Evaluation of Decision Tree Pruning Algorithms for Complexity and Classification Accuracy

Article *in* International Journal of Computer Applications · December 2010

DOI: 10.5120/1554-2074 · Source: DOAJ

3 authors, including:

Dipti Durgesh Patil
Cummins College of Engineering for Women

Evaluation of Decision Tree Pruning Algorithms for Complexity and Classification Accuracy

Dipti D. Patil
Assistant Professor,
MITCOE, Pune, INDIA

V.M. Wadhai
Professor and Dean of Research,
MITSOT, MAE, Pune, INDIA

J.A. Gokhale
Professor
VESIT, Mumbai INDIA

ABSTRACT
Classification is an important problem in data mining. Given a database of records, each with a class label, a classifier generates a concise and meaningful description for each class that can be used to classify subsequent records. A number of popular classifiers construct decision trees to generate class models. These classifiers first build a decision tree and then prune subtrees from the decision tree in a subsequent pruning phase to improve accuracy and prevent “overfitting”. In this paper, the different pruning methodologies available & their various features are discussed. Also the effectiveness of pruning is evaluated in terms of complexity and classification accuracy by applying C4.5 decision tree classification algorithm on Credit Card Database with pruning and without pruning. Instead of classifying the transactions either fraud or non-fraud the transactions are classified in four risk levels which is an innovative concept.

Keywords
Decision tree classification, Pruning, Data Mining

1. INTRODUCTION
Classification is an important problem in data mining. It has been studied extensively by the machine learning community as a possible solution to the knowledge acquisition or knowledge extraction problem. The input to a classifier is a training set of records, each of which is tagged with a class label. A set of attribute values defines each record. Attributes with discrete domains are referred to as categorical, while those with ordered domains are referred to as numeric. The goal is to induce a model or description for each class in terms of the attributes. The model is then used to classify future records whose classes are unknown.
Data classification [1] is a two step process as shown in figure 1. In the first step, a model is built describing a predetermined set of data classes or concepts. The model is constructed by analyzing database tuples described by attributes. Each tuple is assumed to belong to a predefined class, as determined by one of the attributes, called the class label attribute. The data tuples analyzed to build the model collectively form the training data. This process is called as machine learning [2, 3]. Since the class label of each training sample is provided, this step is also known as supervised learning. It contrasts with unsupervised learning (or clustering), in which the class label of each training sample is not known, and the number or set of classes to be learned may not be known in advance. Typically, learn model is represented in the form of classification rules, decision rules, or mathematical formulae. The rules can be used to categorize future data samples, as well as provide a better understanding of the database contents. In the second step, the model is used for classification. First, the predictive accuracy of the model is estimated. If the accuracy of the model is considered acceptable, the model can be used to classify future data tuples or objects for which the class label is not known. For example in figure 1 given a database of customers credit information, classification rules can be learned to identify customers as having either excellent or fair credit ratings. With this analysis of data from existing customers can be used to predict the credit rating of new or future customer.
When decision tree is formed with decision tree classification algorithm sometimes it happens that it generates some unwanted & meaningless rules as it grows deeper, it is called as overfitting [4]. This can be avoided by only considering those attributes which will have big contribution in forming the particular rule.
This is done by stopping the growth of decision tree at particular level so that the rules formed give better classification. There are two types of Pruning methods, first is pre-pruning [4, 5], i.e. while building the decision tree keep on checking whether tree is over fitting based on different measures like Laplace error [4], MDL [6] length, cost etc and second method is post pruning, in which the tree is built first & then reduction of branches & levels of decision tree is done. In this paper we have discussed various decision tree pruning methodologies. Also the effectiveness of pruning is evaluated by applying C4.5 algorithm [7] with and without pruning on credit card database. As human behavior is unpredictable classifying any transaction either as fraud or non-fraud is not acceptable. In all of the previous studies [8], [9], [10] the transactions were classified in only two levels either fraud or legitimate. Our approach classifies the credit card transactions in various fraud levels depending on different fraudulent situations mined from the historical behavior of the customers. By considering risk involved in the transactions, banks can take necessary preventive actions and provide services to customers accordingly. So the classifier developed by us does multi level classification of transactions rather than just binary classification. So here onwards the C4.5 algorithm with pruning is referred as **M**ulti-**L**evel **P**runed **C**lassifier (**MLPC**) algorithm. Section-2 discusses various pruning techniques. Section 3 gives implementation details of C4.5 & MLPC. Section 4 evaluates results with both the algorithms. Section 5 represents concluding remarks.

