Cognitive Computation (2019) 11:418–433
https://doi.org/10.1007/s12559-019-09630-6

Feature Selection and Evolutionary Rule Learning for Big Data in Smart Building Energy Management

Pablo Rodriguez-Mier[1] · **Manuel Mucientes**[1] · **Alberto Bugarín**[1]

Received: 11 April 2018 / Accepted: 21 February 2019 / Published online: 9 March 2019

Abstract

Since buildings are one of the largest sources of energy consumption in most cities of the world, energy management is one of the major concerns in their design. To ameliorate this problem, buildings are becoming smarter by the incorporation of intelligent supervision and control systems. Data captured by the sensors can be interpreted and processed by rule-based computation methods of biological inspiration (such as genetic fuzzy systems, GFS) for predicting the future behavior of the building in a knowledge-based interpretable human-like manner. GFS are computational models inspired in human cognition which use evolutionary computation (inspired in the natural evolution) to automatically learn fuzzy rules which contain explicit imprecise knowledge about a system or process. This knowledge, represented using fuzzy rules that involve fuzzy linguistic variables and values, is used to perform approximate reasoning on the input values for obtaining inferred values for the output variables. In energy management of buildings, these rules allow a smart control of the system actuators to reduce the building average energy consumption. However, the large amount of data produced on a per second basis complicates the generation of accurate and interpretable models by means of traditional methods. In this paper, we present an evolutionary computation-based approach, namely a genetic fuzzy system, to build scalable and interpretable knowledge bases for predicting energy consumption in smart buildings. For accomplishing this task, we propose a cognitive computation system for multi-step prediction based on S-FRULER, a state-of-the-art scalable distributed GFS, coupled with a feature subset selection method to automatically select the most relevant features for different time steps. S-FRULER is able to learn a fuzzy rule-based system made up of Takagi-Sugeno-Kang (TSK) rules that are able to predict the output values using both linguistic imprecise knowledge (represented by fuzzy sets) and fuzzy inference. Experiments with real data on two different problems related with the energy management revealed an average improvement of 6% on accuracy with respect to S-FRULER without feature selection, and with knowledge bases with a lower number of variables.

Keywords Regression methods for Big Data · Feature selection · Fuzzy TSK rules

Introduction

Energy efficiency is becoming a major worldwide concern. It is estimated that buildings account for 40% of the total energy use in the EU [29], with a global consumption that has been facing a steadily increasing year after year. Concretely, it is estimated that almost half of the consumption is entirely used for heating and cooling. For this reason, improving the energy efficiency in buildings and especially the performance and management of the heating systems is key for environmental sustainability.

Advanced energy management in buildings can be achieved through a smarter control of their heating systems. This requires the use of sensors and actuators that can act accordingly to turn on or off the systems or to regulate the amount of heat needed at each moment. For example, heating usage can be optimized by taking into account multiple factors that affect the consumption, such as the thermal dynamics of the building, current and future weather conditions, in order to decide in a smarter way how to regulate the heating. Buildings that incorporate such level of automation through the use of supervision and control devices are called smart buildings. Cognitive systems can play an important role in smart buildings, not only from the performance point of view but, also, from

✉ Pablo Rodriguez-Mier
pablo.rodriguez.mier@usc.es

[1] Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS), University of Santiago de Compostela, Santiago de Compostela, Spain

the interpretability perspective, as they can assist human operators in their decision-making process. Interpretability is a key issue here, since the linguistic representation of variables included in the rules, together with its simplicity, usually allow human users a better understanding of the processes, providing them with valuable knowledge about the system behavior. Smart building technologies enable greater opportunities for the development of more sophisticated methods for improving energy management. Concretely, one of the current challenges is how to manage the information produced by the building to improve the overall energy consumption. This problem requires (1) the use of automatic methods for extracting and interpreting relevant knowledge from the big amounts of data collected by multiple sensors over time and (2) the generation of predictions of the future state of the system to design better energy policies [6, 11, 40, 47]. These challenges can be tackled from the perspective of cognitive computation by generating accurate and interpretable models that learn from data, in a way that experts can understand the interaction and the impact of the variables in the model.

Current methods for energy prediction [14, 46] are mostly based on physical model simulations [23, 41] and cognitive computation methods that learn black-box models from data [11, 22, 26, 33, 39, 42]. Physical models describe the building behavior by solving theoretical equations that describe to a certain extent the different dynamics and interactions between the variables. Although these methods are very powerful to simulate the different dynamics of a building, especially when there is no real data available, in general they are (1) very time-consuming since they require many simulation hours, which prevents their application for short-term forecasting; and (2) complex to formulate, since it is very difficult to produce a detailed model of a complex building, especially when there are many unknown factors that can affect the thermal dynamics. A cognitive computation approach can overcome some of these limitations by developing methods that are able to learn the behavior from real data. However, current techniques, which are mostly black-box models [4, 5, 10, 20], are hard to interpret and, thus, the interaction of the different variables of the system are hard to understand and explain to experts.

Fuzzy rule-based systems are one of the most well-known cognitive computing techniques, due to its linguistic-based orientation. They have been extensively used with great success in many fields and, particularly in systems supervision and control [28] since they are able to provide both accurate and interpretable models that in many cases can be directly understood by human users. Biologically inspired approaches such as genetic fuzzy systems (GFS) mimic the process of natural evolution to automatically build fuzzy rule bases from data. Although GFS have proved their validity in many fields of applications, there are some challenges associated, such as the impact of the problem size, which has a huge influence in the performance of these algorithms [9, 17]. As the number of variables increases, the computational cost of generating accurate rules for the data grows exponentially.

One way to cope with scalability issues from a Big Data perspective [13, 19, 30, 44] is to adopt the distributed computing paradigm for scaling GFS [31]. However, there is a lack of approaches—with only a few exceptions [12]—that use Big Data frameworks, such as Spark [45] or Hadoop [43] to deal with the scalability issues for regression problems. Concretely, the use of Spark is closely related to the success of Hadoop, which enables the processing of vast amounts of data in parallel on large clusters, usually implemented using the Hadoop Distributed File System (HDFS). Spark adds to the Hadoop ecosystem the capability to use advanced data-flow computations with an improvement of in-memory computing and high-level functions that facilitate to build parallel applications.

It was not until recently that the use of GFS for solving large-scale regression problems has started to attract attention in the field [2, 3, 25, 35, 36]. However, the size of the training data used in these works is not large enough to be considered Big Data. Among the different approaches, FRULER [36] obtains 1-order Takagi-Sugeno-Kang (TSK-1) fuzzy rule bases with high accuracy and the lowest number of rules. Although the runtime of this approach is acceptable for the most simple datasets, it does not scale properly when solving large-scale problems and may not converge to a good solution in reasonable time. These problems motivated the development S-FRULER [37], the distributed version of FRULER.

In this work, we address the problem of generating knowledge bases for the prediction of energy consumption in smart buildings in a big data setting. Our proposal is based on cognitive computing techniques to generate both accurate and interpretable knowledge bases. We extend our previous research using evolutionary computation methods [38] to improve the generation of the models by coupling the S-FRULER GFS with a feature selection mechanism. We observed that when in combination, S-FRULER generates consistently more accurate knowledge bases for the two prediction problems related with the energy management of the building considered in this work. These models have been used in the EU LIFE-OPERE project [21] to predict the behavior of the building under different conditions in order to find new control strategies that lead to even further energy savings.

FRULER: Fuzzy RUle Learning Through Evolution for Regression

FRULER (Fuzzy RUle Learning through Evolution for Regression) [36] is a novel GFS that obtains accurate and simple linguistic order-1 TSK fuzzy rule base models for regression problems. FRULER (Fig. 1) is composed of a new instance selection method for regression, a novel multi-granularity fuzzy discretization of the input variables, and an evolutionary algorithm that uses a fast and scalable method with Elastic Net regularization to generate accurate and simple TSK-1 rules.

Instance Selection

The objective of the instance selection module is to reduce the variance of the models, focusing the generated rules on the representative examples. The instance selection method for regression is an improvement of the CCISR (Class Conditional Instance Selection for Regression) algorithm [34], which is an adaptation for regression of the instance selection method for classification CCIS (Class Conditional Instance Selection) [24].

Multi-granularity Fuzzy Discretization

In a multi-granularity proposal, each granularity has a different fuzzy partition. The generation of the fuzzy linguistic labels can be divided into two stages. First, the variable must be discretized to obtain a set of split points C^g for each granularity g. Then, given the split points, the fuzzy labels can be defined for each granularity. In regression problems (TSK-1 in our case), the discretization process must search for the split point that minimizes the error when a linear model is applied to each of the resulting intervals.

Evolutionary Algorithm

The evolutionary algorithm learns a linguistic TSK model. The integration of the evolutionary algorithm with the preprocessing stage is as follows (Fig. 1):

- First, the instance selection process is executed over the training examples E_{tra} in order to obtain a subset of representative examples E_S.
- Then, the multi-granularity fuzzy discretization process obtains the fuzzy partitions for each input variable.
- Finally, the evolutionary algorithm searches for the best data base configuration using the obtained fuzzy partitions, generates the entire linguistic TSK rule base using E_S and evaluates the different rule bases using E_{tra}.

The chromosome is codified with a double coding scheme ($C = C_1 + C_2$). C_1 represents the granularity of each input variable. C_2 represents the lateral displacements of the split points of the input variables fuzzy partitions.

