Volume 4, No. 4, March-April 2013
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Decision Trees For Training Data Sets Containing Numerical Attributes With Measurement Errors

C. Sudarsana Reddy*
Department of Computer Science and Engineering,
S.V. University College of Engineering, Tirupati.
cheruku1sudarsana2reddy3@gmail.com

Dr. V. VASU
Department of Mathematics,
S.V. University, Tirupati
vasuvaralakshmi@gmail.com

B. Kumara Swamy Achari
Department of Mathematics,
S.V. University, Tirupati
acharykumaraswamy44@gmail.com

Abstract: Classification is one of the most important techniques in data analysis. Decision tree is the most commonly used data classification technique. Training data sets are not error free due to measurement errors in data collection process. In general, values of numerical attributes in training data sets are always inherently associated with errors.

Measurement errors in training data sets can be properly handled by assuming an appropriate error correction model such as Gaussian error distribution. Data errors are corrected by fitting appropriate error correction model to the training data set. Different types of errors in the training data sets are not considered during the construction of existing decision tree classifiers. Hence, classification results of existing decision tree classifiers are less accurate or inaccurate in many cases because of different types of data errors present in the training data sets.

It is proposed to employ existing decision tree classifier construction algorithm using error corrected numerical attributes of the training data sets to construct new effective decision tree classifier. Errors in numerical attributes of the training data sets are corrected by using truncated Gaussian distribution. This new decision tree classifier construction algorithm is called error corrected decision tree classifier construction algorithm. It proves to be more effective regarding classification accuracy when compared with the existing decision tree classifier construction algorithm.

Computational complexity of error corrected decision tree classifier construction algorithm is approximately same as that of existing decision tree classifier construction algorithm but the classification accuracy of error corrected decision tree classifier construction algorithm is much more than the existing decision tree classifier construction algorithm.

Keywords: decision tree, error corrected values of the numerical attributes of the training data sets, training data sets containing numerical attributes, measurement errors in training data sets, types of errors in the training data sets, training data sets, classification, data mining, machine learning

I. INTRODUCTION

Data mining is the process of analyzing large data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Decision tree is the most popular classification method because it is more interpretable.

Data mining has many applications in research, science, engineering and business. Classification is an important technique in machine learning and data mining. Two most important features of decision tree are comprehensibility and interpretability [1]. When decision trees are used for classification they are called classification trees [2]. Decision tree learning algorithms can generate a decision tree model from a training data set [2]. Decision trees are popular because they learn and respond quickly, accurate in many domains, and they can handle high dimensional data [2].

Existing decision trees are constructed by using training data sets as it is without considering inherent errors present in the training data sets. But in real life many training data sets contain errors in the values of the attributes of the training datasets.

This paper introduces a new decision tree classifier construction algorithm that uses Gaussian error distribution model to correct errors in the numerical attributes of the training data sets. New decision tree describes an approach to correct errors in the values of the numerical attributes of the training data sets and then error corrected values of the numerical attributes of the training data sets are used in the process of new decision tree classifier construction. New decision tree classifier, based on error correction principle, is much more effective when compared with the existing decision tree classifier.

We introduce a new decision tree called error corrected decision (ECDT) classifier construction algorithm by using error corrected numerical attributes in the training data sets. In contrast to the existing decision tree (EDT) classifier, our approach, a new error corrected decision tree (ECDT) classifier construction algorithm, produces more accurate decision tree classifiers by modeling errors in the training data sets appropriately. No data set is available without error. That is, there are no error free data sets particularly when the training data sets contain numerical attributes.

Error corrected decision (ECDT) classifier construction algorithm uses error modeling technique based on the assumption that data sets are not always error free and it is likely that some sort of measurement errors are present in the collection process of data sets. Errors are inevitable in all

training data sets particularly in the training data sets with numerical attributes.

Measurement errors often have a distribution very close to normal. Measurement errors in physical experiments are often modeled by a normal distribution.

A. Types of Errors

Errors and their sources are a very important part of every scientific experiment. There are three types of errors that can occur within the experiment: personal error, systematic error, and random error.

In this paper we try to find and correct random errors by using truncated Gaussian distribution model. Random errors are also known as statistical uncertainties, and are a series of small, unknown, and uncontrollable events. Statistical uncertainties are much easier to assign, because there are rules for estimating the size. If you are reading a ruler, the statistical uncertainty is half of the smallest division on the ruler. Thus, if a ruler is marked to the nearest millimeter, the statistical uncertainty associated with any measurement made with that ruler is ± 0.5 millimeters. Even if you are recording a digital readout, the uncertainty is half of the smallest place given. This type of error should always be recorded for any measurement.