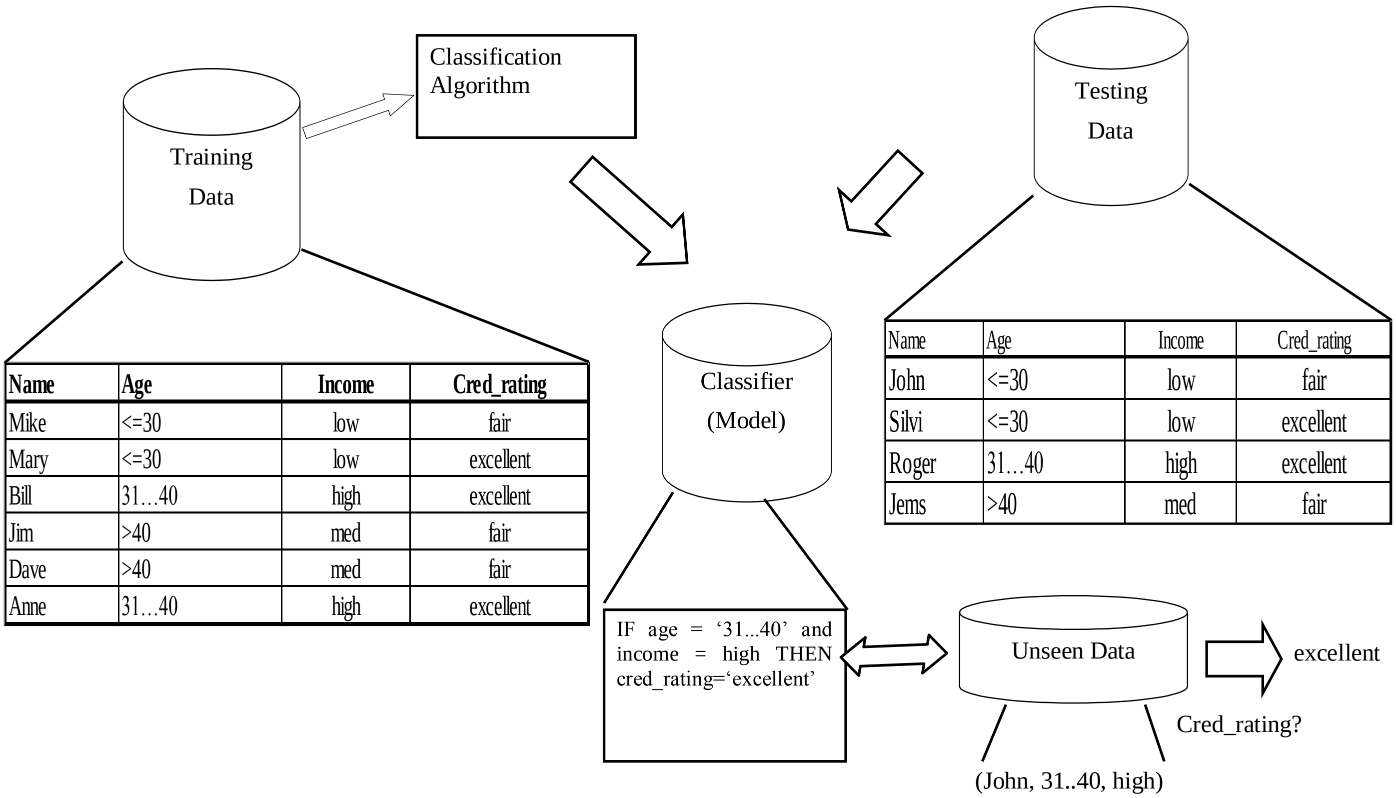

Figure 1: Decision tree Learning and Classification Process

2. PRUNING TECHNIQUES

Although the decision trees generated by methods such as ID3 and C4.5 are accurate and efficient, they often suffer the disadvantage of providing very large trees that make them incomprehensible to experts [11]. To solve this problem, researchers in the field have considerable interest in tree pruning. Tree pruning methods convert a large tree into a small tree, making it easier to understand. Such methods typically use statistical measures to remove the least reliable branches, generally resulting in faster classification and an improvement in the ability of the tree to correctly classify independent test data". It is necessary to know the advantage and disadvantage of every decision tree pruning method before it is decided that which pruning method will be selected. The following are some main methods to simplify decision trees.