FRULER uses the Wang & Mendel algorithm to create the antecedent part of the rule base for each individual. The consequent part of the rules is learned using the Elastic Net method [48] in order to obtain the coefficients of the degree 1 polynomial for each rule. Elastic Net combines the ℓ_1 (lasso regularization) and ℓ_2 (ridge regularization) penalties of the lasso and ridge methods, minimizing the following equation:

$$\hat{\beta} = \arg\min_{\beta} ||Y - X \cdot \beta||^2 + \lambda \cdot \alpha \cdot \|\beta\|^2 + \lambda \cdot (1-\alpha) \cdot \|\beta\|_1 \quad (1)$$

where β is the coefficients vector, Y is the outputs vector, X is the inputs matrix, λ is the regularization parameter, and α represents the trade-off between ℓ_1 and ℓ_2 penalization. In order to solve the minimization problem of Elastic Net (1), we used stochastic gradient descent (SGD).

The rule base is generated using only those examples in E_s. In this manner, those examples that are not representative are not taken into account, the method avoids the generation of too specific rules, and reduces the time needed to create the rule base. The fitness function is as follows:

$$\text{fitness} = \text{MSE}(E_{\text{tra}}) = \frac{1}{2 \cdot |E|} \sum_{i=1}^{|E|} (F(x^i) - y^i)^2, \quad (2)$$

where E_{tra} is the full training dataset and $F(x^i)$ is the output obtained by the knowledge base for input x^i. Using all the examples for evaluation can be seen, in some way, as a validation process, as the rule base was constructed with a subset of them (E_S).

S-FRULER

S-FRULER [37] (Scalable Fuzzy Rule Learning through Evolution for Regression), is the distributed version of FRULER designed to improve the current scalability issues that hampers the use of FRULER with large-sized problems. To cope with these limitations, S-FRULER, instead of processing the entire dataset, divides the original problem into a set of smaller problems that are more tractable using a distributed approach (*map phase*). Each of these divisions is then independently solved in the *map phase* using the FRULER algorithm, as described in the previous section. Finally, the solutions obtained in each map are combined in the *aggregation phase* in order to obtain a final solution for the original problem.

The algorithm structure is shown in Fig. 2. The first step is the multi-granularity fuzzy discretization process

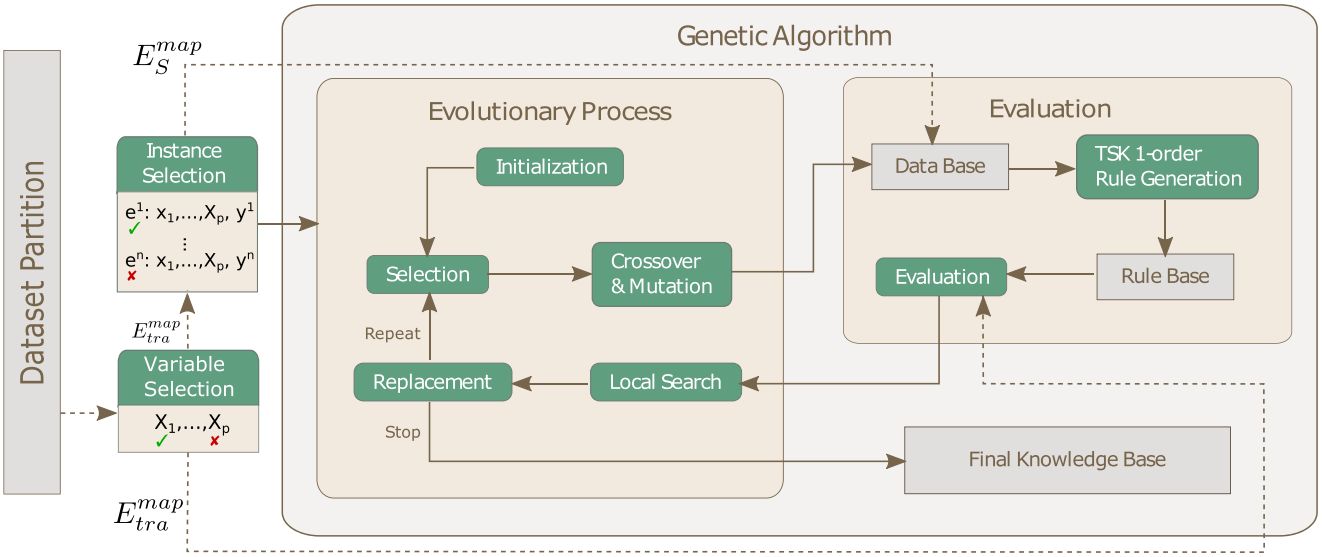

Fig. 1 FRULER architecture. Dashed lines indicate flow of datasets, dotted lines multi-granularity information, and solid lines represent process flow

of FRULER, and it is performed using the whole training dataset. Then, the training dataset is splitted into n partitions during the *map phase*. Those partitions generated during the *map phase* correspond with the tasks that are distributed as independent sets of processes to be processed in the worker nodes using *Apache Spark*. For each partition, only a subset of randomly selected variables is taken into account. The probability of selecting a particular input variable X_j in a dataset partition is as follows:

$$P(X_j \in X_s^i) = \frac{p_m}{p}, \tag{3}$$

where X_s^i is the selected subset of input variables in the dataset partition i, p_m is the subset size of selected input variables, and p is the total number of input variables. Thus, the probability that a particular input variable is not selected for all the dataset partitions is as follows:

$$P(X_j \notin X_s^i, \forall i = 1, \ldots, n) = \left(\frac{p - p_m}{p}\right)^{n_{\text{map}}}. \tag{4}$$

where n_{map} is the total number of dataset partitions.

Each partition is solved using FRULER, considering each partition as an independent problem, where only the instance selection and the genetic algorithm are executed. Finally, each independent solution for each sub-problem is combined in the *aggregation phase*, where the missing variables that were not selected in some of the partitions

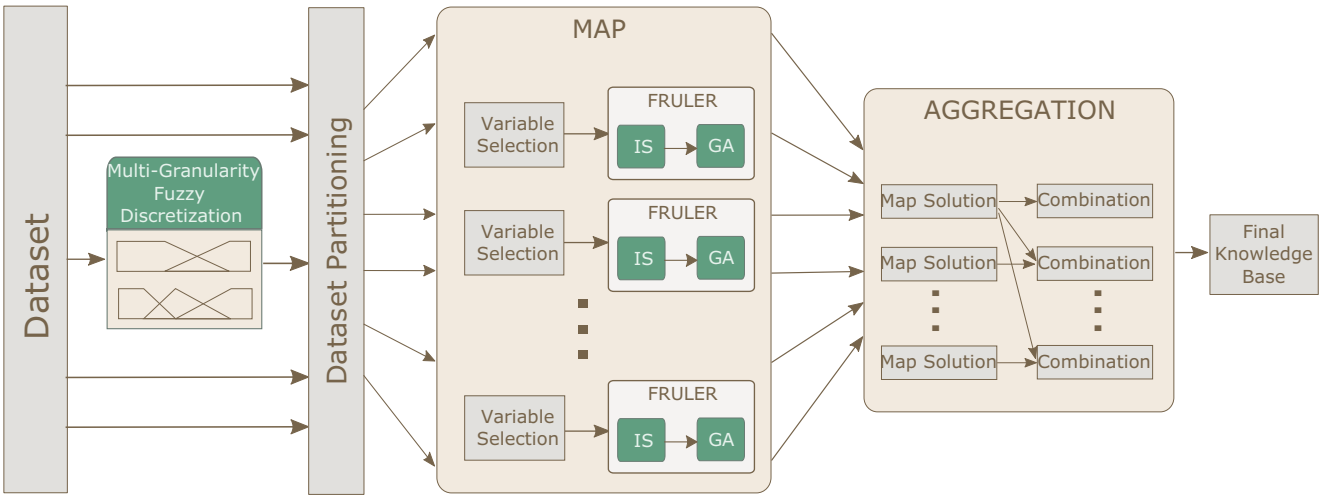

Fig. 2 S-FRULER architecture

are combined with the information of the other partitions to produce the final knowledge base.

Smart Building Energy Management

In this section, we describe in detail the problem of the energy management in smart buildings in the context of the LIFE-OPERE EU project [21]. This project aims to reduce the environmental impact derived from the high energy consumption of the University of Santiago de Compostela by implementing a series of actions that span from the installation of new control and sensor systems to the fine-tuning of the different programs and actuators. Concretely, one of this actions consists of the intelligent optimization of the thermal and electrical energy grids, following a data-driven approach for modeling the complex behavior of the system. This implies the generation of predictive models using the data collected from the sensors to simulate the behavior of the building but also for understanding how the different variables affect the power consumption. For this purpose, in this work, we focus our study in the Monte da Condesa, located in the University of Santiago de Compostela (USC), which was selected as the first pilot plant with the aim of validating this proposal for its use in other smart buildings with similar characteristics.

The complex of Monte da Condesa, located in the Campus Vida of the USC comprises the facilities of four different buildings: the Faculty of Optics, the Faculty of Physics, the University Residence of Monte da Condesa, and the Institute of Orthopaedics. Particularly, the university residence is one of the largest buildings in the complex, with a building area of 25,000 m^2 distributed in six floors, ground floor, and basement, and with an annual power consumption of around 5,700 MWh. Previous analysis reported that most of the energy produced is used for heating either the university residence or the domestic hot water. This highlights the importance of improving the heating system to reduce the total energy consumption.

