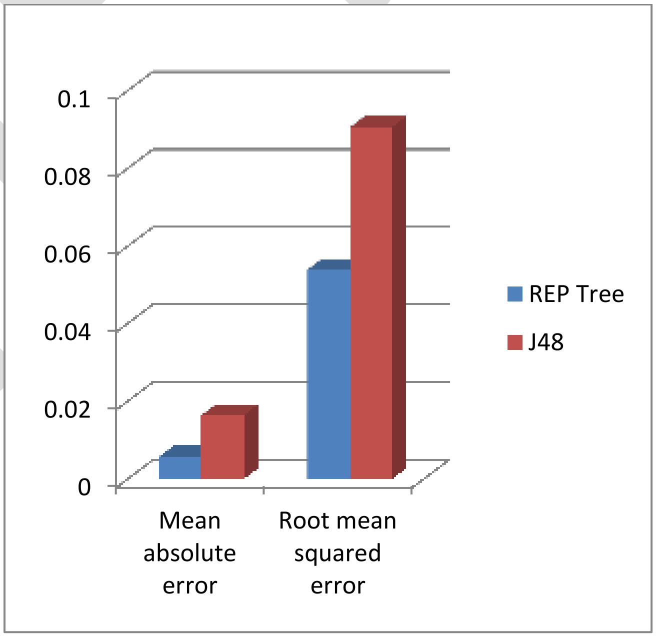

Figure 4. Comparisons of MAE and RMSE

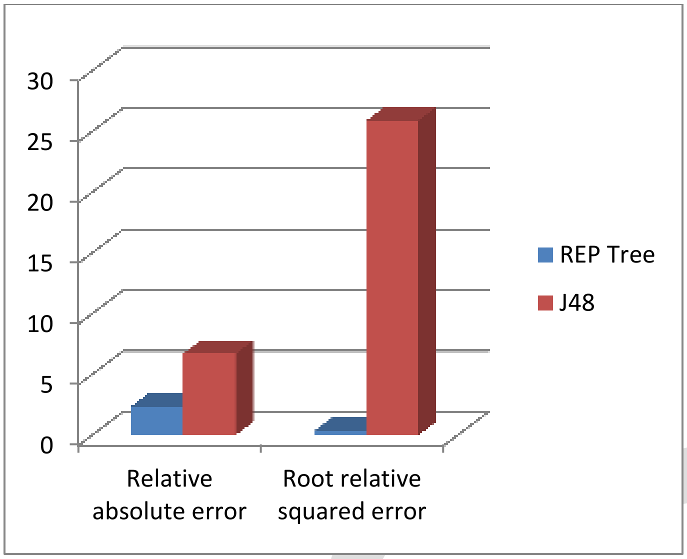

Figure 5. Comparisons of RAE and RRSE

According to above comparisons, Classification error is reduced. Entropy or Randomness measurement in J48 is much higher than REPTree as shown in fig 6. Sizes of the trees are shown in fig 7. Mean Entropy is showed in fig 8.