2.1 Reduced Error Pruning

This method was proposed by Quinlan [11]. It is the simplest and most understandable method in decision tree pruning. For every non-leaf subtree *S* of the original decision tree, the change in misclassification over the test set is examined. The misclassification would occur if this subtree were replaced by the best possible leaf which is the majority of leaf. If the error rate of the new tree would be equal to or smaller than that of the original tree and that subtree *S* contains no subtree with the same property, *S* is replaced by the leaf. Otherwise, stop the process. The constraint that the subtree *S* contains no subtree with the same property guarantees reduced error pruning in bottom-up induction [12]. Since each node is visited only once to evaluate the opportunity of pruning it, the advantage of this method is its linear computational complexity [12]. However, this method requires a test set separate from the cases in the training set from which the tree was constructed [11]. When the test set is much smaller than the training set, this method may lead to over pruning. Many researchers found that Reduced Error Pruning performed as well as most of the other pruning methods in terms of accuracy and better than most in terms of tree size [12].

2.2 Pessimistic Error Pruning

This method was also proposed by Quinlan [11] and was developed in the context of ID3. Quinlan found that it is too optimistic for us to use a training set to test the error rate of a decision tree, because decision trees have been tailored to the training set. In this case, the error rate can be 0. But some data other than the training set is used; the error rate will increase dramatically. To solve this problem, Quinlan used continuity correction for the binomial distribution to get an error rate which is more realistic. In statistics, continuity correction is a useful method in the application of the normal distribution to the computation of binomial probabilities. When the normal distribution (a continuous distribution) is used to find approximate answers to problems arising from the binomial distributions (discrete distribution), an adjustment is made for the mismatch of types of distribution. This is called the continuity correction." Quinlan uses the following equations to obtain the number of misclassifications:

$$n'(t) = e(t) + (1/2) \qquad \ldots\ldots\ldots\ldots(1)$$

$$n'(T_t) = e(T_t) + (N_T/2) \qquad \ldots\ldots\ldots\ldots(2)$$

Equation(1) is the number of misclassifications for node *t* and equation(2) is the number of misclassifications for subtree *T*.

where:

NT is the number of leaves for subtree *T*,
e(*t*) is the number of misclassifications at node *t*,
e(*Tt*) is the number of misclassifications for subtree *T*.
The 1/2 in the equation (1) and (2) is a constant which indicates the contribution of a leaf to the complexity of the tree.

This pruning method only keeps the subtree if its corrected figure (from equation 2) is more than one standard error better than the figure for the node (from equation 1). This method is much faster than Reduced Error Pruning and also provides higher accuracies. Its disadvantage is that, in the worst case, when the tree does not need pruning at all, each node will still be visited once [12].

2.3 Cost-Complexity Pruning

This method was proposed by Breiman et al., [13]. It takes account of both the number of errors and the complexity of the tree. The size of the tree is used to represent the complexity of the tree. It is also known as the CART pruning method and Floriana Esposito, et al, describes it in two steps [12]:

1) Selection of a parametric family of subtrees of { *T*0*; T*1*;.....; TL }*, according to some heuristics. *T*0 is the original decision tree and each *Ti*+1 is obtained by replacing one or more subtrees of *Ti* with leaves by pruning those branches that show the lowest increase in apparent error rate per pruned leaf until the final tree *TL* is just a leaf.
2) Choice of the best tree according to an estimate of the true error rates of the trees in the parametric family.

For example, consider subtree *T* used to classify each of the *N* cases in the training set and *E* of *N* examples are wrongly classified if subtree *T* is replaced by the best leaf. Let N_T be the number of leaves in subtree *T*, the following equation is used to define the total cost-complexity of subtree *T*:

$$\text{Cost-complexity} = (E/N) + \alpha * N_T \qquad \ldots\ldots\ldots. (3)$$

where α is the cost of one extra leaf in the tree and gives the reduction in error per leaf.

If the subtree is pruned, the new tree would misclassify *M* more of the cases in the training set but would contain N_T -1 fewer leaves. The same cost-complexity will be obtained when

$$\alpha = (M / (N^* (N_T - 1)) \qquad \ldots\ldots\ldots \qquad (4)$$

From the above equation, α can be calculated for each subtree and the subtree(s) with the smallest value of α is selected for pruning. Continue to process this until the leaf is obtained. The next job is to select one of the trees. The standard error (*SE*) of the misclassification rate is

$$SE = (R * (100 - R)) / N \qquad \ldots\ldots\ldots \qquad (5)$$

where:

R = misclassification rate of the pruned tree,
N = number of examples in the test data.

The smallest tree whose observed number of errors on the test set does not exceed *R* + *SE* is selected.

