Quest Journals
Journal of Software Engineering and Simulation
Volume 8 ~ Issue 4 (2022) pp: 31-37
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

An Efficient Data Mining Method for Distributed Databases using Decision Trees

[1]Eberechi Omaegbu, [2]Bennett, E. O, [3]Whyte, S. T.
Department of Computer Science, Rivers State University, Port Harcourt, Nigeria

Abstract
Data mining from large volume of data has given rise to increased use of distributed computing, which in turn has led to distributed data mining. Distributed data mining is faced with challenges ranging fromcommunication cost, computational complexities, to quality and efficiency of the data mining algorithms. One critical problem therefore is developing methods and algorithms best suited for solving complex, real-world data mining problems. This research presents a decision tree algorithm for solving classification problem. The solution utilizes feature set decomposition using genetic algorithm which partitions the features of the dataset into subsets which collectively improve accuracy and minimize the complexities associated with distributed data mining using decision trees.The execution time and accuracy rate were evaluated and the algorithm's ability to scale across several datasets with varying increasing data sizes; breast cancer dataset with 569 observations, churn test with 10000 observations and credit card dataset with 30,000 was observed. The obtained results from the (DTClassifier)algorithmwere compared with some already existing decision tree induction algorithms. The(DTClassifier) algorithm maintained steady classification accuracy with a slight increase as the data size increased.

Keywords: *distributeddata mining, decision tree algorithm, genetic algorithm, feature set decomposition*

***R**eceived 14 Apr, 2022; **R**evised 28 Apr, 2022; **A**ccepted 30 Apr, 2022 © The author(s) 2022.*
P**ublished with open access at **www.questjournals.org

I. INTRODUCTION

Before now, data mining and its research have been focused on the traditional approach to data mining, which assumes that the dataset is memory-resident, static, and centrally stored at a single location. The required data is downloaded and aggregated to a centralized location in this manner, and data mining activities are then performed on the data [1]. This assumption however,no longer stands true, this is as a result of the rapid growth of large-scale data in recent decades and the development of the Internet, where data mining techniques must meet two challenges: first, the amounts of data are produced at very rapid rate that they can only be processed by supercomputers; second, data is now being stored in various locations, making it increasingly expensive to consolidate information in one location.

Also, as technology advances, and begins to shift towards the use of distributed systems; a direct application of traditional mining algorithms to distributed databases is not effective because it implies a significant amount of communication and computation overhead. As data continues to grow in quantity and complexity, parallel and distributed data mining algorithms offer a more viable alternative for extracting knowledge with high accuracy, efficiency while assuring system scalability and interactivity [2].

Distributed Data Mining (DDM) extracts information from data located at heterogeneous sites, it uses decentralized mining strategy which makes the entire working system scalable by distributing workload across heterogeneous sites.[3].

Whereas, centralized data mining leads to increased computational cost and privacy due to centralized data mining, DDM paved way for decrease in computational cost as well as enhanced data privacy by distributing resources across distributed sites.

Albeit its numerous benefits, there are some concerns that may arise in distributed data mining, including how to effectively reduce the amount of communication needed for distributed computation, how to mine across various heterogeneous data sources, such as multisource databases, and how to execute multi-relational mining in a distributed context, how to consolidate the data mining results obtained from multiple sources and also how to address data security issues. These challenges make the distributed data mining process

complex and complicated [4]. One way to deal with the complexities that may arise in distributed data mining is to reduce the volume of data that is processed at a time, this can be achieved by applying a decomposition technique [5]. Decomposition is the process of breaking down a complex classification task into several smaller, less complex and more manageable, sub-tasks that are solvable by using existing data mining tools, then joining their solutions together in order to solve the original problem [6].

Decomposition in data mining attempts to solve some of the complexities that may arise from mining in large databases by reducing the amount of data to be processed at a time. By breaking a task into several smaller sub-problems, decomposition is able to minimize training time and making it possible to apply standard data mining algorithms to large databases. Instead of getting a single and complicated model, decomposition methods create several sub-models, which are more datacomprehensible.

Decomposition methodologies in data mining provide the added benefits of: increased performance in terms of classification accuracy, simplification of the problem; enhanced feasibility for huge databases; clearer and more comprehensible results; reduced runtime by solving smaller problems and by using parallel/distributed computation; and the opportunity of using different techniques for individual subproblems [7].

Decisions trees when compared with other algorithms stands as a well-suited alternative for classification problems in data mining. Decision trees require less effort for data preparation during pre-processing, it is very intuitive and easy to explain to technical teams and other stakeholders.