The Monte da Condesa complex shares a centralized heating system that combines the use of two gas boilers with a power of 2615 kW and a cogenerator plant. The complex has also five thermally stratified hot water storage tanks that are used to store the hot water produced by the boiler and to recover also the residual heat produced by the cogenerator plant. The hot water in the tanks is distributed for heating across the different buildings, and it is regulated by different pumps in each building that mix cold and hot water at different proportions. When there is no enough hot water in the tanks to fulfill the demand, the gas boilers start heating water, which is costly. In this setting, there are two main strategies to reduce the power consumption: (1) optimizing the schedule of the cogenerator to turn it on earlier or later depending on the predicted demand and (2) improving the regulation of the hot water pumps to reduce the heating without affecting the comfort.

Thermally Stratified Hot Water Storage Tanks

The complex is equipped with five thermally stratified hot water storage tanks that are used to store the hot water produced by the boiler and to recover also the residual heat produced by the cogenerator plant. This is represented in Fig. 3. The water in the top of the tank is hotter than the water in the bottom, and it is used for different purposes. The water of the bottom is mainly used to refrigerate the cogenerator (using a heat exchanger), whereas the water in the top is used to heat either the buildings or the domestic hot water (DHW). When this water is consumed, the tanks are refilled with cold water pumped at the bottom, and when the temperature at the top decreases below a certain established threshold, the gas boilers are pumped with cold water from the bottom to be heated and pumped back in the upper part of the tanks. This process usually occurs in moments of high demand, for example, at mornings when the students are using the showers or at night when the temperature inside of the building decreases and needs to be heated.

The water in the tanks can be heated as well by the cogenerator plant. When the cogenerator plant is producing electricity, part of the heat is transferred to the top of the tank. Thus, deciding when to switch on or off the cogenerator has a direct impact on the temperature of the water stored in the tanks. On one hand, if the cogenerator is switched on when the tanks are at its maximum capacity (with no cold water at the bottom), the waste heat

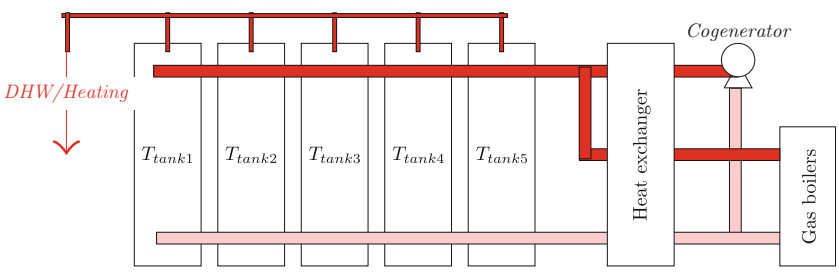

Fig. 3 Representation of the five thermally stratified hot water storage tanks attached to the cogeneration plant and the gas boilers

produced cannot be recovered, and the cogenerator is at risk of overheating since there is no enough cold water to refrigerate it. On the other hand, if the cogenerator is switched off before a period of high demand, the gas boilers need to be used to compensate the lack of enough hot water. Since the schedule has to be decided in advance (due to operational issues), a good schedule of the cogenerator is critical for the optimization of the energy consumption. Currently, this schedule is decided by the operator in charge, who designs it 2 h in advance, based on his own experience. In order to improve this, in the next Section ("Scalable multi-step prediction using S-FRULER with Feature Selection"), we present a model that is able to predict the upper temperature of the tanks as a function of the cogenerator status (and other variables) for different time windows.

Indoor Temperatures and Heating

In order to heat the different buildings in the complex, the water is independently pumped from the top of the water storage tanks to each facility. For the case of the university residence, the water is regulated by two independent pumps and valves, one for the first two floors, and the other one for the remaining floors, as represented Fig. 4. The mixture of the pumps is regulated by a pre-programmed heating curve using the temperature outside the building as a reference. When the temperature is cold, the pump mixes more hot water to heat the building, and when it is hot, it mixes more cold water. This is very inefficient, because it does not take into account the different factors that affect the temperature inside the building, including the current temperature. In Section "Scalable multi-step prediction using S-FRULER with Feature Selection," we present a model that is able to predict the temperature inside the building as a function of the pump status and other weather variables.

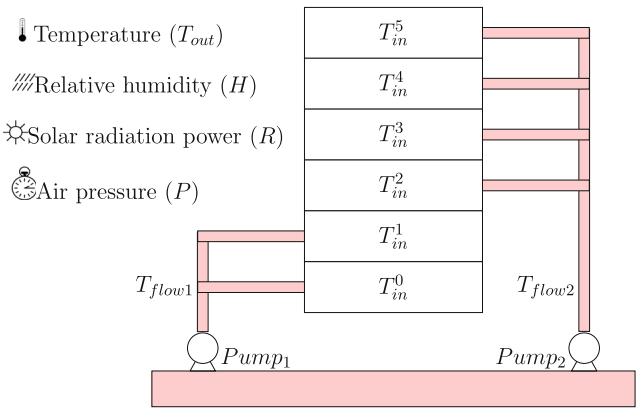

Fig. 4 Schema of the Monte da Condesa residential college with the related variables

Sensors and Data

In order to measure and control the different parts of the system, each building in the complex is equipped with a set of sensors and actuators connected to a PLC (programmable logic controller), one per smart building. These PLCs are in turn connected to a central control system using the BACnet protocol. All these sensors and actuators are supervised by a SCADA system that allows experts to visualize and control the heating production and the radiators, among others. In total, more than 450 sensors and actuators related to the primary heating circuits and power generation system are connected and supervised by the SCADA platform. All this data is stored in a distributed database. This database stores two types of signals:

- Synchronous signals: These signals are recorded at a constant rate of 10 s. This is the case for the temperature sensors installed in the water tanks and across different parts of the system, and also for the mixture of hot and cold water for each pump.
- Asynchronous signals: These signals are recorded when a change of value above a prefixed threshold is detected. Examples of these signals are the indoor temperatures, error signals, and binary status of many low level parts of the system, like pumps, valves, engine status, and so on.

In addition to these SCADA variables, we also collect the humidity (H), solar radiation power (R), and pressure (P) from *Santiago-EOAS*, a weather station of the Galician Meteorological Agency (Meteogalicia) situated approximately 100 m from the building [27]. These variables give relevant information about weather conditions that may directly affect the indoor temperatures. We also collect the temperature ($\bar{T}_o$), relative humidity ($\bar{H}$), and pressure ($\bar{P}$) predictions from *MeteoSIX*, a numerical weather prediction service by Meteogalicia that provides hourly predictions for the next 4 days.

All these variables are also stored in a distributed database for its posterior analysis. For the generation of the predictive models, synchronous measures were downsampled to 1-h bins and asynchronous measures were converted into time series by applying linear interpolation and 1-h resampling.

Scalable Multi-step Prediction Using S-FRULER with Feature Selection

In order to optimize the power consumption in the building, we need to build predictive models to simulate the future state of the building at different time horizons, given its current state, the local weather information, the weather

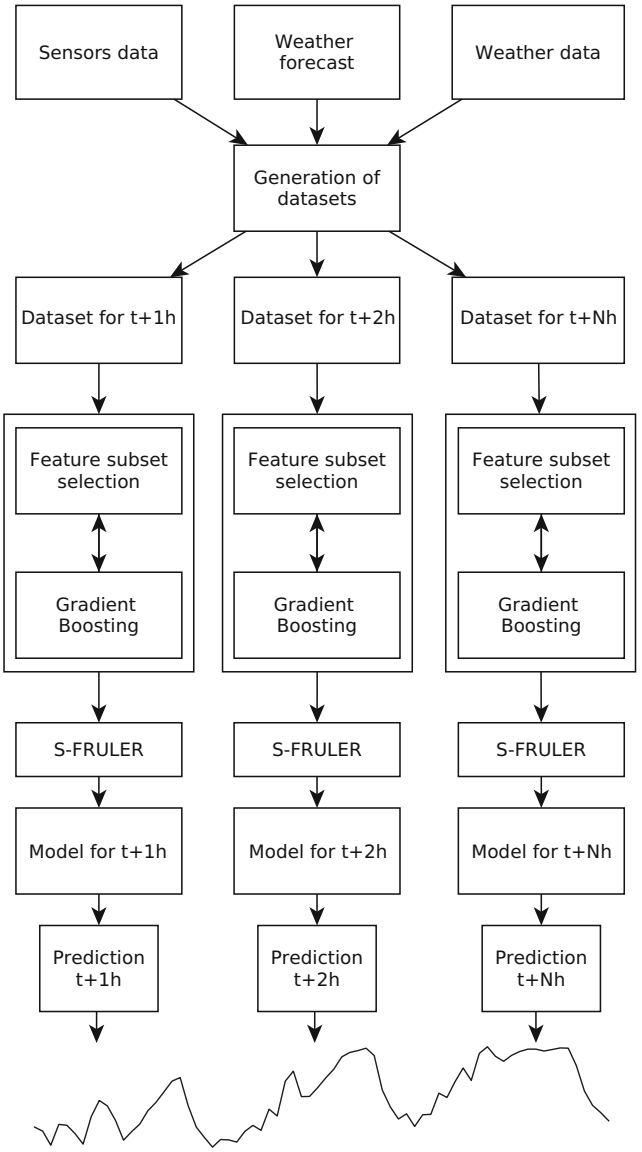

Fig. 5 Schema of the proposed approach for the scalable construction of the predictive models for multi-step prediction

predictions and the schedule for the systems to be controlled (the cogenerator and the pumps). The models for long-term prediction we need to generate for this problem are of two types: one to predict the evolution of the temperature in the upper part of the tanks and other to predict the evolution of the temperatures inside the building. The purpose of these models is twofold: (1) to make accurate predictions of the future states of the building, in order to simulate and optimize the management of the building; (2) to interpret the effect and interaction of the different variables in order to help the experts understand the dynamics of the building and to prepare future actions to improve the efficiency.

