Comparative Analysis of Serial Decision Tree Classification Algorithms

Matthew N. Anyanwu manyanwu @memphis.edu
Department of Computer Science
The University of Memphis,
Memphis, TN 38152, U.S.A

Sajjan G. Shiva sshiva @memphis.edu
Department of Computer Science
The University of Memphis,
Memphis, TN 38152, U.S.A

Abstract

Classification of data objects based on a predefined knowledge of the objects is a data mining and knowledge management technique used in grouping similar data objects together. It can be defined as supervised learning algorithms as it assigns class labels to data objects based on the relationship between the data items with a pre-defined class label. Classification algorithms have a wide range of applications like churn pre-diction, fraud detection, artificial intelligence, and credit card rating etc. Also there are many classification algorithms available in literature but decision trees is the most commonly used because of its ease of implementation and easier to understand compared to other classification algorithms. Decision Tree classification algorithm can be implemented in a serial or parallel fashion based on the volume of data, memory space available on the computer resource and scalability of the algorithm. In this paper we will review the serial implementations of the decision tree algorithms, identify those that are commonly used. We will also use experimental analysis based on sample data records (Statlog data sets) to evaluate the performance of the commonly used serial decision tree algorithms.

Keywords: Decision tree, Classification Algorithm, data mining, SPRINT, SLIQ, CART, C4.5, IDE3

1. INTRODUCTION

Classification can be described as a supervised learning algorithm in the machine learning process. It assigns class labels to data objects based on prior knowledge of class which the data records belong. It is a data mining techniqueago, has made it possible to co-design and co-develop software and hardware, and hence, such components [1]. However, integration of an that deals with knowledge extraction from database records and prediction of class label from unknown data set of records (Tan et al, 2006). In classification a given set of data records is divided into training and test data sets. The training data set is used in building the classification model, while the test data record is used in validating the model. The model is then used to classify and predict new set of data records that is different from both the training and test data sets (Garofalakis et al, 2000 and Gehrke et al, 1998). Supervised learning algorithm (like classification) is preferred to unsupervised learning algorithm (like clustering) because its prior knowledge of the class labels of data records makes feature/attribute selection easy and this leads to good prediction/classification accuracy. Some of the common classification algorithms used in data mining and decision support systems are: neural networks (Lippmann, 1987), logistic regression (Khoshgoftaar et al, 1999), Decision trees (Quinlan, 1993) etc. Among these classification algorithms decision tree algorithms is the most commonly used because of it is easy to understand and cheap to implement. It provides a modeling technique that is easy for human to comprehend and simplifies the classification process (Utgoff and Brodley, 1990). Most Decision tree algorithms can be implemented in both serial and parallel form while others can only be implemented in either serial or parallel form. Parallel implementation of decision tree algorithms is desirable in-order to ensure fast generation of results especially with the classification/prediction of large data sets, it also exploits the underlying computer architecture (Shafer et al, 1996). But serial implementation of decision algorithm is easy to implement and desirable when small-medium data sets are involved. In this paper we will review the most common decision tree algorithms implemented serially and perform an experiment to compare their classification and prediction accuracy. In Section 2 we did a review of decision tree algorithms, the phases of decision tree construction and its implementation patterns. In Section 3 we review the serial implementation decision tree algorithms and compare their features. In Section 4 we conduct a survey of the implementation statistics of the commonly used serially implemented decision tree algorithms and compare their frequency of usage. In section 5 we perform an experimental analysis of the commonly used decision tree algorithms and evaluate their performance based on execution time and classification accuracy.

