# Parallel and Scalable Map Reduce and Pipeline Tree Classifiers for Massive Dataset Using Map Reduce and Data Flow Pipeline

A. M. James Raj[1], J. Prema[1], P. Xavier[2] and F. Sagayaraj Francis[1]

[1] Department of Computer Science and Engineering, Pondicherry Engineering College, Pondicherry University, Pondicherry, 605 014, India

[2] Department of Computer Applications, Sacred Heart College, Tirupattur, 635 601, India

## Abstract

One of the important research areas in today's scenario is classification of Big Data. While there are a lot of traditional classification methods, extending them to Big Data is quite challenging. Decision Tree Classifier is one of the effective traditional classification techniques. The combination of Hadoop and Map Reduce has been adapted by many researchers both commercially and academically to process Big Data. Of late, Google cloud dataflow paradigm has sneaked into the Big Data scenario that augments the earlier systems with stream processing. This paper presents two algorithms based on Map Reduce and Google cloud data flow for implementing decision trees for classification is presented. The performances of both algorithms on various parameters have been compared and presented.

***Keywords:*** *Decision Tree, Hadoop Distributed File System, Map Reduce Classifier, Pipeline Tree Classifier, Google Dataflow.*

## 1. Introduction

Big Data refers to a collection of technologies that deals with large volume of data that are normally beyond the capabilities of commonly used software tools to capture, manage, and process data within a tolerable elapsed time [1]. Hadoop is one of the main software framework for handling big data using MapReduce. Hadoop is an open-source for storing data and running applications on clusters of commodity with high processing power and the ability to handle virtually limitless concurrent tasks or jobs [2]. The main advantage of using Hadoop is ability to store and process huge amount of mixed data quickly, with high computing power, less fault tolerance, high flexibility, low cost, and scalability. The main two parts of Hadoop is Hadoop Distributed File System (HDFS) and Map Reduce. HDFS is the storage component of the Hadoop framework, which is designed for maintaining and processing huge datasets efficiently among cluster nodes. Hadoop takes care of running code across a cluster of machines by HDFS [3]. The strength of Hadoop is that it has built in fault tolerance and fault compensation capabilities.

Map Reduce is a programming paradigm that deals with a massive scalability across hundreds or thousands of servers in Hadoop cluster [4]. Map Reduce, originally designed and implemented by Google is a distributed programming model for processing and generating large data sets [5]. The term Map Reduce actually refers to two separate and distinct tasks that Hadoop program performs [6]. The first is the *map* job, which takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Next the *reduce* job takes the output from a map as input and combines those data tuples into a smaller set to tuples. In a Hadoop cluster, a Map Reduce program is referred to as job. A job is executed by subsequently breaking it down into pieces called tasks.

Google Dataflow is a unified programming model for developing and executing a wide range of data processing patterns. It includes batch and continuous computation. It frees from an operational tasks like resource management and performance optimization. Dataflow resources are allocated on-demand, providing a nearly limitless resource capacity to solve any big data processing challenges. Dataflow provides programming primitives such as powerful windowing and correctness controls that can be applied across both batch and stream based data sources. This model effectively eliminates switching cost between batch and continuous stream processing by enabling developers to express computational requirements regardless of data source.

A pipeline encapsulates an entire series of computations that accepts some input data from external sources, transforms that data to provide a useful intelligence, and produces some output data. That output data is often written to an external data sink. The input source and output sink can be the same, or they can be of different types, allowing you to easily convert data from one format to another. Each pipeline represents a single, potentially repeatable job, from start to finish, in the Dataflow service.

Classification is a form of data mining that constructs models describing important data classes called as classifiers. Classification is carried out in two steps. The first is the learning step and the next is the classification step. In the learning step a classification model is constructed by describing a predetermined set of data classes or concepts. Subsequently the classification step uses the classes constructed in the first step to classify the further inputs.

Since the Big data mining is the recent technology, the research focus on this domain is very less and hence the traditional approaches are often inadequate and complicated to process large volume of data [7]. In this paper two efficient classification approaches based on the parallel computing Map Reduce programming model and data flow model are presented. Their performances are also evaluated and the results are tabulated.

