Fuzzy Information and Engineering

ISSN: 1616-8658 (Print) 1616-8666 (Online) Journal homepage: https://www.tandfonline.com/loi/tfie20

A MapReduce C4.5 Decision Tree Algorithm Based on Fuzzy Rule-Based System

Fatima Es-sabery & Abdellatif Hair

To cite this article: Fatima Es-sabery & Abdellatif Hair (2020): A MapReduce C4.5 Decision Tree Algorithm Based on Fuzzy Rule-Based System, Fuzzy Information and Engineering, DOI: 10.1080/16168658.2020.1756099

To link to this article: https://doi.org/10.1080/16168658.2020.1756099

Published online: 23 Jun 2020.

FUZZY INFORMATION AND ENGINEERING
https://doi.org/10.1080/16168658.2020.1756099

Taylor & Francis
Taylor & Francis Group

ORIGINAL ARTICLE

OPEN ACCESS Check for updates

A MapReduce C4.5 Decision Tree Algorithm Based on Fuzzy Rule-Based System

Fatima Es-sabery and Abdellatif Hair

Faculty of Sciences and Technology, Sultan Moulay Slimane University, Beni Mellal, Morocco

ABSTRACT

Decision tree is the most efficient and fast technology of data mining that is frequently used in data analysis and prediction. According to the development in science and technology in the last years, the data is growing faster, and the principle of the decision tree algorithms become not efficient in respect runtime and speed-up ratio. In view of the above problem, we propose a new method of classification based on framework Hadoop and Fuzzy logic. Our proposed hybrid approach is designed to propose a new C4.5 decision tree algorithm using fuzzy logic and fuzzy set theory to handle uncertainty and imprecision in data, and Hadoop framework (MapReduce + HDFS) to parallelize our work. This combination of big data technologies, fuzzy systems and C4.5 decision tree algorithm has produced a parallel fuzzy decision tree model, which takes advantage of these three techniques (hadoop + fuzzy logic + C4.5) to produce a decision tree with higher predictive accuracy. In this paper, an experiment is presented to compare our approach with other approaches from the literature. Experiments were carried out using three datasets, and the results show that our new method outperforms the other approaches in terms of accuracy and execution time.

ARTICLE HISTORY
Received 27 January 2020
Revised 25 March 2020
Accepted 2 April 2020

KEYWORDS
BigData; fuzzy logic; decision tree; MapReduce; HDFS; Hadoop

1. Introduction

The classification component is the primary technique in data mining and vastly used in diverse areas. Classification is a data mining function that attributes items in a collection to decision categories or classes. Generally, the dominant principle of classification is to accurately predict the decision or target class for each element in the dataset by using the constructed model [1]. The historical data for a classification project is characteristically split into two data sets: one for building the model; the other for testing the model. The most used classification algorithm is the decision tree algorithm [2].

A C4.5 decision tree algorithm is an oriented tree comprised of a root node, as well as decision nodes all the other nodes each with exactly one incoming edge. In order to construct a decision tree, the process is as follows: Given a dataset of training data, apply a measure function on all available attributes, find the better splitting attribute based on the

CONTACT Fatima Es-sabery fatima.essabery@gmail.com

result obtained by the calculation of measure function, once the best attribute is determined, the dataset is divided into numerous partitions according to the ranges of values or number of values associated with the best attribute. Within each partition, if all samples appertain to a single class, the algorithm stops [3,4]. Otherwise, the splitting procedure is recursively executed until each partition appertains to a single class, or no attribute is left. In this domain of scientific research, all researches deal with the problem of finding the better splitting criteria of decision tree algorithm in order to construct small, accurate trees, and to decrease execution time for a given dataset [1].

One of the excellent characteristics of the decision tree is that; it doesn't require a lot of background cognization in the learning procedure since the training dataset can donate expression by the attribute that is the conclusion of the model [3]. After this, use the algorithm for learning. Decision tree algorithms have advantages as follows: (1) the structure of the algorithm is simple, easy to comprehend; (2) the algorithm has high predictive accuracy. But nowadays, the traditional decision tree algorithms have encountered many challenges because of the faster growth of data. First, as the quantity of data becomes hugely massive, the process of constructing a decision tree model can be quite time-consuming. Second, several computations moved to external storage because the memory storage capacity is limited. Therefore expand the I/O cost. In our work, to overcome these challenges, we are used the big data framework Hadoop with its component MapReduce computational model and distributed file system HDFS.

Currently, big data is the capability of extracting useful patterns or information from large-scale data [5]. For handling this huge quantity of data using a single computer node it's inefficient in real-time. To resolve this problem the big data processing framework is deployed on cluster computers with a high-performance computing platform, and the data mining tasks are deployed on this cluster of computers by running the high-level data-parallel framework Hadoop. Apache Hadoop is an open-source software framework that efficaciously facilitates writing distributed applications. It contains two components, the distributed file system HDFS, and the MapReduce programming model.

HDFS is a distributed, portable and scalable file system written in Java. Up to now, it is a highly fault-tolerant storage system, which stores huge amounts of data reliably on multiple low-cost machines redundantly. Thus rescue the system from eventual subsequent data losses in case of failure [5,6]. The input data of a Hadoop job are stored as files in HDFS. Such as it stores the file metadata on the NameNode server and application data is stored on other servers called DataNodes. MapReduce is a style of parallel computing that has been deployed in multiple systems, which the computation in this model takes a set of input key/value pairs, and produces a set of output key/value pairs. The user specifies a map function that processes a set of input key/value pairs in order to generate a set of intermediate key/value pairs, finally, the reduce function merges all intermediate values associated with the same intermediate key. Programmes written in this functional style are automatically parallelised and executed on a large cluster of commodity computers [6,7].

Fuzzy Systems (FS) can be defined as systems that use the fuzzy set theory proposed by prof. Lofti A. Zadeh [8] to represent at least one of its variables. The fuzzy set theory allows the computational representation and processing of imprecise and uncertain information, which are abundant in the real world. In fact, most of the available computer approaches cannot directly process information with imprecision and uncertainty, making fuzzy systems a valuable alternative to work with domains presenting such characteristics. Rule-based fuzzy systems, a particular type of fuzzy system, use a reasoning mechanism based on approximate reasoning that has the ability to express the ambiguity and subjectivity present in human reasoning. The rule bases on fuzzy systems store knowledge represented by means of rules [9]. A fuzzy system consists of a Knowledge Base (KB) and an Inference Mechanism (IM). The KB contains a Fuzzy Rule Base (FRB) and a Fuzzy Database (FDB). The FRB has the rules that form the core of the system. These rules are constructed based on the fuzzy sets defining the attributes of the system, stored in the FDB. The FDB and FRB are used by the IM to classify new examples [9].

In this article we propose a new approach to classify the data, using the notions of Fuzzy Logic, C4.5 decision tree algorithm based on fuzzy information gain, and the open-source software Hadoop. The first step is to fuzzify the data to be classified (transform the crisp set to fuzzy set) using the fuzzification methods (trapezoidal shaped membership function or triangular membership function) and store it in HDFS. After the data is stored in HDFS we parallelise the instructions of the fuzzy C4.5 algorithm applied on data using the MapReduce programming model. We can deduce that the goal of our new method is to fuzzify the C4.5 algorithm in order to handle uncertainty and imprecision data, and in order to classify the huge dataset using this fuzzified algorithm without having the problem of the execution time, we parallelise our method using Hadoop framework.

The remainder of this paper is organised as follows: Section 2 defines some literature review. Section 3 describes the motivation of our work, Section 4 presents our research methodology. Section 5 describes the experiment results and comparisons, followed by the conclusions and future work in Section 6.

2. Related Works

Several research papers in the literature pursue to study, construe and identify the issues of text classification using fuzzy logic methods, and their applications in diverse areas [10–19]. Fuzzy logic (FL) [8,9,20] is one of the soft computing techniques that takes a crucial role in the construction of hybrid classification models in the last years. FL suggested by prof. Zadeh [8] explains the manner of representation of human thinking and perception especially in various scopes such as Datamining, Information abstraction, Machine Learning, Pattern Recognition, Natural Language processing, and other domains that resolves uncertainty problems. These ambiguous and uncertainty issues can be solved by different fuzzification methods that are applied to transform the input crisp set into fuzzified sets.

Ducange et al. [10] propose an effective distributed fuzzy associative classification model based on the MapReduce programming model. The first step of their approach aims to extract a set of fuzzy association classification rules using the fuzzy extension of the learning algorithm FP-Growth, then they prune the resulted set of rules through using tools of pruning such as fuzzysuppConfL, minFuzzysupp, and minFuzzyConf. The aims of this pruning process is to reduce the redundant and noise rules generated in the first phase of the proposed approach. They implemented their work using the Hadoop framework, also they study the scalability of their work by carrying out a lot of experiments on a real-world huge dataset.

Authors of the research paper [11] proposed a fuzzy system that can extract the principle aspects from tourist opinions and then classify these extracted aspects into the positive or negative category they employ algorithms based on fuzzy logic in both phases: aspect classification and aspects extraction. They evaluated five prevalent algorithms based on fuzzy logic, FURIA, FLR, FNN, VQNN, and FRNN in order to choose the best one. According to the presented result, the FURIA algorithm gave good results as compared to other fuzzy learning algorithms with the 90.12% accuracy on the restaurant's dataset and the FLR classifier achieved a better result with the 86.02% accuracy on the hotel's dataset. In general, their work is carrying out through four phases, data collection, data pre-processing, fuzzy rules extracted, and classification step using fuzzy logic algorithms.

Abdul-Jaleel et al. [12] proposed an approach combined genetic algorithm and theory of fuzzy logic to resolve the issue of text classification based on the membership degree. The inputs for their proposed classification application are a set of features obtained from a tweet and the outcome of this classification system is the class (negative, neutral, positive) which the tweet belonging to it. The results obtained from this proposed system are compared with the technique of fuzzy logic and the technique of keyword searches. This comparison is based on both rates, which are correction rate and incremental rate. In the incremental rate, their classification system is more efficient than these techniques (keyword search and fuzzy logic), where the number of tweets extracted using the proposed approach is 160 tweets compared to 98 and 141 using the other techniques. Also, the proposed classification system achieved a better result with the 98.75% correction rate compared to 97.9% and 95.7% correction rate obtained by other techniques.

Authors of [13] present a hybrid methodology to classify the soil using Munsell Soil Colour Charts. In their proposed approach, they resolve the issue of soil classification by combining Fuzzy Logic Systems and Artificial Neural Networks. Melin et al. [14] develop a new approach for dynamic parameter adaptation in particle swarm optimisation (PSO), where PSO is a metaheuristic inspired in social behaviours. The authors also in this work used fuzzy logic in order to ameliorate the variety and the convergence of the swarm in PSO. Experiment outcomes prove that their proposal gave good results in terms of the performance of PSO. The authors Rubio et al. [15] present a new clustering algorithm called Fuzzy Possibilistic C-Means (FPCM). This proposed algorithm is based on the technique of Type-2 Fuzzy Logic. The objective of this work is to improve the performance of the FPCM. Several simulations were made by applied the Interval Type-2 Fuzzy C-Means algorithm and FPCM on 6 well-known datasets. The authors of these research papers [16,18,19] proposed the new machine learning techniques to solve the issue of classification in some areas such as pattern recognition and diabetes disease classification.

Authors of [17] present a work that combines both the company's stakeholders and decision-makers in order to choose the better supplier. In their work, the authors convert the set of extracted opinions into a fuzzy soft set, then combine the obtained fuzzy soft set with the rough approximation theory. The attributes in this work are represented by linguistic terms. To evaluate the effectiveness and the performance of their proposed method, the authors gave a case study using their improved technique. Also, many works in the literature have exploited the possibility of combining the fuzzy set theory with the decision tree algorithms to handle uncertainty data. And these fuzzy Decision tree algorithms have been successfully used in several areas such as industrial applications, decision making, machine learning, knowledge engineering, and data mining. In this section, I will describe some of these research works.