II. PROBLEM STATEMENT

Existing decision tree (EDT) classifiers are constructed by using training data sets directly without considering inherent data errors associated with values of numerical attributes in the training data sets. Hence, existing decision tree (EDT) classifiers produce incorrect or less accurate data mining results. As errors are associated with training data sets containing numerical attributes, it is important to develop more accurate data mining techniques by taking error corrected values of numerical attributes of the training data sets.

Also, for preserving data privacy sometimes training data sets are modified or injected certain error values into the values of attributes in the training data sets in a systematic or controlled way. So, in such cases data sets contain errors with modified attribute values. Such modified data sets must be reconstructed before applying them in any data mining technique. During data set reconstruction errors must corrected by using appropriate error correction model. In the existing decision tree classifier construction each tuple t_i is associated with a set of values of attributes of training data sets and the i^{th} tuple is represented as

$$t_i = (t_{i,1}, t_{i,2}, t_{i,3}, ...t_{i,k}, classlabel)$$

where

a. 'i' is the tuple number and 'k' is the number of attributes in the training data set.
b. $t_{i,1}$ is the value of the first attribute of the i^{th} tuple.
c. $t_{i,2}$ is the value of the second attribute of the i^{th} tuple and so on.

It is required to traverse the decision tree from the root node to a specific leaf node to find the class label of an unseen (new) test tuple.

$$t_{test} = (a_1, a_2, a_3, ...a_k, ?)$$

The present study proposes an algorithm to improve accuracy and performance of the existing decision tree (EDT) classifier algorithm. This new algorithm is called error corrected decision tree (ECDT) classifier construction algorithm. ECDT corrects or models errors in the numerical attribute values by using Gaussian error modeling technique or truncated Gaussian distribution and modifies values of numerical attributes in the training data sets accordingly before computing entropy.

III. EXISTING ALGORITHM

A. Existing Decision Tree (EDT) Algorithm Description:

Existing decision tree (EDT) classifier algorithm constructs a decision tree by splitting each node into left and right nodes. Initially, the root node contains all the training tuples. The process of partitioning the tuples in a node into two sets based on the value of an attribute and storing the resulting tuples in its left and right nodes is referred to as splitting. Whenever further split of a node is not required then it becomes a leaf node referred to as an external node. All other nodes except root node are referred as internal nodes. The splitting process at each internal node is carried out recursively until no further split is required. Further splitting of an internal node is stopped if one of the stopping criteria given hereunder is met.

a. All the tuples in an internal node have the same class label.
b. Splitting does not result nonempty left and right nodes.

In the first case, the probability for that class label is set to 1 whereas in the second case, the internal node becomes external node. The empirical probabilities are computed for all the class labels of that node.

Best attribute and best value of that attribute is called best split pair. Best split pair is associated with minimum entropy. In the existing decision tree classifier construction the goodness of split is quantified by an impurity measure. One possible function to measure impurity is entropy. Entropy is an information based measure and it is based only on proportions of tuples of each class label in training data set.

Entropy is predominantly used for constructing decision tree classifiers because in most of the cases entropy finds the best split and balanced node sizes after split so that both left and right nodes are as much pure as possible.

Entropy is computed for each value of each attribute in the training data set. Best split pair is selected corresponding to the minimum entropy among all the computed entropy values. Entropy is basically an information measure. Entropy is zero when there is no impurity in the data set; otherwise entropy value is any value between 0 and 1 inclusive.

Accuracy and execution time of existing decision tree (EDT) classifier algorithm for 9 data sets are shown in Table 5.2. Execution times of EDT are charted in Figure 5.1.

Entropy in the training data set is calculated using the formula

$$entropy(S) = \sum_{i=1}^{C} -p_i . log_2(p_i) \qquad 3.1$$

Where p_i = number of tuples belongs to the i^{th} class and C is the total number of distinct class labels in the training data set. Entropy after splitting the tuples into two groups is

$$H(A_j, z) = \sum_{X=L,R} \frac{|X|}{|S|}\left(\sum_{c\in C} -\frac{p_c}{X} log_2\left(\frac{p_c}{X}\right)\right)$$

$$H(A_j, z) = \frac{|L|}{|S|}\left(\sum_{c\in C} -\frac{p_c}{L} log_2\left(\frac{p_c}{L}\right)\right) + \frac{|R|}{|S|}\left(\sum_{c\in C} -\frac{p_c}{R} log_2\left(\frac{p_c}{R}\right)\right) \quad 3.2$$

$$H(A_j, z) = \frac{|L|}{|S|}\big(Entropy(L)\big) + \frac{|R|}{|S|}\big(Entropy(R)\big)$$

