A Fuzzy Mining Approach for Energy Efficiency in a Big Data Framework

Carlos Fernandez-Basso, M. Dolores Ruiz, and Maria J. Martin-Bautista, *Member, IEEE*

***Abstract*— The discovery and exploitation of hidden information in collected data have gained attention in many areas, particularly in the energy field due to their economic and environmental impact. Data mining techniques have then emerged as a suitable toolbox for analyzing the data collected in modern network management systems in order to obtain a meaningful insight into consumption patterns and equipment operation. However, the enormous amount of data generated by sensors, occupational, and meteorological data involve the use of new management systems and data processing. Big Data presents great opportunities for implementing new solutions to manage these massive data sets. In addition, these data present values whose nature complicates and hides the understanding and interpretation of the data and results. Therefore, the use of fuzzy methods to adequately transform the data can improve their interpretability. This article presents an automatic fuzzification method implemented using the Big Data paradigm, which enables, in a later step, the detection of interrelations and patterns among different sensors and weather data recovered from an office building.**

***Index Terms*—Big Data, data mining, energy building, fuzzy association rules, operational research.**

I. Introduction

NOWADAYS, modern management systems can generate thousands of measurements every minute from the different detection devices that record their operation. Companies in the energy sector are increasingly aware of the great opportunity that the analysis and exploitation of these data can bring (see for instance [1] and [2]). To this end, there is a tendency to store and process this type of data in order to obtain a meaningful insight into consumption patterns and the operation of equipment.

However, the enormous amount of data generated by sensors and other data sources, such as meteorological and occupancy data, require new infrastructures and algorithms capable of storing, processing and analyzing them. Big Data presents great opportunities for implementing new solutions to manage these massive data sets. Moreover, the nature of these data can be diverse and can be described in numerical, categorical, imprecise forms. To improve the interpretability of the data, we can modify the knowledge extraction algorithm through the use of fuzzy logic to create, for example, linguistic labels for sensors with numerical values that also provide meaningful semantics for the user.

The main challenge when processing large energy data is therefore to provide adequate methods and techniques capable of improving the quality of the data generated by the sensor metering of buildings. In this regard, different methods can be applied during the preprocessing phase in order to detect outliers [3] or by applying other cleaning data procedures. In general, these data are massive because they are generated with low frequencies by a large number of sensors. This massive quantity gives grounds to use Big Data techniques to process the data in a distributed way, thus improving the efficiency of the processes.

Sensor metering data are very often collected by means of numerical measurements taking values within a continuous range. This increases the difficulty of analyzing them on a larger scale due to their fine granularity. A primary approach is to divide the range of possible values into intervals in order to help the algorithms to process the data. But this division suffers from some drawbacks: first, the results can vary a lot depending on the applied division, and second, this division may not be very intuitive for its later analysis of results. Fuzzy sets have been proven to adequately represent data with soft borders, increasing the interpretability of results by associating meaningful linguistic labels to the generated fuzzy sets. Other approaches use interval programming methods to tackle the uncertainty that the data may contain [4].

In this article, we propose a fuzzification algorithm to adequately preprocess the data in order to apply, in a later step, fuzzy data mining techniques to discover potentially useful information that maybe hidden in the data.

In particular, we have applied association-rule discovery, an unsupervised technique able to find existing relationships between variables and their values. In addition, these results allow the operators to carry out procedures and the use of the equipment in the building better, thus improving its energy efficiency. On the other hand, the use of weather and occupation patterns for the energy use of the building would allow the optimization of the needs of the building in specific situations.

Manuscript received November 8, 2019; revised February 7, 2020 and April 23, 2020; accepted April 27, 2020. Date of publication May 4, 2020; date of current version October 30, 2020. This work was supported in part by the Spanish Ministries of Science, Innovation and Universities under Grant TIN2017-91223-EXP and in part by the Economy and Competitiveness under Grant TIN2015-64776-C3-1-R); and in part by the European Union (Energy IN TIME EeB.NMP.2013-4, No. 608981). *(Corresponding author: Carlos Fernandez-Basso.)*

Carlos Fernandez-Basso and Maria J. Martin-Bautista are with the Department of Computer Science and A.I. and CITIC-UGR, University of Granada, 18071 Granada, Spain (e-mail: cjferba@decsai.ugr.es; mbautis@decsai.ugr.es).

M. Dolores Ruiz is with the Department of Statistics and Operative Research, University of Granada, 18071 Granada, Spain (e-mail: mdruiz@decsai.ugr.es).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2020.2992180

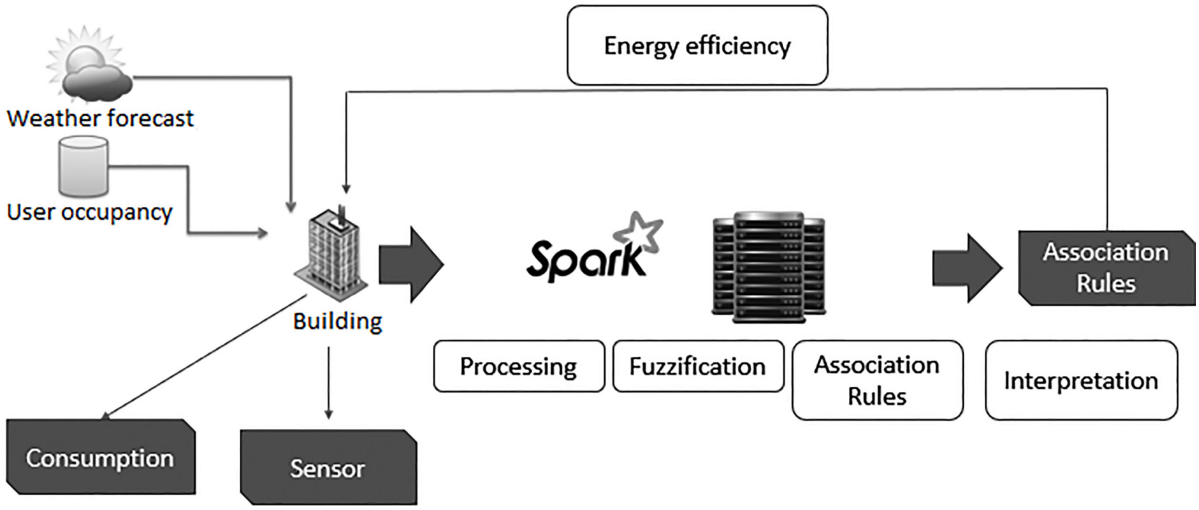

Fig. 1. General process of our proposal.

To this end, we have deployed the whole system following the map-reduce paradigm, which allows the distributed computation of large volumes of data. Specifically, we used the Spark platform [5] together with an unstructured storage following NoSQL specifications, which enables an efficient storage of sensor data collected in buildings.

The whole system has been successfully applied in an office building located in Bucharest, obtaining a set of patterns describing the operational and energetic functioning of the building. Nevertheless, the presented approach could also be applied in other types of buildings.

The obtained patterns describe the day-to-day working of the building, but they could also help to discover the poor functioning of some systems due to abnormal circumstances.

The rest of this article is organized as follows. Section II reviews previous related research and introduces the necessary background of related concepts. Section III describes the design of our system focusing on the developed fuzzification algorithm. Section IV presents our results for the office building located in Bucharest. Finally, Section VI concludes this article.

II. Background

Data mining techniques are widely used in the field of energy as can be observed in [6]–[9]. In [6] and [7], the authors reviewed how some of the traditional data mining techniques have been used to obtain construction-related information. Additionally, an association-rule discovery tool was used to explore correlations between building data in two different time periods: a day and a year. This study allowed the detection of some equipment failures in two ventilation units and to propose low-cost strategies for saving energy. In [8], the authors focused on the different data mining techniques used for energy management, especially in the construction sector, discussing the main challenges and opportunities that will arise with the advent of new computational technologies such as Big Data.

A more recent revision can be found in [9], where unsupervised data mining techniques are presented, paying special attention to the operational data mining of massive data collected from buildings in order to find significant patterns. More recent reviews such as [10], focus on the use of data mining tools to predict the consumption of the building by taking into account sensor metering, time, and building occupancy data.

There are very few studies that have investigated fuzzy data in the field of energy management. In this regard, we can highlight the research in [11], where the authors look for anomalies in sensor data under uncertainty.