This method requires a pruning set distinct from the original training set. Its disadvantage is that it can only choose a tree in the set {*T*0*; T*1,….. *TL}*, which is obtained in the first step, instead of the set of all possible subtrees [12]. It also seems anomalous that the cost-complexity model used to generate the sequence of subtrees is abandoned when the best tree is selected [11].

2.4 Minimum Error Pruning

This method was developed by Niblett and Brotko [14]. It is a bottom-up approach which seeks a single tree that minimizes the expected error rate on an independent data set. Assume that there are *k* classes for a set of data which number is *n* and *nc* is the class *c* with the greatest number of data. If it is predicted that all future examples will be in class *c*, the following equation is used to predict the expected error rate:

$$E_k = (n- n_c + k -1) / (n+k) \qquad \ldots\ldots\ldots(6)$$

where:

k is the number of classes for all data,
Ek is the expected error rate if we predict that all future examples will be in class *c*.

The method consists of three steps [15]:

1) At each non-leaf node in the decision tree, use equation (6) to calculate the expected error rate if that subtree is pruned.
2) Calculate the expected error rate if the node is not pruned, combined by weighting according to the proportion of observations along each branch [12][14].
3) If pruning the node leads to a greater expected error rate, then keep the subtree; otherwise, prune it.

J. Mingers [16] points out that there are several disadvantages in this method. First, it is seldom true in practice that all the classes are equally likely. Second, this method produces only a single tree. This is a disadvantage in the context of expert systems, where it will be more helpful if several trees, pruned to different degrees, are available. Third, the number of classes strongly affects the degree of pruning, leading to unstable results. Minimum error pruning was improved by Cestnik and Bratko [14] and the most recent version of minimum error pruning overcomes two problems of original method:

Optimistic bias and dependence of the expected error rate on the number of classes.

2.5 Critical Value Pruning

This method was proposed by Mingers [17]. In this method, a threshold, named the critical value, is set to estimate the importance or strength of a node. When the node does not reach the critical value, it will be pruned. But when a node meets the pruning condition but its children do not all meet the pruning condition, this branch should be kept because it contains relevant nodes. If a larger critical value is selected, a smaller resulting tree will be obtained because of the more drastic pruning.

Mingers describes the critical value pruning as two main steps [17]:

1) Prune subtree for increasing critical values,
2) Measure the significance of the pruned trees as a whole and their predictive ability and choose the best tree among them.

The disadvantage of this method is its strong tendency to under prune and this method selects trees with comparatively low predictive accuracy [18].

2.6 Optimal Pruning

Breiman et al., introduce a convenient terminology used to state and verify the mathematical properties of optimal pruning [13]. They also introduce an algorithm to select a particular optimally pruned subtree from among the *k* candidates [13].

Bratko and Bohanec [19] and Almuallim [20] address the issue of finding optimal pruning in another way. Bohanes et al., [19] introduced an algorithm guaranteeing optimal pruning (OPT), and Almuallim [20] further improved OPT in terms of the computational complexity. Their motivation for simplifying decision trees is different from the typical motivation for pruning decision trees when learning from noisy data. Both [19] and [20] assume that the initial, unpruned decision trees are completely correct. However, in learning from noisy data, which is our case, it is assumed that the initial, unpruned decision tree is inaccurate and appropriate pruning would improve its accuracy.

2.7 Cost-Sensitive Decision Tree Pruning

One main problem for many decision tree pruning methods is that when a decision tree is pruned, it is always assumed that all the classes are equally probable and equally important. However, in real-world classification problems, there is also a cost associated with misclassifying examples from each class. Currently, the most common method for cost-sensitive pruning method is to use techniques in statistics to deal with the problem. The use of probability models and statistical methods for analyzing data has become common practice in virtually all scientific disciplines. For example, M. Jordan used a statistical approach to build a decision tree model [21]. A parameter can be estimated from sample data either by a single number (a point estimate) or an entire interval of plausible values (a confidence interval). Frequently, however, the objective of an investigation is not to estimate a parameter but to decide which of two contradictory claims about the parameter is correct (some cost-sensitive pruning method makes use of this [22]). Methods for accomplishing this comprise the part of statistical inference called hypothesis testing. The null hypothesis, denoted by H_0, is the claim about one or more population characteristics that is initially assumed to be true. The alternative hypothesis, denoted by Ha, is the assertion that is contradictory to H_0. The null hypothesis will be rejected in favor of the alternative hypothesis only if sample evidence suggests that H_0 is false. If the sample does not strongly contradict H_0, it will continue to believe in the truth of the null hypothesis.

3. IMPLEMENTATION

This section discusses the C4.5 and MLPC decision tree induction algorithm which are applied on credit card database.