Where

a. A_j is the splitting attribute and z is the split point and (A_j, z) is the best split pair.
b. L is the total number of tuples to the left side of the split point z.
c. R is the total number of tuples to the right side of the split point z.
d. $\frac{p_c}{L}$ is the number of tuples belongs to the class label c to the left side of the split point z.
e. $\frac{p_c}{R}$ is the number of tuples belongs to the class label c to the right side of the split point z.
f. S is the total number of tuples in the node.

B. Pseudo code for Existing Decision Tree (EDT) Algorithm:

```
Existing_Decision_Tree (T)
a.  If all the training tuples in the node T have the
b.   same class label then
c.      set p_T(c) = 1.0
d.      return(T)
e.  End if
f.  If tuples in the node T have more than one
g.    Class label then
h.    Find_Best_Split(T)
i.  End if
j.  For i ← 1 to datasize[T] do
k.    If split_atribute_value[t_i] <= split_point[T] then
l.        Add tuple t_i to left[T]
m.  Else
n.          Add tuple t_i to right[T]
o.  End if
p.  End for
q.  If left[T] = NIL or right[T] = NIL then
r.      Create empirical probability distribution of the
s.        node T
t.      return(T)
u.  End if
v.  If left[T] != NIL and right[T] != NIL then
w.      Existing_Decision_Tree ( left[T] )
x.      Existing_Decision_Tree ( right[T] )
y.    return(T)
z.  end if
```

In the existing decision tree (EDT) classifier construction entropy values are computed for f(g-1) split points where f is the number of attributes and g is the total number of tuples at the current node.

IV. PROPOSED ALGORITHAM

A. Proposed Error Corrected Decision Tree (ECDT) Algorithm Description:

The new decision tree called error corrected decision tree (ECDT) classifier construction algorithm works exactly similar to the existing decision tree (EDT) classifier construction algorithm but main difference is that the error corrected decision tree (ECDT) classifier construction algorithm computes entropy values for error corrected values of numerical attributes of training data sets, whereas the existing decision tree (EDT) classifier construction algorithm computes entropy values directly for the values of numerical attributes of the training data sets without considering data errors in the values of the numerical attributes of training data sets.

The error corrected decision tree (ECDT) classifier algorithm constructs a decisions tree classifier by splitting each node into left and right nodes. Initially, the root node contains all the training data tuples. Entropy values are computed for f(g–1) split points where f is the number attributes of the training data set, g is the number of training data tuples at the current node T. The process of partitioning the training data tuples in a node into two subsets based on the best split point value z_T of best split attribute A_{j_T} and storing the resulting tuples in its left and right nodes is referred to as splitting.

Whenever further split of a node is not required then it becomes a leaf node referred to as an external node. All other nodes except root rode are referred as internal nodes. The splitting process at each internal node is carried out recursively until no further split is required. Further splitting of an internal node is stopped if one of the stop splitting criteria given hereunder is met.

a. All the tuples in an internal node have the same class label
b. Splitting does not result nonempty left and right nodes

In the first case, the probability for that class label is set to 1.0 whereas in the second case, the internal node becomes external node. The empirical probabilities are computed for all the class labels of that node.

The best split pair comprising an attribute and its value is that associated with minimum entropy. Best split point value of the best split attribute is used to partition the tuples of the internal node into two parts.

Entropy is a metric or function that is used to find the degree of dispersion of training data tuples in a node. In decision tree construction the goodness of a split is quantified by an impurity measure. One possible function to measure impurity is entropy. Entropy is an information based measure and it is based only on the proportions of tuples of each class in the training data set. Entropy is used for finding how much information content is there in a given data.

Entropy is taken as dispersion measure because it is predominantly used for constructing decision trees. In most of the cases, entropy finds the best split and balanced node sizes after split in such a way that both left and right nodes are as much pure as possible.

Entropy is computed for each value of each attribute in the training data set. Best split pair is selected corresponding to the minimum entropy among all the computed entropy values. Entropy is basically an information measure.

Entropy is zero when there is no impurity in the data otherwise entropy value may be any value between 0 and 1 inclusive.