Therefore,this research, provides an algorithm that constructs a decision tree and then applies a decomposition method to partition the features of the dataset into several smaller subsets and constructs a classifier for each of the subsets. The algorithm will be compared with other existing decision tree induction algorithms such as C4.5, CART and SPRINTto measure the classification accuracy and execution time across multiple datasets with varying sizes and number of observations.

The rest of the paper is organized as follows:section 2 describes the related works. section 3 describes the design of the proposed algorithm. section 4 shows some experimental results. Finally, section 5 presents the conclusions and future work.

II. RELATED LITERATURES

Several methods have been developed fordistributed data mining, including the use of decision trees. Over the years, several decision tree algorithms have been developed that are particularly useful for solving classification problems in data mining. They include ID3, C4.5, C5.0 etc, these algorithms however, have to keep in memory the entire training set for building a decision tree.

Some other algorithms have been developed for building decision trees from large training sets, such as SLIQ, SPRINT, CLOUDS, Rainforest, these algorithms however, use lists in keeping a dataset in the main memory[8].

The proposed algorithm focuses on minimizing the computing complexities associated with distributed data mining. The algorithm builds a decision tree based on the CART algorithm; it applies a decomposition method using genetic algorithm which partitions the features of the dataset into multiple feature subset and a classifier is built for each feature subset.

III. SYSTEM DESIGN

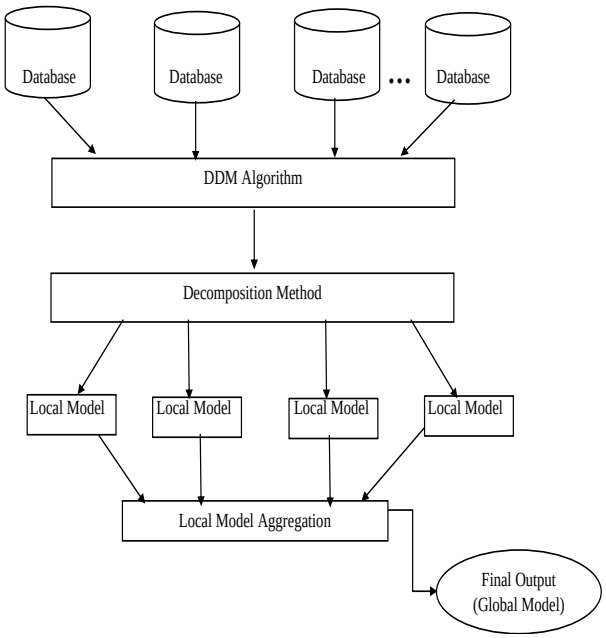

Figure 1. Architectural Design of the Proposed System

Database: This is where the datasets are stored. The datasets are retrieved for the data mining application and the results are relayed back to the databases after the mining process.
DDM Algorithm: The algorithm constructs a decision tree out of the dataset received from the database in order to extract meaningful patterns.
Decomposition Tool: The decomposition tool will partition the attributes(features) of the tree into smaller subsets. The DDM algorithm will then construct a classifier for each feature subset.