## 2. Overview of Classifiers

Data mining is the analysis of observational datasets to find unsuspected relationship and to summarize large amounts of data in novel ways that are both understandable and useful to data owner in proactive decision making. Data mining software allows users to analyze data from different dimensions or angles and to categorize them. It is the process of finding correlations or patterns among dozens of fields in large databases. There are various kinds of data mining tasks like association rule, classification, clustering, prediction, etc., used to extract meaningful information. The classification methods were used to extract nuggets of knowledge from large sets of data [8]. Among the many classification algorithms that exist, some are based on Decision tree induction (ID3, C4.5,CART), Bayes classification, Rule based classification [9].

### 2.1 ID3 Algorithm

ID3 is one of the decision tree induction method for classification. A decision tree is a flowchart like tree structure, where each internal node (non-leaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The topmost node in a tree is the root node. Given a tuple X, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple. Decision trees can easily be converted to classification rules [2][10].

Constructing decision trees do not require any domain knowledge or parameter setting, and hence it is appropriate for exploratory knowledge discovery. Decision trees can handle multidimensional data. Their representation of acquired knowledge in tree form is intuitive and generally easy to assimilate by humans. The learning and classification steps of decision tree induction are simple and fast. In general, the decision tree classifiers have good accuracy but, successful use may depend on the data at hand [11]. Decision tree induction algorithms have been used for classification in many applications such as medicine, manufacturing and production, financial analysis, astronomy and molecular biology.

### 2.2 C4.5 Algorithm

C4.5 is also one of the decision tree induction algorithm adopts a greedy approach in which decision trees are constructed in a top down recursive divide and conquer manner. Decision tree induction algorithm starts with a training set of tuples and their associated class labels. The training set is recursively partitioned into smaller subsets as the tree is being built. C4.5, a successor of ID3, uses an extension of information gain known as gain ratio [12].

#### 2.2.1 Classifiers with Map Reduce Model

The map reduce concept is incorporated in the existing classifiers in order to make the classification process simple and sophisticated one. This C4.5 algorithm for Map Reduce uses three data structures [13]. They are Attribute table, Count Table and Hash table. An Attribute table contains the attributes or feature extraction information for the map reduce model. Count table computes the number of instances with specific class labels if split by attribute. Hash table stores the link information between tree nodes as well as the link between parent node and its branches.

The entire process of combining map reduce concept with C4.5 algorithm is carried out in four steps: Initially, it prepares the required data for the proposed work, i.e. The given Web KB data set is converted into the above said 3 data structures for Map Reduce. Map function transforms the input into attribute/feature table. Reduce function computes the number of instances with specific class labels into count table. Next, the selection step has been performed with one map and two reduce functions. First the reduce function takes the number of instances for each attribute/value pair to aggregate the total size of records for given attribute. Next, map function is called to compute the information and then the reduce function computes the information gain ratio and the maximum of Gain ratio value will be selected as attribute for splitting the tree. The update step is followed by the selection process in order to update count and hash tables. Finally, tree growing step is utilized to obtain a complete decision trees by establishing link between nodes.

### 2.3 Information gain and gain ratio

When the rapid growth property is enabled node splits are determined by information gain ratio instead of information gain. The information gain chooses a split based on which attribute provides the greatest information gain. It can be calculated by the Eq. (1) as.

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i) \qquad (1)$$

where, $p_i$ is the nonzero probability that an arbitrary tuple in D.

The information gain ratio method provides the value of a split and in general the greatest information gain ratio is chosen to split the decision tree. In other words, the attribute with the maximum gain ratio is selected as the splitting attribute [14]. ID3 is a decision tree algorithm (Iterative Dichotomizing) which is modified and utilized in the work depicted in Fig.1. The potential information generated by splitting the training data set T into v partitions using Info (D), corresponding to the v outcomes of a test on attributes A. Information gain using a split information value defined analogously with Info (D) in Eq. (2) as,

$$SplitInfoA(D) = -\sum_{j=1}^{v}\left(\frac{|D_j|}{|D|} * Info\left(\frac{|D_j|}{|D|}\right)\right) \qquad (2)$$

The gain ratio is defined in Eq. (3) as,

$$Gain\,Ratio(A) = \frac{Gain(A)}{SplitInfoA(D)} \qquad (3)$$

**Input :**Training dataset T with S attributes.
**Output:**Decision tree.
// Info (D) is the expected information needed to classify a tuple in D and also known as the entropy(D).
// InfoA (D) is the expected information required to classify a tuple from D based on the partitioning by A.
// Gain(A) is the information gain for all attributes A, and the difference of the new information (i.e. obtained after partitioning on A) and the original one //

```
i.  If (S=1) then
        Single node tree S.
    else
        Calculate Info(D) using Eqn(1).
ii. Find the entropy for all the attributes in the table using Eqn(2).
iii. Calculate Information gain, Gain(A)= Info(D) - InfoA(D).
iv. Choose the attribute for splitting which has the highest
    information gain and split the training dataset.
v. Repeat the steps i to iv for all the divided sub training dataset.
```

Fig. 1 Proposed ID3 algorithm.

