# VLSI Implementation of Decision Tree Classification Using C4.5 Algorithm

[1]Delna Davis, [2]J Samson Immanuel
[1]School of Electrical Sciences, Karunya University, Coimbatore, Tamilnadu, India

**Abstract**
In recent years, the size of the data being collected and analysed increased tremendously. Because of this, Data mining techniques have become important to researchers in science, engineering, medicine, business and security domains. The main difficulty in datamining is classification. Decision Tree Classification (DTC) is the most accepted solution for this problem, that gives high precision while handling very large amount of data. This paper presents VLSI implementation of flexible architecture for Decision Tree classification in data mining using c4.5 algorithm.

**Keywords**
Data mining, Decision Tree Classifi-Cation, C4.5 Algorithm

## I. Introduction

Data mining is the process of converting unprocessed data into valuable information. Data mining results finds application in a number of fields like biotechnology, marketing and internet searches. Current advances in data processing techniques have resulted in complicated data sets. As a result, ever more complex data mining algorithms are required for extracting information from these vast data.

The most leading and established approaches in knowledge discovery and data mining is decision tree classification method. It is the science and technology of exploring enormous and multiple data sets in order to find out useful patterns from training set. Now a day, the field of data mining achieves more importance because it enables the information mining from the large amounts of data available. Practitioners and theoreticians are constantly in the search for techniques to make the process more efficient, cost-effective and precise [2]. The Decision trees are created in the basis of decision theory and statistics, are very much helpful tools in other fields such as data mining, information mining, and pattern detection.

Many techniques are used in data mining. Aprori algorithm[3] and K-Means Clustering [5] are some of them. Decision tree algorithms builds a tree structure using if-else-then rules. In this paper we propose a decision tree classification using c4.5 algorithm.

## II. Architecture

### A. Decision Tree Classification

Data mining techniques can be applied to a variety of data sets and Information produced by data mining techniques can be characterized in many different ways. Decision tree structures are widely used to organize classification techniques. Decision trees can be used to find out the steps that reaches a classification. In general decision trees are starts with a root node or parent node from which the tree growth begins. Then it evaluates each attribute using an algorithm and finds out the path through which the tree grows. Typically Decision making process is done through if else statements. The process is repeated until the tree reaches the leaf node.

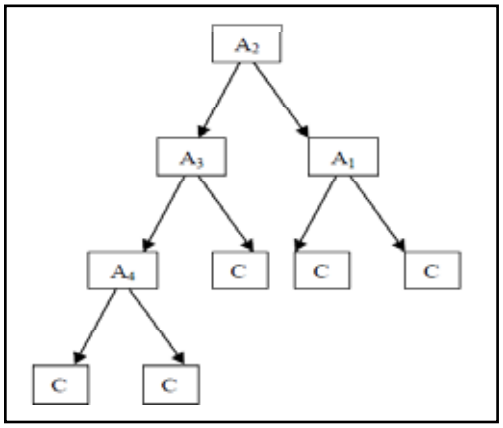


Fig. 1: Decision Tree Model

### B. System Architecture

In this architecture shown in fig. 2 the main components are Block-RAM, split info module and a find min Gini gain module. Block-RAM is used to store the instances(input) values. Split info module consists of comparators, counters, adders, multipliers and memory elements according to the need. It computes the Gini gain values for each attribute. This values are given to the find minimum Gini gain module. This module find out which has lower value and gives the attribute name according to the lower value at the output.

Block Random Access Memory (BRAM) is an highly developed memory constructor that is used to generate area and performance-optimized memories with the help of embedded block RAM resources in Xilinx FPGAs. It cannot be used to implement other functions like digital logic. So BRAM is a dedicated memory. It is a two port memory consisting of several kbits of RAM. Depending on how advance the FPGA, there may be several of them. Block-RAM is used to store the data inputs from pc. They are fixed RAM modules. Usually they are available in 9 Kbits or 18 Kbits in size. If a small RAM is implemented with a block RAM then its wastes the rest of the space in RAM. Usually block RAMs are used for large sized memories and distributed RAM for small sized memories or FIFO's. Block RAM memory is synchronous. Block-RAM is used to store the data inputs from pc.