Authors in [21] proposed a new fuzzy logic-based method for multi-label classification. The new algorithm utilises generalised fuzzy entropy, aggregate overall labels, to select the best attribute for growing the tree. The reasons adopted by the authors for improving this new fuzzy decision are two-fold: firstly, the ingrained interpretability of fuzzy systems give some anticipation or explication about the classification. Which is a very important feature in several knowledge discoveries and data mining tasks. Second, the new method has several degrees of ambiguity among the labels boundaries, which cannot be properly discovered by classical crisp classifiers.

Another work that uses fuzzy sets in the decision tree is that presented in [22]; in this article, the authors introduced an approach of using cumulative information estimations for fuzzy decision tree induction. They proposed a novel type of fuzzy decision tree called an ordered tree. This tree is used to process the attributes in a parallel manner with differing costs. Unordered tree dissents from ordered fuzzy decision tree in the manner of testing attributes. In the ordered tree the order of tested attribute is unrelated from the outcomes of preceding tests, therefore we can examine the next attributes in a parallel way. This leads to the diminishing of costs for test attributes.

Suryawanshi and Thakore [23] proposed a method that integrates fuzzy set theory with the ID3 decision tree algorithm. This paper essentially focuses on the classification method of data mining to recognise the class of an attribute using the ID3 decision tree algorithm, and then to add the fuzzification principle to ameliorate the performance of ID3.

Authors of [24] present a hybrid approach, which combines maximum ambiguity based sample selection and fuzzy decision tree induction. This paper introduces a novel sample selection technique, i.e. the maximum ambiguity-based sample selection in fuzzy decision tree induction. The experimental results show that the generalisation ability of the tree using this new selection method is more performance than that found on the random selection technique.

3. Motivation

The idea of fuzzy logic theory aims to analyse the collected data from different areas in a way that is similar to the human beings feelings [20], unlike traditional analysis strategy. The output of a fuzzy system is obtained through the application of the membership functions on both inputs and outputs, this process is called the fuzzification process. A crisp input will be transformed into the various members of the related membership functions founded on its value. Furthermore, the output of the fuzzy logic system is derived from its memberships of the various membership functions, which can be treated as a set of inputs [25].

Fuzzy logic ideas are often used in our routine life that none even pays attention to them. For example, to respond to a few questions in some surveys, in all the time the person could reply with ‘Not Satisfied’ or ‘Fully Satisfied’, that are also vague ambiguous or fuzzy answers. Precisely to what a degree is a person contented or discontented with certain products or services for those surveys. These ambiguous answers can only be created by human beings, but not machines [20]. Is it possible for a machine to respond to those survey questions immediately as human beings did? It is definitely impossible. Machines can only comprehend either ‘FALSE’ or ‘TRUE’, and ‘0’ or ‘1’. Those pieces of information are called crisp data and can be treated by all computers. Is it possible the human being help the machines to treat those vague data? If so, how can machines and computers treat those ambiguous data? Yes, inspired by human being feeling's, professor L. A. Zadeh proposed the fuzzy logic that can help the computers to handle those vague/ambiguous data as human beings do [8].

Fuzzy logic is considered as an extension of classical logic. in other words, the truth value takes a real number from the interval [0, 1] in fuzzy logic rather than a binary value '0' or '1' in classical logic. the main objective of the theory of fuzzy logic is converting a white and black problem into a grey issue [8]. In the definitions of set theory, classical or deterministic logic is considering the set of elements as the crisp set, which denotes that the membership degree of each element in a set is equal to 1 i.e. the element entirely belongs to the set. Unlike, fuzzy logic is considering the set of elements as the fuzzy set, which denotes that the membership degree of each element in a fuzzy set is ranged from 0 to 1, i.e. the element belongs partially to the set. The membership degree is computed by a specific membership function such as triangular membership function, Gaussian membership function, and trapezoidal membership function [26].

Generally, features in a learning process can be divided into two categories, namely, continuous-valued features and discrete-valued features. The first category is regarded as nominal concepts while the second, as real numbers. The C4.5 decision tree algorithm supposes that all feature values are nominal. Therefore the continuous-valued attributes should be discretized before the C4.5 measures the splitting criterion. there are several manners for discretization but an effective one is a binary split which denotes that a continuous-valued feature is discretized at the beginning of the learning algorithm process by dividing its range into two intervals [27] binary split is generally performed by selecting the threshold value which decreases the impurity measure (C4.5 gain ratio) utilised as the splitting criterion [28]. Once the threshold value T is determined for the continuous-valued attribute A, the instances of the training set with A $\leq$ T are assigned to the left node's branch, whereas the instances of the training set with A $>$ T are assigned to the right node's branch.

C4.5 handles continuous-valued features by putting real numbers into two different intervals using the binary split technique, each interval is utilised as a condition judgment by the current node toward the next node. In the literature, there are several research works [22,23,27,28] criticise this way of dealing with continuous-valued feature and consider it as judgment bias. Motivated by the effectiveness and advantage of fuzzy logic techniques to resolve the judgment bias problem in several problems, we proposed a new version of C4.5 by representing the continuous-valued features utilising fuzzy linguistic terms instead of the split binary technique. In the next section, I will describe how we use the fuzzy logic technique with the C4.5 algorithm to handle the continuous-valued features.

From the point of view of some research papers [28–30], the rule-based fuzzy system (RBFS) is the most important field of fuzzy sets theory. This kind of system is regarded as an extension of traditional rule-based systems, taking into consideration IF THEN rules whose consequent block and antecedent block are constituted of fuzzy logic terms, instead of traditional logic ones. As argued in [26] RBFS can raise the interpretability rate of learning algorithms for text-classification than computational models. Generally, the RBFS is a particular kind of expert systems, which typically be composed of a set of fuzzy rules. Each rule is a set of linguistic terms, which are called conditions or antecedents. In the literature, There are three common kinds of RBFS, namely Sugeno, Tsukamoto and

Mamdani [31]. Both Mamdani and Sugeno rule-based fuzzy systems are used in cases of regression problems, and the Tsukamoto rule-based fuzzy system generally used for classification problems. Tsukamoto consists of three phases, which are fuzzification, Inference, and defuzzification. In fuzzification step, the Tsukamoto use one of the three popular fuzzification functions, such as Triangular membership function, Gaussian membership function, and Trapezoidal membership function [26], the inference mechanism is based on expert knowledge, and in the defuzzification step, one of the most popular functions is used such as Max membership function, Centroid function, and Weighted average function. Similar to Tsukamoto model, our proposed method consists of three phases, such as fuzzification step performed by using triangular membership function, Inference step carried out by applying the Fuzzy C4.5 algorithm [32] to fuzzified dataset, and in the last phase, we applied the classic and general reasoning methods on extracted fuzzy rules to classify the new instances.

4. Research Methodology

Our proposal pursues to resolve one of the problems encounter by The C4.5 decision tree algorithm by using fuzzy logic techniques. The problem is how C4.5 handles continuous-valued attributes. Generally classical C4.5 uses the binary split to deal with continuous-valued attributes as explained in the motivation section. After numerous experiments, analysis and studies carried out by researchers, it turns out that the binary split technique is not more efficient and they consider it as judgment bias. Finding another way to overcome this judgment bias in the C4.5 learning process is the first phase of our proposal. After our studies of fuzzy logic theory, we deduced that this theory is more efficient to resolve the judgment bias problem in several problems as presented in the related works section. Therefore, we decided in the first phase of our proposal to fuzzify the dataset using the fuzzification techniques. This step allows us to improve the C4.5 decision tree, instead of the discretized process using the binary split technique for the continuous-valued attribute, we replace the continuous value of such attribute with the linguistic term with the highest membership degree with it.

In the second phase of our work, we propose a new rule-based fuzzy system to handle the uncertainty and imprecision data in the classification process. This system consists of three steps such as the fuzzification step presented earlier in the first phase of our proposal, the Inference phase, and the classification phase. The inference phase is the component that extracts the set of fuzzy rules from the fuzzified dataset according to the application of the parallel fuzzy C4.5 algorithm on the fuzzified data. The classification phase aims to classify new instances by using classic and general reasoning methods. Therefore, the integration of fuzzy logic (using the fuzzy linguistic term to represent the continuous-valued features) and rule-based fuzzy system (designed by parallel fuzzy C4.5 algorithm) can make rules appeared in a form that is extremely identical to natural language and can thus make the knowledge generated from rules more interpretable and understandable.

For more details, in this section, we going to present the different steps of our work and to describe the methodology of our hybrid system. As we have presented previously, the aim of our proposed hybrid system is to improve the C4.5 decision tree algorithm using Fuzzy logic and to propose a new fuzzy rule-based system using our improved C4.5,

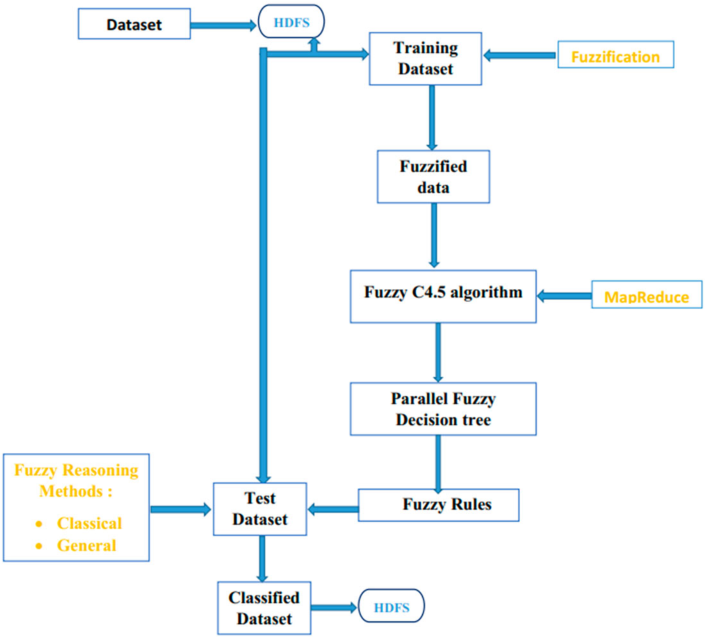


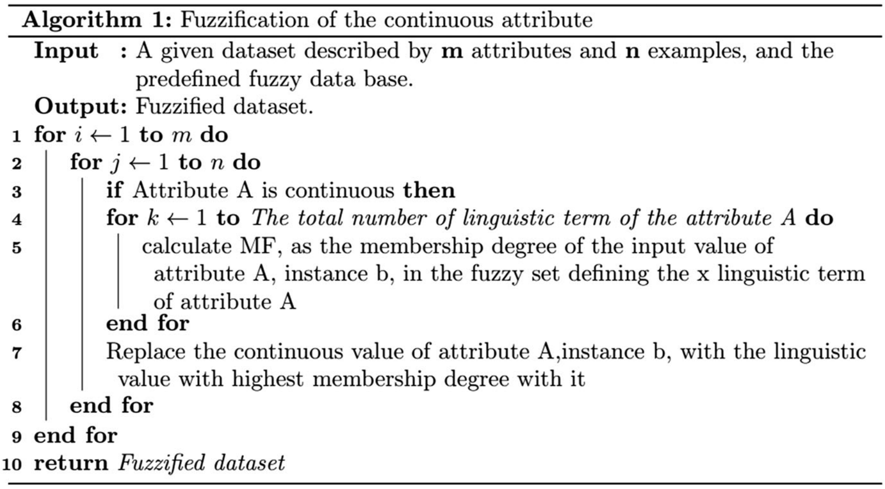
Figure 1. Flow chart of our improved algorithm.

in order to handle the uncertainty and imprecision data. The classification is made using the fuzzy C4.5 algorithm, fuzzy rule-based system and the Hadoop framework, which parallelises the classification tasks between five machines; one master node and four slave nodes, using its distributed file system (HDFS) for storing the dataset to classify and the classified dataset (the result of the classification), and MapReduce programming model for the process and development of our work. We can summarise our work in the following steps:

- Store the dataset in the Hadoop distributed file system.
- Apply the fuzzification method to fuzzify the dataset to fuzzy set, in this work we use the fuzzification method called triangular membership function (MFs).
- After the fuzzification process is done, we store the fuzzified data set in HDFS file system.
- Apply our parallel fuzzy C4.5 algorithm.
- After the implementation and execution of our parallel fuzzy C4.5, the parallel fuzzy decision tree is created, and we use this resulted decision tree to deduce the fuzzy rule (That is called Inference rule in the fuzzy system).
- After the Inference rule step, we use the classic and general reasoning methods to classify the new examples.
- Finally, store the classification result in the HDFS file system. Figure 1 presents the flow chart of our improved algorithm.