For each value of each numerical attribute Gaussian error corrected value is generated taking given attribute value as a mean and the difference of upper and lower bounds of an interval around the mean divided by 4 or 3 as a standard deviation.

Let x is numerical value of an attribute. An interval around x is constructed by using the range of the attribute. Now (x – d, x + d) is the interval around x and x is the mean and interval length divided by 4 or 3 is taken as the standard deviation of the Gaussian distribution. Here error in the x value is corrected by Gaussian distribution and then entropy is calculated for that error corrected value.

Accuracies and execution times of error corrected decision tree (ECDT) algorithm for 9 data sets are shown in Table 5.3. Execution times of error corrected (ECDT) algorithm for 9 data sets are charted in the figure 5.2. Comparison of execution times and accuracies for EDT and ECDT algorithms for 9 data sets are shown in Table 5.4 and charted in Figure 5.3 and Figure 5.4 respectively.

Entropy of the training data set is calculated using the formula

$$entropy(S) = \sum_{i=1}^{C} -p_i . log_2(p_i) \qquad 4.1$$

Where p_i = number of tuples belongs to the i^{th} class and C is the total number of distinct class labels in the training data set. Entropy after splitting the tuples into two groups is

$$H(A_j, z) = \sum_{X=L,R} \frac{|X|}{|S|}\left(\sum_{c\in C} -\frac{p_c}{X} log_2\left(\frac{p_c}{X}\right)\right)$$

$$H(A_j, z) = \frac{|L|}{|S|}\left(\sum_{c\in C} -\frac{p_c}{L} log_2\left(\frac{p_c}{L}\right)\right) + \frac{|R|}{|S|}\left(\sum_{c\in C} -\frac{p_c}{R} log_2\left(\frac{p_c}{R}\right)\right) \qquad 4.2$$

$$H(A_j, z) = \frac{|L|}{|S|}\big(Entropy(L)\big) + \frac{|R|}{|S|}\big(Entropy(R)\big)$$

Where

a. A_j is the splitting attribute and z is the the split point and (A_j, z) is the best split pair.
b. L is the total number of tuples to the left side of the split point z.
c. R is the total number of tuples to the right side of the split point z.
d. $\frac{p_c}{L}$ is the number of tuples belongs to the class label c to the left side of the split point z.
e. $\frac{p_c}{R}$ is the number of tuples belongs to the class label c to the right side of the split point z.
f. S is the total number of tuples in the node.

B. Pseudo code for Error Corrected Decision Tree (ECDT) Algorithm:

Error_Corrected_Decision_Tree(T)

a. If all the training tuples in the node T have
b. the same class label then
c. set $p_T(c) = 1.0$
d. return(T)
e. End if
f. If tuples in the node T have more than one class label
g. then
h. compute entropy for all the error corrected values of
i. all the attributes of the training data set
j. End if
k. Find_Best_Split(T)
l. For i ← 1 to datasize[T] do
m. If split_atribute_value[t_i] <= split_point[T] then
n. Add tuple t_i to left[T]
o. Else
p. Add tuple t_i to right[T]
q. End if
r. End for
s. If left[T] = NIL or right[T] = NIL then
t. create empirical probability distribution of the node T
u. return(T)
v. End if
w. If left[T] != NIL and right[T] != NIL then
x. Error_Corrected_Decision_Tree(left[T])
y. Error_Corrected_Decision_Tree(right[T])
z. return(T)
aa. end if;

In error corrected decision tree (ECDT) classifier construction entropy values are computed for f(g-1) split points where f is the number of attributes and g is the total number of tuples at current node.

Computational complexity of new decision tree classifier construction algorithm is f(g-1), which is approximately similar to the computational complexity of the existing decision tree classifier construction algorithm but the classification accuracy of the error corrected decision tree (ECDT) classifier is much more than the classification accuracy of the existing decision tree (EDT) classifier algorithm.

V. EXPERIMENTS ON EFFICIENCY

A simulation model is developed for evaluating the performance of two algorithms: existing decision tree (EDT) classifier and error corrected decision tree (ECDT) classifier experimentally. The data sets shown in Table 5.1 from University of California (UCI) Machine Learning Repository are employed for evaluating the performance of the above said algorithms.