2. Decision Tree Algorithm

Decision tree algorithm is a data mining induction techniques that recursively partitions a data set of records using depth-first greedy approach (Hunts et al, 1966) or breadth-first approach (Shafer et al, 1996) until all the data items belong to a particular class. A decision tree structure is made of root, internal and leaf nodes. The tree structure is used in classifying unknown data records. At each internal node of the tree, a decision of best split is made using impurity measures (Quinlan, 1993). The tree leaves is made up of the class labels which the data items have been group. Decision tree classification technique is performed in two phases: tree building and tree pruning. Tree building is done in top-down manner. It is during this phase that the tree is recursively partitioned till all the data items belong to the same class label (Hunts et al, 1966). It is very tasking and computationally intensive as the training data set is traversed repeatedly. Tree pruning is done is a bottom-up fashion. It is used to improve the prediction and classification accuracy of the algorithm by minimizing over-fitting (noise or much detail in the training data set) (Mehta et al, 1996). Over-fitting in decision tree algorithm results in misclassification error. Tree pruning is less tasking compared to the tree growth phase as the training data set is scanned only once. In this study we will review Decision tree algorithms implemented in a serial pattern, identify the algorithms commonly used and compare their classification accuracy and execution time by experimental analysis

3. Serial Implementation of Decision Tree Algorithm

Decision tree algorithm can be implemented in a parallel form based on its scalability with respect to the input data. Parallel implementation tends to be scalable, fast and disk resident and can be implemented in computer architecture with many processors (Shafer et al, 1996). Serial

implementation on the other hand is fast, memory resident and easy to understand. In this paper we will focus on serial implementation of decision tree algorithm by Hunt's algorithms and other serial decision tree algorithms that does not obey Hunt' algorithm (SLIQ and SPRINT). Hunt's method of decision tree construction (Quinlan, 1993 and Hunts et al, 1966) is as stated below: Given a training set T of data records denoted by the classes C= C1, C2, •••, Ck
The decision tree is constructed recursively using depth-first divide-and-conquer greedy strategy by the following cases:
• Case1: T contains all the cases that belong to the same class Cj. The leaf node for T is created and it is known by the class Cj
• Case2: T contains cases that belong to one class or more. The best splitting single attribute is chosen, which will test and split T in to a single-class that contains many cases. The split of T gives the subsets of T which are: T1, T2, •••, Tn. The split on T is chosen in order to obtain mutually exclusive results:

O1, O2, •••, On Ti T having the result Oi

• Case3: T contains no cases. The leaf created for the decision T has a class from other source which is not T. In C4.5 this is regarded as the most frequent class and is chosen as the parent not of the constructed tree. With Hunt's method decision tree is constructed in two phases: tree growth and pruning phases which have been explained in Section II. Most serial decision tree algorithms (IDE3, CART and C4.5) are based Hunt's method for tree construction (Srivastava et al, 1998). In Hunt's algorithm for decision tree construction, training data set is recursively partitioned using depth-first greedy technique, till all the record data sets belong to the class label (Hunts et al, 1966). The data sets are memory resident and the data sets are sorted at every node in-order to determine the best splitting attribute (Shafer et al, 1996). One of the disadvantages of serial decision tree implementation is low classification accuracy when the training data is large. In order to reduce the high computational complexity associated with large training data set, the whole training data set is loaded into the memory at the same time which leads to low classification accuracy (Srivastava et al, 1998). This short coming of serial decision tree implementation is addressed by SLIQ and SPRINT algorithm. In serial implementation of SPRINT and SLIQ, the training data set is recursively partitioned using breadth-first technique till all the data set belongs to the same class label and there is one time sort of the data set using list data structure. Also the training data set is not memory resident but disk resident, which makes data scalability possible. This approach improves the classification accuracy and reduces misclassification errors. The following sections give a review of the commonly used decision tree algorithms based on serial implementation. Decision trees based on Hunt's algorithm can be classified as classical decision trees which can only be implemented serially. But there have been on-going researches to implement them in a parallel pattern. Peng et al,(n.d) implemented the parallel version of IDE3. The disadvantages associated with classical decision tree algorithms are as enumerated below by Podgorelec et al (2002):
• Handling Noise Data: Classical decision tree algorithms do not always produce decision models with accurate classification when the training data contains noise or too much detail. But C4.5 and enhanced processing technique that handles this deficiency

• Production of Same Type of Decision tree: Given the same training data set and the same condition, classical algorithm always produce the same tree, instead of producing multiple trees with a flexibility to choose the one that is less prone to error.