4.1. *Fuzzification Methods*

As shown in Figure 1 and as presented earlier, the first step of our proposed method is to store the data in HDFS distributed system, after the storing is done, we divide the data into two subsets training dataset and test dataset using a10-fold Cross-Validation strategy. The following step is to fuzzify the training dataset using the membership function (MF). The aim of this step is to take the crisp input and calculate the degree to which the crisp input belongs to each of the suitable fuzzy sets (linguistic terms). In our case, the crisp inputs are the values taking by each attribute in our using dataset and we use the triangular MF to determine the membership degree. The Algorithm 1 illustrates the steps of the fuzzification of the training dataset.



Algorithm 1: Fuzzification of the continuous attribute

```
Input  : A given dataset described by m attributes and n examples, and the
         predefined fuzzy data base.
Output: Fuzzified dataset.
for $i \leftarrow 1$ to $m$ do
    for $j \leftarrow 1$ to $n$ do
        if Attribute A is continuous then
        for $k \leftarrow 1$ to The total number of linguistic term of the attribute A do
            calculate MF, as the membership degree of the input value of
              attribute A, instance b, in the fuzzy set defining the x linguistic term
              of attribute A
        end for
        Replace the continuous value of attribute A,instance b, with the linguistic
          value with highest membership degree with it
    end for
end for
return Fuzzified dataset
```

Our Fuzzification algorithm takes into input the training dataset described by **m** attributes and **n** examples and the predefined fuzzy database which contains the set of linguistic terms. The first step of our algorithm is to verify if the attribute is continuous, then and for each linguistic value we calculate the membership degree of the input value of the attribute. After we replace the continuous value of the attribute, with the linguistic term with the highest membership degree with it. And as we said earlier to calculate the membership degree, we use the triangular MF, which is determined by three parameters a, b and c is defined by Equation (1)

$$f(x) = \begin{cases} 0 & \text{if} \quad x \leq a \\ \dfrac{x-a}{b-a} & \text{if} \quad a \leq x \leq b \\ \dfrac{c-x}{c-b} & \text{if} \quad b \leq x \leq c \\ 0 & \text{if} \quad c \leq x \end{cases} \tag{1}$$

In order to calculate the membership function for each linguistic value, it is necessary to determine the values of scalar parameters a, b and c. In our case, we calculate these parameters using the maximum(max) and minimum (min) value of each attribute in all examples of the training dataset. The first step is to determine for each attribute the max and min values, then we calculate the mean of these two values. After we determine the value of scalar parameters as **a** = **min**, **b** = **mean** and **c** = **max**.

After all continuous attributes in our training dataset are fuzzified, the next step is to define the fuzzy rules. And to achieve this step we apply the C4.5 decision tree algorithm based on fuzzy information gain, which is executed in a parallel manner using the MapReduce programming model.

4.2. Parallel Fuzzy C4.5 Decision Tree Algorithm

The next step after the fuzzification process of the crisp inputs is the step of the definition of the rules base. For that, we apply the parallel fuzzy C4.5 decision tree algorithm at the fuzzified training dataset. Our proposed approach integrates the principle of Fuzzy logic, Decision tree, and Hadoop framework.

As we presented earlier, the C4.5 decision tree algorithm is an oriented tree comprised of a root node, as well as decision nodes all the other nodes each with exactly one incoming edge. In order to construct a decision tree, the process is as follows: Given a dataset of training data, apply a measure function on all available attributes, find the better splitting attribute based on the obtained result by the calculation of measure function, once the best attribute is determined. The dataset is divided into numerous partitions according to the ranges of values or number of values associated with the best attribute. Within each partition, if all samples appertain to a single class, the algorithm stops. Otherwise, the splitting procedure is recursively executed until each partition appertains to a single class, or no attribute is left.

On the other hand, Fuzzy C4.5 integrates decision trees with convergent reasoning given by fuzzy logic to handle measurement and language uncertainties. Fuzzy C4.5 utilises fuzzy linguistic terms to designate the splitting conditions of nodes and authorise instances to simultaneously follow down various branches with different membership degrees ranged on [0, 1]. The construction of Fuzzy C4.5 decision tree is identical to that of the classical C4.5 with the difference is that, in the learning process to choose the best splitting attribute, while the classical C4.5 calculates the information gain ratio based on the probability of the ordinary examples, the Fuzzy C4.5 calculates the information gain ratio using the probability of the membership degrees of the examples. In the next paragraph, we will describe how we calculate the fuzzy information gain ratio as described in the article [32].

As known, in each dataset, an attribute could take several values. And with fuzzy logic, these values expressed in linguistic terms (fuzzy set). Each fuzzy set is described by a MF. Let X is the set of instances, $A^{(k)} = \{k = 1, 2, \ldots, n\}$ is the set of attributes which has fuzzy values described by fuzzy set $A_i^{(k)} = \{i = 1, 2, \ldots, m\}$, $M_{A_i^{(k)}}$ is the MF of the fuzzy set $A_i^{(k)}$, and the training examples are to be classified into fuzzy classes described by fuzzy sets $y_j = \{j = 1, 2, \ldots, c\}$. Let M_{y_j} denote the MF of the fuzzy set y_j. The class degree (CD) of the ith fuzzy set of the kth attribute $A_i^{(k)}$ with respect to the jth fuzzy class y_j is defined as

$$CD_{A_i^{(k)}}(y_j) = \frac{\sum_{x \in X_j} M_{A_i^{(k)}}(x^{(k)})}{\sum_{x \in X} M_{A_i^{(k)}}(x^{(k)})} \tag{2}$$

where X_j is all the members in the set of training instances that possess the kth attribute in the sense of falling in the support of the fuzzy set $A_i^{(k)}$, and also belong to class y_j, X is all the members in the set of training instances that possess the kth attribute in the sense of falling in the support of the fuzzy set $A_i^{(k)}$. $x^{(k)}$ Is the value of the kth attribute of instance x and $M_{A_i^{(k)}}(x^{(k)})$ is the membership degree of the value of the kth attribute of instance x represented by the fuzzy set $A_i^{(k)}$. So the fuzzy entropy (FE) of the ith fuzzy set of the kth attribute $A_i^{(k)}$ is defined as follow:

$$FE_{A_i^{(k)}} = -\sum_{j=1}^{c} CD_{A_i^{(k)}}(y_j) \log CD_{A_i^{(k)}}(y_j) \tag{3}$$

where $CD_{A_i^{(k)}}(y_j)$ is the class degree (CD) of the ith fuzzy set of the kth attribute $A_i^{(k)}$ with respect to the jth fuzzy class y_j calculated using Equation (2). Furthermore, the fuzzy entropy (FE) of the kth attribute $A^{(k)}$ is defined as a weighted sum of the $FE_{A_i^{(k)}}$:

$$FE_{A^{(k)}} = \sum_{i=1}^{m} \frac{\sum_{x\in X} M_{A_i^{(k)}}(x^{(k)})}{\sum_{l=1}^{m} \sum_{x\in X} M_{A_l^{(k)}}(x^{(k)})} * FE_{A_i^{(k)}} \tag{4}$$

where X is all the members in the set of training instances that possess the kth attribute in the sense of falling in the support of the fuzzy set $A_i^{(k)}$, m is the total number of fuzzy set $A_i^{(k)}$, $M_{A_i^{(k)}}(x^{(k)})$ is the membership degree of the value of the kth attribute of instance x represented by the fuzzy set $A_i^{(k)}$, and $FE_{A_i^{(k)}}$ is the fuzzy entropy (FE) of the ith fuzzy set of the kth attribute $A_i^{(k)}$ calculated using Equation (3). On the other hand, the class degree (CD) of the training instances with respect to the jth fuzzy class y_j is defined as

$$CD(y_j) = \frac{\sum_{x\in X_j} M_{y_j}(x)}{\sum_{x\in X} M_{y_j}(x)} \tag{5}$$

where X is all the members in the set of training instances, X_j is all the members in the set of training instances that belong to class y_j and $M_{y_j}(x)$ is the membership degree of the class j represented by the fuzzy set y_j in the instance x. The fuzzy entropy (FE) of the training instances is defined accordingly as

$$FE = -\sum_{j=1}^{c} CD(y_j) \log CD(y_j) \tag{6}$$

where $CD(y_j)$ is the class degree (CD) of the training instances with respect to the jth fuzzy class y_j calculated using Equation (5). Therefore, the fuzzy information gain (FIG) of the kth attribute with respect to a set of training instances is finally defined as

$$FIG_{A^{(k)}} = FE - FE_{A^{(k)}} \tag{7}$$

So, the fuzzy information gain (FIG) is the difference between the fuzzy entropy (FE) of the training instances calculated using Equation (6) and the fuzzy entropy ($FE_{A^{(k)}}$) of the kth attribute calculated using Equation (4). The split information $SI_{A^{(k)}}$ of the kth attribute defined as

$$SI_{A^{(k)}} = \sum_{i=1}^{m} - \left(\frac{\sum_{x\in X} M_{A_i^{(k)}}(x^{(k)})}{\sum_{l=1}^{m} \sum_{x\in X} M_{A_l^{(k)}}(x^{(k)})} \right) \log \left(\frac{\sum_{x\in X} M_{A_i^{(k)}}(x^{(k)})}{\sum_{l=1}^{m} \sum_{x\in X} M_{A_i^{(k)}}(x^{(k)})} \right) \tag{8}$$

where X is all the members in the set of training instances that possess the kth attribute in the sense of falling in the support of the fuzzy set $A_i^{(k)}$, m is the total number of fuzzy set $A_i^{(k)}$, and $M_{A_i^{(k)}}(x^{(k)})$ is the membership degree of the value of the kth attribute of instance x represented by the fuzzy set $A_i^{(k)}$. Therefore the fuzzy information gain ratio (*FIGR*) of the kth attribute defined as follow:

$$FIGR_{A^{(k)}} = \frac{FIG_{A^{(k)}}}{SI_{A^{(k)}}} \tag{9}$$

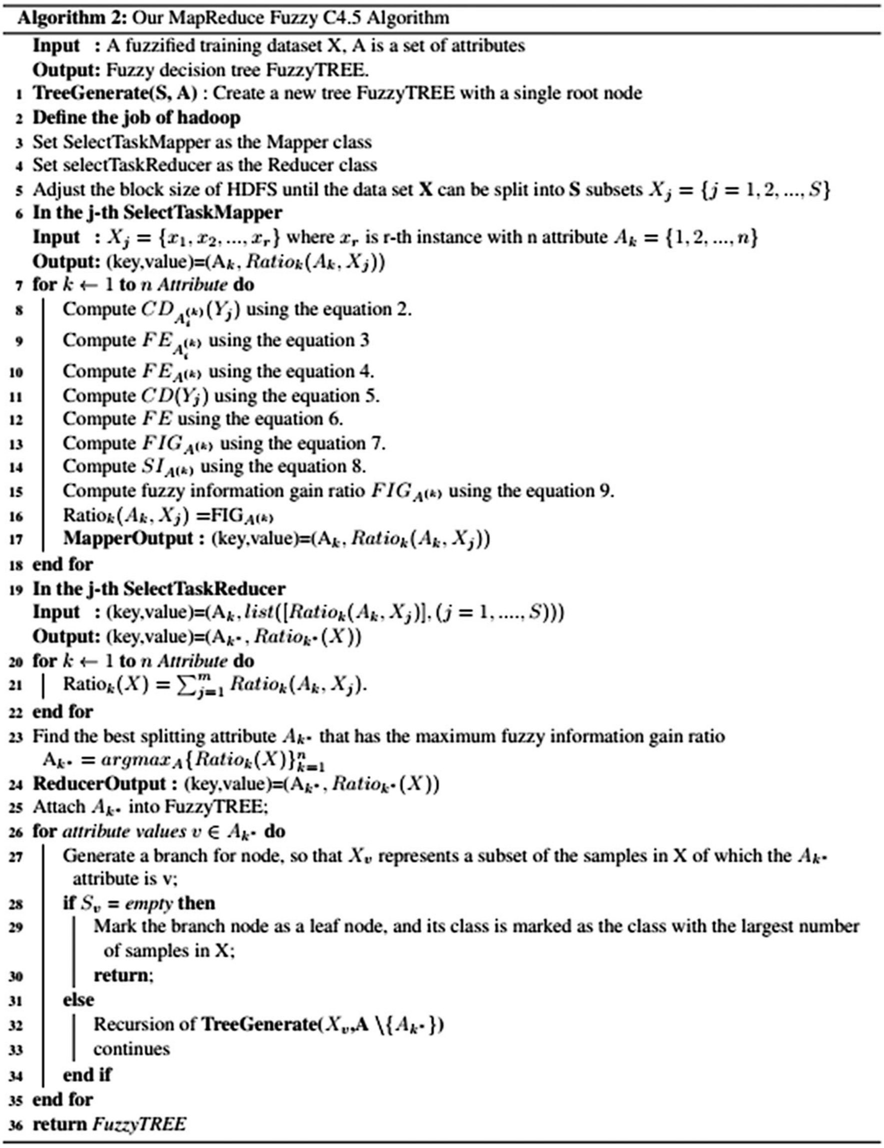
where $FIGR_{A^{(k)}}$ is the fuzzy information gain (*FIG*) of the kth attribute with respect to a set of training instances calculated using Equation (7), and $SI_{A^{(k)}}$ is the split information calculated using Equation (8).

The building of a decision tree is a repeated process, if the classic serial algorithm is applied to realise the process, a lot of resources are spent on a small amount of data, not to mention a huge amount of data. To remedy this problem we work with the parallel programming method. The fuzzy C4.5 decision tree is also produced through the iterative process. In the situation of big amounts of data, it is hard to reach the goal of the classification using fuzzy C4.5 with a single node. In particular, calculating the fuzzy information gain ratio in the process of building the fuzzy decision tree, is the most time-consuming process and used a lot of resources. In our work, to handle this problem we apply the MapReduce programming model, which parallelises the classification tasks between five machines; one master node and four slave nodes. The following Algorithm 2 illustrates the steps of our parallel fuzzy C4.5 decision tree algorithm.

4.3. *Fuzzy Rules*

We create the rule base by first transforming the training dataset into fuzzified data using the fuzzification method (triangular MF). Then, we apply the parallel fuzzy C4.5 algorithm to the fuzzified dataset for producing a fuzzy decision tree. Finally, we extract the rule base from the produced fuzzy decision tree. The rule base contains the fuzzy rules that are to be used in making decisions. The process of generating these rules is usually based on some approaches such as neural networks, decision trees (that used in our work), genetic algorithms or other empirical methods. However, in some situations, the rule can be produced using intuition and personal experience. Rules are among the first techniques used to represent knowledge. In fact, rules are still widely used due to the fact that they make it possible to clearly express directives and strategies, as well as capturing the knowledge from human experts. Rules also have the advantage of their linguistic format, which is easily understandable. Fuzzy rules are a facile manner to formulate vague knowledge. In general, fuzzy rules have the following form:

$$\textbf{IF}(\text{antecedent})\textbf{THEN}(\text{consequent})$$



```
Algorithm 2: Our MapReduce Fuzzy C4.5 Algorithm
Input  : A fuzzified training dataset X, A is a set of attributes
Output: Fuzzy decision tree FuzzyTREE.
TreeGenerate(S, A) : Create a new tree FuzzyTREE with a single root node
Define the job of hadoop
Set SelectTaskMapper as the Mapper class
Set selectTaskReducer as the Reducer class
Adjust the block size of HDFS until the data set X can be split into S subsets $X_j = \{j = 1, 2, ..., S\}$
In the j-th SelectTaskMapper
Input  : $X_j = \{x_1, x_2, ..., x_r\}$ where $x_r$ is r-th instance with n attribute $A_k = \{1, 2, ..., n\}$
Output: (key,value)=($A_k$, $Ratio_k(A_k, X_j)$)
for $k \leftarrow 1$ to $n$ Attribute do
    Compute $CD_{A_i^{(k)}}(Y_j)$ using the equation 2.
    Compute $FE_{A_i^{(k)}}$ using the equation 3
    Compute $FE_{A^{(k)}}$ using the equation 4.
    Compute $CD(Y_j)$ using the equation 5.
    Compute $FE$ using the equation 6.
    Compute $FIG_{A^{(k)}}$ using the equation 7.
    Compute $SI_{A^{(k)}}$ using the equation 8.
    Compute fuzzy information gain ratio $FIG_{A^{(k)}}$ using the equation 9.
    Ratio$_k(A_k, X_j)$ =FIG$_{A^{(k)}}$
    MapperOutput : (key,value)=($A_k$, $Ratio_k(A_k, X_j)$)
end for
In the j-th SelectTaskReducer
Input  : (key,value)=($A_k$, $list([Ratio_k(A_k, X_j)], (j = 1, ...., S))$)
Output: (key,value)=($A_{k^*}$, $Ratio_{k^*}(X)$)
for $k \leftarrow 1$ to $n$ Attribute do
    Ratio$_k(X) = \sum_{j=1}^{m} Ratio_k(A_k, X_j)$.
end for
Find the best splitting attribute $A_{k^*}$ that has the maximum fuzzy information gain ratio
  $A_{k^*} = argmax_A\{Ratio_k(X)\}_{k=1}^{n}$
ReducerOutput : (key,value)=($A_{k^*}$, $Ratio_{k^*}(X)$)
Attach $A_{k^*}$ into FuzzyTREE;
for attribute values $v \in A_{k^*}$ do
    Generate a branch for node, so that $X_v$ represents a subset of the samples in X of which the $A_{k^*}$
      attribute is v;
    if $S_v$ = empty then
        Mark the branch node as a leaf node, and its class is marked as the class with the largest number
          of samples in X;
        return;
    else
        Recursion of TreeGenerate($X_v$,A \{$A_{k^*}$})
        continues
    end if
end for
return FuzzyTREE
```

A rule is made up of two principal parts: an antecedent block (between If and Then) and a consequent block (following Then). As we said earlier, in our work, we use the parallel fuzzy C4.5 decision tree algorithm to generate the fuzzy rules. Rules are generated from each path from the root to a leaf node of the produced decision tree. Figure 2 shows an example of a fuzzy decision tree produced by the application of the parallel fuzzy C4.5 decision tree to a fuzzified training dataset characterised by two classes (Y_1, Y_2) and six attributes. And each attribute has 5 fuzzy sets.

From the fuzzy decision tree illustrated by Figure 2 we can deduce the set of fuzzy rules, such as the number of rules will correspond to the number of possible paths from the root to the leaf nodes, and from Figure 2, the number of paths is thirteen so the number of rules will be thirteen as describe below:

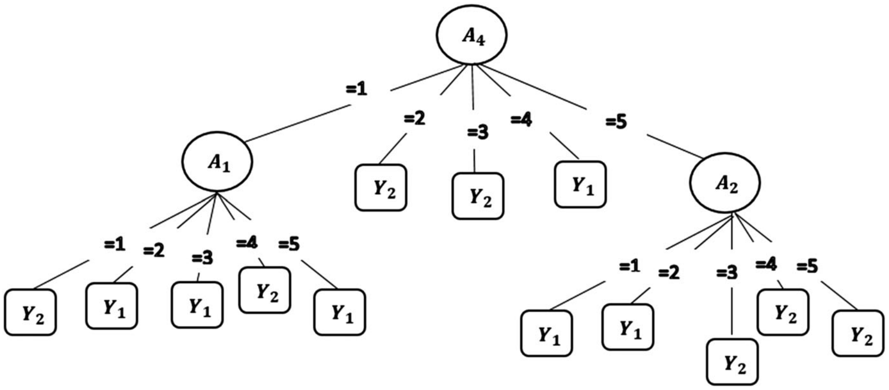


Figure 2. An example of fuzzy decision tree produced by the parallel fuzzy C4.5 algorithm applied to the fuzzified training dataset.

Rule 1: **IF** A_4 is $v_{1;4}$ **AND** A_1 is $v_{1;1}$ **THEN** C is Y_2
Rule 2: **IF** A_4 is $v_{1;4}$ **AND** A_1 is $v_{2;1}$ **THEN** C is Y_1
Rule 3: **IF** A_4 is $v_{1;4}$ **AND** A_1 is $v_{3;1}$ **THEN** C is Y_1
Rule 4: **IF** A_4 is $v_{1;4}$ **AND** A_1 is $v_{4;1}$ **THEN** C is Y_2
Rule 5: **IF** A_4 is $v_{1;4}$ **AND** A_1 is $v_{5;1}$ **THEN** C is Y_1
Rule 6: **IF** A_4 is $v_{2;4}$ **THEN** C is Y_2
Rule 7: **IF** A_4 is $v_{3;4}$ **THEN** C is Y_2
Rule 8: **IF** A_4 is $v_{4;4}$ **THEN** C is Y_1
Rule 9: **IF** A_4 is $v_{5;4}$ **AND** A_2 is $v_{1;2}$ **THEN** C is Y_1
Rule 10: **IF** A_4 is $v_{5;4}$ **AND** A_2 is $v_{2;2}$ **THEN** C is Y_1
Rule 11: **IF** A_4 is $v_{5;4}$ **AND** A_2 is $v_{3;2}$ **THEN** C is Y_2
Rule 12: **IF** A_4 is $v_{5;4}$ **AND** A_2 is $v_{4;2}$ **THEN** C is Y_2
Rule 13: **IF** A_4 is $v_{5;4}$ **AND** A_2 is $v_{5;2}$ **THEN** C is Y_2

4.4. Fuzzy Reasoning Methods

After the step of the extraction of fuzzy rules by applying the parallel fuzzy C4.5 to the fuzzified training dataset, the next step is the test of our generated learning model. That is to say, we use our generated decision tree to classify the new input. In our work, to classify the new instance or to apply the resulted set of fuzzy rules to a new input instance in order to determine the class it belongs to. We use two inference mechanisms. The general and classic fuzzy reasoning methods, which are vastly used in the literature.

4.4.1. Classic Fuzzy Reasoning Method

Many fuzzy classification systems utilise the Classic Fuzzy Reasoning Method (CFRM), which chooses the rule with greatest compatibility degree to classify the new given instance. Let $e_p = \{a_{p1}, a_{p2}, \ldots, a_{pm}\}$ a new instance to be classified and $\{R_1, R_2, \ldots., R_s\}$a set of s fuzzy classification rules. Let $M_i(a_{pi})$ and $\{i = 1, \ldots., m\}$ be the membership degree of attribute value. The (CFRM), applies the following steps to classify a new instance:

(1) Calculate the compatibility degree between example e_p and each rule R_k for $k = 1, 2, \ldots, s$ and a t-norm t, given by

$$\text{compat}(e_p, R_k) = t[M_1(a_{p1}), M_2(a_{p2}), \ldots., M_m(a_{pm})] \tag{10}$$

(2) Find rule $R_{k\max}$ as the rule with the greatest compatibility degree with the instance, i.e.

$$\text{compat}(e_p, R_{k\max}) = \max[\text{compat}(e_p, R_k)]; \quad k = 1, 2, \ldots, s. \tag{11}$$

(3) Assign the class c_j to the instance e_p, where c_j the class predicted by the rule $R_{k\max}$ found in the previous step.