3.1 Credit Card Database

The credit card database used for training and classification is developed based on the snapshot of the credit card database given by the bank. For security purpose the bank did not allow us to reveal the real data, due to this the database was manually preprocessed from the given information and the overall survey of the credit card world. The credit card transaction table built for learning contains 101580 records.

The transaction table is built based on the current transaction information such as amount, transaction time, transaction location, expiry date entered, card limit, in addition to that some historical information is also combined with these fields like average purchase of previous three months, average purchase of previous twelve months, customer's preferred transaction location and time, limit of number of transactions within a day to trace the customer's normal behavior. The transaction record does not contain customer account number because instead of learning, behavior models of individual customer accounts, overall models that try to differentiate legitimate transactions from fraudulent ones is built. So the model is customer-independent.

3.2 Types of Fraud

Instead of classifying the given transactions in only two types that is either fraud or non-fraud, in the system implemented transaction gets classified in four different types of class levels (L1, L2, L3, L4) which are decided based on different fraudulent situations traced out from given snapshot of database by bank and survey done on credit card world. The fraudulent situations based on which class levels have been assigned to the transactions.

3.3 Decision Tree Induction Algorithm

MLPC algorithm is implemented with pre-pruning where while constructing the tree growth of the tree is stopped at the set pruned level. The algorithm considers some base cases which are listed below:

Base Cases:

- All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.
- None of the features provide any information gain. In this case, algorithm creates a decision node higher up the tree using the expected value of the class.
- Instance of previously-unseen class encountered. Again, algorithm creates a decision node higher up the tree using the expected value.

MLPC Algorithm

a) Construct the tree in a top-down recursive divide-and -conquer manner.
b) In the beginning, keep all the training examples at the root.
c) Partition examples recursively based on selected attributes.
d) Select the splitting attribute on the basis of entropy measure.
e) Repeat all the steps until one of the three conditions get satisfied:
    i. All samples for a given node belong to the same class.
    ii. There are no remaining attributes for further partitioning.
    iii. There are no samples left.
    iv. Set prune level is reached.

Entropy Measure

Entropy measure is given by following equation. For a set of record S,

$$\textbf{Entropy } \mathbf{E(S) = -\Sigma\, p_j \log p_j} \qquad \text{…………………….. (7)}$$

Where, j= 1, 2,…., m

p_j is the relative frequency of class j in S

Entropy divides S with n records in two sets, S1 with n1 records and S2 with n2 records.

$$E(S1, S2) = \frac{n_1}{n} E(S1) + \frac{n_2}{n} E(S2) \quad \text{……………… (8)}$$

In the context of decision trees, if the outcome of a node is to classify the records into two classes, C1 and C2, the outcome can be viewed as message that is being generated and the entropy gives the measure of information for a message to be C1 or C2. If a set of records T is partitioned into a set of disjoint exhaustive classes C1,C2,…., Cn on the basis of a value of the class attribute, then the information needed to identify the class of an element of T is

Info (T) = Entropy (P) .... (9)

Where, P is probability distribution of the partition C1, C2, …, Cn.

P is computed based on their relative frequencies, that is,

P = ((|C1|/|T|, |C2|/|T| , …|Cn|/|T|) .... (10)

The goal is to lower the Entropy.

3.4 Classification Algorithm

There are two phases in decision tree classification, first is to generate the decision tree from the given training data and second is actual classification where decision rules of formed decision tree is applied to the transaction having unknown class label to classify it in one of the classes. The algorithm for this classification is given below:

1. For each transaction to be classified, read one by one the decision rule from the Decision table.
2. Match the fields from the transaction with each decision rule. (Fields having blank entries in decision table indicate don't care condition).
3. First try to find out perfect match and fill the Class field of the transaction with the class of matched rule.
4. If perfect match is not found then among matched rules the rule having highest risk level is chosen and the class field of the transaction is filled with that class of matched rule.

4. RESULTS

4.1 Data Set

The data used in this paper is real world data which is provided by a nationalized bank. As the data contains sensitive information, the database cannot be revealed as per the agreement with the bank. Around 1 lac credit card transactions are used based on the different fraudulent cases. The transactions are then divided into different test sets.

The classifier is trained with different transaction sets and used for the classification of each of these sets. For comparison purpose basic C4.5 algorithm and MLPC algorithm are used for training. As classes of these transactions are already known, the classification accuracy is evaluated by comparing the classified transactions with the original class value of the transactions. Classification measures used for results evaluation are True Positive Rate (TPR), False Positive Rate (FPR), False Negative Rate (FNR) and Accuracy [2], [15]. C4.5 is a basic decision tree classification algorithm. Credit card fraud detection system is one of the applications of it which has been developed. The application is useful for inter-banking where banks can share their fraud detecting rules with each other to overcome the threat of fraud which is spreading widely in world of credit cards.