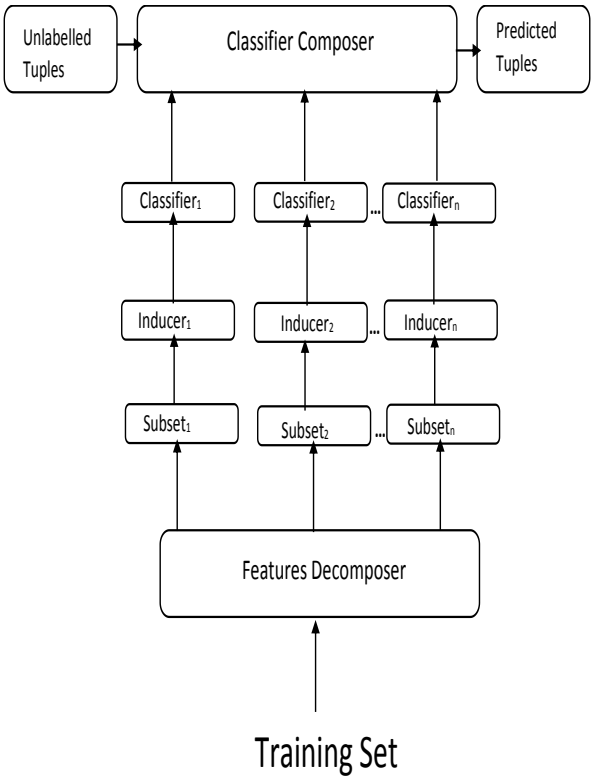

Figure 2. Decomposition Tool Design

Training Set: This refers to the dataset that is retrieved from the databases. The datasets are retrieved for the data mining application and the results are relayed back to the databases after the mining process.
Feature Decomposer: This is the algorithm that partitions the features of the dataset into several smaller feature subset. A genetic algorithm will be used in the system for feature decomposition.
Subset: The subset refers to a portion of the data based on a group of selected features.
Inducer: It provides a common platform for analysing the decision tree algorithm.
Classifier: This learns the various categories or classes in the dataset, and identifies what attributes belong to the various classes.
Unlabelled Tuples: Tuples represent a list of ordered elements, the unlabelled tuples are the elements in their raw form, that is, just be they are classified.
Predicted Tuples: These are the list of elements in a dataset that have been identified and categorized into the various classes.
Classifier Composer: The classifier composer integrates the individual subset classifiers into one to form a global decision tree.
Local Model: During the distributed data mining process, each site takes a portion of the dataset and mines it or performs a section of the mining task on a replicated data. The result generated from the individual sites are called local models.
Aggregated Local Model: The results from the individual sites (local models) are combined to form an aggregated local model.
Final Output (Global Model): The aggregated local model is sent back to the database as the final output (global model). The overall classification algorithm is shown below:
Output: A decision tree – DTClassifier
Method:
(1) create a node N;
(2) if tuples in D are all of the same class, C, then
(3) return N as a leaf node labeled with the class C;
(4) if attribute list is empty then
(5) return N as a leaf node labeled with the majority class in D; // majority voting
(6) apply Attribute selection method(D, attribute list) to find the bestsplitting criterion;
(7) label node N with splitting criterion;
(8) if splitting attribute is discrete-valued andmultiway splits allowed then // not restricted to binary trees
(9) attribute list attribute list splitting attribute; // remove splitting attribute
(10) for each outcome j of splitting criterion
// partition the tuples and grow subtrees for each partition
(11) let Dj be the set of data tuples in D satisfying outcome j; // a partition
(12) if Dj is empty then
(13) attach a leaf labeled with the majority class in D to node N;
(14) else attach the node returned by Generate decision tree(Dj, attribute list) to node N;
End for
(15) return N;

IV. RESULTS AND DISCUSSIONS

To demonstrate the behaviour of the algorithm, the execution time, accuracy rate and the algorithm's ability to scale across several datasetswith varying increasing data sizes; breast cancer dataset with 569 observations were evaluated [9], churn prediction dataset with 10,000 [10] observations and credit card payment default dataset with 30,000 observations [11]. We compared the obtained results of the proposed algorithm, DTClassifier with some already existing decision tree induction algorithms. C4.5, CART, SPRINT. The experiments were carried out on an Intel core i5 processor with 4GB RAM, running on Microsoft Windows 7.

First, accuracy across three different sizes was measured, i.e., number of observations.As, the size increases, the classification accuracy for the existing algorithms decreases, as these algorithms are not able to cater for large datasets. However, DTClassifier algorithm on the other hand, maintains steady classification accuracy and even increases slightly as the data size increases.

Next, the processing/execution time for the same algorithmswere measured, using the same datasets and it was observed that though the execution time grew longer as the data size increases for all the algorithms, our algorithm took the longest time.

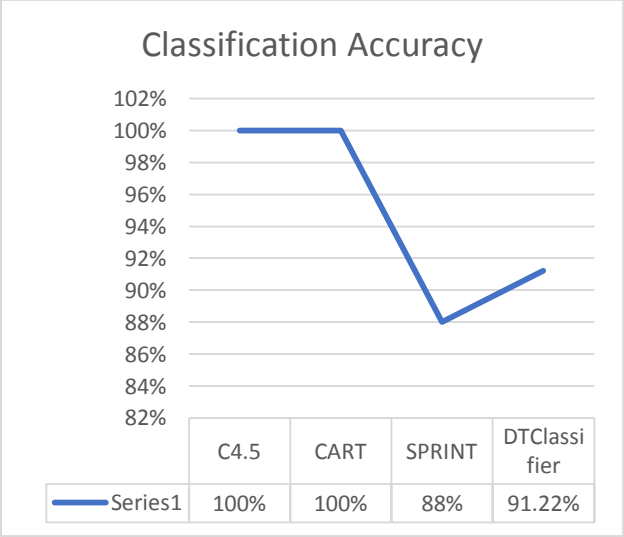

Figure 3. Classification Accuracy for Breast Cancer dataset.