For example, we have $e_p = \{a, b, c, ?\}$ a new instance with unknown class '?' to be classified. Where a, b and c are fuzzy sets. Let $M(a) = 0.65$, $M(b) = 0.32$ and $M(c) = 0.82$ are the membership degree of a, b, and c respectively. And we have two rules such as

- **R1: IF** A is a_1 **AND** B is b_1 **AND** C is c_1 **THEN** D is Y_1. With $M\,(a_1) = 0.52$, $M\,(b_1) = 0.21$ and $M\,(c_1) = 0.92$.
- **R2: IF** A is a_2 **AND** B is b_2 **AND** C is c_2 **THEN** D is Y_2. With $M\,(a_2) = 0.13$, $M\,(b_2) = 0.85$ and $M\,(c_2) = 0.63$.

Step 1: Calculate the degree that input instance (a, b, c) matches each rule term $(a_1, a_2, b_1, b_2, c_1, c_2)$, and then we will use these calculated degrees to compute the compatibility degree for each rule.

$$\left.\begin{aligned} d(a, a_1) &= \min[M(a), M(a_1)] = \min(0.65, 0.52) = 0.52 \\ d(b, b_1) &= \min[M(b), M(b_1)] = \min(0.32, 0.21) = 0.21 \\ d(c, c_1) &= \min[M(c), M(c_1)] = \min(0.82, 0.92) = 0.82 \end{aligned}\right\} = \min(0.52, 0.21, 0.82) = 0.21$$

$$\left.\begin{aligned} d(a, a_2) &= \min[M(a), M(a_2)] = \min(0.65, 0.13) = 0.13 \\ d(b, b_2) &= \min[M(b), M(b_2)] = \min(0.32, 0.85) = 0.32 \\ d(c, c_2) &= \min[M(c), M(c_2)] = \min(0.82, 0.63) = 0.63 \end{aligned}\right\} = \min(0.13, 0.32, 0.63) = 0.13$$

Therefore the compatibility degree between example e_p and rule R_1 is equal to $\text{compat}(e_p, R_1) = 0.21$, and $\text{compat}(e_p, R_2) = 0.13$ is the compatibility degree between example e_p and rule R_2. To compute the compatibility degree we used **t-norm = minimum** because we have the **AND** in the rules, and in the case where we have **OR** in the rules, we must use **t-norm = maximum**.

Step 2: Find rule $R_{k\max}$ as the rule with the greatest compatibility degree with the instance e_p, i.e.: $\text{compat}(e_p, R_{k\max}) =$ max $[\text{compat}(e_p, R_1)$,$\text{compat}(e_p, R_2)] =$ max (0.21, 0.13) = 0.21. Therefore the rule with the greatest compatibility degree is the rule **R1: IF** A is a_1 **AND** B is b_1 **AND** C is c_1 **THEN** D is Y_1.

Step 3: Assign the class Y_1 to instance $e_p = \{a, b, c, Y_1\}$, where Y_1 is the class predicted by the rule **R1**: **IF** A is a_1 **AND** B is b_1 **AND** C is c_1 **THEN** D is Y_1. Found as $R_{k\max}$ in the previous step.

Figure 3 illustrates graphically the (CFRM). The compatibility degree of the new input instance is computed in relation to all s fuzzy rules, and because the class c_j from rule $R_{k\max}$ has the greatest compatibility degree, it assigned to the input example.

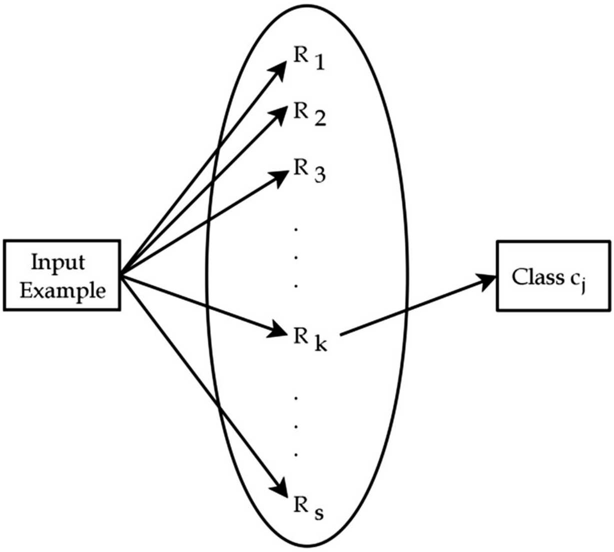


Figure 3. A classic fuzzy reasoning method.

4.4.2. General Fuzzy Reasoning Method

The General Fuzzy Reasoning Method (GFRM) follows the below indicated steps to classify a given example e_p:

(1) Calculate the compatibility degree between example e_p and each rule R_k for $k = 1, 2, \ldots, s$ and a t-norm **t**, given by

$$\text{compat}(e_p, R_k) = t[M_1(a_{p1}), M_2(a_{p2}), \ldots., M_m(a_{pm})] \tag{12}$$

(2) For each class, calculate the classification value class_c. class_c is defined as the aggregation of the compatibility degree, computed in the preceding step, of all rules with class c_j and represents the compatibility degree of the instance with all the rule whose predicted class is c_j, given by: $\text{class}_{c_j} = \mathbf{f}\{\text{compat}(e_p, R_k)|\ c_j \text{ is the class of } R_k\}$. Where **f** is an aggregation operator.

For example, we have $e_p = \{a, b, c, ?\}$ a new instance with unknown class '?' to be classified. Where *a, b* and *c* are fuzzy sets. Let $M\,(a) = 0.65$, $M\,(b) = 0.32$ and $M\,(c) = 0.82$ are the membership degree of *a, b*, and *c* respectively. And we have four rules such as

- **R1: IF** A is a_1 **AND** B is b_1 **AND** C is c_1 **THEN** D is Y_1. With $M(a_1) = 0.52$, $M(b_1) = 0.21$ and $M(c_1) = 0.92$.
- **R2: IF** A is a_2 **AND** B is b_2 **AND** C is c_2 **THEN** D is Y_2. With $M(a_2) = 0.13$, $M(b_2) = 0.85$ and $M(c_2) = 0.63$.
- **R3**: **IF** A is a_3 **AND** B is b_3 **AND** C is c_3 **THEN** D is Y_1. With $M(a_3) = 0.19$, $M(b_3) = 0.97$ and $M(c_3) = 0.38$.
- **R4**: **IF** A is a_4 **AND** B is b_4 **AND** C is c_4 **THEN** D is Y_2. With $M(a_4) = 0.75$, $M(b_4) = 0.53$ and $M(c_4) = 0.20$.

Step 1: Calculate the degree that input instance (*a, b, c*) matches each rule term $(a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, c_1, c_2, c_3, c_4)$, and then we will use these calculated degrees to compute the compatibility degree for each rule.

$$\left.\begin{aligned} d(a,a_1) &= \min[M(a),M(a_1)] = \min(0.65,0.52) = 0.52 \\ d(b,b_1) &= \min[M(b),M(b_1)] = \min(0.32,0.21) = 0.21 \\ d(c,c_1) &= \min[M(c),M(c_1)] = \min(0.82,0.92) = 0.82 \end{aligned}\right\} = \min(0.52,0.21,0.82) = 0.21$$

$$\left.\begin{aligned} d(a,a_2) &= \min[M(a),M(a_2)] = \min(0.65,0.13) = 0.13 \\ d(b,b_2) &= \min[M(b),M(b_2)] = \min(0.32,0.85) = 0.32 \\ d(c,c_2) &= \min[M(c),M(c_2)] = \min(0.82,0.63) = 0.63 \end{aligned}\right\} = \min(0.13,0.32,0.63) = 0.13$$

$$\left.\begin{aligned} d(a,a_3) &= \min[M(a),M(a_3)] = \min(0.65,0.19) = 0.19 \\ d(b,b_3) &= \min[M(b),M(b_3)] = \min(0.32,0.97) = 0.32 \\ d(c,c_3) &= \min[M(c),M(c_3)] = \min(0.82,0.38) = 0.38 \end{aligned}\right\} = \min(0.19,0.32,0.38) = 0.19$$

$$\left.\begin{aligned} d(a,a_4) &= \min[M(a),M(a_4)] = \min(0.65,0.75) = 0.65 \\ d(b,b_4) &= \min[M(b),M(b_4)] = \min(0.32,0.53) = 0.32 \\ d(c,c_4) &= \min[M(c),M(c_4)] = \min(0.82,0.20) = 0.20 \end{aligned}\right\} = \min(0.65,0.32,0.20) = 0.20$$

Therefore the compatibility degree between example e_p and each rule R_1, R_2, R_3 and R_4, is equal to:compat$(e_p, R_1) = 0.21$, compat$(e_p, R_2) = 0.13$, compat$(e_p, R_3) = 0.19$, and compat$(e_p, R_4) = 0.20$ respectively.

Step 2: For each class, calculate the classification value class$_c$ in our example we have two class Y_1 and Y_2.

$$\begin{aligned} \text{class}_{Y_1} &= \mathbf{f}\,\{\text{compat}(e_p,R_k)|Y_1\} = \text{compat}(e_p,R_1) + \text{compat}(e_p,R_3) = 0.21 + 0.19 = 0.40 \\ \text{class}_{Y_2} &= \mathbf{f}\,\{\text{compat}(e_p,R_k)|Y_2\} = \text{compat}(e_p,R_2) + \text{compat}(e_p,R_4) = 0.13 + 0.20 = 0.33 \end{aligned}$$

Step 3: Assign the class Y_1 to instance $e_p = \{a, b, c, Y_1\}$, where Y_1 the class with highest sum is (class$_{Y_1}$ =0.40) found in the previous step.

Figure 4 describes graphically the GFRM. The compatibility degree of the new input instance is computed in relation to all *s* fuzzy rules, and because the class c_j is the class that obtained the greatest classification degree among all classes, it assigned to the input example.

5. Simulation Experiments and Analysis

In our approach, we divided the dataset into two subsets (training dataset and test dataset) using a 10fold Cross-Validation strategy and store it in HDFS. Then we used the fuzzification method, especially the triangular MF, to fuzzify the training dataset. After we applied our proposed algorithm (parallel fuzzy C4.5 decision tree) to the fuzzified data, we obtained a fuzzy decision tree. Further, we used the generated fuzzy decision tree to extract a set of rules. Finally, we applied the two fuzzy reasoning methods on the set of rules to classify the test dataset and store the classified data into HDFS. To assess the effectiveness of our improved algorithm, we have applied it to three data sets chosen from the UCI dataset

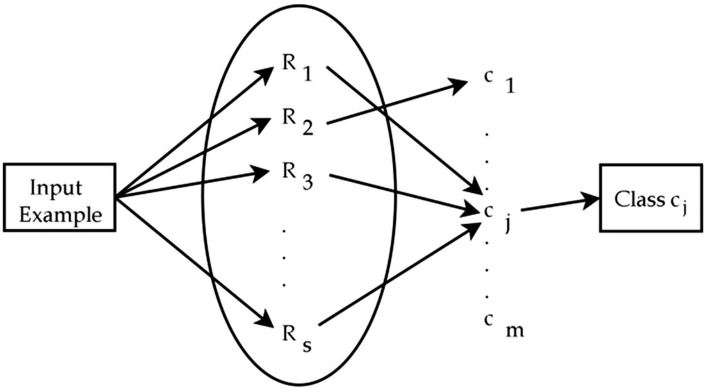


Figure 4. A general fuzzy reasoning method.

Table 1. Data sets properties.