Classifier measures for Accuracy Evaluation

The accuracy of the classifier in case of credit card fraud detector is evaluated based on the following measures [5]:

True positive rate (TPR)

$$\text{TPR} = \frac{\text{Total number of samples correctly classified as fraud}}{\text{Actual number of fraud samples}}$$

False Positive Rate (FPR)

$$\text{FPR} = \frac{\text{Total number of samples incorrectly classified as fraud}}{\text{Total number of samples}}$$

False Negative Rate (FNR)

$$\text{FNR} = \frac{\text{(Total number of samples incorrectly classified as Legitimate)}}{\text{Total number of samples}}$$

Accuracy

$$\text{Accuracy} = \frac{\text{Total no. of samples correctly classified}}{\text{Total no. of samples}}$$

4.2 Classification Results at Different Prune Levels

From the whole transaction set, some transactions are taken for training and part of it are taken for testing purpose and then this procedure is repeated for the whole transaction database.

The results are evaluated with both the training algorithms C4.5 and MLPC on all the combination of sets, but due to space constraints only some of the results are listed and compared. The specifications of transaction set are given in table1.

Pre-Pruning method is implemented in the developed system, where the tree growing is stopped at particular level to prevent forming meaningless rule or more specific rules. Results are evaluated by truncating the tree growing at different levels and accuracy at each pruned level is compared. Table 2 shows comparison of classification accuracy with different prune levels for 4 risk levels (L1, L2, L3, L4) as class value with Set1 as Test file & Main Set as Training file.

Table2 shows evaluation of accuracy at each prune level. Results shows that when prune level is low, means considering small number of attributes for classification, accuracy of class type L1 is high as most of the transactions get classified as non-fraudulent causing low accuracy of other class type representing fraudulent levels. Whereas as prune level increases considering more attributes for classification the accuracy of class type L1 slightly lowers but accuracy of other class types improves. Figure 2 depicts the same result graphically.

Table 1: Specification of each transaction set

Test Set Name	No. Of Transactions
Test1	10000
Test2	20000
Test3	30000
Test4	40000
Test5	10000
Test6	20000
Test7	30000
Test8	40000
Set 1	50780
Set 2	50780
Main Set	101560

Table2: Class Type vise accuracy evaluation at each Prune level

Prune Level	L1	L2	L3	L4
1	88.84	70.51	79.12	75.79
2	87.65	70.51	96.03	75.79
3	87.51	75.32	96.03	75.79
4	85.38	80.6	96.03	75.79
5	88.69	72.53	82.35	75.79
6	88.43	81.2	82.35	75.79
7	88.43	82.1	82.35	75.79
8	86.6	82.32	80.73	75.79
9	87.57	82.33	79.12	75.79
10	86.74	82.33	79.12	75.79
11	86.74	82.32	79.12	75.79

Table 3 gives overall comparison of accuracy, TPR, FPR & FNR for different prune levels.

Table 3: Prune Level vise Overall accuracy, FPR, TPR & FNR

Prune Level	FPR	FNR	TPR	Accuracy
1	11.16	19.98	80.02	79.24
2	12.35	15.45	84.55	81.72
3	12.49	13.51	86.49	82.9
4	14.62	11.53	88.47	83.55
5	11.31	18.41	81.59	80.25
6	11.57	14.78	85.22	82.38
7	11.57	14.38	85.62	82.61
8	13.4	14.23	85.77	81.79
9	12.43	14.23	85.77	81.84
10	13.26	14.23	85.77	81.57
11	13.26	14.23	85.77	81.57

Results show that FPR provided by middle levels are good, though lowest level gives lowest FPR it doesn't give better overall accuracy. Also the middle prune levels give better TPR than other levels.

4.3 Comparison of decision table size

As C4.5 algorithm grows the decision tree for all combinations of each attribute, the level of the decision tree generated is high, which in terms generates large number of decision rules whereas the MLPC considers only attributes which have highest contribution in classification so the number of rules generated are less. This helps in distributed data mining where data of a company is distributed over different sites & instead of passing the data they share the rules generated from the database. Here if the decision table size is high it will require higher network bandwidth and transmission time will also be high. The results in table 4 show the same that the number of rules generated by MLPC algorithm is 10 times lesser than C4.5 algorithm.