Our article is in a similar direction but we propose to automatically fuzzify data collected by sensors and afterward apply fuzzy association rule mining to discover potentially useful patterns in the field of energy management in buildings.

A. Our Approach

In Fig. 1, we depict the complete system of our proposal. It can be divided into three big blocks. In the first step, the data collection are carried out. Depending on the building different types of data are generated with more or less reliability. For example, in the case of nonresidential buildings (the example of our case study) the agenda containing the occupation of the building will be more reliable than that of a hotel, where the guests do not have fixed schedules.

The second step comprises the core of the computation with different phases: preprocessing, fuzzification, and application of data mining techniques. In our approach, distributed processing tools are used to improve efficiency and computational capacity.

Once we have obtained the results of our analytical techniques, the last step is the interpretation of results by experts or operators of the system in order to obtain knowledge that maybe useful for improving the maintenance processes and the energy efficiency of the building.

This whole process has been applied in the field of the Energy IN TIME project to nonresidential buildings: a hotel, an airport, and two office buildings, although we just present here the results obtained from one of the office buildings. Nevertheless, the proposed system is general and can be applied to other types of buildings.

B. Fuzzy Association Rules

In the data mining field, association rules are used to discover facts that often occur together within a particular data set. A typical example of this type of problem is figuring out which products from a supermarket are normally bought together. Association rules were formally defined for the first time by Agrawal *et al.* [12], although the analysis of associations was investigated much earlier in the more general framework of observational calculi in [13]. The problem consists of discovering implications of the form $A \to B$ where A, B are subsets of items from $I = \{i_1, i_2, \ldots, i_m\}$ fulfilling that $A \cap B = \emptyset$ in a database formed by a set of n transactions $D = \{t_1, t_2, \ldots, t_n\}$ each of them containing subsets of items from I. A is usually referred as the antecedent and B to the consequent of the rule.

The problem of discovering association rules is divided into two subtasks.

1) Finding all the item sets above the minimum support threshold, where support is provided by the percentage of transactions containing the items. These sets of items, or itemsets, are known as frequent itemsets.
2) On the basis of the found frequent itemsets, rules are discovered as those exceeding the minimum threshold for confidence or another assessment measure generally established by the user.

However, the nature of the data can be diverse and can be described numerically, categorically, or imprecisely. In the case of numerical elements, a first approximation could be to categorize them so that, for example, the temperature of a room can be given by a range to which it belongs, such as [24°, 30°]. However, depending on how these intervals are defined, the results obtained can vary a lot. To avoid this, the use of linguistic tags such as "warm" represented by a fuzzy set is a good option to represent the temperature of a room, having at the same time significant semantics for the user [14]. Beside this, we may also have a data set with inherent imprecise knowledge, where ordinary crisp methods cannot be directly applied (see for instance [15]).

To deal with this kind of data, the concept of fuzzy transaction and fuzzy association rule are defined in [16] and [17].

Definition 1: Let I be a set of items. A fuzzy transaction, t, is a nonempty fuzzy subset of I in which the membership degree of an item $i \in I$ in t is represented by a number in the range [0, 1] and denoted by $t(i)$.

By this definition, a crisp transaction is a special case of fuzzy transaction. We denote by $\tilde{D}$ a database consisting in a set of fuzzy transactions.

Definition 2: Let $A \subseteq I$ be an itemset, i.e., a subset of items in I. The degree of membership of A in a fuzzy transaction $t \in \tilde{D}$ is defined as the minimum of the membership degree of all its items

$$t(A) = \min_{i \in A} t(i). \tag{1}$$

Definition 3: Let $A, B \subseteq I$ be itemsets in the fuzzy database $\tilde{D}$. Then, a fuzzy association rule $A \to B$ is satisfied in $\tilde{D}$ if and only if $t(A) \leq t(B)\ \forall t \in \tilde{D}$, that is, the degree of satisfiability of B in $\tilde{D}$ is greater than or equal to the degree of satisfiability of A for all fuzzy transactions t in in $\tilde{D}$.

Assessment measures for fuzzy association rules have been studied in numerous papers according to different perspectives (see a review in [18]). The approach followed here it is a cardinality-based generalization presented in [17] and extended in other works (see for instance [19], [20]) by considering a finite set of α-cuts for the unit interval.

Definition 4: The support of an itemset A in a fuzzy database $\tilde{D}$ is defined a follows:

$$F\text{Supp}(A) = \sum_{\alpha_i \in \Lambda} (\alpha_i - \alpha_{i+1}) \frac{|\{t \in \tilde{D} : t(A) \geq \alpha_i\}|}{|\tilde{D}|} \tag{2}$$

where $\Lambda = \{1 = \alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of α-cuts.

Definition 5: The support of a fuzzy rule $A \to B$ in a fuzzy database $\tilde{D}$ is defined as follows:

$$F\text{Supp}(A \to B) = \sum_{\alpha_i \in \Lambda} (\alpha_i - \alpha_{i+1}) \frac{|\{t \in \tilde{D} : t(A) \geq \alpha_i \text{ and } t(B) \geq \alpha_i\}|}{|\tilde{D}|} \tag{3}$$

where $\Lambda = \{1 = \alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of α-cuts.

Definition 6: The confidence of a fuzzy rule $A \to B$ is defined as follows:

$$F\text{Conf}(A \to B) = \sum_{\alpha_i \in \Lambda} (\alpha_i - \alpha_{i+1}) \frac{|\{t \in \tilde{D} : t(A) \geq \alpha_i \text{ and } t(B) \geq \alpha_i\}|}{|\{t \in \tilde{D} : t(A) \geq \alpha_i\}|} \tag{4}$$

where $\Lambda = \{1 = \alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of α-cuts.

By using support and confidence measures and setting appropriated thresholds, fuzzy association rules can be discovered by fixing a set of predefined α-cuts [19]. Note that considering a sufficiently dense set of α-cuts in the unit interval, the obtained measure will be a good approximation of the real measure that should consider every $\alpha \in [0, 1]$ appearing in the data set.

C. Big Data Paradigm

The most famous framework for Big Data is *MapReduce* designed by Google in 2003 [21]. It has become one of the most relevant tools for processing large data sets with parallel and distributed algorithms in a cluster. The MapReduce framework manages all data transfers and communications among the systems. It also provides redundancy, fault-tolerance and job scheduling. In this programming paradigm, we usually have two

phases. First, the $Map()$ function, which makes the processing of data and returns the data transformed into key value pairs depending on our necessities. Second, the $Reduce()$ function that aggregates the lists of $< key, value >$ pairs sharing the same key to obtain a piece of processed data.

MapReduce algorithms can be programmed in different frameworks. One of the most used that have been proven and which works quite fast is Apache Spark [22]. It appeared as an open-source framework built around speed, ease of use, and sophisticated analytics [5]. The most important feature of Spark is that it allows in-memory computing, and, as a consequence more complex algorithms can be developed. This is because Spark supports an advanced directed acyclical graphics execution engine that allows more complex data flows using several MapReduce phases, a procedure that is not possible with other tools such as Hadoop.

Apache Spark has implemented a data structure to abstract the concept of data partition. This structure is called the resilient distributed data set (RDD) [23], meaning that data are distributed across the clusters. The RDD has two different types of operations. The first type of transformation converts the RDD structure into a different RDD, which is called *Transformation operations*. The second type is *evaluation Actions* performed over the above transformations, which return a final value for each RDD partition. The programmer has to take into account that evaluations are not executed until a specific *Action operation* is specified in the code. This is due to the "lazy" evaluation of Spark that strongly distinguishes between transformations and actions.

For the implementation of the proposed methodology, the Spark tool has been used due to its large computing capacity and compared to Hadoop because it uses memory storage, thus improving its efficiency (see the complete comparison made in [22] where Hadoop and Spark frameworks are compared in several machine learning algorithms).

III. Methodology

In our proposal, before applying fuzzy association-rule mining, we have to preprocess the data collected from the building. We can observe the workflow of the proposal in Fig. 2, where we have distinguished several phases.

The first phase comprises the collection and storage of building data, which are e.g., data from sensors and building equipment, weather, and occupation. All these data are stored in a NoSQL database MongoDB [24] for their efficient management in the insertion and search of documents. Moreover, this MongoDB enables the storage of data with different structures and different fields [25]. This choice is due to the fact that the data from the sensors arrive with a very low frequency, so the insertions in the database must be fast, and also each sensor can have values with different structures (e.g., XML, real values, strings, etc.)