## 3. Proposed Classification Methods

### 3.1 MR Tree Classifier

Applying the data mining methods on Big data is difficult and uncomfortable only by using traditional statistical methods because of its large and complex datasets [7] [15]. So that in the proposed work decision trees are constructed using Map Reduce classification algorithm to mine the big data set by 5 Mappers and 5 Reducers. Fig. 2 shows the block diagram of MR tree classifier and its procedure is given in Fig. 3.

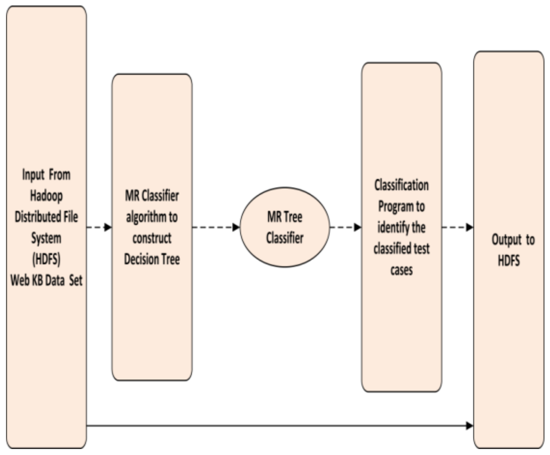
Input From Hadoop Distributed File System (HDFS) Web KB Data Set
MR Classifier algorithm to construct Decision Tree
MR Tree Classifier
Classification Program to identify the classified test cases
Output to HDFS

Fig. 2 MR tree classifier.

1. Mapper 1: Assigns different key value for all the features in the dataset.
2. Reducer 1: Counts the number of target class labels of the dataset and calculates the Info (D) and InfoA (D).
3. Mapper 2: Assigns a same key for all Info (D) and InfoA (D).
4. Reducer 2: Information gain is calculated and attribute is selected by the maximum information gain.
5. Mapper 3: The divided dataset is collected in different keys.
6. Reducer 3: Checks the completeness of the tree. If not complete, then the reducer1 task is applied for the divided datasets.
7. Mapper 4: The key is reassigned for all datasets.
8. Reducer 4: Information gain is calculated and the tree is grown by identifying the non-leaf node (i.e., the attribute which has highest information gain value). The dataset is divided according the selected non-leaf node.
9. Mapper 5: The key is assigned for all divided datasets.
10. Reducer 5: Checks the completeness of the tree. If not repeat the procedure till the tree gets completed.

Fig.3 Procedure for MR tree classifier.

## 3.2 Pipeline Tree Classifier

Pipeline tree classifier using Google dataflow is also implemented by two phases, viz., training and classification. In training, decision tree is constructed using Pipeline method using modified ID3 algorithm presented in Fig.1 and the given data set is executed with the help of 4 transforms. The Pipeline Tree construction procedure is summarized below and its corresponding dataflow is depicted in Fig. 4 and 5.

1.Create P collection

The given Web KB dataset is created as a PCollection to perform various transforms in pipeline. A new pipeline can be created by the following the syntax

```
Pipeline(pipeline_object)=Pipeline.create(PipelineOptions
Factory.fromArgs(args). withValidation().create());
```

2. Level- 1 Transform

It counts the number of class labels of each and every value in each and every attribute of the dataset and then this transform calculates the Info(D) and InfoA(D).

3. Level-2 Transform

In which, all the operation of level-1 transform is calculated for each and every dataset split. Check for the completeness of tree. If not identify the maximum information gain and split the dataset. Now second level of the tree is formed.

4. Level- 3 Transform

It checks the completeness of the tree. Split the dataset using split command and check every class of the record in the particular dataset is same, then that path of the tree is completed.

5. Print Transform

Print transform is used to log the outputs either in separate file or output console.

6. Run the pipeline

After forming the pipeline, and run the pipeline by using pipeline-object.run() method to start the pipeline service.