• Importance of Error: Different errors arises during application of classical decision tree algorithms, but some errors have higher priority than others and need to be minimized to achieve accurate classification. The errors occur as a result of the decisions made in building the tree which reduces classification accuracy.

3.1 IDE3

IDE3 (Iterative Dichotomiser 3) decision tree algorithm was introduced in 1986 by Quinlan Ross (Quinlan, 1986 and 1987). It is based on Hunt's algorithm and it is serially implemented. Like other decision tree algorithms the tree is constructed in two phases; tree growth and tree pruning. Data is sorted at every node during the tree building phase in-order to select the best splitting single attribute (Shafer et al, 1996). IDE3 uses information gain measure in choosing the splitting attribute. It only accepts categorical attributes in building a tree model (Quinlan, 1986 and 1987). IDE3 does not give accurate result when there is too-much noise or details in the training data set, thus a an intensive pre-processing of data is carried out before building a decision tree model with IDE3

3.2 C4.5

C4.5 algorithm is an improvement of IDE3 algorithm, developed by Quinlan Ross (1993). It is based on Hunt's algorithm and also like IDE3, it is serially implemented. Pruning takes place in C4.5 by replacing the internal node with a leaf node thereby reducing the error rate (Podgorelec et al, 2002). Unlike IDE3, C4.5 accepts both continuous and categorical attributes in building the decision tree. It has an enhanced method of tree pruning that reduces misclassification errors due noise or too-much details in the training data set. Like IDE3 the data is sorted at every node of the tree in order to determine the best splitting attribute. It uses gain ratio impurity method to evaluate the splitting attribute (Quinlan, 1993).

3.3 CART

CART (Classification and regression trees) was introduced by Breiman, (1984). It builds both classifications and regressions trees. The classification tree construction by CART is based on binary splitting of the attributes. It is also based on Hunt's model of decision tree construction and can be implemented serially (Breiman, 1984). It uses gini index splitting measure in selecting the splitting attribute. Pruning is done in CART by using a portion of the training data set (Podgorelec et al, 2002). CART uses both numeric and categorical attributes for building the decision tree and has in-built features that deal with missing attributes (Lewis, 200). CART is unique from other Hunt's based algorithm as it is also use for regression analysis with the help of the regression trees. The regression analysis feature is used in forecasting a dependent variable (result) given a set of predictor variables over a given period of time (Breiman, 1984). It uses many single-variable splitting criteria like gini index, symgini etc and one multi-variable (linear combinations) in determining the best split point and data is sorted at every node to determine the best splitting point. The linear combination splitting criteria is used during regression analysis. SALFORD SYSTEMS implemented a version of CART called CART® using the original code of Breiman, (1984). CART® has enhanced features and capabilities that address the short-comings of CART giving rise to a modern decision tree classifier with high classification and prediction accuracy.

3.4 SLIQ

SLIQ (Supervised Learning In Ques) was introduced by Mehta et al, (1996). It is a fast, scalable decision tree algorithm that can be implemented in serial and parallel pattern. It is not based on Hunt's algorithm for decision tree classification. It partitions a training data set recursively using breadth-first greedy strategy that is integrated with pre-sorting technique during the tree building phase (Mehta et al, 1996). With the pre-sorting technique sorting at decision tree nodes is eliminated and replaced with one-time sort, with the use of list data structure for each attribute to determine the best split point (Mehta et al, 1996 and Shafer et al, 1996). In building a decision tree model SLIQ handles both numeric and categorical attributes. One of the disadvantages of SLIQ is that it uses a class list data structure that is memory resident thereby imposing memory restrictions on the data (Shafer et al, 1996). It uses Minimum Description length Principle (MDL) in pruning the tree after constructing it MDL is an inexpensive technique in tree pruning that uses the least amount of coding in producing tree that are small in size using bottom-up technique (Anyanwu et al, 2009 and Mehta et al, 1996).