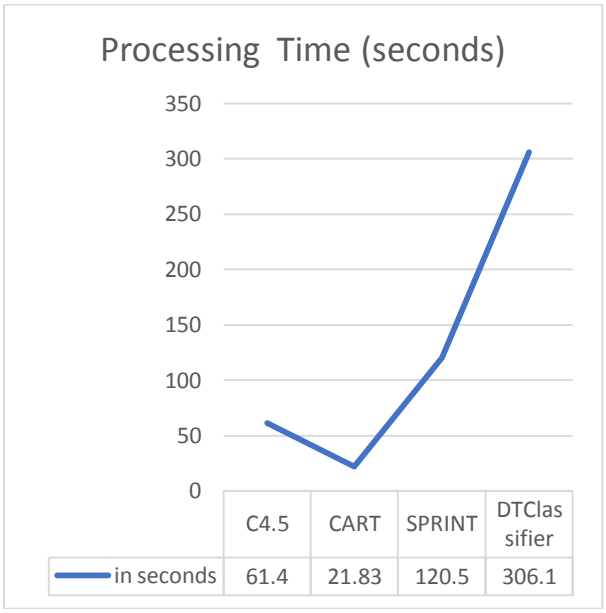

Figure 4. Processing Time for Breast Cancer Dataset

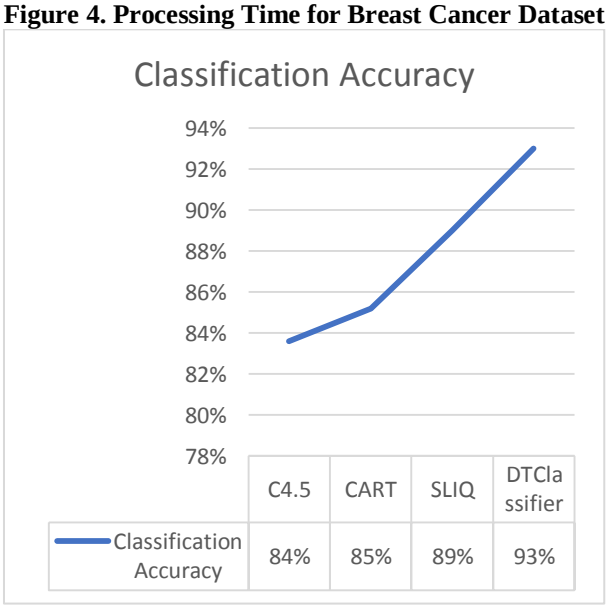

Figure 5. Classification Accuracy for Churn Dataset

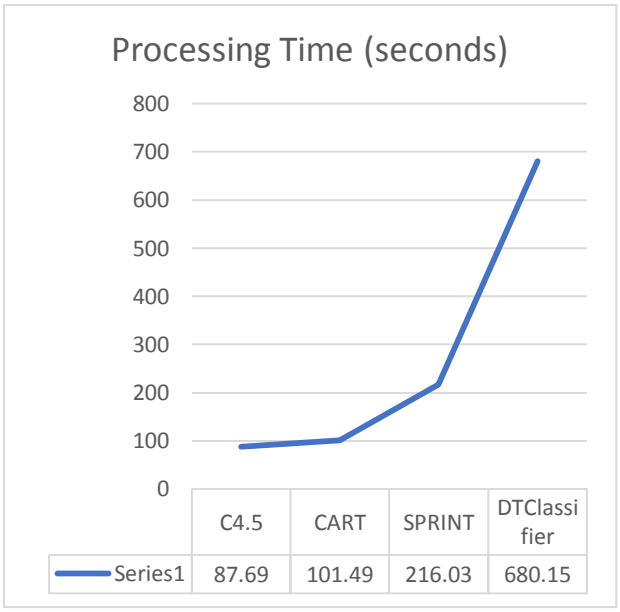

Figure 6. Processing Time for Churn Dataset

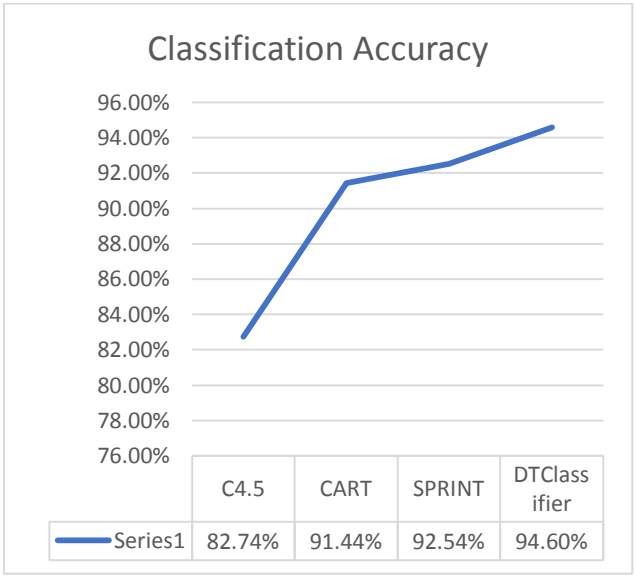

Figure 7. Classification Accuracy for Credit Card Dataset

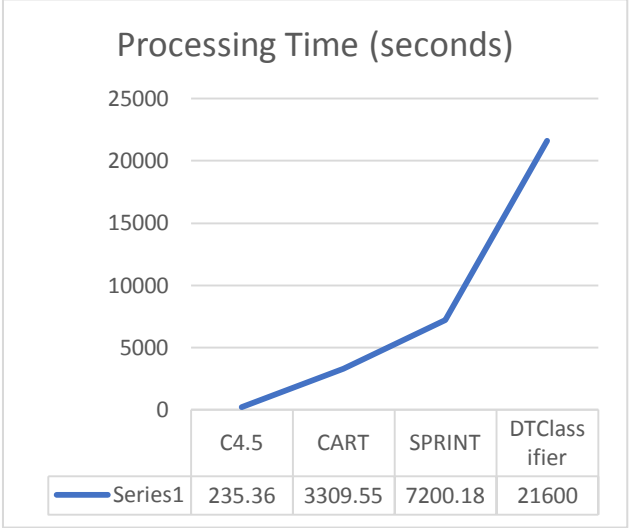

Figure 8. Classification Accuracy for Credit Card Dataset

V. CONCLUSIONAND RECOMMENDATIONS

One of the most important performance measures for a classification is its classification accuracy. From our experiments, our proposed algorithm outperforms all other decision tree algorithms in terms of classification accuracy. However, there is a very serious lag in its execution time when compared to the other algorithms.

Data mining remains is an ongoing and active research area, we recommend that future work research areas should dive deeper into various decomposition techniques as a means to improve the data mining process, especially for large datasets.

REFERENCES

[1]. Rekha, S. T., Sabu, T. M. (2012) Survey of distributed data mining in P2P networks.

[2]. Zaki, M. J. (2000). Large- scale parallel data mining: Parallel and Distributed Data Mining, an introduction. *Springer-Verlag*, pp 1 – 23.