Table 5.1 Data Sets from the UCI Machine Learning Repository

S No.	Data Set	Training Tuples	No. of Attributes	No of Classes	Test Tuples
1	Iris	150	4	3	10-fold
2	Glass	214	9	6	10-fold
3	IonoSphere	351	32	2	10-fold
4	BreastCancer	569	30	2	10-fold
5	Vehicle	846	18	4	10-fold
6	Segment	2310	14	7	10-fold
7	Satellite	4435	36	6	2000
8	PageBlock	5473	10	5	10-fold
9	PenDigits	7494	16	10	3498

10-fold cross-validation technique is used for test tuples for all the training data sets with numerical attributes except Satellite and PenDigits training data sets. For Satellite and PenDigits training data sets with numerical attributes a separate test data set is used for testing.

The simulation model is implemented in Java 1.6 on a Personal Computer with 3.22 GHz Pentium Dual Core processor (CPU), and 2 GB of main memory (RAM). The performance measures, accuracy and execution time, for the above said algorithms are presented in Table 5.2 to Table 5.5 and Figure 5.1 to Figure 5.4.

Table 5.2 Accuracy and Execution Time of EDT Algorithm for 9 Data sets

Data Set Name	Total Training Tuples	Correctly Classified Tuples	Accuracy on Training data	k-fold Cross Validation	Execution Time
Iris	150	147	98	97.3333	0.1
Glass	214	183	85.51402	92.381	0.172
Iono Sphere	351	292	83.1908	83.1429	0.281
Breast Cancer	569	559	98.2425	97.3214	0.75
Vehicle	846	673	79.5508	79.0476	3.532
Segment	2310	2255	97.619048	96.5801	24.563
Satellite	4435 (test 2000)	1665	83.25	Not applied	128.359
Page Block	5473	5472	99.98173	98.2369	28.969
PenDigits	7494 (test 3498)	3215	91.91	Not applied	622.765

Table 5.3 Accuracy and Execution Time of ECDT Algorithm for 9 Data sets

Data Set Name	Total Training Tuples	Correctly Classified Tuples	Accuracy on Training data	k-fold Cross Validation	Execution Time
Iris	150	149	99.3333	98.6666	0.1
Glass	214	211	98.5981	94.2857	0.281
Iono Sphere	351	351	100	98.28571	1.75
Breast Cancer	569	569	100	96.964286	2.781
Vehicle	846	843	99.6454	97.50	5.953
Segment	2310	2282	98.7878	97.5325	51.891
Satellite	4435 (test 2000)	1687	84.40	Not applied	479.32
Page Block	5473	5424	99.1047	99.012	383.079
PenDigits	7494 (test 3498)	3217	92.0	Not applied	902.3

Table 5.4 Comparison of Execution Times and accuracies for EDT and ECDT Algorithms for 9 Data Sets

Data Set Name	EDT Execution Time (seconds)	ECDT Execution Time (seconds)	EDT Accuracy	ECDT Accuracy
Iris	0.1	0.1	97.3333	98.6666
Glass	0.172	0.281	92.381	94.2857
Iono Sphere	0.281	1.75	83.1429	98.28571
Breast Cancer	0.75	2.781	97.3214	96.964286
Vehicle	3.532	5.953	79.0476	97.50
Segment	24.563	51.891	96.5801	97.5325
Satellite	128.359	479.32	83.25	84.40
Page Block	28.969	383.079	98.2369	99.012
PenDigits	622.765	902.3	91.91	92.0

Table 5.5 Comparisons of Accuracies for EDT, and ECDT Algorithms

Data Set Name	EDT Accuracy	ECDT Accuracy
Iris	97.3333	**98.6666**
Glass	92.381	**94.2857**
Iono Sphere	83.1429	**98.28571**
Breast Cancer	**97.3214**	96.964286
Vehicle	79.0476	**97.50**
Segment	96.5801	**97.5325**
Satellite	83.25	**84.40**
Page Block	98.2369	**99.012**
PenDigits	91.91	**92.0**

Bold values show highest classification accuracies.

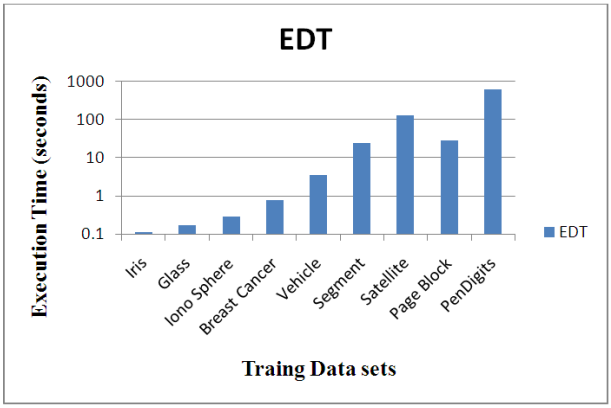

Figure 5.1 Execution Times for EDT Algorithm for 9 Data Sets.