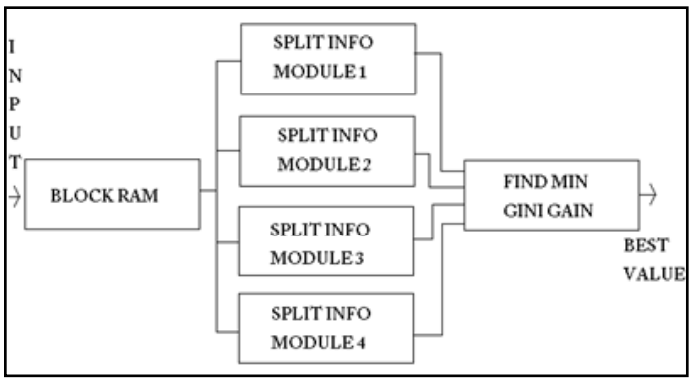


Fig. 2: Proposed Architecture

The Split Info calculation module is the main part of the architecture. It contains memory elements, adders, multipliers

and division elements. Using these it computes the Gini value for all the attributes by applying certain conditions and using equations shown below.

Find Mini Gini Gain Module is used to get the least values from a set of values obtained from the Split Info Module. This block firstly finds out the root node value from the output of the split info sub system. Then this value is given to the memory to store that value. After that it finds out the branches from the other values got from the split info sub system.

The Gini value is a numerical quantity which calculates the inequality of a data set. Computing the Gini score for a typical split involves calculating the occurrence of each class in each of the partitions. The particulars of the Gini calculation can be verified by the following pattern. Assume that there are R records in the present node. Also, assume that there are only two different values of class IDs, hence there is only two partitions into which the root node can be split. The algorithm works over the R records and calculates the number of records belonging to different partitions. The Gini value for each partition is then given by

$$Gini_i = 1 - \sum_{j=0}^{1} Rij/(Ri * Ri)$$

where Ri is the number of total number of instances in partition i. Rij represent the number of instances that holds the class value j. The Gini value of the total split is then computed using the weighted average of the Gini values for each partition, i.e.,

$$\text{Gini total} = \sum_{i=0}^{1} \frac{Ri}{R} * Gini_i$$

The Rij values are stored in a count matrix, in memory. The partitions are formed based on a splitting crieteria, which depends on the value of a particular attribute. Each attribute is a feasible nominee for being the split attribute. Hence this process of calculating the finest split has to be done over all attributes. Categorical attributes have a limited number of different class ID values, so there is little advantage in optimizing Gini value computation for such attributes.

The C4.5 algorithm is used to construct a decision tree, given a set of non-categorical attributes C1, C2, .., Cn, the categorical attribute C, and a training data set T of records.

Function C4.5 (R: set of non-categorical attributes,
C: set of the categorical attribute,
S: training data)
gives a decision tree structure;
start
If S is vacant, then produce a single node with value Failure;
If all of the instances present in S have the same value for the categorical attribute, then produce a single node with that value;
If R is vacant , then produce a single node with a value which is the most common value of the categorical attribute that are found in instances of S
Note: If the records are improperly classified then errors will be there.
Let D be the attribute with highest Gain(D,S) compared to all other attributes.
Let {di| i=1,2, .., p} represents the values of attribute D.
Let {Si| i=1,2, .., p} represents subsets of S consisting respectively of instances with value di for attribute D.
Generate a tree with root as D and branches as d1, d2, .., dp.
C4.5(R-{D}, C, S1), C4.5(R-{D}, C, S2), .., C4.5(R-{D}, C, Sp),
end C4.5;

Fig. 3: Algorithm

## IV. Results and Discussions

In this section we describe the results obtained through our work. We did our work in Xilinx 13.2 version with device XC5VLX110T. The example data set used for checking the architecture is shown in table 1. It contains 14 instances, 4 attributes and one class value. For doing this, c4.5 algorithm is used and according to that the output is obtained. Steps used in the algorithm are shown in fig. 3.