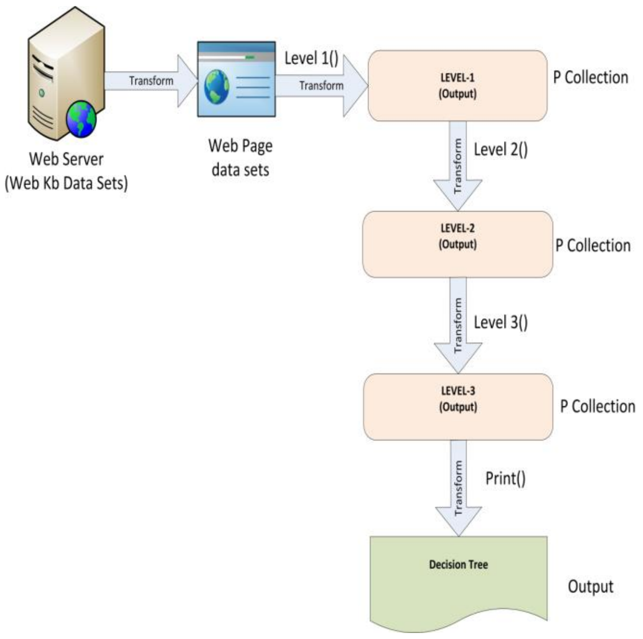

Fig. 4 Dataflow of pipeline tree.

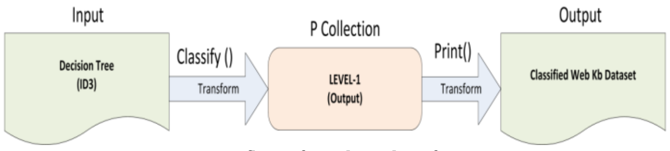

Fig. 5 Dataflow of pipeline classification.

## 4. Experiments

### 4.1 Data Sets

To construct the decision trees for classification using map reduce and pipeline tree methods, the standard Web KB data set is used. It is a benchmarking data set for machine learning problems. Information gain value is utilized to construct decision trees by which the unclassified test cases are identified. The sample outputs and evaluation results are also discussed. With the help of the decision tree, classes are identified for the test case inputs and they are scalable.

In the proposed work two decision tree algorithms were constructed and their performances were analyzed. Initially, Map Reduce decision tree was constructed using the data set by 5 Mappers and 5 Reducers and then Pipeline tree classifier was implemented in Google dataflow programming model. In an Improved version of ID3 Pipeline classifier algorithm the results are shown that it supports scalability when the datasets are increased. But the traditional version of ID3 and C4.5 without pipeline is not scalable and cannot handle massive data (big data). The rule based algorithm is also appropriate for only less number of datasets. The ID3 algorithm with Map Reduce algorithm is too complex because it has many mapper and reducer implementation and its computation time is high. On the other hand ID3 pipeline tree algorithm is all about collections and transforms and however it is very simple and also it takes less computation time when compared to MR Tree algorithm.

## 4.2 Performance Evaluation Metrics

The standard performance evaluation metrics generally used in classification work to evaluate the results are precision, recall, F-measures and accuracy. They are calculated as follows [16] [17].

**Precision (p)**

Precision is the ratio of the number of relevant Web documents that were retrieved and the total number of retrieved web documents. It is measured by the formula.

$$Precision(p) = \frac{TP}{TP + FP} \qquad (4)$$

**Recall (r)**

Recall is the ratio between the number of relevant web documents retrieved and the total number of relevant web documents. It is measured by the following formula.

$$Recall(r) = \frac{TP}{TP+FN} \qquad (5)$$

**F-measure (f)**

It can be employed in Information retrieval to test the performance of the classifier. It is the harmonic mean of precision and recall. It can be calculated by the formula.

$$F-measure(f) = \frac{2\times p\times r}{p+r} \qquad (6)$$

## 4.3 Experimental Results

The results obtained in the experiments show that the computation time is very less in Pipeline Tree Classifier when compared with the MR Tree Classifier. Moreover, the performance of Pipeline Tree classifier is significantly increased when compared to the MR Tree Classifier which is tabulated in Table 1.