3.5 SPRINT

SPRINT (Scalable Parallelizable Induction of decision Tree algorithm) was introduced by Shafer et al, 1996. It is a fast, scalable decision tree classifier. It is not based on Hunt's algorithm in constructing the decision tree, rather it partitions the training data set recursively using breadth-first greedy technique until each partition belong to the same leaf node or class (Anyanwu et al, 2009 and Shafer et al, 1996). It is an enhancement of SLIQ as it can be implemented in both serial and parallel pattern for good data placement andload balancing (Shafer et al, 1996). In this paper we will focus on the serial implementation of SPRINT. Like SLIQ it uses one time sort of the data items and it has no restriction on the input data size. Unlike SLIQ it uses two data structures: attribute list and histogram which is not memory resident making SPRINT suitable for large data set, thus it removes all the data memory restrictions on data (Shafer et al, 1996). It handles both continuous and categorical attributes.

4. Serial Decision Tree Algorithm Implementation Statistics

We reviewed about thirty-two articles in order to determine which of the serial decision tree methods is commonly used in practical applications. The outcomes of our literature survey are as stated in the following tables below: the articles are represented by the last name of the first author and the year of publication and also the decision tree algorithm used

Paper	Decision Tree Algorithm
Quinlan, 1983, 1993	IDE3 and C4.5
Shafer et al, 1996	SPRINT, SLIQ, IDE3 and CART
Hunts et al, 1966	CLS, C4.5, CART and IDE3
Breiman, 1984	CART
Fan et al, 2003	Random Tree
Mehta, et al, 1996	SLIQ
Gehrke et al, 1998	RainForest
Peng et al	IDE3
Srivastava et al. 1997	SPRINT, IDE3 and C4.5
Srivastava et al. 1998	SPRINT, IDE3, C4.5 and SLIQ
Rastog et al, 1998	PUBLIC, CLS, IDE3, C4.5 and CART
Sattler and Dunemann, 2001	ID3, C4.5, SPRINT, SLIQ and PUBLIC
Kufrin, 1997	ID3 and CART
BA˘DULESCU, (n.d)	Rainforest, IDE3, C4.5 and SLIQ CART and SPRINT
Srivastava and Singh (n.d)	ID3, C4.5, ClOUDS and SPRINT
Sattler and Dunemann, 2001	ID3, C4.5, SPRINT, SLIQ and PUBLIC
Podgorelec et al, 2002	ID3, C4.5, CART and OCI
Ling et al, 2004	C4.5
Du and Zhan, 2002	ID3 and CART
Pješivac-Grbović et al, 2006	C4.5

Wen et al, 2008	CART and C5.0
Xu et al, 2006	IDE3 and IDE3+

Table 1: Literature review of Decision Tree Algorithms

Table 1 show a literature of decision tree algorithms that is implemented serially.
Table 2 shows the frequency usage of the serial implementation of decision tree algorithms. Table 2 shows that IDE3 is the most frequently used classifier, followed by C4.5 and then SPRINT. ID3 was one the earliest classifiers but as researchers and scientists discovered its flaws they switch to CART, C4.5 and SPRINT

5. Experimental Analysis

We carried out some experiments using Statlog data sets (Michie et al, 1994) as shown in Table 3. The Stalog data set include large scale data sets of various disciplines like financial (Australian and German data sets); transportation (Vehicle data sets), science (Shuttle data set) and health (Heart data set). We did a performance evaluation of the decision tree classifiers. Number of records, number of attributes and class size of the different data sets are varied in-order to determine their effect on the performance of each classifier. Tables 4, 5, and figures (1), (2) shows the result of the analysis.