N.	Name of dataset	N. Instances	N. Attributes
1	PAMAP2 Physical Activity Monitoring	3850505	52
2	Gas sensor array under dynamic gas mixtures	4178504	19
3	Record Linkage Comparison Patterns	5749132	12

(Machine Learning repository) [33]. Table 1 describes these dataset properties. And to evaluate its effectiveness, we have chosen nine evaluation metrics are shown in Table 3. The nine metrics are True Positive Rate (TPR) or Sensitivity or Recall, True Negative Rate (TNR) or Specificity, False Positive Rate (FPR), False Negative Rate (FNR), Error Rate (ER), Precision (PR), Classification Rate (CR), kappa statistic (KS), and F1-score (FS). Without forgetting the execution time rate.

5.1. Experiment Platform

- Computer Performance: Hardware environment is: Intel(R) Core(TM) i7- 6500U CPU @2.50 GHz 2.59 GHz, Installed memory(RAM) 16.0GB, Two Hard disk SSD 500G, System type 64-bit Operating System, Window 10 system.
- Virtual Box: Virtual Box Graphical User Interface Version 5.2.22 r126460 (Qt5.6.2), which released within 9 November 2018.
- Operating System: in our work we used Ubuntu 16.04.5 LTS (Xenial Xerus).
- Eclipse Software: in our work we used Eclipse 2018- 12 (4.10), with package Eclipse IDE for Java Developers.
- Hadoop machine: the cluster of our work contains five Hadoop machines, four slave nodes and master node.

5.2. Experiment Data Sets

To evaluate the performance of our approach (fuzzyLogic + MapReduce + C4.5) compared with other methods like ID3, C4.5, MapReduce + C4.5, Fuzzy + C4.5, Damanik et al., Cherfi et al., and Lee, we have considered three datasets selected from UCI dataset (Machine

Learning repository) [33] with the number of instances range from 3850505 to 5749132 as described in Table 1.

5.3. Evaluation Metrics

The major concept of the classification process is linked an unknown instance into appropriate predefined class labels. This linked process takes place according to the type of classification desired (Binary, Multi-class, Multi-labelled, and Hierarchical classification). In our work, we used Binary and Multi-class classification, which pushed us to focus on them in this section.

Multi-Class Classification: The income instance in predicting model is to be classified into one, and only one of l non-overlapping classes. As the binary classification, multi-class categorisation can be thematic or particular, well defined, or fuzzy. How to compute the nine selected evaluation criteria for any multi-class classification problem: Most of them can be calculated by using the confusion matrix. It is a way of classifying true positives, true negatives, false positives, and false negatives when there are more than two classes. It is used for computing the evaluation criteria for multi-class problems.

Binary Classification: Positive or Negative: the binary classification system is the most popular task. Its idea is to classify the input instances into two possible non-overlapping categories positive C1 or negative C2. The effectiveness of this type of classification can be examined by calculating the correctly detected positive class instances rate (TPR) and the correctly recognised negative class instances rate (TNR). We could have instances that are actually positive but are predicted to be negative (FNR) and instances that are actually negative and predicted to be positive (FPR). These four possible outcomes constitute a confusion matrix, as shown in Table 2.

- **True Positive (tp)**: instance that is actually positive and predicted to be positive
- **False Negative (fn)**: instance that is actually positive and predicted to be negative
- **True Negative (tn)**: instance that is actually negative and predicted to be negative
- **False Positive (fp)**: instance that is actually negative and predicted to be positive

Therefore, we are going to use these four outcomes for discussing the ten selected evaluation metrics of the Binary and Multi-class classification tasks.

- **TPR:** estimates the effectiveness of a classifier to recognise the instances have positive labels, TP corresponds to the number of the true positive instances, and TP + FN is the total number of positive instances.
- **TNR:** measures how efficaciously a classifier identifies the instances have negative labels. Where TN matches the number of the true negative samples, and TN + FP is the total number of examples that is negative.

Table 2. Confusion matrix: for binary classification and the corresponding array representation used in this paper.

Data class	Classified as positive	Classified as negative
Positive	True Positive(TP)	False Negative(FN)
Negative	False Positive(FP)	True Negative(TN)

- **FPR:** is the rate to measure the ineffectiveness of a classifier and to estimate the misclassification rate by determinate the number of examples actually are negative and the classifier is predicted it positives.
- **FNR:** is the rate to measure the inability of a classifier and to estimate the misclassification rate by specify the number of examples actually are positives and the classifier is classified it negatives.
- **ER:** incorrect classification rate is the misclassification instances over all instances in the distribution its objective is to measure the classifiers ability to prevent false classification
- **PR:** Gauges how many instances predicted as a positive class are actually positive. This measure is valuable for appreciating fragile classifiers that are used to classify an entire dataset.
- **CR**: The classification accuracy is an overall measure for assessing the correctness and righteousness of learning systems. The accuracy of a decision tree is calculated using a test set that is separate from the training set. Generally, is the rate of all true classified instances overall classified instances.
- **FS:** F1-Score or F-measure is the harmonic mean between precision and recall. Supplies a better notion of average, the range for F1 Score is [0, 1]. It tells you how accurate your classifier is, as well as how robust it is. The higher the F1 Score, the better is the performance of our model. That means the precision is high, but the recall is lower, gives you an extremely accurate.
- **KS:** The Kappa statistic is an evaluation criterion that makes a comparison between an Expected Accuracy (random chance) and an Observed Accuracy. It is applied not only to assess one classifier but also to examine classifiers amongst themselves. Where:

$$P_0 = \frac{tp + tn}{100} \text{ and} P_e = \left[\frac{tp + fn}{100} * \frac{tp + fp}{100}\right] + \left[\frac{fp + tn}{100} * \frac{fn + tn}{100}\right]$$

5.4. Results and Discussion

In this section, we are going to present the experimental results of our approach (MapReduce + Fuzzy Logic + C4.5). These experimental results are obtained by applying our approach and other approaches like ID3, C4.5, MapReduce + C4.5, Fuzzy + C4.5, Damanik et al., Cherfi et al., and Lee, on three selected dataset as shown in Table 1. To verify which of these approaches more efficient and better, we compute nine evaluation metrics as described earlier in Table 2. The classification using our approach will be done in a parallel manner using the Hadoop framework with HDFS and the MapReduce programming model. The Hadoop cluster contains four salve nodes and one master node.

Figure 5 shows the result of the classification accuracy (AC) after the application of Fuzzy + C4.5 using the general reasoning method(approach 1) and Fuzzy + C4.5 using the classical reasoning method(approach 2) on three select dataset number 1,2 and 3.

From Figure 5, we notice that the approach1 outperforms the approach2 in all selected datasets with accuracy rate equal to 72.23, 86.56 and 78.47 respectively to dataset numbers 1, 2 and 3. That is to say; the general reasoning method is more efficient in the classification of new instances than the classic reasoning method. So in the rest of this work, we will use the general reasoning method.

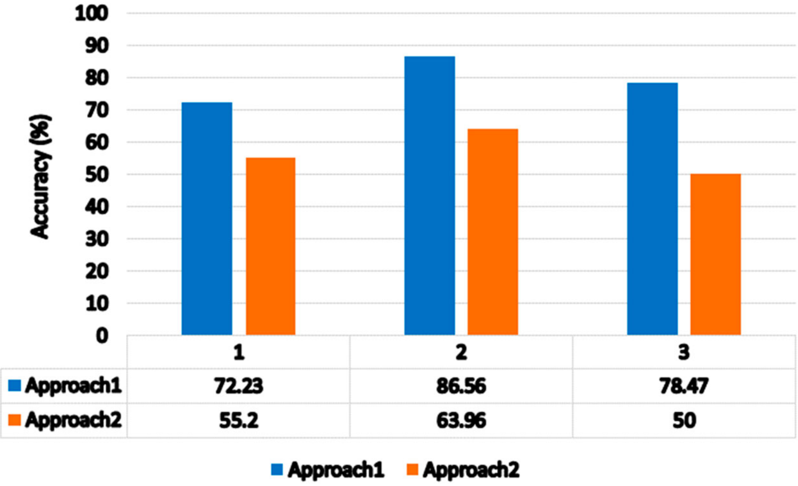


Figure 5. Accuracy rate using approach1 and approach2.

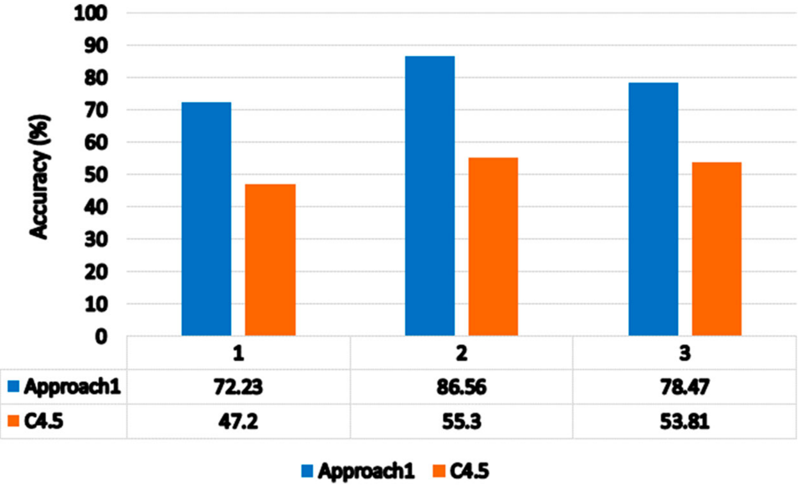


Figure 6. Accuracy rate using approach1 and C4.5.

Another experiment is made to compare the classical C4.5 and approach1, to demonstrate if the application of fuzzy logic influences the classification result, as the first experiment, we applied both approaches on the three chosen dataset. Figure 6 shows the result of the accuracy rate using both algorithms: fuzzy and classical.

From Figure 6, we deduce that the application of fuzzy logic on the C4.5 algorithm improves the performance of the classification. Which is it increases the accuracy rate by 25.03, 31.26 and 24.66 respectively to dataset numbers 1, 2 and 3, compared to C4.5.

To evaluate our work, we have selected three datasets that contain a huge amount of data, such as the dataset n.1 has 3850505 instances, the dataset n.2 has 4178504 instances, and the dataset n.3 has 5749132 instances. The application of classical C4.5 takes a lot of time, which can be varied from one hour to 2.5 h according to the size of the dataset used. To remedy this problem, we use the MapReduce programming model, which shares the

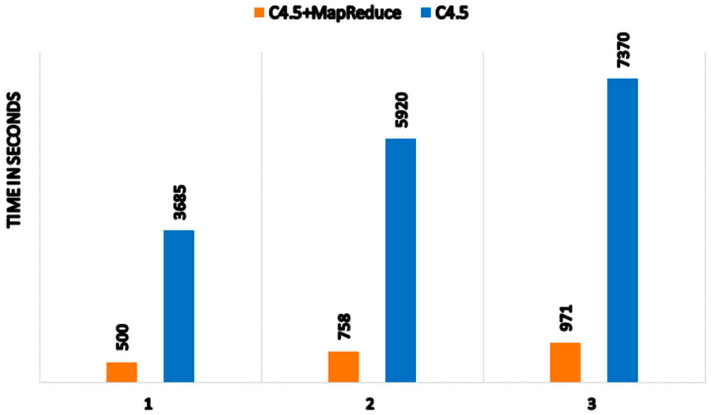


Figure 7. Execution time using C4.5 + MapReduce and C4.5.