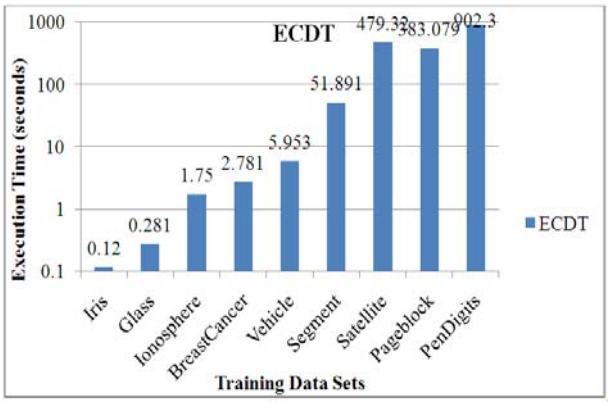

Figure 5.2 Execution Times for ECDT Algorithm for 9 Data Sets.

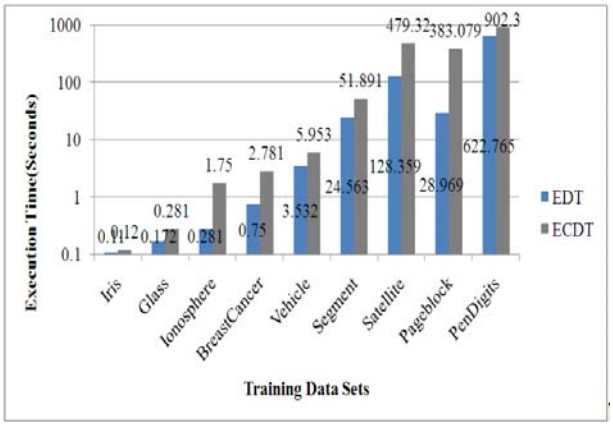

Figure 5.3 Comparisons of Execution Times for EDT and ECDT Algorithms for 9 Data Sets

Figure 5.4 Comparisons of Accuracies for EDT and ECDT Algorithms for 9 Data Sets

Accuracy Comparisons of existing decision tree (EDT) and error corrected decision tree (ECDT) algorithms are shown in Table 5.5. From the Table 5.5 it is clear that for most of the datasets accuracies obtained by Error Corrected Decision Tree (ECDT) algorithm are more accurate than existing decision tree (EDT).

Execution time of error corrected decision tree (ECDT) classifier is slightly more than the execution times of existing decision tree (EDT) classifier but ECDT classification accuracies are far better than EDT.

VI. CONCLUSIONS

A. Contributions:

The performance of existing decision tree (EDT) classifier algorithm is verified experimentally. A new algorithm, Error Corrected Decision Tree (ECDT) classifier, is proposed and compared with existing decision tree (EDT). It is found that the classification accuracy of ECDT algorithm is better than EDT algorithm.

B. Limitations:

For large training data sets construction of decision tree classifiers is less efficient and less scalable. Execution time of new decision tree classifier, called Error Corrected Decision Tree (ECDT) classifier, is slightly more than the execution time of existing decision tree classifier (EDT) algorithm. Some privacy preserving techniques cause reduced utility of classification results.

C. Suggestions for future work:

Special techniques are needed to decrease the execution time and space complexity of ECDT. Special error correcting techniques are needed to find and correct different types of errors inherently present in the values of numerical attributes of the training data sets.

Special privacy preserving techniques are needed to maintain training data sets without loss of utility and accuracy when privacy preserving techniques are applied to training data sets.

Also effective, efficient, and simple techniques are needed to reconstruct the modified training datasets before applying data mining techniques.

VII. REFERENCES

[1]. Jiawei Han, Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2006. Page Numbers: 285, 288, 292

[2]. Introduction to Machine Learning Ethem Alpaydin PHI MIT Press. Page Numbers: 185,187, 188

[3]. U.M. Fayyad and K.B. Irani, "On the Handling of Continuous –Valued Attributes in Decision tree Generation", Machine Learning, vol. 8, pp. 87-102, 1996.

[4]. R.E.Walpole and R.H. Myers, Probability and Statistics For Engineers and Scientists. Macmillan Publishing Company, 1993.

[5]. A. Asuncion and D. Newman, UCI Machine Learning Repository, http:/ / www. ics. uci. edu/ mlearn/ MLRepository.html, 2007.