Table 1: Comparison of pipeline and MR tree classifiers

| ***Name*** | ***Pipeline tree*** | MR tree |
|---|---|---|
| Correctly Classified Instances % | 97.81 | 94.21 |
| Precision | 98.81 | 96.61 |
| Recall | 97.71 | 93.13 |
| F-measure | 98.23 | 93.84 |
| Computation | Less | High |

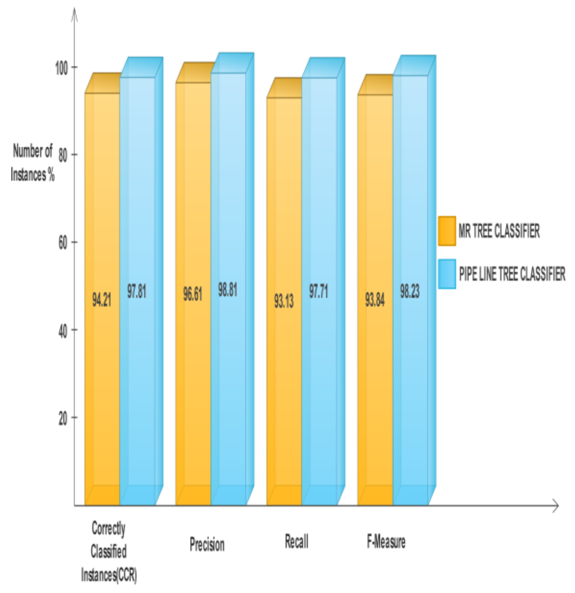

Fig. 6 Comparison of Map Reduce and Pipeline Tree Classifiers performance

Fig. 6 compares the performance of Pipeline Tree Classifier with Map Reduce Classifier using different number of instances. From the above table and figure, one can make the following observations. First of all, when the dataset increases, computation time also increases in both MR Tree and pipeline tree classifiers for constructing decision trees. Second, the computation time of Pipeline classifier is much less than the Map Reduce tree classifier even when the dataset increases exponentially. Thirdly, the Map Reduce implementation uses too many Mapper and Reducer Procedures which leads to high computation time. So that more number of Job configuration is required. Finally, MR Tree method is more complicated one to understand whereas Pipeline Tree Classifier uses minimum number of collections and transforms and also provides better results.

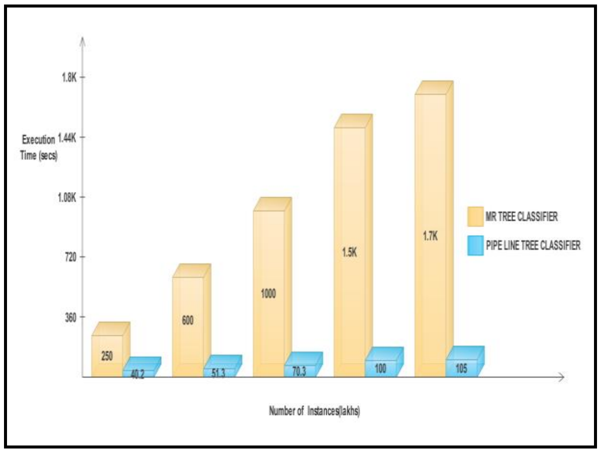

Fig. 7 Comparison of Map Reduce and Pipeline Tree classifiers computation time

Fig. 7 depicts that when the data set increased, the classification time and the time taken to read the input file is also increased in MR tree classifier and also the computation time is higher than pipeline tree classifier.

## 5. Conclusion

In this work, two classifiers, based on Map Reduce and Pipeline paradigms were developed and they handled the massive dataset with large number of attributes efficiently and brought down the computation time significantly. In the first experiment, a decision tree was constructed for classification with the help of Map Reduce method using Hadoop. The MR Tree classification model was tested using the WebKB data sets. The results show that the method is scalable for large data sets. This MR Tree Map Reduce algorithm is best suitable for less number of attributes (features) because if the dataset has more number of attributes, then obviously the depth of the tree is increased. Moreover, number of mapper and reducers are increased and it is a tedious task to construct the decision trees. In order to increase the efficiency of this work another classifier named as Pipeline tree using Google Dataflow was developed which can handle massive dataset with any number attributes with less execution time. Thus the pipeline tree classifiers computation time is lesser than the MR Tree classifier.