Decision Tree Algorithm	Frequency Usage
CLS	9%
IDE	68 %
IDE3+	4.5 %
C4.5	54.55 %
C5.0	9%
CART	40.9 %
Random Tree	4.5 %
Random Forest	9%
SLIQ	27.27 %
PUBLIC	13.6 %
OCI	4.5 %
CLOUDS	4.5 %
SPRINT	31.84 %

Table 2: Frequency use of Decision Tree Algorithm

5.1 Experimental Results

Figure (1) shows that for all the classifiers the execution time increases as the number of records increases. The steady portion of the graph is the effect of varying the number of attributes of the classifiers. Also Figure (2) shows that the execution time (time to build the model) of the classifiers decreases as the attributes of the classifiers increases and becomes steady at some point due to the change in the number of records of the data sets and also class size change. Table 4 shows that SPRINT classifiers have the fastest execution time among all the classifiers, irrespective of the class size, number of attributes and records of the data sets volume. This is closely followed by C4.5. The table also showed that execution time for IDE3 is faster that CART but CART is preferred by researches and scientists as it handles both categorical and continuous attributes while IDE3 does not handle continuous attributes. Table 5 shows that SPRINT classifier has the highest classification accuracy among all the classifiers, this is followed by C4.5. The class size, attribute number and record number do not affect the classification accuracy of SPRINT and C4.5 compared to other classifiers. The classification accuracy of the IDE3 and CART classifiers depends to a large extent the class size, attribute number and record number of the data sets. As shown in Table 5 for a large data set (shuttle data set), the classification accuracy of IDE3 is better than that of CART as ID3 has a high accuracy for large data that have been pre-processed (noise and outliers removed) and loaded into the memory at the same time. But for other data sets (Vehicle, Australian, German and Heart) that are not too large (small-medium data sets), the classification accuracy of CART is more than that of IDE3.

Dataset	Category	No. of Attributes	No. of Classes	No. of Records
Australian	Credit Analysis	14	2	690
Shuttle	Space Shuttle Radiation	9	7	43499
German	Credit Analysis	24	2	1000
Heart	Heart Disease Screening	13	2	270
Vehicle	Vehicle Identification	18	4	753

Table 3: Statlog Datasets

Dataset	IDE3	CART	C4.5	SPRINT
Austrilian	0.08secs	11.41secs	0.02secs	0.02secs
Shuttle	1.48secs	38.31secs	0.17secs	0.15secs
German	0.03secs	2.17secs	0.06secs	0.04secs
Heart	0.03secs	0.61secs	0.1secs	0.03secs
Vehicle	0.03secs	1.64secs	0.1secs	0.02secs

Table 4: Execution Time to build Model

Dataset	IDE3	CART	C4.5	SPRINT
Austrilian	71.5 %	85.4 %	84.2 %	85.8 %
Shuttle	99.2 %	94 %	98.00 %	99.63 %
German	32.1 %	70 %	69.2 %	70 %
Heart	35.2%	56.67%	76.7%	80%