The implementation of the system also requires updating the models over time, using the new acquired data. This means that, every year, the training set increases by 8,760 h of new data, which makes it necessary to use scalable machine learning techniques to prevent the whole system suffering from scalability issues over time as the training set grows. Given these requirements, we adopted a direct prediction forecasting strategy [7] using S-FRULER to generate fuzzy knowledge bases in an scalable fashion that are both interpretable and accurate. The direct strategy learns independently N models, one for each hour, and the forecasting is generated by joining the predictions of each model. This has many advantages: (1) since the models are independently learned, estimations of models are not propagated and thus is not prone to accumulation of errors; (2) the generation of the models for each time horizon can be distributed, which increases the level of parallelism; and (3) models are also easy to interpret for the experts as they can choose to focus on just one model, for example, to explore the effect of the variables in the prediction of the temperatures for the next 24 h, ignoring the rest of time lags.

Figure 5 shows the long-term prediction framework proposed for the generation of the tank temperature and indoor temperature models. Starting from the available data from the sensors, the weather station and the MeteoSIX service, we identified a subset of variables of interest in successive talks with the experts in the energy domain involved in the project. With this variables, we generate N different datasets, one for each hour we want to predict. Then, we reduce the set of initial features using a wrapper-based feature subset selection method by combining recursive feature elimination (RFE) with gradient boosting trees (GBTs). After that, we build models for each hour using S-FRULER to generate interpretable and accurate fuzzy knowledge bases. Finally, a multi-step prediction is generated by using the first model to predict t+1h, the second model to predict t+2h, and so on.

Definition of the Variables

After a discussion with the experts in the domain of energy involved in the project, we ended up with a list of the relevant features for the prediction of the indoor temperatures and tank temperatures. Table 1 summarizes the list of selected variables. Among all the variables selected, there are two important variables, Cog_M^N and Pump_M^N, that are related with the explicit control of the system.

Variable Cog_M^N refers to the total number of hours the cogenerator is working between $[t+M, t+N]$. This variable is known in advance and provided by the operator in charge of the schedule of the cogenerator plant. This allows us to use the model to simulate the temperature in the upper part of the tanks for different schedules of the cogenerator with the aim to find the best schedule that guarantees that the temperature of the tanks is not too hot or too cold.

Table 1 List of features

Name (abbrev.)	Description
Pi_temp (T^{Pi})	Indoor temperature in the ith floor.
Outdoor_temp (T°)	Outdoor temperature.
Upper_tank (T^{up})	Average temperature of the tanks (measured at the top)
Mid_tank (T^{mid})	Average temperature of the tanks (measure in the middle)
Lower_tank (T^{low})	Average temperature of the tanks (measure in the bottom)
Humidity (H)	Relative humidity (%).
Pressure (P)	Air pressure (hPa)
Radiation (R)	Global solar radiation power (W/m^2)
Outdoor_temp_pred ($\bar{T}^{\circ}$)	Predicted outdoor temperature
Humidity_pred ($\bar{H}$)	Predicted relative humidity (%)
Pressure_pred ($\bar{P}$)	Predicted air pressure (hPa)
Cogenerator_status_N_M (Cog_M^N)	Total number of hours the cogenerator was ON between $[t+M, t+N]$, $M < N$
z8_pump_N_M (Pump_M^N)	Total number of hours the heating pump was pumping hot water to the first and second floors of the building, between $[t+M, t+N]$, $M < N$
acs_consumption_N_M ($\bar{\text{DHW}}$)	Estimation of the previous $N-M$ h of hot water consumption, using data from the previous week as the estimation

In the same way, the variable Pump_M^N refers to the total number of hours the pump was pumping hot water to the building, which is equivalent to say that the heating of the building was working. Again, we can use these variables to simulate the indoor temperature for different working hours of the heating in order to find the best schedule that minimizes the total usage without affecting the thermal comfort.

Since all variables are sampled at 1-h interval, we decided to apply the same criteria for these two control variables. Concretely, we use the hourly information of the previous 24 h. The main drawback of this approach is that we need to use 24 features, one for each hour. This increases the complexity of the dataset but requires almost no domain knowledge. A better way would be to compact the representation by taking larger groups and summing up the total hours in each group. However, this requires domain knowledge and is not generalizable for other buildings.

Figure 6 shows the linear correlation analysis of the variables initially included for the tank problem (left) and indoor temperature (right) for the time horizon of 12h. Variables Cog_M^N and Pump_M^N are represented as *cogenerator_status_N_M* and *z8_pump_N_M* respectively. There are some interesting observations that can be derived from this graph. For example, the amount of hot water consumed in the previous 24 h is negatively correlated with the temperature of the water in the middle part and the lower part of the tank (left figure). This makes sense since the domestic water is heated with the water in the middle of the tank. There is also a strong correlation between the usage of the cogenerator and the tank temperature. Weather variables, like radiation, pressure, and temperature are also correlated with the temperature of the tanks and with the indoor temperatures. This confirms the importance of these variables for its inclusion in the initial set of features.

Feature Selection

After training each model for each hour, we perform a wrapper-based feature subset selection using GBTs [15]. GBT is a supervised learning algorithm that generates a predictive model in the form of an ensemble of weak prediction models using in this case regression trees. After training a GBT model, the importance of each feature can be calculated by averaging the importance of the features within each tree by the weight of each decision tree used as a weak learner. This can be used to produce a list of features ranked by their predictive power. Wrapper methods for feature selection use the machine learning algorithm of interest as a black-box function that produces a ranking of features given the training data. Then, a search algorithm is used to find a subset of features that perform well for a particular problem. Algorithm 1 shows the pseudo-code of the approach used to find a subset of features for the temperature prediction. Starting with the full set of features, the algorithm trains a GBT using the training data, and removes the worst feature according to its importance. This process is repeated after a total of *max_feats* is selected.

In order to select the number of *max_feats*, we gradually increase the number of features selected from 1 to the number of available features in each dataset, and we plotted the cross-validation error curves. We finally selected the number of features for each dataset by observing the curves and selecting the number of features from which the error no longer improves significantly. A more detailed explanation of this is offered in Section "Experimental Study."

Although wrapper methods are commonly used with the same machine learning technique selected for the final model, a combination of different techniques is also common in the literature and usually with better results [16]. Although S-FRULER can be also used as the base model

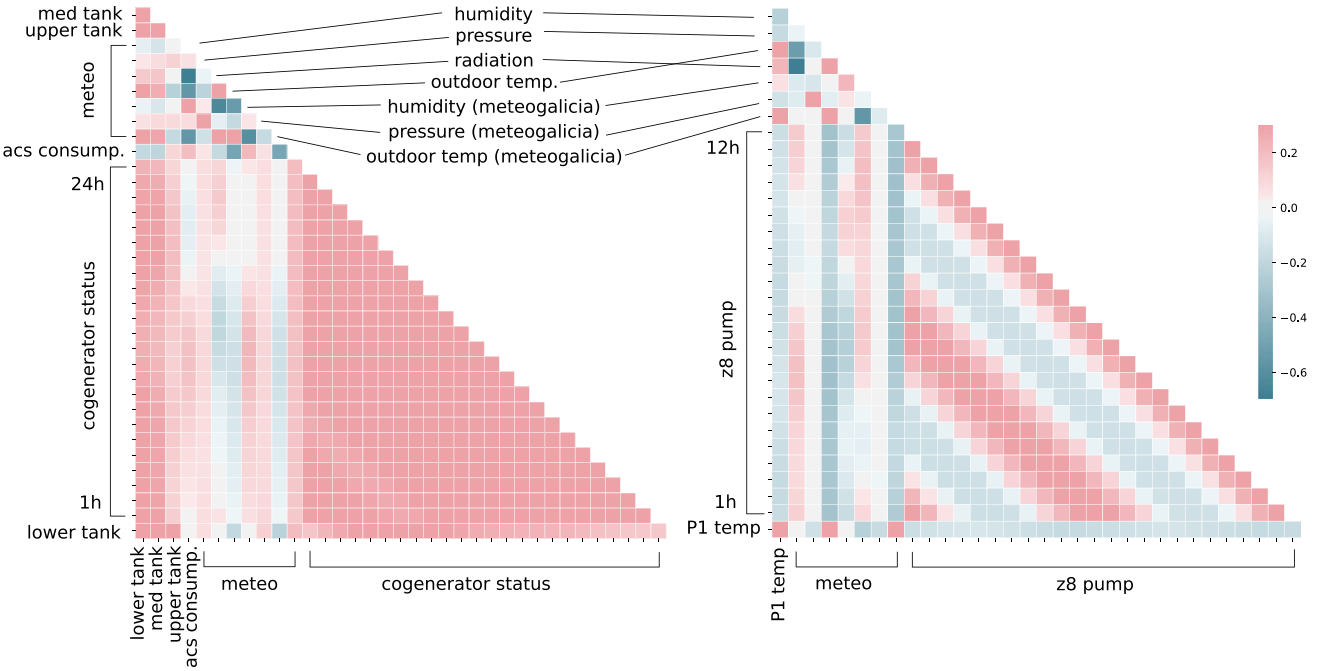

Fig. 6 Diagonal correlation matrix of the features selected for the upper temperature of the tanks (left image) and the indoor temperature of the first floor (right image), for a time horizon of 12 h, measured using the Pearson correlation. The target variable y in the left image corresponds with $y = T^{\text{up}}_{t+12}$, and the target variable on the image on the right corresponds with $y = T^{P1}_{t+12}$

for wrapper feature subset selection, a combination of GBT and S-FRULER is preferred since it is less computationally demanding.