Subsequently, the preprocessing phase of the data is carried out. The collected data from sensors may contain lost values, outliers, etc., and therefore some transformation to be used by the algorithms is needed.

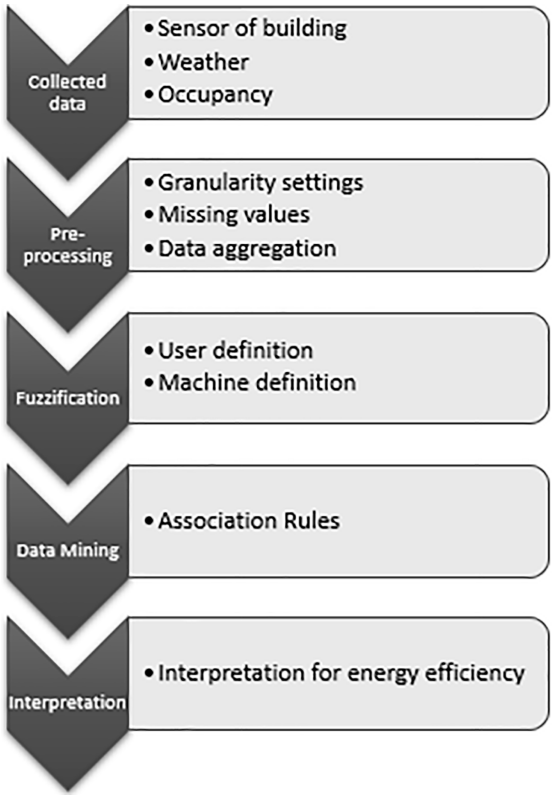

Fig. 2. General workflow of the proposed data mining framework.

After this, the continuous numerical values are transformed following a fuzzification method explained in Section III-C. By means of this process, the range of values are divided into meaningful fuzzy sets, where some linguistic labels can be associated to each fuzzy set. This step improves the interpretability of the data and the obtained results will adjust better to the nature of the variables. Depending on the type of variable, the fuzzification procedure can either be carried out automatically or by the help of expert knowledge.

Finally, when we have the preprocessed and transformed the data, we proceed to apply data mining techniques. In this case study, extraction of fuzzy association rules has been applied in order to extract hidden relationships from the sensors, occupation, and the environment of the building. Once the results have been obtained, the discovered patterns are interpreted with the help of end users to improve the energy efficiency of the building.

A. Data Collection

The developed methodology has been used in a large office building in Bucharest. In Fig. 3, the different data sources collected and added to the database can be seen.

Each of these pieces of data were collected using different procedures. On the one hand, data provided by building management systems (BMS) [26] such as sensors, actuators, and building equipment data have been considered. In particular, sensors and BMS data have been sent via a real-time messaging system application programming interface (API) (rabbitmq [27]) to the

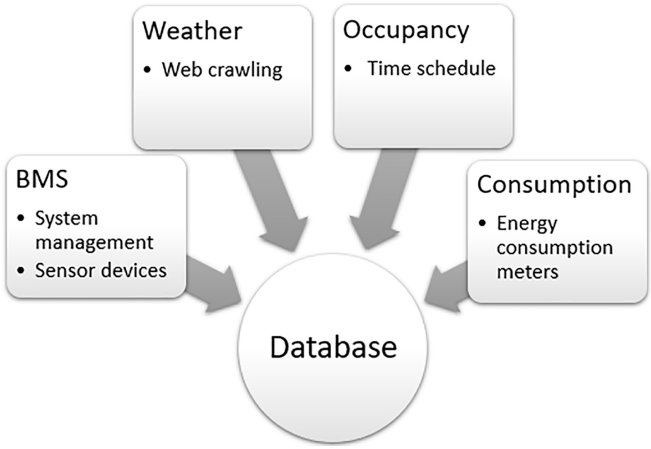

Fig. 3. Schema of the type of data collected for the office building.

TABLE I
EXAMPLE OF THE TEMPORAL GRANULARITY PROCESSING OF SOME TEMPERATURE DATA

Time	Temperature
1/1/2016 15:14:30	16
1/1/2016 15:16:45	17
↓	
Time	Temperature
1/1/2016 15:15:00	16.5

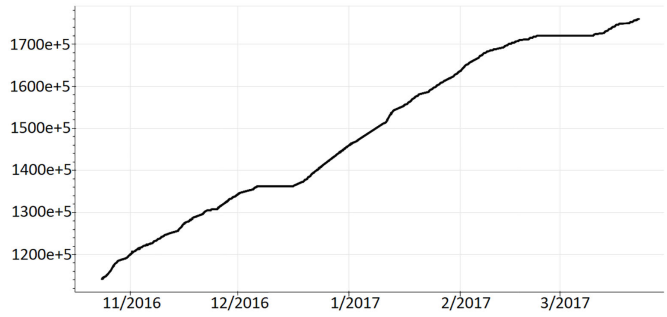

Fig. 4. Heating consumption counter.

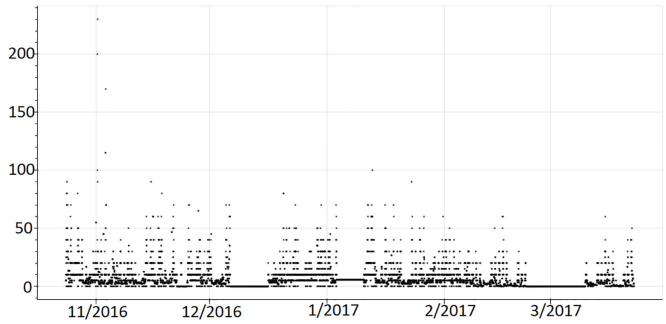

Fig. 5. Heating consumption (every 15 min).

database. This includes energy consumption metering from the building. On the other hand, the occupancy data have been obtained by processing the working hours of the offices in the building. Finally, the weather has been obtained using a web crawler that stores, in a structured way, the meteorological forecast foreach day.

B. Preprocessing

Different techniques are commonly used in data preprocessing. First of all, the granularity of the data is standardized. Since the data from the sensors are collected in different instants of time it is necessary to group these measurements in the same instants of time. Table I shows an example of this transformation. In the example, the values of the time instants are grouped every 15 min (this parameter can be changed according to user preferences). Depending on the type of values, the method used can be fixed according to different criteria. For instance, for continuous data, we may be interested in considering the average (as in the example of Table I) or consider the last value sent by the sensor. This transformation allows us to generate a set of transactional data suitable for applying data mining techniques. Some of the collected data come from energy meters, which function as accumulators. You can see an example of this type of data in Figs. 4 and 5, where its value is always increasing and represents the energy consumption can be seen. This type of variable has been processed through a transformation function to represent the consumption of the building in different time slots. That is, each temporal piece of data represents the consumption in that interval of time. We can see an example after applying this transformation in Fig. 5.

After processing the data and obtaining the desired granularity, transactions with lost values are eliminated, as well as transactions containing outlier measurements made by sensors. The *sklearn* python library [28] was used to determine these outliers, specifically using the elliptic envelope fitting function [29]. At the end of this procedure, the transactional database comprises sensor, weather, and occupation data for each fraction of time without anomalous values.

C. Fuzzification

The collection of data from sensors, counters, weather, or building occupation variables has a nature that is difficult to represent and interpret by end users, since they are continuous values with often complex measures to interpret and understand. Fuzzification of these data can improve the results found by the mining algorithms, and at the same time increase the interpretability of the results.

We propose a fuzzification algorithm that allows an automatic processing defined by the machine by using the data values according to their distribution. In addition, the algorithm allows the definition of the fuzzy labels by a user (see the two different types of input that can be provided in Algorithm 1). For this, we have developed a distributed algorithm in Spark following the MapReduce philosophy. This enables the processing of large amounts of data, as in the case of sensor generated data in buildings. The general process is described in Algorithm 1. For this, we used Spark for the distribution of data throughout the

Algorithm 1: Main Spark Procedure for Fuzzification Preprocessing Algorithm.