Table 4: Comparison of size of decision table

Training File	No. Of Decision rules generated	
	C4.5	MLPC
Test1	134	15
Test2	131	12
Test3	237	16
Test4	238	15
Test5	96	15
Test6	95	15
Test7	156	18
Test8	156	15
Set1	237	15
Set2	156	15
Main Set	207	16

4.4 Comparison of Accuracy and Classification measures with Main_set as training file

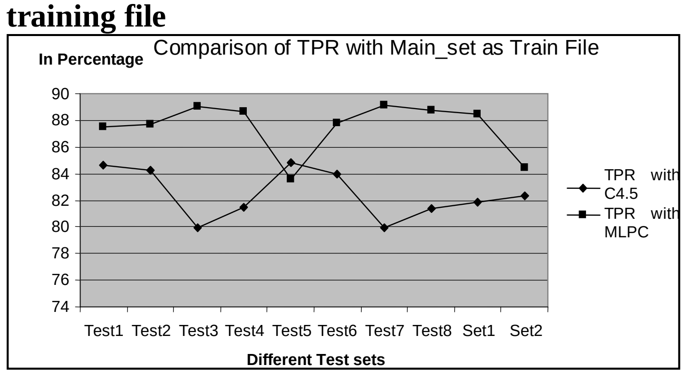

Figure 2: Comparison of True Positive Rate (TPR)

Figure 2 shows that MLPC gives average 87% of TPR whereas C4.5 gives 82% of TPR. Figure 3 gives comparative results of False Positive Rate with C4.5 and MLPC algorithm.
Results show that MLPC gives lower FPR around average 12% than C4.5 which gives average FPR of 30%.

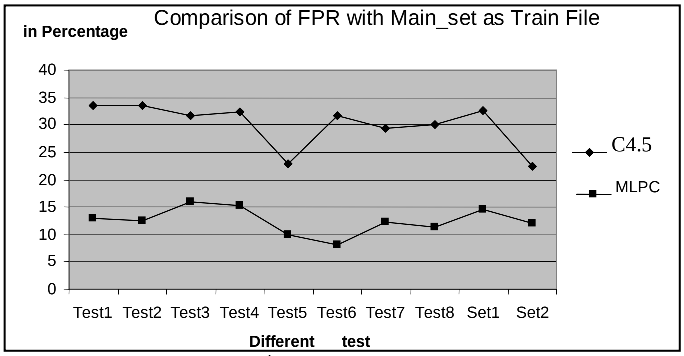

Figure 3: Comparison of False Positive Rate

Figure 4 represents results of overall accuracy evaluation. Overall accuracy of each transaction getting classified to the correct class level is very important for fulfilling the objective of the system. Considering the main set as base classifier and classifying different data sets MLPC gives highest average accuracy of 80% which is much better than C4.5 giving overall accuracy of 62%.

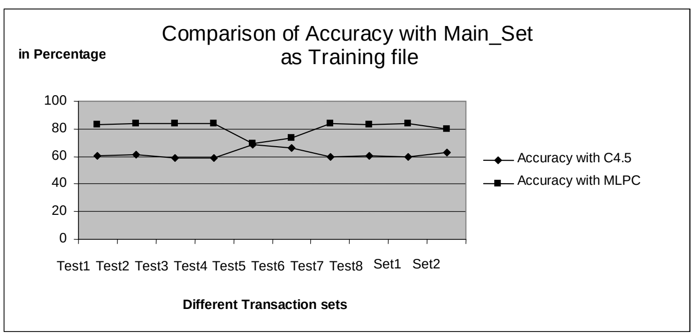

Figure 4: Comparison of Overall Accuracy

5. CONCLUSIONS

In this paper, MLPC algorithm that integrates the pruning phase into the building phase is proposed. At reaching to set prune level nodes are not expanded during the building phase. As a result, fewer nodes are expanded during the building phase, and thus the complexity of constructing the decision tree is reduced.

Number of decision rules generated by MLPC is much lesser than rules generated by C4.5 algorithm as listed in table 4. Many times rules generated by C4.5 are redundant and meaningless. In MLPC algorithm, rules are lesser which directly affects the size of the decision table. This directly decreases the time complexity of classification. As size of the decision table generated with MLPC algorithm is small, the required network bandwidth while transferring the decision table also reduces. Thus Pruning technique proves effective efficient and scalable in decision tree induction.

The performance based on Accuracy and True Positive Rate is compared between basic C4.5 algorithm and newly developed MLPC algorithm. MLPC gives on average 80% accuracy whereas C4.5 algorithm gives on an average 62% accuracy. Thus pruning algorithm is effective from classification accuracy perspective.