Algorithm 1 Recursive feature elimination.

```
function RFE(GBT, DATASET, FEATS, MAX_FEATS)
    # Train the model with all features
    model ← train(GBT, dataset, Feats)
    # Rank features by importance
    Rank ← ranking(model, Feats)
    while size(Feats) > max_feats do
        # Remove the worst performing feature
        Feats ← Feats − {worst(Rank)}
        # Re-train the model with the new set
        model ← train(GBT, dataset, Feats)
        Rank ← ranking(model, Feats)
```

Model for Prediction of Indoor Temperatures

Given the set of features described before, the problem of one-step forecasting for the indoor temperatures of the ith floor is formulated as a regression problem:

$$\hat{T}^{\text{Pi}}_{t+k} = f_k(T^{\text{Pi}}_t, T^o_t, \bar{T}^o_{t+k}, H_t, \bar{H}_{t+k}, P_t, \bar{P}_{t+k}, \text{D}\bar{\text{H}}\text{W}, R_t, \text{Pump}^k_{k-1}, \text{Pump}^{k-1}_{k-2}, \ldots, \text{Pump}^{k-23}_{k-24})$$

Then, a RFE process is performed using GBTs as the estimator for each f_k, trained with the whole training set, to retain a different number of features. After that, we apply S-FRULER to learn a knowledge base for each f_k using the selected subset of features.

Model for the Water Storage Tanks

For the case of the average upper temperature of the water in the tanks, we define the base model as follows:

$$\hat{T}^{\text{up}}_{t+k} = f_k(T^{\text{up}}_t, T^{\text{mid}}_t, T^{\text{low}}_t, T^o_t, \bar{T}^o_{t+k}, H_t, \bar{H}_{t+k}, P_t, \bar{P}_{t+k}, \text{D}\bar{\text{H}}\text{W}, R_t, \text{Cog}^k_{k-1}, \text{Cog}^{k-1}_{k-2}, \ldots, \text{Cog}^{k-23}_{k-24})$$

Again, we repeat the same procedure as before to select a subset of features and, then, we use S-FRULER to learn the final models.

Experimental Study

In this section, we analyze the performance of our proposed approach with real data collected from the building. In total, we collected 8,760 h of training data from February 1, 2016 to January 31, 2017 for the indoor temperature models, and

6,161 h from May 19, 2016 to January 31, 2017 for the tank models. We generate models for a total of six time horizons $k \in \{1, 2, 4, 8, 12, 24\}$ (hours).

We also compare how well S-FRULER performs with feature selection in terms of accuracy against two black-box models, which produce accurate results with low interpretability, and two state-of-the-art fuzzy rule-based systems, which produce more interpretable models at the expenses of the accuracy. For the black-box models, we chose a multi-layer perceptron (MLP) as a supervised parametric model, and support vector machine for regression (SVR) as a non-parametric model. MLPs are feedforwad artificial neural networks which have the ability to learn very complex nonlinear relationships with good accuracy, but they do not produce interpretable models. For this comparison, we used a MLP with only one hidden layer, as in our preliminary tests we did not observe any benefit on adding more layers. The number of neurons in the hidden layer of the MLP was tuned using grid search with tenfold cross validation for each time horizon, searching for values between $[5, 2 \cdot \text{max\_features}]$ in five-step increments. The training was performed for a maximum of 20,000 epochs using the L-BFGS quasi-Newton optimization algorithm for backpropagation. SVR is an extension of the classical SVR problems. For training the SVR, a radial basis function kernel was used, and the C and γ parameters were adjusted using grid search with tenfold cross validation.

For the fuzzy rule-based methods, we chose GFS.LT.RS [1], a genetic lateral tuning and rule selection method for linguistic fuzzy systems, and FS.HGD [18], a TSK fuzzy rule generation method using heuristics and gradient descent. Both implementations are available in the FRBS library [32] for R programming language. The selection of those algorithms was done based on their performance on a set of preliminary tests. As the methods implemented in FRBS are computational demanding, instead of performing a grid search for the selection of the hyperparameters, we started with very low values for the number of iterations and the number of individuals in the population. Algorithms that did not make any progress after a few hours or methods that produced execution errors in the process were discarded. These hyperparameters were gradually increased to improve the accuracy of the methods until the training time exceeded 1 h. Also, methods that were not able to obtain models with reasonable accuracy in a small version of the dataset were also filtered out. After this process, only GFS.LT.RS and FS.HGD were selected. The following parameters were used for training:

- GFS.LT.RS: population size = 5, num. labels = 5, mutation = 20%, max. generations = 3, tuning mode = local, t-norm = minimum, s-norm = maximum, implication = minimum, defuzzifier = center of gravity
- FS.HGD: num. labels = 5, max. iterations = 10, step size = 0.1, alpha heuristic = 1, t-norm = minimum, s-norm = maximum, implication = minimum

Increasing further the number of individuals or iterations resulted in prohibitively computation times for a very modest decrease in errors. Those parameters remained fixed for all the tests. It should be noted that we did not perform any hyperparameter optimization for S-FRULER. Default parameters across all tests were used for S-FRULER as well.

Indoor Temperatures

Figure 7 shows the estimated error using a shuffled tenfold cross validation for different number of selected features and for different time horizons using RFE with GBTs. For training the trees, we used XGBoost [8], and we optimized the hyperparameters with grid search and cross validation. The hyperparameters selected were as follows:

- Learning rate: 0.05, shrinkage factor for the contribution of each tree
- Max depth: 6, maximum depth of the individual regression estimators
- Number of boosting stages: 600, number of weak learners used
- Column sample by tree: 0.8, sub-sample ratio of features when constructing each tree.

As expected, the easier problem was the prediction of the indoor temperature for the next hour ($k = 1$). In this case, the performance does not improve significantly after selecting the top 5 most performing features. The top 5 features selected to predict the indoor temperature for $k = 1$, sorted by decreasing weight importance, are T^{P1}, T°, H, P, and $\bar{T}^{\circ}$. The most important features are the current indoor temperature and the current outdoor temperature as

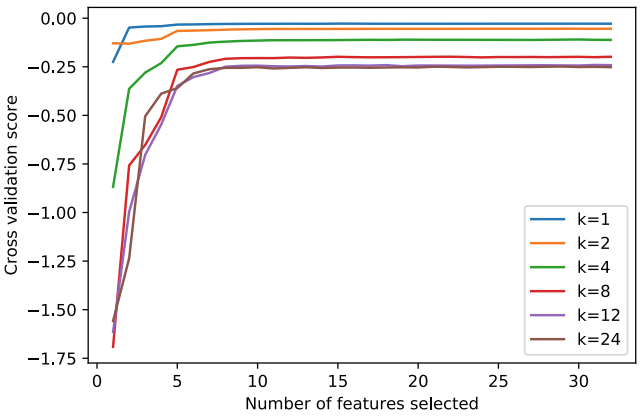

Fig. 7 Model score (negative MSE) estimated with tenfold cross validation for different number of features, selected with RFE, for the prediction of the indoor temperature for each time horizon k

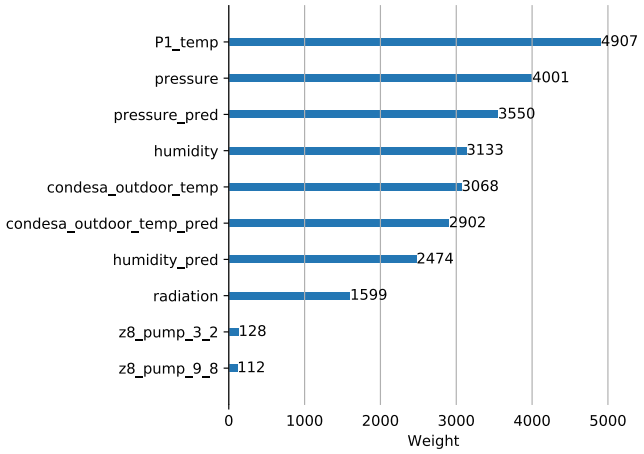

Fig. 8 Top 10 features selected by RFE for $k = 24$ h for the indoor temperature problem

expected. It should be highlighted that the current humidity and pressure are also relevant for the prediction of the indoor temperature.

As the time horizon increases, the number of features needed to improve the error also increases. For the time horizon $k = 24$ h, the estimated error stabilizes around ten features. The top ten features for $k = 24$ h selected by RFE are shown in Fig. 8.

Again, weather-related variables were among the top most informative features, including now also the radiation. Unfortunately, MeteoSIX does not offer a predicted radiation value, but from this analysis, we can expect that the predicted radiation at $t + 24$ h would be also an interesting feature to include in the model. Only two variables related with the status of the heating were included, corresponding with the status between $[t+2$ h, $t+3$ h$]$ and $[t + 8$ h, $t + 9$ h$]$. Although at first glance it may seem that the heating system does not affect the indoor temperature, a closer look to the correlation matrix (Fig. 6) reveals that the heating status is strongly correlated with the weather variables. This is reasonable since the heating system of the building is used only when needed, i.e., when the building is cold, and so there is a strong causal-effect relationship. Collecting more data could help to leverage the true influence of the heating independently of weather conditions. This also highlights the importance of updating the models with more building data collected in the coming years.

Using the features of Fig. 8, we learned different predictive models with S-FRULER, the SVR and the MLP. Figure 9 shows an example for the prediction of the indoor temperatures for the training period and for a hold-out dataset of unseen data (from February 1, 2017 to May 31, 2017, separated by the vertical red line), using a knowledge base generated by S-FRULER with feature selection for $k = 4$ h. We focus only on the first floor (P1), as the same analysis applies for the remaining floors. Figure 10 shows the average error (RMSE) using shuffled tenfold cross validation without feature selection and with feature selection including the top 10 features for all time horizons. We decided to use the top 10 features based on the observation that, after the first ten features, there is no substantial change in the error model.