1: **Input:** *Data:* RDD transactions: $\{t_1, \ldots, t_n\}$
2: **Input:** *DefaultIntervals:* number of intervals automatically generated by the algorithm
3: **Input:** *Intervals:* Hash-list of intervals for each variable: $\{Variable_i : [\{Intervals\}, \{Labels\}]$ $, \ldots, Variable_p : [\{Intervals\}, \{Labels\}]\}$
4: **Output:** Fuzzy transactions containing fuzzified values

Start Algorithm

5: Features = Dataset.NameFeatures()
6: $broadcast(Global\_Features)$ #Create a broadcast variable for its use across the cluster
7: **DCS in** q **chunks of Data:** $\{S_1, \ldots, S_q\}$
8: $FuzzyData_{S_i} \leftarrow S_i$.**Map** $(Fuzzification$ $(t_k \in S_i))$
Map function computes independently each transaction in S_i
9: $FuzzyDatabase ==$ **ReduceByKey**$(Aggregation$ $(FuzzyData_{S_1}, \ldots, FuzzyData_{S_q}))$
10: **return** $FuzzyDatabase$

cluster. The algorithm has as input a data set, a python dictionary (hash) and an integer. The dictionary is used to store the intervals and labels for variables that have been defined by experts. On the other hand, the default number of tags is used for variables that have not been defined by users and will be automatically created depending on their distribution. Note that Spark automatically divides data into chunks for distributed calculation. We have specified this with the acronym DCS (distribute computing using Spark) and representing each piece of data by S_i. In line 6, a global variable is used throughout the whole cluster can be seen, which is then used by the function that distributes the computation through MapReduce (line 8 of Algorithm 1).

Additionally, in line 8, the procedure calls the fuzzification function described in Algorithm 2. This function is divided into different parts. First, it checks if the name of the variable is found in the $Intervals$ hash-list, if it is found in the python dictionary the new fuzzified variables are created using the names of the labels specified by $Intervals$ and its configuration (i.e., computation of membership degrees) attending to the specified interval (see lines 10–16 of Algorithm 2). If the variable is not found in the dictionary, an automatic procedure is used that divides the values of the variable in a number of intervals defined in $DefaultIntervals$ according to the percentiles of the variable. Fig. 6 presents an example with the value of $DefaultIntervals = 3$ where the y-axis represents the degree of membership and x-axis the percentile of the variable. In this example, percentiles employed have been 25 and 37.5 for defining the trapezoidal form of the first label and left part of second label, and, 62.5 and 75 to define the right part of second label and the third label. So, the $GenerateIntervals$ function divides the set into k equidistributed fuzzy sets using the corresponding

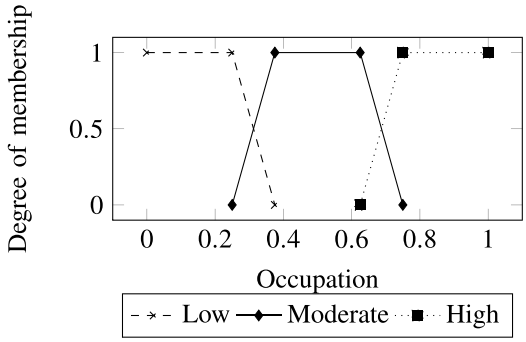

Fig. 6. Example of automatic execution with three default intervals.

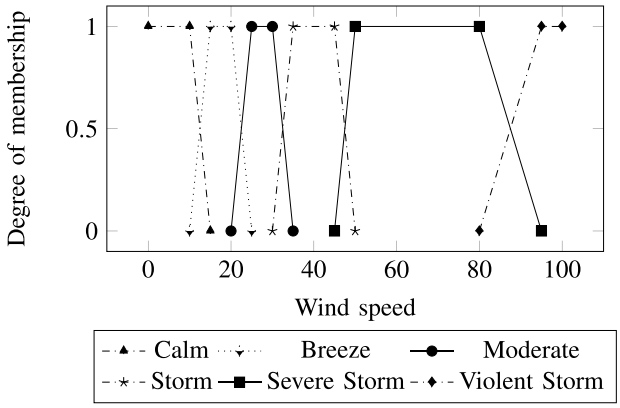

Fig. 7. Wind speed.

percentiles. For instance, for $k = 4$, the considered percentiles are computed as follows:

$$\left\{\frac{100}{k+1}, \frac{100}{k+1} + \frac{100}{(k+1)(k-1)}, \frac{2 \cdot 100}{k+1} + \frac{100}{(k+1)(k-1)} \right.$$
$$\frac{2 \cdot 100}{k+1} + \frac{2 \cdot 100}{(k+1)(k-1)}, \frac{3 \cdot 100}{k+1} + \frac{2 \cdot 100}{(k+1)(k-1)}$$
$$\left.\frac{3 \cdot 100}{k+1} + \frac{3 \cdot 100}{(k+1)(k-1)}\right\}$$

which results in

$$\{p_{20}, p_{26.6}, p_{46.6}, p_{53.3}, p_{73.3}, p_{80}\}.$$

On the contrary, the $FuzzyDivision$ function uses the defined intervals contained in the global variable $Intervals$.

For the use case under study, the experts determined different intervals for generating the fuzzy labels. These depend on the nature of the variable, e.g., external temperature, humidity, occupation, etc. Examples of these fuzzy intervals can be seen in Figs. 7–10. Figs. 11–13 show an example of the heating consumption data (see Fig. 4 before fuzzification) after their transformation into three different fuzzy sets with the labels: low, medium, and high.

D. Data Mining: Fuzzy Association Rules

After data preprocessing and fuzzification, data mining techniques were applied to the processed data. In particular, an

Algorithm 2: Fuzzification Function.

Input: *Data:* A transaction: $t_k = \{item_1, \ldots, item_m\}$
Global distributed variable: *Intervals:* Hash-list of intervals for each variable : $\{Variable_1 : [\{Intervals\}, \{Labels\}], \ldots, Variable_p : [\{Intervals\}, \{Labels\}]\}$
Input: *DefaultIntervals:* number of intervals automatically generated by the algorithm
Output: Fuzzy transaction

Start Algorithm

```
Features = Dataset.NameFeatures()
DistributeVariable(Features)
DCS in q chunks of Data: {S_1, ..., S_q}
i=0
do
    # Check if the variable exists in the hash list
    if Feature[i] ∈ Intervals then
        Interval=Intervals[Feature[i][0]]
        Labels=Intervals[Feature[i][1]]
    else
        Interval = GenerateIntervals(DefaultIntervals,Data[Feature[i]])
        Labels = GenerateLabels(DefaultIntervals)
    end if
    for j = 0; j < |Labels|; j++ do
        FuzzyData[Label]=FuzzyDivision(Interval[j], Interval[j+1], type)
        # type ="linear," "exponential," "logarithmic"...
        i++
    end for
while |Feature| > i
return FuzzyData
```

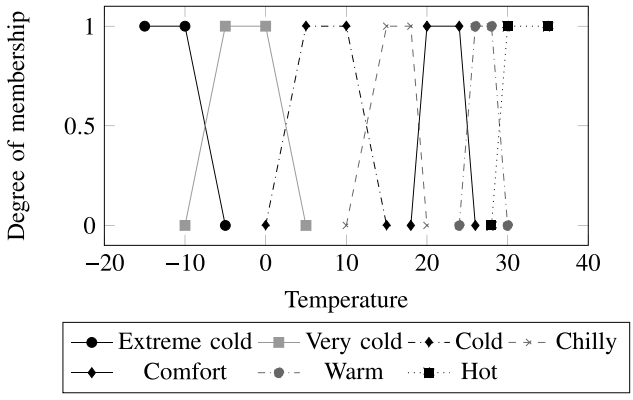

Fig. 8. Temperature.

algorithm for association-rule discovery was applied in Big Data (BDFARE Apriori-TID Big Data fuzzy association-rule extraction [30], [31]). This algorithm was also implemented following the MapReduce paradigm under the spark framework and enables the processing of huge sets of fuzzy transactions,

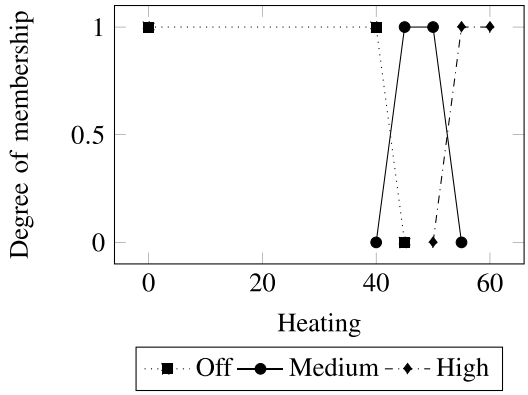

Fig. 9. Heating.