Vehicle 54.3% 65% 66.5% 67%

Table5: Classification Accuracy

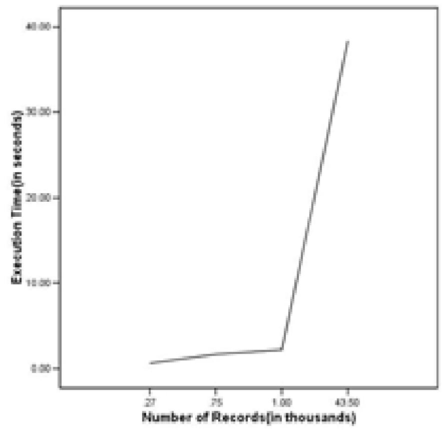

Figure 1: Execution time and Number of Records

6. Conclusion and Future Work

Decision tree induction is one of the classification techniques used in decision support systems and machine learning process. With decision tree technique the training data set is recursively partitioned using depth- first (Hunt's method) or breadth-first greedy technique (Shafer et al , 1996) until each partition is pure or belong to the same class/leaf node (Hunts et al, 1966 and Shafer et al , 1996). Decision tree model is preferred among other classification algorithms because it is an eager learning algorithm and easy to implement. Decision tree algorithms can be implemented serially or in parallel. Despite the implementation method adopted, most decision tree algorithms in literature are constructed in two phases: tree growth and tree pruning phase. Tree pruning is an important part of decision tree construction as it is used improving the classification/prediction accuracy by ensuring that the constructed tree model does not overfit the data set (Mehta et al, 1996). In this study we focused on serial implementation of decision tree algorithms which are memory resident, fast and easy to implement compared to parallel implementation of decision that is complex to implement. The disadvantages of serial decision tree implementation is that it is not scalable (disk resident) and its inability to exploit the underlying parallel architecture of computer system processors. Our experimental analysis of performance evaluation of the commonly used decision tree algorithms using Statlog data sets (Michie et al, 1994) shows that there is a direct relationship between execution time in building the tree model and the volume of data records. Also there is an indirect relationship between execution time in building the model and attribute size of the data sets. The experimental analysis also shows that SPRINT and C4.5 algorithms have a good classification accuracy compared to other algorithms used in the study. The variation of data sets class size, number of attributes and volume of data records is used to determine which algorithm has better classification accuracy between IDE3 and CART algorithms. In future we will perform experimental analysis of commonly used parallel implementation tree algorithms and them compare it that serial implementation of decision tree algorithms and determine which one is better, based on practical implementation.

References

[1] Anyanwu, M., and Shiva, S. (2009). Application of Enhanced Decision Tree Algorithm to Churn Analysis. 2009 International Conference on Artificial Intelligence and Pattern Recognition (AIPR-09), Orlando Florida
[2] BA˘DULESCU, L. A. (n.d). Data mining algorithms based on decision trees. Annals of the ORADEA university. From http://www.imtuoradea.ro/auo.fmte/files2006/MIE files/Laviniu%20Aurelian%20Badulescu%201.pdf Retrieved date: May 13, 2009
[3] Breiman, L., Friedman, J., Olshen, L and Stone, J. (1984). Classification and Regression trees. Wadsworth Statistics/Probability series. CRC press Boca Raton, Florida, USA.
[4] Du, W., Zhan, Z. (2002). Building decision tree classifier on private data, Proceedings of the IEEE international conference on Privacy, security and data mining, pp.1-8, Maebashi City, Japan
[5] Garofalakis, M., Hyun, D., Rastogi, R. and Shim, K. (200). Efficient algorithms for constructing decision trees with constraints. Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 335 - 339
[6] Gehrke, J., Ramakrishnan, R., Ganti, V. (1998). RainForest - a Framework for Fast Decision Tree Construction of Large Datasets.Proceedings of the 24th VLDB conference, New York, USA. pp.416-427
[7] Gehrke, J., Ramakrishnan, R., Ganti, V. (1998). RainForest - a Framework for Fast Decision Tree Construction of Large Datasets, Proceedings of the 24th VLDB conference, New York, USA. pp.416-427

[8] Hunt, E.B., Marin. and Stone,P.J. (1966). Experiments in induction, Academic Press, New York.
[9] Khoshgoftaar, T.M and Allen, E.B. (1999). Logistic regression modeling of software quality. International Journal of Reliability, Quality and Safety Engineering, vol. 6(4, pp. 303-317.
[10] Kufrin, R. (1997). Decision trees on parallel processors. In J. Geller, H. Kitano, and C. B. Suttner, editors, parallel Processing for Artificial Intelligence 3 Elsevier Science
[11] Lewis, R.J. (200). An Introduction to Classification and Regression Tree (CART) Analysis. 2000 Annual Meeting of the Society for Academic Emergency Medicine, Francisco, California
[12] Ling, X. C., Yang, Q., Wang, J and Zhang, S. (2004). Decision trees with minimal costs. Proceedings of the 21st International Conference on Machine Learning, Banff, Canada