A first look at the curves reveals that the performance of S-FRULER improves when coupled with the feature selection mechanism for all time horizons, with an average gain in accuracy of 6.8% when compared with S-FRULER without feature selection. Also, it can be seen that both the SVR and the MLP decrease in accuracy when using feature selection. We hypothesize that the proposed feature selection method is not effective for SVR and MLP but works fine for S-FRULER as it is based on trees, which

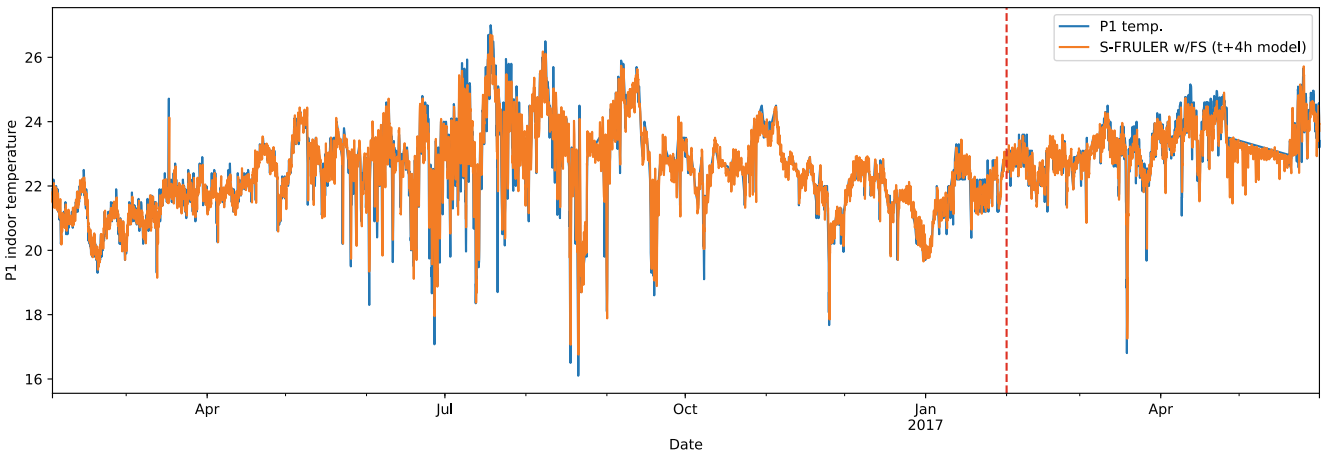

Fig. 9 P1 indoor temperatures vs. predicted temperatures, using one knowledge base generated by S-FRULER for the time horizon $k = 4$ h. The vertical red line separates the data used for cross validation (left) from the hold-out test data (right)

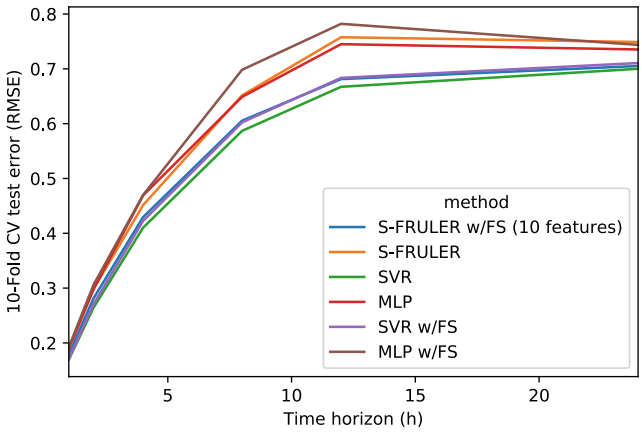

Fig. 10 Performance of the models (tenfold estimation of the test RMSE) for different time horizons using S-FRULER, MLP, and SVR for the indoor temperatures, with and without feature selection. Note that the S-FRULER w/FS curve (blue curve) is occluded by the SVR w/FS curve (purple curve) as they have a very similar performance

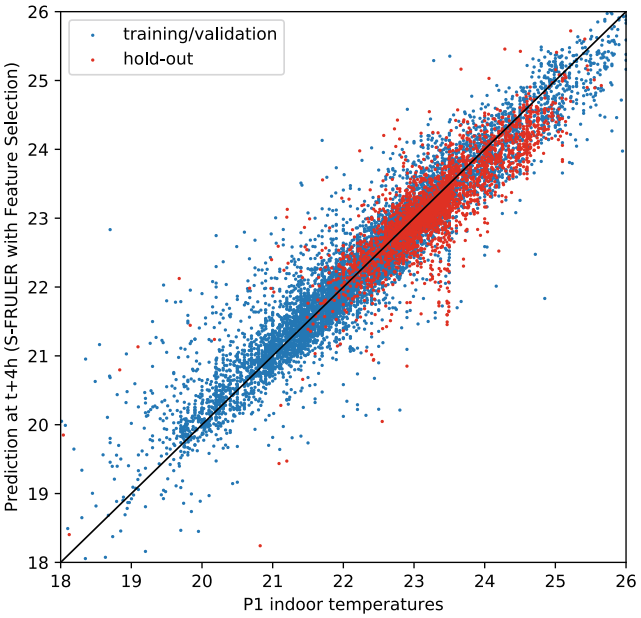

Fig. 11 Observations vs predictions for $k = 4$ h for one of the models generated with S-FRULER, for the training set and for a hold-out set with unseen data from February 1, 2017 to May 31, 2017

partition the input space in a way that is very similar to that of a fuzzy rule base. If we compare the best results achieved by the MLP and the SVR (i.e., without FS) against S-FRULER with FS, we observe that the error of the SVR is 3.7% better than S-FRULER with FS (an average of 0.015 °C, which is negligible), but at expenses of generating a non-interpretable model. On contrast, the MLP is 6.9% worse on average (an average of 0.034 °C). Table 2 shows the average RMSE and the average number of rules for each time horizon in detail. One interesting outcome is that S-FRULER achieves a consistent increase in accuracy with an average of 4.2 rules for the knowledge bases generated by S-FRULER and 8.5 for S-FRULER with feature selection.

As can be seen, the accuracy on both training and test is similar, with no signs of overfitting. This is better shown in Fig. 11. It should be noted that the prediction on the hold-out set is especially difficult, since there is no data for the same period of time on previous years in the training set. Thus, a smaller error in the predictions is expected in the following years, as more data is used to train the models.

Table 2 Comparative between S-FRULER and S-FRULER with feature selection (FS, ten features) for the indoor temperatures. Values are averaged across all folds

	S-FRULER			S-FRULER w/FS 10		
Dataset	RMSE	Rules	Time (s)	RMSE	Rules	Time (s)
P1 (1 h)	0.189	2.6	658.3	0.180	6.8	719.7
P1 (2 h)	0.298	4.2	644.8	0.281	5.5	703.5
P1 (4 h)	0.451	4.5	645.6	0.429	8.3	736.8
P1 (8 h)	0.652	3.5	639.7	0.605	9.6	693.3
P1 (12 h)	0.758	7.6	678.5	0.681	11.4	731.7
P1 (24 h)	0.749	2.7	799.3	0.705	9.5	738.8

We also evaluated the selected FRBS algorithms GFS.LT.RS and FS.HGD with and without feature selection (FS). These results are shown in Table 3. As we found during the experimentation that the methods had problems for learning, we also included tests with only five features to measure the effect that the number of features has in the learning process. Given the large amount of time taken by these methods, we performed only a fivefold cross validation for the estimation of the errors, and we limited the tests only to $t + 1$ h and $t + 8$ h.

For datasets P1-1 h and P1-8 h without feature selection, we could not get any result as the methods terminated with errors. In all the other cases, the average RMSE and time of the methods are higher than the results obtained by S-FRULER. It should be noted that the performance of the methods is very sensible to the number of features, as the errors were lower using five features instead of ten.

Table 3 Comparative between GFS.LT.RS and FS.HGD with and without feature selection for the indoor temperatures

	GFS.LT.RS		FS.HGD	
	RMSE	Time (s)	RMSE	Time (s)
P1, 1 h	N/A	N/A	N/A	N/A
P1, 1 h (FS 5)	0.56	4290.7	0.45	2096.2
P1, 1 h (FS 10)	4.87	1178.4	6.25	1223.5
P1, 8 h	N/A	N/A	N/A	N/A
P1, 8 h (FS 5)	0.89	6179.6	0.71	2417.7
P1, 8 h (FS 10)	5.04	301.9	6.37	409.7

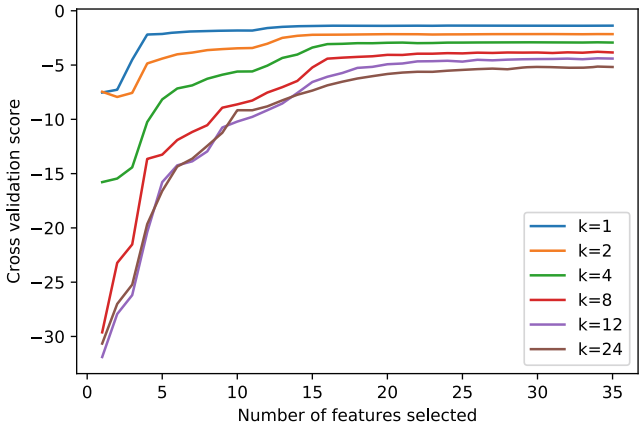

Fig. 12 Model score (negative MSE) estimated with shuffled tenfold cross validation for different number of features, selected by RFE, for the prediction of the upper tank temperature