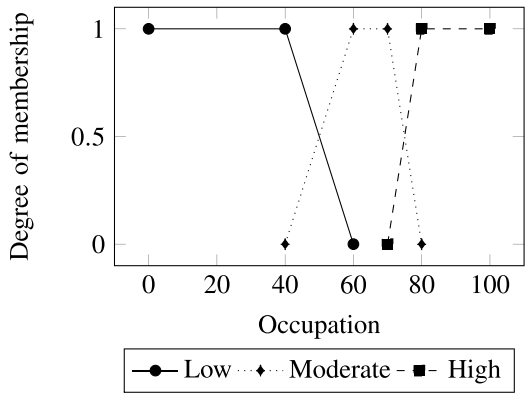

Fig. 10. Occupation.

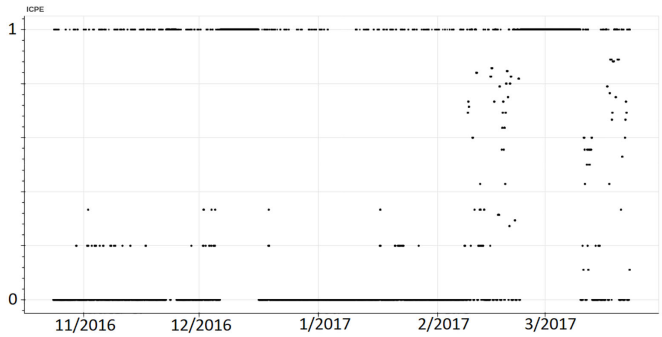

Fig. 11. Low-fuzzy label (heating consumption).

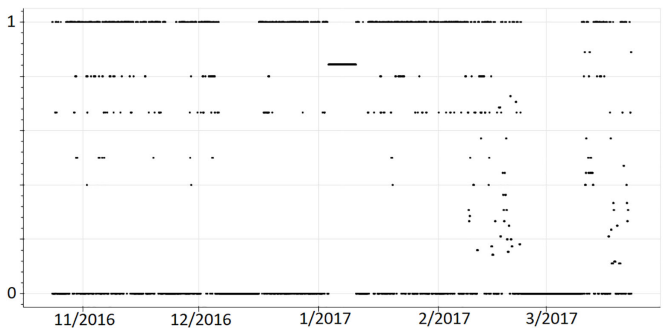

Fig. 12. Medium-fuzzy label (heating consumption).

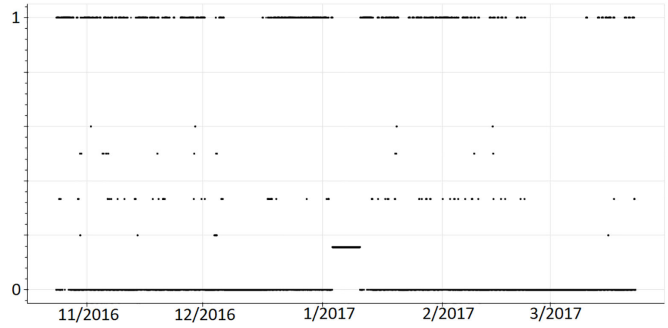

Fig. 13. High-fuzzy label (heating consumption).

finding frequent itemsets and fuzzy association rules exceeding the imposed thresholds for support and confidence, given a set of α-cuts.

IV. Results

The results from applying our proposal must be analyzed from two points of view. On the one hand, the efficiency and capabilities of our distributed processing that allows the more efficient processing of larger data sets. On the other hand, by means of this processing, fuzzification of the variables and their application to discover fuzzy association rules. The aim is to extract energy patterns in order to improve the knowledge, we have about the functioning of the building and to be able to improve tasks such as maintenance and energy efficiency by means of an improvement in the use of the systems.

Data collected from an office building in Romania were retrieved for the analysis. The building is located in Bucharest, a city with warm summers and very cold and dry winters. The building is comprised of offices with a constant flow of people and fixed-scheduled plans for indoor conditions.

The considered set of data comprises 273 sensors containing different metering data with a total of 3 649 678 transactions corresponding to data collected from September 2016 to September 2017. The procedure to collect the data by the system was described in Section III-A. The set of sensors can be roughly classified into meters and sensor status. More specifically, we distinguished the following groups: 1) electric energy, 2) heating agent, 3) domestic water, 4) air-conditioning, 5) temperature, and 6) humidity meters. From the setup and status category, there are different sensors related to heating, lightning, windows, etc.

A. Efficiency Analysis

As previously mentioned, the proposal was implemented using Spark, which enables MapReduce implementation in large data sets. We have carried out different tests in order to be able to analyze the improvement obtained by processing and fuzzifying these data using this framework. The experimental evaluation has been made on a 64-b architecture server with 16 cores on 2 Intel Xeon E5 and with 120 GB of RAM, functioning over an operative system with CentOS 6.8. Disabling intel's hyperthreading functionality for testing. The Spark version was 2.2 using a fully distributed mode with Cloudera Manager.

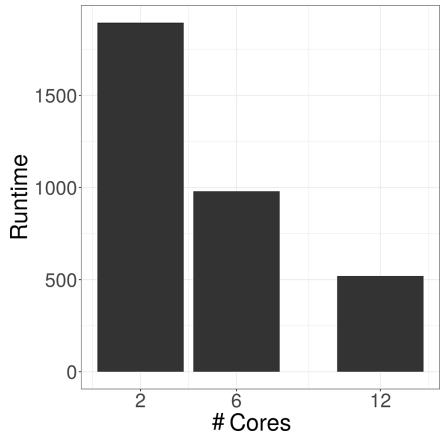

Fig. 14. Time in seconds with different core configurations.

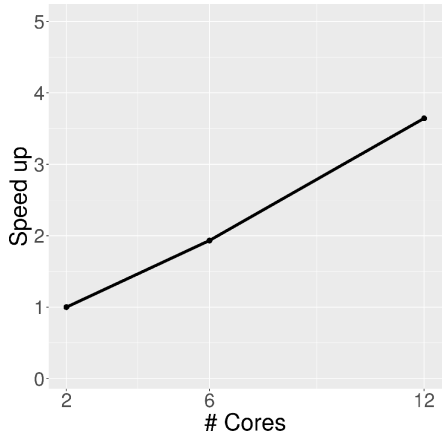

Fig. 15. Speedup versus number of processing cores.

Depending on the number of processors, different percentages of improvement can be achieved (regarding the computation time). In Fig. 14, the improvement achieved with different configurations can be seen (2, 6, and 12 cores).

With the purpose of analyzing the *speed up* and the *efficiency* [32]–[34] according to the number of cores, we have employed the known measure of speed up defined as [34], [35]

$$S_n = T_1/T_n \tag{5}$$

where T_1 is the time of the sequential algorithm and T_n is the execution time of the parallel algorithm using several cores. The efficiency [32]–[34] can be defined in a similar way as

$$E_n = S_n/n = T_1/(n \cdot T_n). \tag{6}$$

Fig. 15 and 16 show that the efficiency and speedup are improved as the number of cores increases, even if they are not optimal. The decrease in the efficiency is due to the core workloads and the network congestion used for communication among the cores.

In addition, Fig. 15 shows the speedup and evolution of the execution times consumed by the proposal. In Fig. 15, it is clearly observed that the greatest reduction in calculation time is achieved when the number of processors is 12.

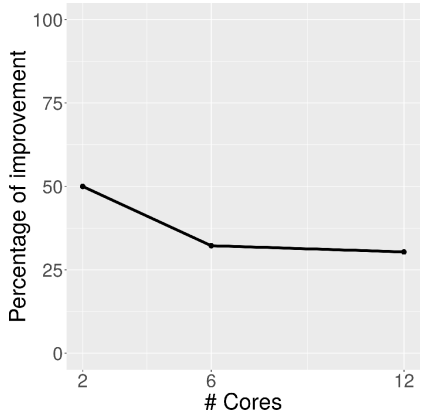

Fig. 16. Efficiency versus number of processing cores.