## References

[1] Karthik Kambatla, GiorgosKollias, Vipin Kumar, Ananth Grama, “Trends in Big Data Analytics”, Elsevier Journal of Parallel and Distributed Computing, No.74, 2014.
[2] Ishwarappa, Anuradha J, “A Brief Introduction of Big data 5 V’s Chararacteristics, Hadoop Technology”, Elsevier – International conference on intelligent computing, communication and convergence, No .48, 2015,pp.319-324.
[3] R.A.Fadnavis, Samrudhi Tabhane, ” Big Data Processing Using Hadoop”, International Journal of Computer Science and Information Technologies,Vol.6, No.1,2015,pp.443-445.
[4] ShahrukhTeli, PrashastiKanikar, “A Survey on Decision Tree Based Approaches in Data Mining”, International Journal of Advanced Research in Computer Science and Software Engineering, Vol.5, No.4, 2015,pp.613-617.
[5] Xiao guang Qi and Brain D. Davidson, “Web Page Classification. Features and Algorithms”, ACM Computing Surveys, Vol.41, No.2, 2009.
[6] Hwanjo Yu, Jiawei Han and Kevin chen chuan Chang, “PEBL: Web Page Classification without Negative Examples”, IEEE Transactions on Knowledge and Data Engineering, Vol.16, No.1, 2004,pp.71-81.
[7] Prashasti Kanikar, “Data mining with Big data”, IEEE transactions and Data Engineering , Vol.26, No.1, 2014.
[8] Malarvizhi et.al. “Distributed approach to web page Categorization using Map-Reduce programming model”, International Journal of Engineering and Technology, Vol.3, No.1, 2012,pp.373-386.
[9] Thair Nu Phyu, “Survey of Classification Techniques in Data Mining”, International Multi Conference of Engineers and Computer Scientists, 2009, Vol. 1.
[10] Aytug Onan, “Classifier and feature set ensembles for web page classification”, Journal of Information Science, Vol.42, No.2, 2015.
[11] Wei Dai and Wei Ji, “A Map Reduce Implementation of C4.5 Decision Tree Algorithm”, International Journal of Database Theory and Application, Vol.7, No.1, 2014, pp.49-60.
[12] Bhardwaj et. al, “Implementation of ID3 algorithm”, International Journal of Advanced Research in Computer Science and Software Engineering, Vol.3, No.6, 2013, pp.845-851.
[13] Seyed Reza Pakie, AbolfazlGandomi, “Comparative Study of Classification Algorithms Based on Map Reduce Model”, International Journal of Innovative Research in Advanced Engineering, Vol.1, No.7, 2014, pp.251-254.
[14] Kumar Ashok, Taneja H C, Chitkara Ashok K and Kumar Vikas, “ Classification of Census Using Information Theoretic Measure Based ID3 Algorithm”, International Journal of Mathematics Analysis, Vol.6, No.51, 2014, pp.2511-2518.
[15] Chris Snijders, Uwe Matzat, Ulf-Dietrich Reips, “Big Data: Big Gaps of Knowledge in the Field of Internet Science”, International Journal of Internet Science, Vol.7,No.1, 2012, pp.1-5.
[16] A.M. James Raj, F. Sagayaraj Francis, P.Julian Benadit, “Optimal Web Page Classification Technique Based on Informative Content Extraction and FA-NBC”, Computer Science and Engineering,Vol.6, No.1, 2016 , pp.7-13.
[17] K.Pranitha Kumari et. al “Performance Improvement of Web Page Genre Classification”, International Journal of Computer Applications, Vol.53, No.10, 2012, pp.24-27.

**A.M.James Raj,** received M.Sc. in Computer Science from Bharathidasan University,Trichy and M.Phil from Alagappa Univeristy, Karaikudi from Tamil Nadu, India and M.Tech in Information Technology from AAI–DU Allahabad, India. He also cleared National Eligibility Test (NET), a qualifying examination for college/university professors, conducted by central Government of India. Presently he is working as an associate professor in computer science and applications in Pope John Paul II College of Education, affiliated to Pondicherry University. His research interest includes in Data Mining, in particular Web mining and Data Bases.

**J Prema,** received B.Tech. and M.Tech. degrees from Pondicherry University and currently working at Tata Consultancy Services, Chennai.

**P. Xavier** obtained his Ph.D. degree from Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya University, Kanchipuram and is currently working as a Professor of Computer Applications at Sacred Heart College, Tirupattur, Tamil Nadu, India.

**F. Sagayaraj Francis** obtained his Ph.D. degree from Pondicherry University and is currently working as a Professor of Computer Science and Engineering, Pondicherry Engineering College, Pondicherry, India.