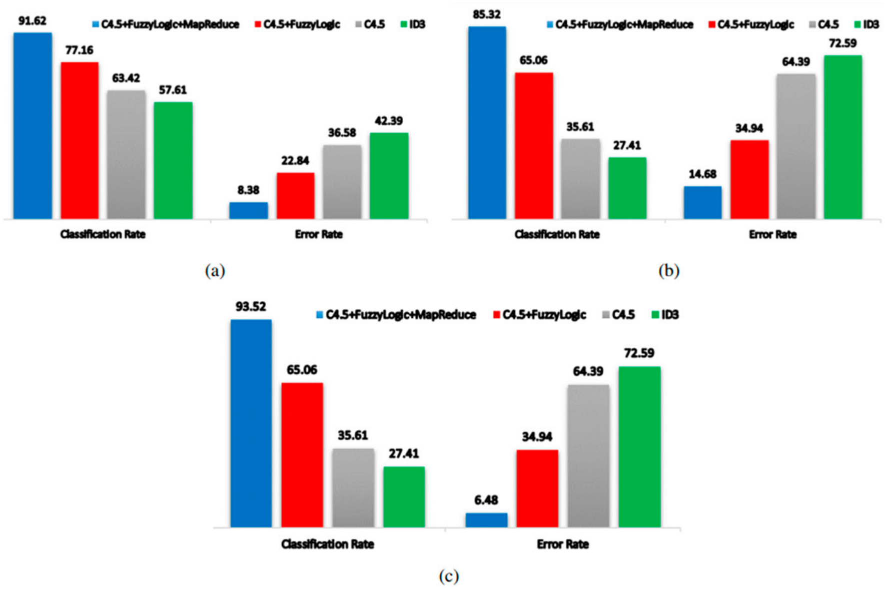
work on five machines (four slave nodes and one master node). Another experiment is done to compare the C4.5 and C4.5+MapReduce. Figure 7 shows the execution time of C4.5 and C4.5 + MapReduce algorithms.

From Figure 7, we notice that the C4.5 + MapReduce algorithm has less execution time than the C4.5 algorithm, which makes the application of MapReduce on C4.5 more efficient. For example, the consuming time by applying the C4.5 on the dataset n.1 is 3685 s, on the other hand, the used time by executing the C4.5 + MapReduce on the same dataset is 500 s. We remark that the C4.5 + MapReduce reduces the consuming time by 7.37 times compared to C4.5. This reduction is due to the use of five machines(four slave nodes and one master node) in the execution of the algorithm C4.5 + MapReduce. Also, the C4.5 + MapReduce algorithm decreases the consuming time for dataset n.2 and dataset n.3 by 7.81 and 9.43 times, respectively compared to C4.5.

In summary, from the first experience, as shown in Figure 5 Fuzzy + C4.5 using the general reasoning method outperforms the Fuzzy + C4.5 using the classical reasoning method, So in our work, we will use the powerful method to classify the new instances. Also, from the second experience, as described in Figure 6, we deduce that the application of fuzzy logic on C4.5 allows us to improve the classification accuracy of the classical C4.5. Therefore in our work, we will apply the fuzzy logic. Finally, from the third experience, as illustrated in Figure 7, we notice that the utilisation of the MapReduce programming model decreases the consuming time used by C4.5 on a huge amount of data. Accordingly, in our work, we have combined C4.5, fuzzy logic, and MapReduce, so in the next experiences, we will evaluate the performance of our approach C4.5 + Fuzzy Logic + MapReduce.

Figure 8, illustrates the result obtained for the classification rate and Error rate using our proposed approach (C4.5 +Fuzzy Logic + MapReduce), and we compare the result obtained with other methods like ID3, C4.5, and C4.5 + FuzzyLogic. Figure 8a shows the result acquired by the application of all approaches on dataset n.1, as well Figure 8b illustrates the result of the classification and error rate obtained by applying all cited methods on the dataset n.2. Finally, Figure 8c presents the result of the dataset n.3.

From Figure 8, the first remark is that our proposed algorithm (C4.5 + FuzzyLogic + MapReduce) outperforms the other algorithms in terms of classification and error rate.



(a)

(b)

(c)

Figure 8. Classification rate and error rate, (a) dataset n.1, (b) dataset n.2, (c) dataset n.3.

And as presented in Figure 8a, if we compare our approach with C4.5, we notice that our approach increases the classification rate from 63.42% (C4.5) to 91.62% (our method) and reduces the error rate from 36.58%(C4.5) to 8.38%(our approach) for the dataset number 1. The second remark, according to this comparison, is that the integration of MapReduce and Fuzzy Logic with C4.5 improves the performance of the classification. As we said earlier, we have evaluated our work by using three datasets, the dataset n.2(4178504 instances) contains more the instances than the dataset n.1(3850505 instances) and also the dataset n.3(5749132 instances) is large than the dataset n.2. The major aim of this variation in the number of instances is to test the scalability of our approach. As we see, in Figure 8a that represents the dataset n.1, the classification rate is 91.62% for our method, 77.16% for C4.5+FuzzyLogic, 63.42% for C4.5 and 57.61% for ID3. As shown in Figure 8b that illustrates the result obtained by the application of all approaches on the dataset n.2, the classification rate is 89.32% for our procedure, 70.06% for C4.5+Fuzzy Logic, 55.61% for C4.5 and 47.41% for ID3. Also for the dataset n.3 (Figure 8c), the classification rate is 93.52% for our approach, 65.06% for C4.5+Fuzzy Logic, 35.61% for C4.5 and 27.61% for ID3. Consequently, the third remark, according to this study, is that the proposed approach is scalable. And because the C4.5+FuzzyLogic is note scalable, we can deduce that this scalability is due to the MapReduce programming model.

For demonstrating the effectiveness of our approach, we have calculated other evaluation metrics like TPR, FNR, TNR, FPR, PR, KS, and FS as earlier explained in Table 3. Table 4 shows the result obtained.

According to Table 4, our approach (C4.5+FuzzyLogic + MapReduce) out-performs the other algorithms in all datasets (1,2,3) and at the level of TPR(92.03%, 89.19%,92.13%), FNR(7.97%, 10.81%,7.87%),TNR(89.71%,87.61%,89.56%), FPR(10.29%, 12.39%, 10.44%) PR(91.45%,75.52%,90.18%), KS(89.96%,88.49%, 85.4%) and FS(88.24%,80.74%,79.05%).

Table 3. Measures for binary and multi-class classification using the notation of Table 1.

Measure	Binary-class formula	Multi-class formula
TPR	$\frac{tp}{tp+fn}$	$\frac{\sum_{i=1}^{l}\frac{tp_i}{tp_i+fn_i}}{l}$
TNR	$\frac{tn}{tn+fp}$	$\frac{\sum_{i=1}^{l}\frac{tn_i}{tn_i+fp_i}}{l}$
FPR	$\frac{fp}{fp+tn}$	$\frac{\sum_{i=1}^{l}\frac{fp_i}{fp_i+tn_i}}{l}$
FNR	$\frac{fn}{fn+tp}$	$\frac{\sum_{i=1}^{l}\frac{fn_i}{fn_i+tp_i}}{l}$
ER	$\frac{fp+fn}{tp+fn+tn+fp}$	$\frac{\sum_{i=1}^{l}\frac{fp_i+fn_i}{tp_i+fn_i+tn_i+fp_i}}{l}$
PR	$\frac{tp}{tp+fp}$	$\frac{\sum_{i=1}^{l}\frac{tp_i}{tp_i+fp_i}}{l}$
CR	$\frac{tp+tn}{tp+fn+tn+fp}$	$\frac{\sum_{i=1}^{l}\frac{tp_i+tn_i}{tp_i+fn_i+tn_i+fp_i}}{l}$
KS	$\frac{p_0+p_e}{1-p_e}$	$\frac{\sum_{i=1}^{l}\frac{p_{0i}-p_{ei}}{1-p_{ei}}}{l}$
FS	$\frac{2*\text{precision}*\text{recall}}{\text{precision}+\text{recall}}$	$\frac{\sum_{\mathrm{i}=1}^{\mathrm{l}}\frac{2*\text{presicion}_{\mathrm{i}}*\text{recall}_{\mathrm{i}}}{\text{presicion}_{\mathrm{i}}+\text{recall}_{\mathrm{i}}}}{\mathrm{l}}$

Table 4. The result of TPR, FNR, TNR, FPR, PR, KS, and FS.

		TPR	FNR	TNR	FPR	PR	KS	FS
Dataset n.1	Our Approach	92.03	7.97	89.71	10.29	91.45	89.96	88.24
	C4.5 + FuzzyLogic	79.19	20.81	78.06	21.94	72.85	68.67	76.49
	C4.5	64.53	35.47	70.11	29.89	63.25	59.46	67.20
	ID3	58.14	41.86	50.89	49.2	42.72	50.52	62.42
Dataset n.2	Our Approach	89.19	10.81	87.61	12.39	75.52	88.49	80.74
	C4.5 + FuzzyLogic	69.23	30.77	60.57	39.42	70.45	58.18	67.92
	C4.5	56.31	34.69	49.82	50.18	59.11	45.68	57.38
	ID3	37.61	62.39	31.36	68.64	40.43	37.02	39.92
Dataset n.3	Our Approach	92.13	7.87	89.56	10.44	90.18	85.4	79.05
	C4.5 + FuzzyLogic	64.15	35.85	60.33	39.67	59.45	62.26	61.05
	C4.5	36.81	63.19	35.09	64.91	30.29	33.67	35.70
	ID3	26.48	73.52	30.00	69.98	29.46	31.52	32.20

Another experiment is made to compare the execution time between our approach and the other techniques. Figure 9 presents the result obtained after the application of all approaches on three selected datasets. Without forgetting that our approach is implemented in a parallel manner on five machines using framework Hadoop.

From Figure 9, we note that our approach has a lower implementation time in all cases. Compared to ID3 our approach decreases the execution time from 4007s to 556s for the dataset n.1, from 7320s to 798s for the dataset n.2, and from 9080s to 1010s for the dataset n.3. That demonstrates that the utilisation of parallelisation is a good idea. Another remark

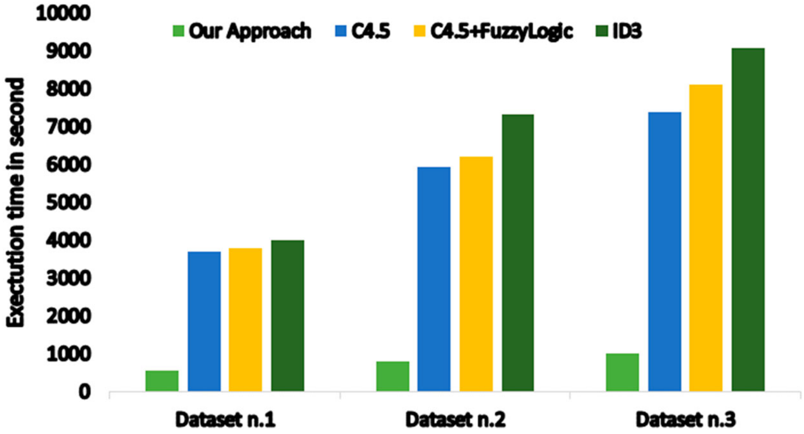


Figure 9. Execution time in seconds of our approach and other techniques.

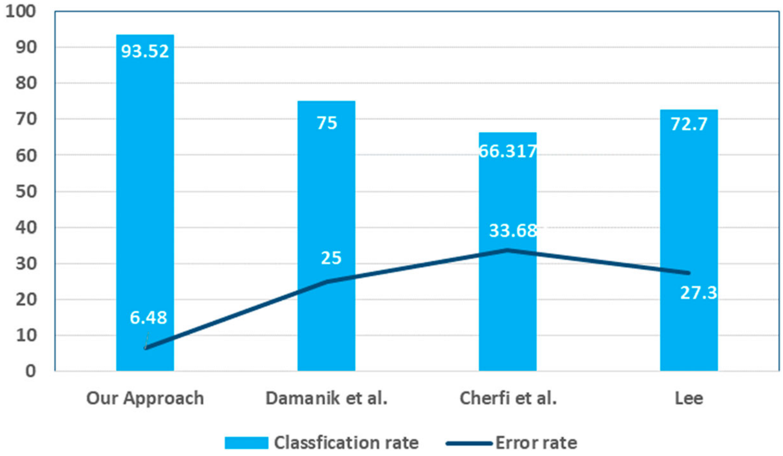


Figure 10. Results obtained by our comparison.

that we had deduced when we implement our work on the Hadoop cluster, the execution time decreases with the increase of the number of nodes in the cluster.