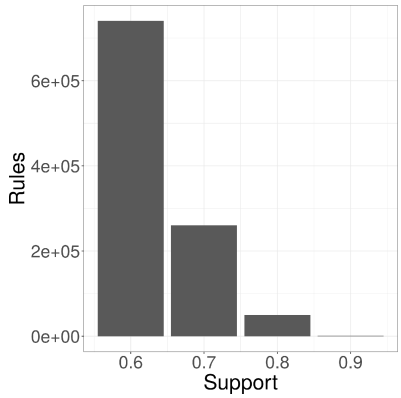

Fig. 17. Number of rules obtained with different parameters of the extraction algorithm.

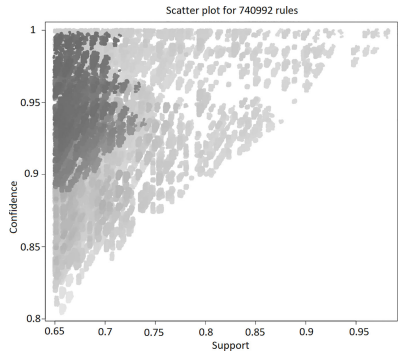

Fig. 18. Rules extracted for the office building in Bucharest according to their support (x-axis) and confidence (y-axis).

B. Knowledge Discovered

The experiments have been applied for different threshold configurations. In particular, we show here the results obtained when the minimum support was set to 0.6 and minimum confidence to 0.8. We also considered a set of ten equidistributed α-cuts. The support and confidence thresholds have been set higher than usual due to the high number of resulting rules obtained for lower values. In Fig. 17, the relationship amount of the support and the number of rules obtained is shown. As

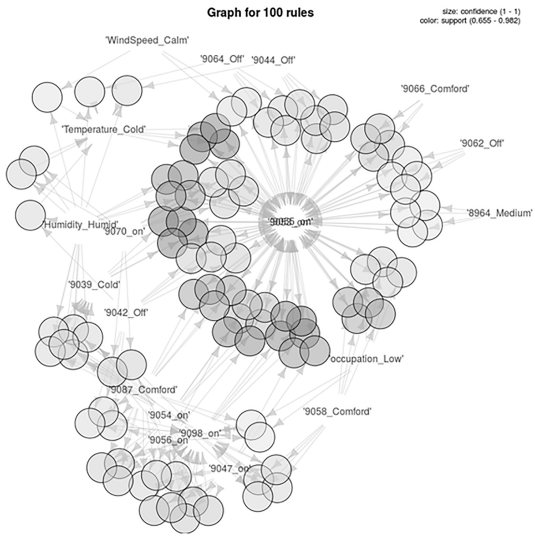

Fig. 19. Graph visualization of some association rules discovered for the office building in Bucharest.

can be observed, the number of rules increases as the support increases. Fig. 18 shows the distribution of the obtained rules (for an experiment with a support of 0.6) according to their confidence.

Fig. 18 and 19 show a summary of the quantity of association rules found, taking into account their support and confidence with the mentioned configuration Fig. 20 shows a subset of the discovered rules in the form of matrix, with the consequent (LHS) and the antecedent (RHS) of the rules.

C. Interpretation of the Results

The obtained set of rules has allowed us to discover hidden patterns in the operation of the building, which experts can then use to improve its efficiency and maintenance.

Having a look at the discovered patterns, we can highlight different rules. For example, in Fig. 20, the rule at the top left of the graph (position column 3, row 1) is

$$\{9098 = \text{on}, 9039 = \text{cold}\} \rightarrow \{9061 = \text{comfort}, 9096 = \text{cold}\}$$

which changing the identifiers of the sensors to a more descriptive name results in

$$\{\text{Setup PAN} = \text{on}, \text{Output temperature} = \text{cold}\} \rightarrow$$
$$\{\text{PAN temperature} = \text{comfort}, \text{PAS temperature} = \text{cold}\}.$$

In this rule, we can observe how the general operation of the building is described, i.e., outside the temperature is cold, the heating setup is ON and for that section of the building (PAN represents north area) the comfort temperature is achieved. In addition, we can see that other sections of the building such as PAS (representing the south area) is cold at the same time, so we could determine that there are two rooms or sections that are not usually occupied at the same time.

On the other hand in Fig. 21 some rules have been selected, where different behaviors of the building can be seen. For

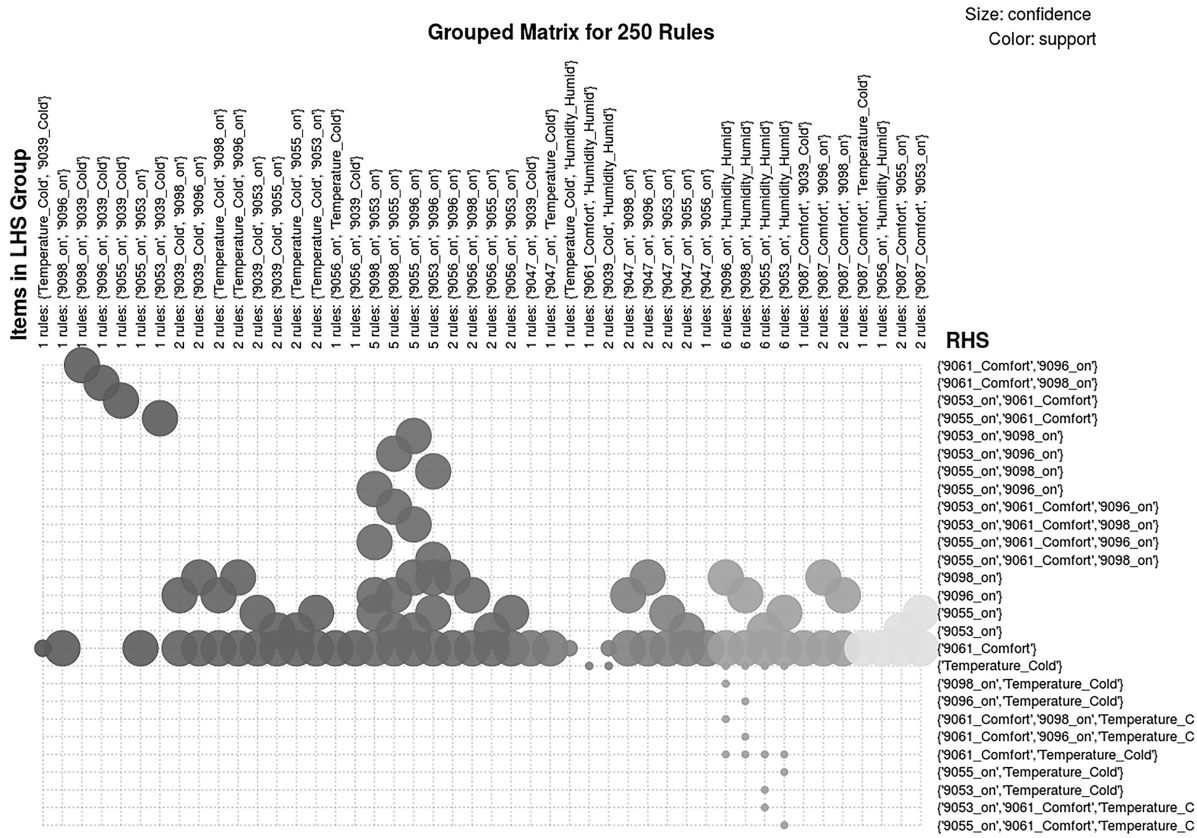

Fig. 20. Some association rules discovered for the office building in Bucharest. LHS stands for left-hand side of the rule or Antecedent and RHS for right-hand side or consequent.

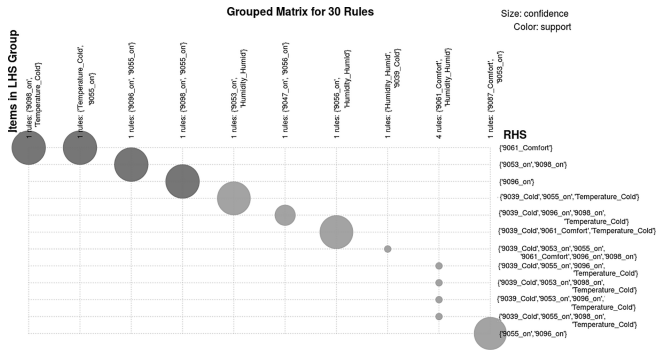

Fig. 21. Some fuzzy association rules discovered for the office building in Bucharest.