Table 1: Sample Data

| Outlook | Temperature | Humidity | Wind | Play ball |
|---|---|---|---|---|
| Sunny | Hot | High | Weak | No |
| Sunny | Hot | High | Strong | No |
| Overcast | Hot | High | Weak | Yes |
| Rain | Mild | High | Weak | Yes |
| Rain | Cool | Normal | Weak | Yes |
| Rain | Cool | Normal | Strong | No |
| Overcast | Cool | Normal | Strong | Yes |
| Sunny | Mild | High | Weak | No |
| Sunny | Cool | Normal | Weak | Yes |
| Rain | Mild | Normal | Weak | Yes |
| Sunny | Mild | Normal | Strong | Yes |
| Overcast | Mild | High | Strong | Yes |
| Overcast | Hot | Normal | Weak | Yes |
| Rain | Mild | High | Strong | No |
| | | | Total | 14 |

Quinlan developed C4.5 algorithm for inducing Classification Models, also known as Decision Trees, from training data set. CLS (concept learning system ) is the basic principle followed by all Decision tree induction algorithms. Hunt developed this principle.

We are given with a set of instances. Each instance has the same structure. It consists of a number of attribute/value pairs. One of these attributes is used to represents the class value of the instance. The problem lies in the constructing process of a decision tree on the basis of answers to questions about the non-category attributes find out correctly the value of class. Commonly the category attribute takes only the values {true, false}, or {success, failure}, or something equivalent.

The final classified tree is shown in fig. 4. Using the c4.5 algorithm we first find out the information gain of each attribute for selecting the root node. Attribute selection for root node is the basic step in constructing a decision tree. Entropy and Information Gain is used in process of attribute selection, using attribute selection, C45 algorithm select which attribute will be selected to become a node of the decision tree and so on. The result obtained through our work is shown in fig. 5.

The final tree structure is shown in fig. 5. It takes outlook as root node and its branches are sunny, rainy and overcast. From sunny it goes to the node humidity and the two branches are high and normal. Then from high it goes to the leaf node NO and from normal it goes to the leaf node YES. Similarly from the branch rainy it goes to the branch node windy. The two branches are true and false. From false branch it goes to the leaf node YES and from true branch it goes to the leaf node NO. From the branch overcast it directly reaches the leaf node YES.

ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

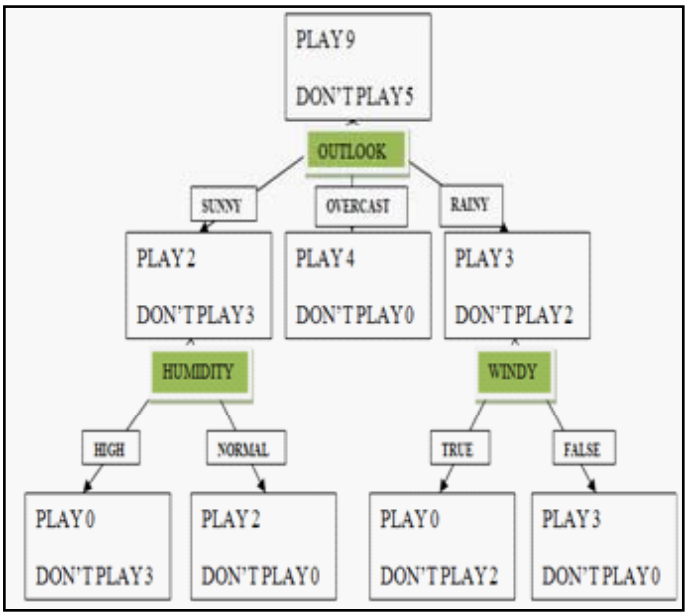


Fig. 4: Final Tree Structure

This architecture can be used for building decision tree for any applications in the data mining process. It is a flexible architecture, which can be used for any data set with 4 attributes, one class value and any number of attribute values.