To evaluate the results obtained by applying our proposed method, we compare our approach with some other techniques from the literature. Such as; a 'Decision Tree Optimization in C4.5 Algorithm Using Genetic Algorithm' proposed by Damanik et al. [34], this integrates decision tree and genetic algorithm to improve the performance of the C4.5 to generate effective rules. a 'Very Fast C4.5 Decision Tree Algorithm' proposed by Cherfi et al. [35], this approach uses the arithmetic mean and median to enhance a reported feebleness of the C4.5 algorithm when it handles the continuous attributes, and an 'AUC4.5: AUC-Based C4.5 Decision Tree Algorithm for Imbalanced Data Classification' proposed by Lee [36]. This approach presents a modification of the C4.5 algorithm, which examines the difference in the AUC (area under the ROC curve) for choosing the better splitting attribute. Figure 10 illustrates the results obtained.

From Figure 10, we remark that our approach based on the decision tree, fuzzy logic and Hadoop framework outperforms the other methods (Damanik et al., Cherfi et al., and Lee) with classification rate equal to 93.52% and error rate equal to 6.48%. This effectiveness and advantage of our proposed method are due to the utilisation of fuzzy logic theory, general reasoning method and Hadoop framework.

6. Conclusion

In this paper, firstly, we have improved the C4.5 decision tree algorithm at the level of handling with continuous-valued attributes. This improvement is performed by using fuzzy logic. Secondly, we have proposed a new rule-based fuzzy model, which is consists of three phases, such as the fuzzification phase, the Inference phase, and the classification phase. This proposed system is proved by several experiments for resolving data classification issues in data mining. Initially, this system applies the fuzzification method to determine the membership degree of each attribute value, and replace the continuous value of the attribute with the linguistic term that has the highest membership degree. This initial phase is carried out to deal with the uncertainty and imprecise data. In the next step, parallel fuzzy C4.5 algorithm is applied to build the fuzzy decision tree, and then to extract the set of fuzzy rules. Finally, the general reasoning method is applied to the set of fuzzy rules to classify the new instances and then to evaluate the effectiveness of our proposed model.

Generally, our proposed approach combines C4.5 decision tree, fuzzy logic and Hadoop framework. To demonstrate the effectiveness of our proposed model (C4.5 + FuzzyLogic + MapReduce). Some other approaches like ID3, C4.5, Fuzzy-Logic + C4.5, Damanik et al., Cherfi et al. and Lee are used to compare with the proposed one. And we have selected three huge datasets from UCI dataset to show the scalability of our improved approach. The experimental result shows that our method outperforms the other approaches in terms of True Positive Rate (TPR = 92.13%) or Sensitivity or Recall, True Negative Rate (TNR = 89.56%) or Specificity, False Positive Rate (FPR = 10.44%), False Negative Rate (FNR = 7.87%), Error Rate (ER = 6.48%), Precision (PR = 90.18%), Classification Rate (CR = 93.52%), kappa statistic (KS = 85.4%), F1-score (FS = 79.05%) and execution time (= 556 s).

Our future work is to integrate the convolution neural network, fuzzy logic and decision tree in order to detect the fake news, taking into account several parameters related to feature extraction.

Acknowledgements

Ms Fatima ES-SABERY reveal its gratitude to the financial support for this research given by the CNRST (Centre National pour la Recherche Scientifique ET Technique) National Center for Scientific and Technical Research, Morocco. And we thank Marilou Ong, Editor-in-Chief of Fuzzy Information and Engineering journal and the reviewers for the time they put in for review as well as constructive suggestions for editing the manuscript.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Fatima Es-sabery received the professional license in computer sciences (option IT development), and master of Business Intelligence degrees from Sultan Moulay Sliman's University, Beni-Mellal, Morocco in 2014 and 2016 respectively. Her general interests span the areas of Data Mining, Big Data, and Wireless Sensor Networks.

Abdellatif Hair works as a full Professor in the Computer Department of FST of Beni Mellal (Morocco), and member of LAMSC laboratory. His research interests include integration of viewpoints and in Object-Oriented Analysis/Design, Security of mobile Agent, CBSE, WSN, and BigData Mining. He has directed several Ph.D. thesis.

ORCID

Fatima Es-sabery http://orcid.org/0000-0002-6158-4148

References

[1] Kaur H. A literature review from 2011 to 2014 on students academic performance prediction and analysis using decision tree algorithm. J Glob Res Comp Sci. 2018;9(5):10–15.

[2] Es-sabery F, Hair A. An improved ID3 classification algorithm based on correlation function and weighted attribute. Proceedings of the 2019 International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS), Taza, Morocco, Dec 26–27; 2019.

[3] I. Abdallah, V. Dertimanis, C. Mylonas, et al., Fault diagnosis of wind turbine structures using decision tree learning algorithms with big data. Proceedings of the 2018 European Safety and Reliability Conference, London, June; 2018.

[4] Goswami S, Chakraborty S, Ghosh S, et al. A review on application of data mining techniques to combat natural disasters. J Ain Shams Eng. 2018;9(3):365–378.

[5] Wu X, Zhu X, Wu GQ, et al. Data mining with big data. J IEEE Trans Knowl Data Eng. 2014;26(1): 97–107.

[6] Es-sabery F, Hair A. Big data solutions proposed for cluster computing systems challenges: a survey. Proceedings of the 3rd International Conference on Networking Information Systems & Security, Marrakech, Morocco, Mar 31–Apr 2; 2020. doi:10.1145/3386723.3387826.

[7] Vaidya M, Deshpande S. Comparative analysis of various distributed file systems performance evaluation using map reduce implementation, Proceedings of the 2016 International Conference on Recent Advances and Innovations in Engineering, Jaipur, India, Dec 23–25; 2016.

[8] Zadeh LA. Fuzzy sets. J Inf Cont. 1965;8(3):338–353.

[9] Pedrycz W, Pedrycz PofCEW, Gomide F. An introduction to fuzzy sets: analysis and design. Berkeley (CA): MIT Press; 1998.

[10] P. Ducange, F. Marcelloni, and A. Segatori, A MapReduce-based fuzzy associative classifier for big data. Proceedings of the 2015 IEEE International Conference on Fuzzy Systems, Istanbul, Turkey, Aug 2–5; 2015.

[11] Afzaal M, Usman M, Fong ACM, et al. Fuzzy aspect based opinion classification system for mining tourist reviews. J Adv Fuzzy Syst Hindawi. 2016;2016:6965725. doi:10.1155/2016/6965725.

[12] M. Abdul-Jaleel, Y. H. Ali and N. J. Ibrahim, Fuzzy logic and genetic algorithm based text classification twitter. Proceedings of the 2nd Scientific Conference of Computer Sciences, Baghdad, Iraq, Mar 27–28; 2019.

[13] Pegalajar MC, Ruiz LGB, Snchez-Maran M, et al. A Munsell colour based approach for soil classification using fuzzy logic and artificial neural networks. J Fuzzy Set Syst. 2019;375:1–196. doi:10.1016/j.fss.2019.11.002.

[14] Melin P, Olivas F, Castillo O, et al. Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic. J Expert Syst Appl. 2013;40:3196–3206. doi:10.1016/j.eswa.2012.12.033.

[15] Rubio E, Castillo O, Valdez F, et al. An extension of the fuzzy possibilistic clustering algorithm using type-2 fuzzy logic techniques. J Adv Fuzzy Syst. 2017;2017:7094046, doi:10.1155/2017/7094046.

[16] Snchez D, Melin P, Castillo O. Optimization of modular granular neural networks using a firefly algorithm for human recognition. J Eng Appl Artif Int. 2017;64:172–186. doi:10.1016/j.engappai.2017.06.007.

[17] Chatterjee A, Mukherjee S, Kar S. A rough approximation of fuzzy soft set-based decision-making approach in supplier selection problem. J Fuzzy Inf Eng. 2018;10(2):178–195. doi:10.1080/16168658.2018.1517973.

[18] Nilashi M, Ibrahim O, Dalvi M, et al. Accuracy improvement for diabetes disease classification: a case on a public medical dataset. J Fuzzy Inf Eng. 2017;9(3):345–357. doi:10.1016/j.fiae.2017.09.006.

[19] Bhamare D, Suryawanshi P. Review on reliable pattern recognition with machine learning techniques. J Fuzzy Inf Eng. 2018;10(3):362–377. doi:10.1080/16168658.2019.1611030.

[20] Sebastian S, Ramakrishnan TV. Multi-fuzzy sets: an extension of fuzzy sets. J Fuzzy Inf Eng. 2011;3(1):35–43. doi:10.1007/s12543-011-0064-y.

[21] R. C. Prati, F. Charte, F. Herrera, A first approach towards a fuzzy decision tree for multilabel classification. Proceedings of 2017 IEEE International Conference on Fuzzy Systems, Naples, Italy, July 9–12; 2017.

[22] Levashenko V, Martincova P. Fuzzy decision tree for parallel processing support, journal of information. Contr Manage Syst. 2005;3(1):4552.

[23] Suryawanshi RD, Thakore DM. Decision tree classification implementation with fuzzy logic. J Comp Sci Netw Secur. 2012;12(10):9397.

[24] Wang XZ, Dong LC, Yan JH. Maximum ambiguity-based sample selection in fuzzy decision tree induction. J IEEE Transs Knowl Data Eng. 2012;24(8):14911505.

[25] Bai Y, Wang D. Fundamentals of fuzzy logic control fuzzy sets, fuzzy rules and defuzzification, chapter. In: Y Bai, H Zhuang, D Wang, editor. Advanced fuzzy logic technologies in industrial applications. London: Springer; 2006. p. 1736. doi:10.1007/978-1-84628-469-42.

[26] Liu H, Cocea M. Fuzzy rule based systems for interpretable sentiment analysis. Proceedings of the Ninth International Conference on Advanced Computational Intelligence (ICACI), Doha, Qatar, Feb 4–6, p. 129136; 2017. doi:10.1109/ICACI.2017.7974497.

[27] Xizhao W, Hong J. On the handling of fuzziness for continuous-valued attributes in decision tree generation. J Fuzzy Set Syst. 1998;99(3):283–290. doi:10.1016/S0165-0114(97)00030-4.

[28] Berzal F, Cubero J-C, Marn N, et al. Numerical attributes in decision trees: a hierarchical approach. Proceedings of the Advances in Intelligent Data Analysis V, Berlin. Heidelberg; 2003. p. 198207. doi:10.1007/978-3-540-45231-719.

[29] Zhang B, Pedrycz W, Fayek AR, et al. Granular aggregation of fuzzy rule-based models in distributed data environment. J IEEE Trans Fuzzy Syst. 2019. doi:10.1109/TFUZZ. 2020.2973956.

[30] Kerr-Wilson J, Pedrycz W. Generating a hierarchical fuzzy rule-based model. J Fuzzy Set Syst. 2020;381:124139. doi:10.1016/ j.fss.2019.07.013.

[31] Ross TJ. Fuzzy logic with engineering applications. New York: Wiley; 2005.

[32] Afsari F, Eftekhari M, Eslami E, et al. Interpretability-based fuzzy decision tree classifier a hybrid of the subtractive clustering and the multi-objective evolutionary algorithm. J Soft Comp. 2013;17(9):1673–1686.

[33] UCI Machine Learning Repository. Available from: https://archive.ics.uci.edu/ml/datasets.php.

[34] Damanik IS, Windarto AP, Wanto A, et al. Decision tree optimization in C4.5 algorithm using genetic algorithm. Proceedings of the International Conference on Computer Science and Applied Mathematics, Medan, Indonesia; 2019. doi:10.1088/1742-596/1255/1/012012.

[35] Cherfi A, Nouira K, Ferchichi A. Very fast C4.5 decision tree algorithm. J Appl Artif Int. 2018;32(2):119–137. doi:10.1080/08839514.2018.1447479.

[36] Lee J-S. AUC4.5: AUC-based C4.5 decision tree algorithm for imbalanced data classification. J IEEE Access. 2019;7:106034106042. doi:10.1109/ACCESS.2019.2931865.