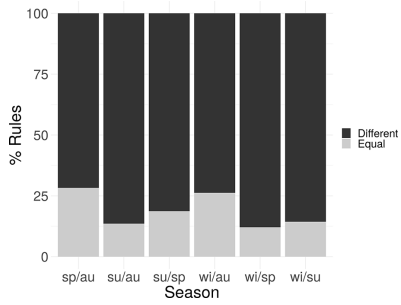

Fig. 22. Comparison of rules obtained by season (sp=spring, au=autumn, su=summer, wi=winter).

example, this rule obtained in winter

$$\{9047 = \text{off}, \text{Humidity} = \text{humid}, \text{temperature} = \text{cold}\} \rightarrow$$
$$\{9039 = \text{comfort}, 9096 = \text{on}, 9098 = \text{on}\}$$

is equivalent to

$$\{\text{Windows PAN} = \text{off}, \text{Humidity} = \text{humid}$$
$$\text{Temperature} = \text{cold}\} \rightarrow$$
$$\{\text{Output temperature PAN} = \text{comfort}, \text{Setup PAS} = \text{on}$$
$$\text{Setup PAN} = \text{on}\}.$$

This rule gives important information about the relationship between the windows being closed, humidity, a cold day, and the heating thermostats ON.

The following rule is obtained in summer, as we can see the heating equipment is not working and the windows are open:

$$\{\text{WindowsPAN} = \text{on}, \text{Windows PAS} = \text{on}\} \rightarrow$$
$$\{\text{Output temperature warm}, \text{Setup PAS} = \text{off}$$
$$\text{Setup PAN} = \text{off}, \text{Temperature} = \text{comfort}\}.$$

In addition, we have studied the number of coincident association rules according to the seasons. In Fig. 22, we can see the

percentage of coinciding rules for every pair of seasons, which are usually those generated by the building system independently of the temperature and weather variables (e.g., windows, lights, security systems, etc.) and those that are different, usually involving temperature and meteorological features.

The following rule occurs equally in summer and winter. It shows us the relationship between the lights on, the temperature of the PAN zone in comfort and the high occupation of the building.

$$\{\text{Occupacy} : \text{high}$$
$$\text{Status} - \text{Lighting} : \text{PAS} = \text{on}\} \rightarrow$$
$$\{\text{Output Status} - \text{Lighting} : \text{PAN} = \text{on}$$
$$\text{TemperaturePAN} = \text{comfort}\}$$

V. Future Challenges

For the data preprocessing, we have incorporated two different ways of using the information provided by the user and an automatic method using the data distribution. The latter could be used to suggest this information to the end users. This could be a precursor of a decision-making system. Different proposals for this type of systems can be found in [36]. In [37], decision support systems are used to improve the operation of building elements or equipment maintenance. Furthermore, some of them use rules obtained from the behavior of the users [38] to improve the building functioning. Therefore, using the results provided by our proposal, a decision system could be implemented incorporating the real functioning of the building.

Additionally, different methods have been observed in the literature for the management of uncertainty such as the use of polyhedral uncertainty [39]. One of the applications of these techniques is to use robustification of multivariate adaptive regression spline under polyhedral uncertainty to predict electricity consumption [40], gas consumption [41], or even finances [39]. In future works, it can be studied how to combine these types of models with the presented approach, taking benefit of both proposals.

VI. Conclusion

The discovery and exploitation of information collected from buildings has attracted attention in the last decade due to its economic and environmental impact. Big Data offers a suitable framework for the efficient implementation of analysis techniques capable of handling large amounts of data, especially those produced in building management systems. In addition, the use of fuzzy logic can improve the interpretability of collected sensor data, offering improved results and interpretation to end users.

In this article, a data mining methodology was implemented using the Big Data framework and applied to different data sets collected from an office building in Romania. In particular, we applied a fuzzification algorithm to improve the application of data mining techniques such as association rules. The whole system was deployed using the Spark platform to enable the analysis of such an amount of data generated by the sensors in the building. This technique allowed the exploitation of different kinds of data collected in the diverse pilot areas of the building. The proposed solution was applied to the static data collected from the building, obtaining different relationships that show the energy behavior of the building and make evident some patterns that can be used to improve the energy efficiency of the building.

However, the barrier that arises to automatically analyze continuously generated data needs to be dealt with. This leads us to propose a future improvement of the proposed system for handling such a continuous flow and to process it in real-time conveniently. To do so, there are recently developed utilities within the spark framework that enables the processing stream data called spark streaming [42], [43]. This extension will enable live data streams to be processed by dividing them into batches, which can be then processed by the spark mining algorithms.

Another future improvement concerns the display of results, which should be more informative and complete for end users. In fact, there are some applications available [44] that can be conveniently adapted to illustrate the discovered patterns.

Acknowledgment

The authors would like to thank the owners of the Bucharest Office building in Romania.

References

[1] M. S. Kiran, E. Özceylan, M. Gündüz, and T. Paksoy, "Swarm intelligence approaches to estimate electricity energy demand in Turkey," *Knowl. Based Syst.*, vol. 36, pp. 93–103, 2012.

[2] G. Nalcaci, A. Özmen, and G. Weber, "Long-term load forecasting: Models based on Mars, ANN and LR methods," *CEJOR*, vol. 27, no. 4, pp. 1033–1049, 2019.

[3] F. Yerlikaya-Özkurt, A. Askan, and G. Weber, "A hybrid computational method based on convex optimization for outlier problems: Application to earthquake ground motion prediction," *Informatica, Lith. Acad. Sci.*, vol. 27, no. 4, pp. 893–910, 2016.

[4] S. Midya and S. K. Roy, "Analysis of interval programming in different environments and its application to fixed-charge transportation problem," *Discrete Math., Algorithms. Appl.*, vol. 9, no. 3, 2017, Art. no. 1750040.

[5] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia, *Learning Spark, Lightning-Fast Big Data Analysis*. Newton, MA, USA: O'Reilly Media, Inc., 2015.

[6] Z. Yu, B. C. Fung, and F. Haghighat, "Extracting knowledge from building-related data. a data mining framework," *Building Simul.*, vol. 6, no. 2, pp. 207–222, 2013.

[7] Z. J. Yu, F. Haghighat, and B. C. Fung, "Advances and challenges in building engineering and data mining applications for energy-efficient communities," *Sustain. Cities Soc.*, vol. 25, pp. 33–38, 2016.

[8] M. Molina-Solana, M. Ros, M. D. Ruiz, J. Gómez-Romero, and M. Martin-Bautista, "Data science for building energy management: A review," *Renew Sustain. Energy Rev.*, vol. 70, pp. 598–609, 2017.

[9] C. Fan and F. Xiao, "Mining gradual patterns in big building operational data for building energy efficiency enhancement," *Energy Procedia*, vol. 143, pp. 119–124, 2017.

[10] T. Ahmad, H. Chen, Y. Guo, and J. Wang, "A comprehensive overview on the data driven and large scale based approaches for forecasting of building energy demand: A review," *Energy Buildings*, vol. 165, pp. 301–320, 2018.

[11] R. U. Islam, M. S. Hossain, and K. Andersson, "A novel anomaly detection algorithm for sensor data under uncertainty," *Soft Comput.*, vol. 22, no. 5, pp. 1623–1639, 2018.

[12] R. Agrawal, T. Imielinski, and A. Swami, "Mining associations between sets of items in large databases," *ACM-SIGMOD Int. Conf. Data*, vol. 22, pp. 207–216, 1993.

[13] P. Hájek, "The question of a general concept of the GUHA method," *Kybernetika*, vol. 4, pp. 505–515, 1968.