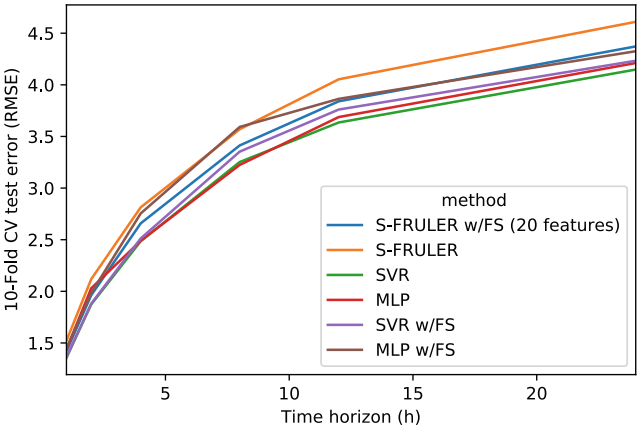

Fig. 14 Performance of the models (tenfold estimation of the test RMSE) for different time horizons using S-FRULER with feature selection (FS), without FS, the MLP and the SVR

Also, in theory, those errors might be further improved by increasing the number of individuals in the population and the generations (for GFS.LT.RS), and the number of iterations (for FS.HGD). However, in our tests, increasing those numbers above the values we used resulted in prohibitive times.

Upper Tank Temperatures

Figure 12 shows the estimated error with shuffled tenfold cross validation for different number of selected features for the upper tank temperature models, using GBT with the following hyperparameters:

- Learning rate: 0.05
- Max depth: 12
- Number of boosting stages: 600
- Column sample by tree: 0.8

Figure 13 shows the feature ranking of the top 20 selected features with RFE. Again, weather variables are among the top most informative features. This is in line with the expected behavior, since more hot water is used when the temperatures are colder. Cogenerator status features are also among the top 20 features, which is a good indicator that the use of the cogenerator is indeed affecting the behavior of the temperatures in the upper tank.

We performed the same comparison as before, with and without feature selection, using in this problem the top 20 features (Fig. 13). As can be seen in Fig. 14, the

Fig. 13 Top 20 features selected by RFE for $k = 24$ h for the upper tank temperature

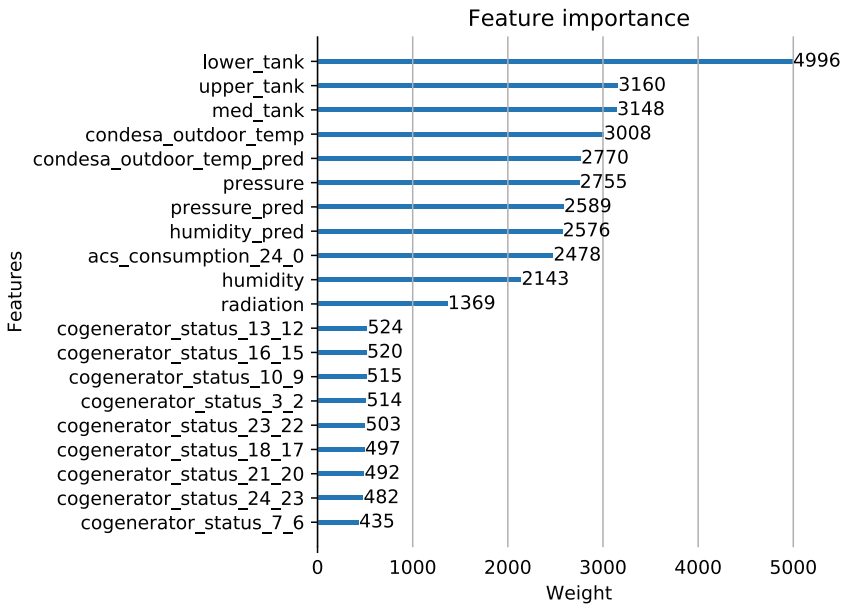

Table 4 Comparative between S-FRULER and S-FRULER with FS (20 features) for the upper tank temperatures

	S-FRULER			S-FRULER w/FS		
Dataset	RMSE	Rules	Time (s)	RMSE	Rules	Time (s)
Tank (1 h)	1.521	3.100	409.4	1.422	6.200	534.0
Tank (2 h)	2.117	3.800	406.8	1.962	7.100	501.2
Tank (4 h)	2.813	7.000	385.2	2.658	7.900	408.6
Tank (8 h)	3.570	6.600	398.3	3.413	7.400	398.4
Tank (12 h)	4.054	7.200	388.2	3.840	8.500	410.1
Tank (24 h)	4.610	5.400	392.7	4.372	8.000	424.0

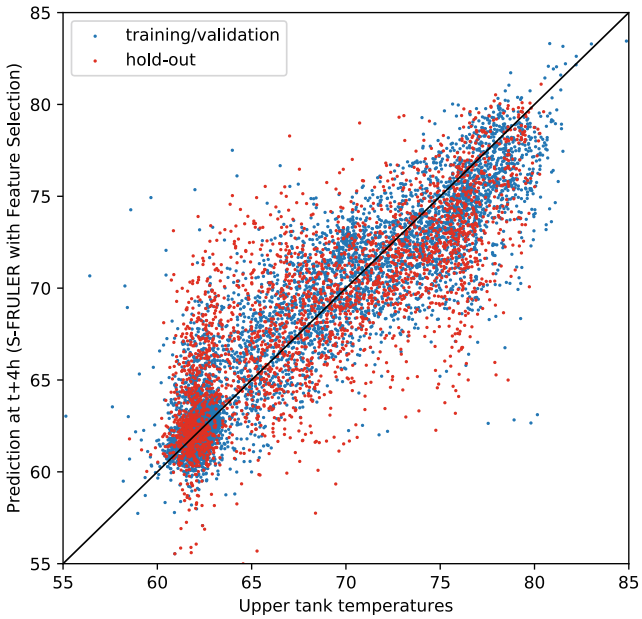

Fig. 15 Observations vs predictions for $k = 4h$ for one of the models generated with S-FRULER, for the training set and for a hold-out set with unseen data from 1 February 2017 to 31 May 2017

performance keeps improving with the number of features, but only slightly after the 20 most informative ones.

Comparing the results of the SVR and the MLP against S-FRULER with FS, we observe that the SVR achieves an error 5.1% better on average than S-FRULER w/FS (an average of 0.15 °C), and the MLP an error 2.5% better on average (an average of 0.10 °C). Both improvements are negligible for the prediction of the upper tank temperatures which are, on average, around 70 °C.

For this problem, the analysis of the performance of S-FRULER represented in Fig. 14 also reveals a greater advantage for the version of S-FRULER with feature selection for all time horizons (Table 4), with an improvement of 6.1%. Again, results for the SVR and the MLP with feature selection are worse than without feature selection. Comparing the results of the SVR and the MLP against S-FRULER with FS, we observe that the SVR achieves an error 5.1% better on average than S-FRULER w/FS (an average of 0.15 °C), and the MLP an error 2.5% better on average (an average of 0.10 °C). Both improvements are negligible for the prediction of the upper tank temperatures which are, on average, around 70 °C. Figures 15 and 16 show, as in the previous problem, the observed temperatures vs. the predicted ones for the 4-h model, with a similar behavior for the training and test data.

As in the case of the indoor temperatures, we performed the same experiment with GFS.LT.RS and FS.HGD. We observed the same behavior as before. The algorithms were not able to learn when all or a large set of features were used. Using only five features, the algorithms were able to learn with relatively low RMSE errors (3.93 and 2.03 for 1 h, 5.42 and 4.61 for 8 h, for GFS.LT.RS and FS.HGD respectively), although the errors obtained with the two methods were larger than the errors obtained by S-FRULER, both with and without FS (Table 5).

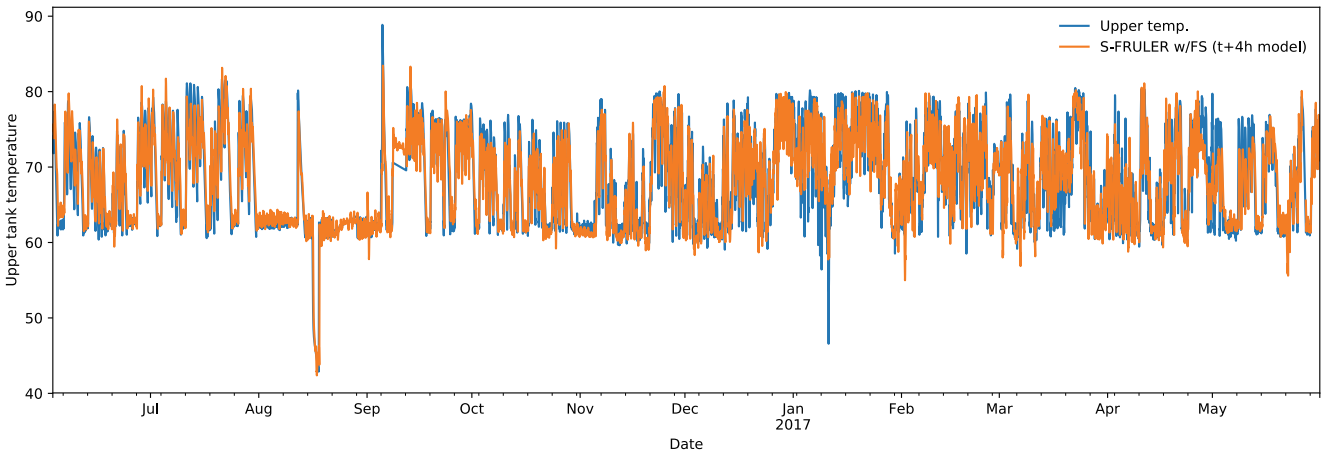

Fig. 16 Upper tank temperatures vs. predicted temperatures, using the model generated by S-FRULER for the time horizon $k = 4h$. The vertical red line separates the data used for cross validation (left) from the hold-out test data (right)