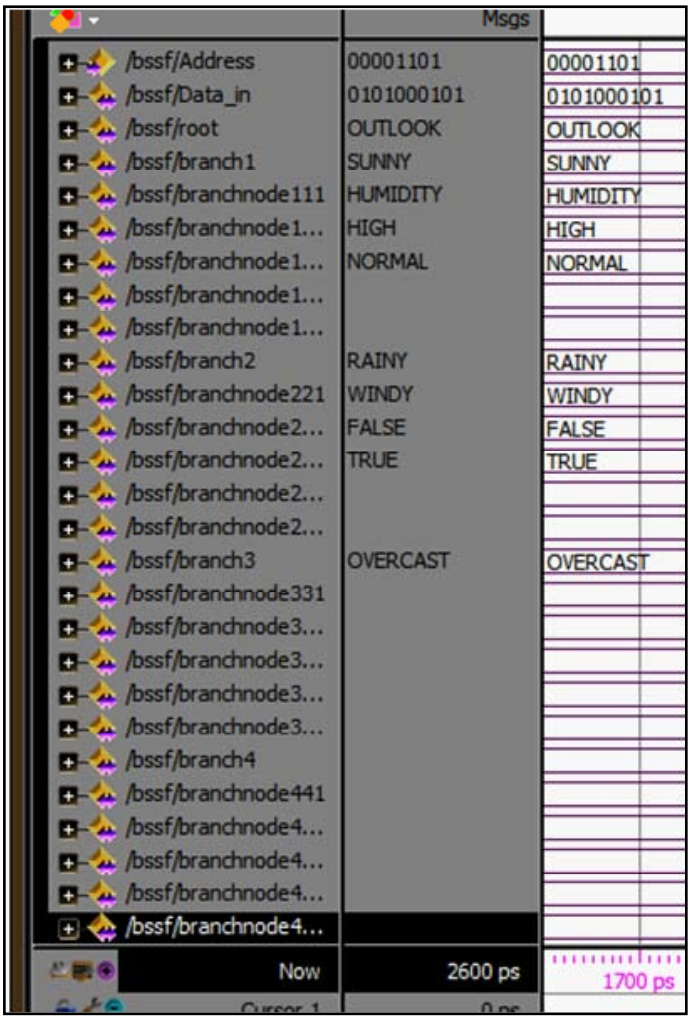


Fig. 5: Final Result

## V. Conclusion

Among the other classification techniques, Decision tree technique is preferred because it is an eager learning algorithm and easy to design. In this paper, we have designed VLSI implementa-tion of Decision Tree Classification method using c4.5 algorithm. The Split Info Module which computes the Gini value is the important component in the system. So we have created an efficient design to implement the Spli Info Module to improve the performance. The arithmetic calculations necessary to calculate the most favorable split point were then simplified to reduce the hardware resources required.

## References

[1] Matthew N. Anyanwu, Sajjan G. Shiva,"Comparative Analysis of Serial Decision Tree Classification Algorithms".

[2] M. Joshi, G. Karypis, V. Kumar,"ScalParC: A new scalable and efficient parallel classification algorithm for mining large datasets", In Proceedings of the 11th International Parallel Processing Symposium (IPPS), 1998.

[3] Z. K. Baker, V. K. Prasanna,"Efficient Hardware Data Mining with the Apriori Algorithm on FPGAs", Field-Programmable Custom Computing Machines, 13th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM), pp. 3-12, 2005.

[4] Y. H. Wen, J. W. Huang, M. S. Chen,"Hardware-Enhanced Association Rule Mining with Hashing and Pipelining", IEEE Transactions on Knowledge and Data Engineering, pp. 784-795, 2008.

[5] M. Estlick, M. Leeser, J. Theiler, J. J. Szymanski, "Algorithmic transformations in the implementation of K-means clustering on reconfigurable hardware", Proceedings of the ACM/ SIGDA 9th international symposium on Field programmable gate arrays, pp. 103-110, 2001.

[6] X. Wang, M. Leeser,"K-means Clustering for Multispectral Images Using Floating-Point Divide", Field-Programmable Custom Computing Machines, 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM), pp. 151-162, 2007.

[7] M. Papadonikolakis, C. Bouganis, G. Constantinides, "Performance comparison of GPU and FPGA architectures for the SVM training problem", International Conference on Field- Programmable Technology, pp. 388-391, 2009.

[8] S. Sun, J. Zambreno,"Mining Association Rules with systolic trees", International Conference on Field Programmable Logic and Applications, pp.143-148, 2008.

Delna Davis received her B.Tech. degree in Electronics and communication Calicut university, Kerala, India, in 2011, the M.Tech degree in VLSI Design from Karunya University, Tamilnadu, India, in 2013.