[13] Lippmann, R. (1987). An Introduction to computing with neural nets. IEEE ASSP Magazine, vol. (22)
[14] Mehta, M., Agrawal, R., and Rissanen, J. (1995). MDL-based decision tree pruning. International conference on knowledge discovery in databases and data mining (KDD-95) Montreal, Canada

[15] Mehta, M., Agrawal, R., and Rissanen, J. (1996). SLIQ: A fast scalable classifier for data mining. In EDBT 96, Avignon, France
[15] Michie, D., Spiegelhalter, D., J., and Taylor, C., C. (1994). Machine Learning, Neural and Statistical Class-

fication, Ellis Horword
[16] Peng, W., Chen, J. and Zhou,H. An Implementation of IDE3 Decision Tree Learning Algorithm. From
web.arch.usyd.edu.au/ wpeng/DecisionTree2.pdf Retrieved date: May 13, 2009
[17] Podgorelec, V., Kokol, P., Stiglic, B., Rozman, I. (2002). Decision trees: an overview and their use in
medicine, Journal of Medical Systems Kluwer Academic/Plenum Press, vol.26, Num. 5, pp.445-463.
[18] Pješivac-Grbović, J., Angskun, T., Bosilca, G., Fagg, G.E., Dongarra, J. J. (2006). Decision trees and MPI
collective algorithm selection problem. Tech. Rep. UT-CS-06-586, The University of Tennessee at Knoxville,
Computer Science Department. From < http : //www.cs.utk.edu/library/2006.html >
[19] Peng, W., Chen, J., and Zhou., H (n.d). An Implementation of ID3–– decision tree learning algorithm.
University of New South Wales, School of Computer Science and Engineering, Sydney, NSW 2032, Australia

[20] Quinlan, J. R. (1986). Induction of decision trees. Machine Leaning, vol (1), pp.81-106
[21] Quinlan, J. R. (1987). Simplifying decision trees, International Journal of Machine Studies, number27,
pp. 221-234.

[22] Quinlan, J. R. (1993). C45: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA.
[23] Utgoff, P and Brodley, C. (1990). An Incremental Method for Finding Multivariate Splits for Decision Trees,
Machine Learning: Proceedings of the Seventh International Conference pp.58.
[24] Rastogi, R., and Shim, K. (1998). PUBLIC: A decision tree classifier that integrates building and pruning.
Proceedings of the 24th VLDB Conference. New York. pp. 404-415
[25] Sattler, K and Dunemann, O. (2001).SQL database primitives for decision tree classifiers Proceedings of the
tenth international conference on Information and knowledge management, pp.379-386
[26] Shafer, J., Agrawal, R., and Mehta, M. (1996). Sprint: A scalable parallel classifier for data mining.
Proceedings of the 22nd international conference on very large data base. Mumbai (Bombay), India
[27] Srivastava, A., Singh, V., Han, E., and Kumar, V .(1997). An efficient, scalable, parallel classifier for data
mining . University of Minnesota, Computer Science, technical report, USA.
[28] Srivastava, A., Singh, V. (n.d). Methods to reduce I/O for decision tree classifiers. IBM T.J Watson Research
center.

[29] Srivastava, A., Singh,V., Han,E., and Kumar, V .(1998). Parallel Formulations of Decision-Tree Classification
Algorithms. Data Mining and Knowledge Discovery, an international journal, pp.237-261

[30] Tan, P., Steinbach, M. and Kumar, V. (2006). Introduction to Data
[31] Wen, X., Hu, G., Yang, X. (2008). CBERS-02 Remote Sensing Data Mining Using Decision Tree Algorithm,
First International Workshop on Knowledge Discovery and Data Mining, pp.86-89

[32] Xu, M., Wang, J. and Chen, T. (2006). Improved decision tree algorithm: ID3+, Intelligent Computing in
Signal Processing and Pattern Recognition, Vol.345, pp.141-149