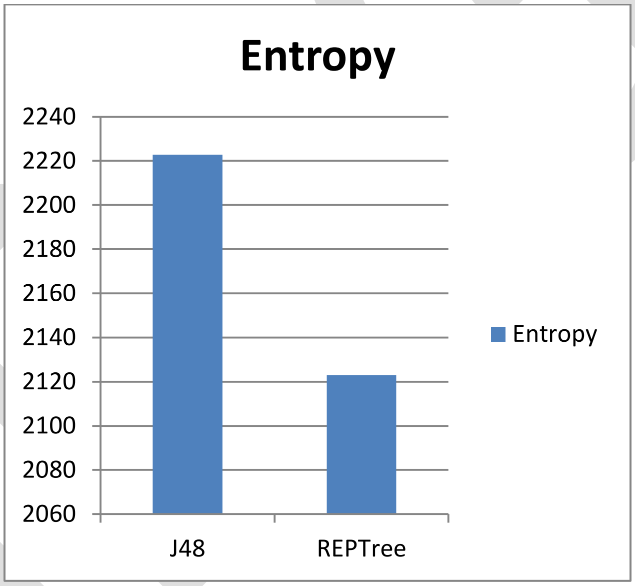

Figure 6. Entropy of the Decision Trees

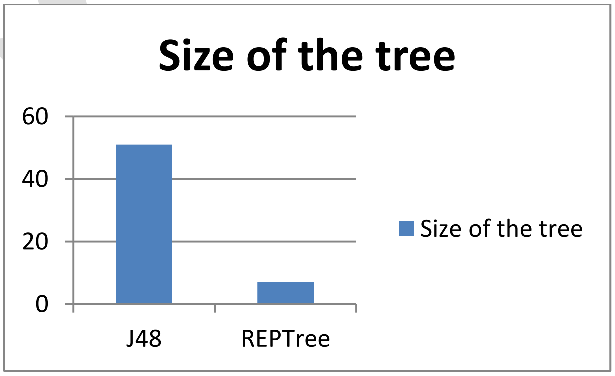

Figure 7. Size of Decision Trees

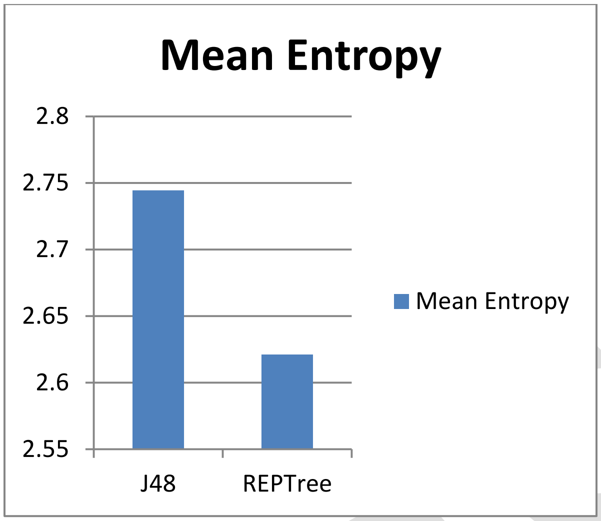

Figure 8. Mean Entropy of the Decision Tree

CONCLUSION

The proposed algorithm is compared alongside the algorithm J48 employing the WEKA. The analogy is completed above Supermarket dataset taken from online. The proposed algorithm cuts the Tree Size as well as the Finished Entropy Mean and Absolute. The relative absolute error and the root relative absolute error are also decreased. The decrease in error results in accurate classification. So the REPTree classify the items on the basis of their attributes more accurately. In the classified tree the Total Entropy of the Decision Tree, Randomness in J48 is much higher than REPTree. Also the size of the tree for the Supermarket Dataset is much less; clearly REPTree performs much better in contrast to J48.

In future we would like work on various issue of missing values. Currently our method is not applied beyond the point at which such values are detected. We can additionally work alongside the subject of imbalance. The counseled method by now performs larger on imbalanced datasets, but we should like to add supplementary improvements such as seizing the comparative imbalance into report after selecting alternative paths and certainty fines.

REFERENCES:

[1] Kohavi, Ronny, and J. Ross Quinlan. "Data mining tasks and methods: Classification: decision-tree discovery." In Handbook of data mining and knowledge discovery, pp. 267-276. Oxford University Press, Inc., 2002.

[2] W. Nor Haizan W. Mohamed, Mohd Najib Mohd Salleh, Abdul Halim Omar "A Comparative Study of Reduced Error Pruning Method in Decision Tree Algorithms" IEEE International Conference on Control System, Computing and Engineering, 23 - 25 Nov. 2012.

[3] Katz, Gilad, Asaf Shabtai, Lior Rokach, and Nir Ofek. "ConfDTree: A Statistical Method for Improving Decision Trees." Journal of Computer Science and Technology 29, no. 3 (2014): 392-407.

[4] Leszek Rutkowski, Lena Pietruczuk, Piotr Duda, and Maciej Jaworski. "Decision trees for mining data streams based on the McDiarmid's bound." Knowledge and Data Engineering, IEEE Transactions on 25, no. 6 (2013): 1272-1279.

[5] Mohammed Abdul Khaleel, Sateesh Kumar Pradham, and G. N. Dash. "A survey of data mining techniques on medical data for finding locally frequent diseases." Int. J. Adv. Res. Comput. Sci. Softw. Eng 3, no. 8 (2013).

[6] Rodrigo C. Barros, Márcio P. Basgalupp, André CPLF de Carvalho, and Alex A. Freitas. "Towards the automatic design of decision tree induction algorithms." In Proceedings of the 13th annual conference companion on Genetic and evolutionary computation, pp. 567-574. ACM, 2011.

[7] Neeraj, Bhargava, Sharma Girja, Dr Bhargava Ritu, and Mathuria Manisha. "Decision Tree Analysis on J48 Algorithm for Data Mining." International journal of advance research in computer science and software engineering 3 (2013)

[8] Tina R. Patil, and M. S. Sherekar. "Performance analysis of Naive Bayes and J48 classification algorithm for data classification." Int J Comput Sci Appl 6 (2013): 256-261.

[9] Lior Rokach and Oded Maimon. . "Top-down Induction of Decision Trees Classifiers-A Survey." IEEE Transaction on systems, man, and cybernetics—Part C: Applications and Reviews, VOL. 35, NO. 4, November 2005

[10] A.S. Galathiya, A. P. Ganatra, and C. K. Bhensdadia. "Improved Decision Tree Induction Algorithm with Feature Selection, Cross Validation, Model Complexity and Reduced Error Pruning." International Journal of Computer Science and Information Technologies 3, no. 2 (2012): 3427-3431.

[11] Smith Tsang, Ben Kao, Kevin Y. Yip, Wai-Shing Ho, and Sau Dan Lee. "Decision trees for uncertain data." Knowledge and Data Engineering, IEEE Transactions on 23, no. 1 (2011): 64-78.

[12] Anuja Priyama, Rahul Guptaa Abhijeeta, Anju Ratheeb, and Saurabh Srivastavab. "Comparative Analysis of Decision Tree Classification Algorithms." International Journal of Current Engineering and Technology 3, no. 2 (2013): 866-883