[14] E. Hüllermeier and Y. Yi, "In defense of fuzzy association analysis," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 37, no. 4, pp. 1039–1043, Aug. 2007.
[15] J. Calero, G. Delgado, M. Sánchez-Marañón, D. Sánchez, M. A. V. Miranda, and J. Serrano, "An experience in management of imprecise soil databases by means of fuzzy association rules and fuzzy approximate dependencies," in *Proc. 6th Int. Conf. Enterprise Inf. Syst.*, Apr. 2004, pp. 138–146.
[16] F. Berzal, M. Delgado, D. Sánchez, and M. Vila, "Measuring accuracy and interest of association rules: A new framework," *Intell. Data Anal.*, vol. 6, no. 3, pp. 221–235, 2002.
[17] M. Delgado, N. Marín, D. Sánchez, and M. Vila, "Fuzzy association rules: General model and applications," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 2, pp. 214–225, Apr. 2003.
[18] N. Marín, M. Ruiz, and D. Sánchez, "Fuzzy frameworks for mining data associations: Fuzzy association rules and beyond," *Wiley Interdisciplinary Rev.: Data Mining Knowl. Discov.*, vol. 6, pp. 50–69, Jan. 2016.
[19] M. Delgado, M. Ruiz, D. Sánchez, and J. Serrano, "A formal model for mining fuzzy rules using the RL representation theory," *Inf. Sci.*, vol. 181, no. 23, pp. 5194–5213, 2011.
[20] M. D. Ruiz, D. Sánchez, M. Delgado, and M. J. Martin-Bautista, "Discovering fuzzy exception and anomalous rules," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 4, pp. 930–944, Aug. 2016.
[21] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.
[22] L. Liu, *Performance Comparison by Running Benchmarks on Hadoop, Spark and Hamr*. Ph.D. dissertation, Univ. Delaware, Newark, DE, USA, 2016.
[23] M. Zaharia *et al.*, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proc. 9th USENIX Conf. Netw. Syst. Design Implementation*, 2012, pp. 2–2.
[24] K. Banker, *MongoDB in Action*. Shelter Island, NY, USA: Manning Publications Co., 2011.
[25] C. Győrödi, R. Győrödi, G. Pecherle, and A. Olah, "A comparative study: MongoDB vs. MySQl," in *Proc. 2015 13th Int. Conf. Eng. Modern Electric Syst.* , 2015, pp. 1–6.
[26] E. J. Knibbe, "Building management system," U.S. Patent 5,565,855, Oct. 15, 1996.
[27] S. Boschi and G. Santomaggio, *RabbitMQ Cookbook*. Birmingham, U.K.: Packt Publishing, 2013.
[28] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[29] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, 2004.
[30] C. Fernandez-Bassso, M. D. Ruiz, and M. J. Martin-Bautista, "Fuzzy association rules mining using spark," in *Proc.Int. Conf. Inf. Process. Manage. Uncertainty Knowl. Based Syst.*, 2018, pp. 15–25.
[31] C. Fernandez-Basso, M. Ruiz, and M. Martin-Bautista, "Spark solutions for discovering fuzzy association rules in Big Data," *Appl. Soft Compt.*, submitted for publication.
[32] V. P. Kumar and A. Gupta, "Analyzing scalability of parallel algorithms and architectures," *J. Parallel Distrib. Comput.*, vol. 22, no. 3, pp. 379–391, 1994.
[33] A. Y. Grama, A. Gupta, and V. Kumar, "Isoefficiency: Measuring the scalability of parallel algorithms and architectures," *IEEE Parallel Distrib. Technol., Syst. Appl.*, vol. 1, no. 3, pp. 12–21, 1993.
[34] C. Barba-González, J. García-Nieto, A. Benítez-Hidalgo, A. J. Nebro, and J. F. Aldana-Montes, "Scalable inference of gene regulatory networks with the spark distributed computing platform," in *Intelligent Distributed Computing XII*, J. Del Ser, E. Osaba, M. N. Bilbao, J. J. Sanchez-Medina, M. Vecchio, and X.-S. Yang, eds., Cham, Switzerland, Springer Int. Publishing, pp. 61–70, 2018.
[35] F. J. Baldán and J. M. Benítez, "Distributed fastshapelet transform: A big data time series classification algorithm," *Inf. Sci.*, vol. 496, pp. 451–463, 2018.
[36] F. Frombo, R. Minciardi, M. Robba, and R. Sacile, "A decision support system for planning biomass-based energy production," *Energy*, vol. 34, no. 3, pp. 362–369, 2009.
[37] A. Mattiussi, M. Rosano, and P. Simeoni, "A decision support system for sustainable energy supply combining multi-objective and multi-attribute analysis: An Australian case study," *Decis. Support Syst.*, vol. 57, pp. 150–159, 2014.
[38] Y.-K. Juan, P. Gao, and J. Wang, "A hybrid decision support system for sustainable office building renovation and energy performance improvement," *Energy Buildings*, vol. 42, no. 3, pp. 290–297, 2010.
[39] A. Özmen and G. W. Weber, "RMARS: Robustification of multivariate adaptive regression spline under polyhedral uncertainty," *J. Comput. Appl. Math.*, vol. 259, pp. 914–924, 2014.
[40] M. H. Yıldırım, A. Özmen, Ö. T. Bayrak, and G. W. Weber, "Electricity price modelling for Turkey," in *Operations Research Proceedings*, New York, NY, USA: Springer, pp. 39–44, 2012.
[41] G.-W. Weber, A. Zmen, and Y. Zinchenko, "MARS Under cross-polytope uncertainty: The case of prediction of natural gas consumption," in *y-BIS 2019: Recent Advances in Data Science and Business*, O. Kocadagli *et al.*, Eds. Istanbul, Turkey: Minar Sinan Fine Arts Univ., 2019, p. 25.
[42] C. Prakash, "Spark Streaming vs Flink vs Storm vs Kafka Streams vs Samza: Choose your stream processing framework," *Medium*, 2018. Accessed: Feb. 6, 2019. [Online]. Available: https://medium.com/@chandanbaranwal/spark-streaming-vs-flink-vs-storm-vs-kafka-streams-vs-samza-choose-your-stream-processing-91ea3f04675b
[43] C. Fernandez-Basso, A. J. Francisco-Agra, M. J. Martin-Bautista, and M. D. Ruiz, "Finding tendencies in streaming data using big data frequent itemset mining," *Knowl. Based Syst.*, vol. 163, pp. 666–674, 2019.
[44] C. Fernandez-Basso, M. D. Ruiz, M. Delgado, and M. J. Martin-Bautista, "A comparative analysis of tools for visualizing association rules: A proposal for visualising fuzzy association rules," in *Proc. Conf. Int. Fuzzy Syst. Assoc. Eur. Soc. Fuzzy Log. Technol.*, Beijing, China: Atlantis Press, 2019.

Carlos Fernandez-Basso received the Informatics Engineering degree in computer science and the M.Sc. degree in data science from the Universidad de Granada, Granada, Spain, in 2014 and 2015, respectively, where he is currently working toward the Ph.D. degree in computer science and energy efficiency.

He was a Lead Developer with the EU FP7 Project Energy IN TIME on the topics of building simulation and control, data analytics, and machine learning. He also collaborates with the Data Science Institute, Imperial College London, where he has carried out research studies, from 2016 to 2018.

M. Dolores Ruiz received the Mathematics degree and the European Ph.D. degree in computer science from the University of Granada, Granada, Spain, in 2005 and 2010, respectively.

She held a nonpermanent Teaching Positions with the Universities of Jaén, Granada and Cádiz. She is currently an Associate Professor with the Statistics and Operations Research Department, University of Granada and she belongs to the Approximate Reasoning and AI research group. She has organized several special sessions about data mining in international conferences and is part of the organization committee of the FQAS'2013 and SUM'2017 conferences. Her expertise involves knowledge extraction from databases involving uncertainty using association rules, exception rules, anomalous rules and gradual dependences, as well as, formal modeling for the representation and evaluation of association rules. Her current research interests include data mining, information retrieval, correlation statistical measures, sentence quantification and fuzzy sets theory.

Maria J. Martin-Bautista (Member, IEEE) received the degree and Ph D. degree in computer science from the University of Granada, Granada, Spain, in 1996 and 2000, respectively.

She has been a Full Professor with the Department of Computer Science and Artificial Intelligence, University of Granada, Spain, since 1997. She is a Member of the IDBIS (Intelligent Data Bases and Information Systems) research group. She has supervised several Ph.D. Thesis and authored or coauthored more than 100 papers in high impact international journals and conferences. She has participated in more than 20 R+D projects and has supervised several research technology transfers with companies. Her current research interests include Big Data analytics in data, text and web mining, intelligent information systems, knowledge representation and uncertainty. Dr. Martin-Bautista was a program committee member for several international conferences.