Fraud catching rate (TPR) of both the classifiers is 85% as per figure 2. False Alarm rate (FPR) of MLPC is 12% and C4.5 gives False alarm rate of 30% depicted in figure 3. MLPC algorithm is decision tree learning algorithm with pre-pruning. Observation shows that at level 4 it gives highest accuracy for different transaction sets of the application. But then also there is always an 'optimum' pruning level for different applications & requirements that one has to identify and select.

Credit card fraud detection system is one of the applications of MLPC which has been developed. In contrast to previously developed credit card fraud detection systems where transactions were getting classified in only two levels either fraud or non-fraud, the system developed can differentiate among different fraudulent situations and classify transactions in four fraud risk levels.

Our ongoing work is focused on incorporating and testing various pruning strategies discussed in paper in the proposed algorithm. Aim is to develop cost effective pruning algorithm on which not much work done so far.

6. REFERENCES

[1] Jiawei Han, Micheline Kamber, “Data Mining Concepts and Techniques”, pp. 279-328, 2001.

[2] Doina Caragea, “Learning classifiers from distributed, semantically heterogeneous, autonomous data sources”, Thesis, Iowa State University, 2004.

[3] Tom. M. Mitchell, “Machine Learning”, McGraw-Hill Publications, 1997

[4] Zhang Yong, “Decision Tree’s Pruning Algorithm Based on Deficient Data Sets”, In Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies, 2005.

[5] Arun Poojari, “Data Mining techniques”, pp 150 -200, 1999

[6] Manish Mehta, Rakesh Agrawal et al:, “SLIQ- A Fast Scalable Classifier for Data Mining.”, In 5th Intl. Conf. on Extending Database Technology, March 1996

[7] J. Quinlan. C4.5 Programs for Machine Learning, San Mateo, CA:Morgan Kaufmann, 1993.

[8] Salvatore, Philip et al:, “Meta learning agents for fraud and intrusion detection in Financial Information Systems.”, Inv paper Proceedings in International conference of Knowledge Discovery and Data mining, 1996.

[9] S. Stolfo et al., “JAM: Java Agents for Metalearning over Distributed Databases,” Proc. Third Int’l Conf. Knowledge Discovery and Data Mining, AAAI Press, Menlo Park,Calif., 1997, pp. 74–81.

[10] Salvatore J. Stolfo, David W. Fan, Wenke Lee and Andreas L. Prodromidis, “Credit Card Fraud Detection Using Meta-Learning: Issues and Initial Results”, DARPA, 1999.

[11] J. Quinlan. Simplifying decision trees, Int. J. Human-Computer Studies, (1999)51, pp. 497-491, 1999.

[12] F. Esposito, D. Malerba, and G. Semeraro. A comparative Analysis of Methods for Pruning Decision Trees, IEEE transactions on pattern analysis and machine intelligence, 19(5): pp. 476-491, 1997.

[13] L. Briemann, J. Friedman, R. Olshen and C. Stone. CART: Classification and Regression Trees, EBelmont, s CA:Wadsworth Statistical Press, 1984.

[14] B. Cestnik, and I. Bratko. On Estimating Probabilities in Tree Pruning, EWSL, pp. 138-150, 1991

[15] Esposito F., Malerba D., Semeraro G. A Comparative Analysis of Methods for Pruning Decision Trees, IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 19, NO. 5, 1997, P. 476-491

[16] J. Mingers. An Empirical Comparison of Selection Measures for Decision Tree Induction, Machine Learning, 3 (3): pp. 319-342, 1989

[17] J. Mingers. An Empirical Comparison of Pruning Methods for Decision Tree Induction, Machine Learning, 4: pp. 227-243, 1989

[18] L. A. Breslow and D. W. Aha. Simplifying Decision Trees: A Survey, Technical Report No. AIC-96-014, Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory Washington, DC.,1996.

[19] I. Bratko and M. Bohanec. Trading accuracy for simplicity in decision trees, Mach. Learn., vol. 15, pp. 223-250, 1994.

[20] H. Almuallim. An efficient algorithm for optimal pruning of decision trees, Artif. Intell., vol. 83, no. 2, pp. 347-362, 1996.

[21] M. I. Jordan. A Statistical Approach to Decision Tree Modeling, Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory, New York: ACM Press, 1994.

[22] B. Andrew, L. Brain C., Cost-Sensitive Decision Tree Pruning Use of the ROC Curve, Proceedings Eight Australian Joint Conference on Artificial Intelligence, pp.1-8, November 1995.