[3]. Urmela, S. & Nandhini, M. (2017). Approaches and techniques of distributed data mining: A comprehensive study. *International Journal of Engineering and Technology*. *9*(1), 63 – 76, DOI: 10.21817/ijet/2017/v9i1/170901408

[4]. Osah, S. &, Bennett, E. O. (2019): A Framework for Implementation of Load Balance Access in Computation(Server) Environment.*International Journal of Computer Science and Mathematical Theory. Vol 5 (1), 56-63*

[5]. Kusiak, A. (2000). Decomposition in Data Mining: An Industrial Case Study, IEEE Transactions on Electronics Packaging Manufacturing, *23(4)*, 345 – 353.

[6]. Rokach, L. (2006). Decomposition methodology for classification tasks – a meta decomposer framework. *Pattern Analysis and Applications.* DOI: 10.1007/s10044-006-0041-y

[7]. Rokach, L. (2008) Decomposition methodology in data mining.

[8]. Ezimora, O. A., & Bennett, E. O. (2019): An Efficient Concept Mining Method of Unstructured Data with Semantics. *Journal of Scientific and Engineering Research. 6 (9), 184-191*

[9]. Franco-Arcega1, A., Carrasco-Ochoa, J. A., Sánchez-Díaz, G., & Martínez-Trinidad, J. A. (2013). Decision Tree based Classifiers for Large Datasets. Vol 17(1), pp 95 – 102.

[10]. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

[11]. https://www.kaggle.com/datasets/santoshd3/bank-customers