Table 5 Comparative between GFS.LT.RS and FS.HGD with and without feature selection for the upper tank temperatures

	GFS.LT.RS		FS.HGD	
	RMSE	Time (s)	RMSE	Time (s)
Tanks, 1 h	20.00	1759.0	25.69	3373.2
Tanks, 1 h (FS 5)	03.93	3088.2	02.03	1442.8
Tanks,1 h (FS 20)	19.81	1327.0	25.67	1843.5
Tanks, 8 h	20.01	2990.8	25.71	3008.4
Tanks, 8 h (FS 5)	05.42	2577.4	04.61	1381.5
Tanks, 8 h (FS 20)	19.74	1974.7	25.54	3097.6

Conclusions

In this work, we presented a novel cognitive computation approach based on genetic fuzzy systems (GFS) for scalable prediction of energy consumption in smart buildings. This method is applied in a large building of the University of Santiago de Compostela with more than 450 sensors and actuators that monitor and control different parts of the building. One of the main issues is the generation of interpretable and accurate fuzzy models in reasonable time, given the large amount of data generated in the building, a problem that is going to grow year after year as more data is available. This requires the use of scalable techniques to be able to cope with the increase in complexity. For this purpose, we used the S-FRULER GFS coupled with a feature selection mechanism that automatically selects the most informative features for each time horizon. The experiments with real data on two different problems related with the energy management revealed an average improvement of 6% on accuracy with respect to S-FRULER without feature selection, and with knowledge bases with a lower number of variables. Moreover, the generated rules are interpretable, allowing the managers of the building to make decisions for the reduction of energy consumption.

Funding Information This research was supported by the European Union LIFE programme (grant LIFE12 ENV/ES/001173). Also, the following support is acknowledged: Spanish Ministry of Economy and Competitiveness (grant TIN2017-84796-C2-1-R), Galician Ministry of Education, Culture and Universities (grants GRC2014/030 and accreditation 2016-2019, ED431G/08). These grants are co-funded by the European Regional Development Fund (ERDF/FEDER program).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Informed Consent Informed consent was not required as no human or animals were involved.

Human and Animal Rights This article does not contain any studies with human or animal subjects performed by any of the authors.

References

1. Alcalá R, Alcalá-Fdez J, Herrera F. A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection. IEEE Trans Fuzzy Syst. 2007;15(4):616–35.
2. Alcalá R, Alcalá-Fdez J, Herrera F, Otero J. Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation. Int J Approx Reason. 2007;44(1):45–64.
3. Alcalá R, Gacto MJ, Herrera F. A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems. IEEE Trans Fuzzy Syst. 2011;19(4):666–81.
4. Aljarah I, Ala'M AZ, Faris H, Hassonah MA, Mirjalili S, Saadeh H. Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm. Cogn Comput. 2018;10(3):478–495.
5. Atli BG, Miche Y, Kalliola A, Oliver I, Holtmanns S, Lendasse A. Anomaly-based intrusion detection using extreme learning machine and aggregation of network traffic statistics in probability space. Cogn Comput. 2018;10(5):848–863.
6. Balac N, Sipes T, Wolter N, Nunes K, Sinkovits B, Karimabadi H. Large scale predictive analytics for real-time energy management. In: IEEE international conference on Big Data, 2013; 2013. p. 657–64.
7. Bontempi G, Taieb S, Borgne YL. Machine learning strategies for time series forecasting. Business Intelligence. 2013;62–77.
8. Chen T, Guestrin C. Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, USA, August 13–17, 2016; 2016. p. 785–794.
9. Cordón O, Herrera F, Hoffmann F, Magdalena L, Cordon O, Herrera F, Hoffmann F. Genetic fuzzy systems. Singapore: World Scientific Publishing Company; 2001.
10. Ding S, Xi X, Liu Z, Qiao H, Zhang B. A novel manifold regularized online semi-supervised learning model. Cogn Comput. 2018;10(1):49–61.
11. Fan C, Xiao F, Wang S. Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques. Appl Energy. 2014;127:1–10.
12. Fernández A, López V, del Jesus MJ, Herrera F. Revisiting evolutionary fuzzy systems: taxonomy, applications, new trends and challenges. Knowl-Based Syst. 80:109–121. 2015.
13. Fernández A, del Río S, López V, Bawakid A, del Jesus MJ, Benítez JM, Herrera F. Big data with cloud computing: an insight on the computing environment, mapreduce, and programming frameworks. Wiley Interdiscip Rev: Data Min Knowl Disc. 2014;4(5):380–409.
14. Foucquier A, Robert S, Suard F, Stéphan L, Jay A. State of the art in building modelling and energy performances prediction: a review. Renew Sust Energ Rev. 2013;23:272–88.
15. Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat. 2001;29(5):1189–1232.
16. Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res. 2003;3:1157–82.
17. Herrera F. Genetic fuzzy systems: taxonomy, current research trends and prospects. Evol Intel. 2008;1(1):27–46.

18. Ishibuchi H, Nozaki K, Tanaka H, Hosaka Y, Matsuda M. Empirical study on learning in fuzzy systems by rice taste analysis. Fuzzy Sets Syst. 1994;64(2):129–44.
19. L'Heureux A, Grolinger K, ElYamany HF, Capretz MAM. Machine learning with big data: challenges and approaches. IEEE Access. 2017;5:7776–97.
20. Li C, Deng C, Zhou S, Zhao B, Huang GB. Conditional random mapping for effective elm feature representation. Cogn Comput. 2018;10(5):827–847.
21. Life-OPERE. http://www.life-opere.org/. Online; Accessed 19 Feb 2018. 2016.
22. Lu T, Viljanen M. Prediction of indoor temperature and relative humidity using neural network models: model comparison. Neural Comput Appl. 2009;18(4):345–57.
23. Luo X, Zhu X, Lim EG, Huang Y. A semi-blind model with parameter identification for building temperature estimation. Cogn Comput. 2018;10(1):105–16.
24. Marchiori E. Class conditional nearest neighbor for large margin instance selection. IEEE Trans Pattern Anal Mach Intell. 2010;32(2):364–70.
25. Márquez AA, Márquez FA, Roldán AM, Peregrín A. An efficient adaptive fuzzy inference system for complex and high dimensional regression problems in linguistic fuzzy modelling. Knowl-Based Syst. 2013;54:42–52.
26. Mechaqrane A, Zouak M. A comparison of linear and neural network ARX models applied to a prediction of the indoor temperature of a building. Neural Comput Appl. 2004;13(1):32–7.
27. Meteogalicia: Galician meteorological web page. http://www.meteogalicia.gal (2016). Online; Accessed 19 Feb 2018.
28. Nian X, Sun M, Guo H, Wang H, Dai L. Observer-based stabilization control of time-delay t-s fuzzy systems via the non-uniform delay partitioning approach. Cogn Comput. 2017;9(1):225–36.
29. Parliament E, Council E. On the energy performance of buildings. Off J Eur Union. 2010;153:13–35.
30. Ramírez-Gallego S, Fernández A, García S, Chen M, Herrera F. Big data: tutorial and guidelines on information and process fusion for analytics algorithms with mapreduce. Inf Fusion. 2018;42:51–61.
31. Reyes-Ortiz JL, Oneto L, Anguita D. Big data analytics in the cloud: spark on Hadoop vs MPI/OpenMP on Beowulf. Procedia Comput Sci. 2015;53:121–30.
32. Riza LS, Bergmeir CN, Herrera F, Benítez Sánchez JM. FRBS: Fuzzy rule-based systems for classification and regression in R. American Statistical Association. 2015.
33. Rodríguez FJ, García A, Pardo PJ, Chávez F, Luque-Baena RM. Study and classification of plum varieties using image analysis and deep learning techniques. Progr Artif Intell. 2018;7(2):119–27.
34. Rodríguez-Fdez I, Mucientes M, Bugarín A. An instance selection algorithm for regression and its application in variance reduction. In: Proceedings of the IEEE international conference on fuzzy systems (FUZZ-IEEE); 2013. p. 1–8.
35. Rodríguez-Fdez I, Mucientes M, Bugarín A. Reducing the complexity in genetic learning of accurate regression TSK rule-based systems. In: Proceedings of the 2015 IEEE international conference on fuzzy systems (FUZZ-IEEE). IEEE; 2015. p. 1–8.
36. Rodríguez-Fdez I, Mucientes M, Bugarín A. FRULER: fuzzy rule learning through evolution for regression. Inf Sci. 2016;354:1–18.
37. Rodríguez-Fdez I, Mucientes M, Bugarín A. S-FRULER: scalable fuzzy rule learning through evolution for regression. Knowl-Based Syst. 2016;110:255–66.
38. Rodriguez-Mier P, Mucientes M, Bugarín A. Scalable modeling of thermal dynamics in buildings using fuzzy rules for regression. In: 2017 IEEE international conference on fuzzy systems (FUZZ-IEEE); 2017. p. 1–6.
39. Ruano A, Crispim E, Conceicao E, Lúcio MM. Prediction of building's temperature using neural networks models. Energy Build. 2006;38(6):682–94.
40. Shaikh PH, Nor NBM, Nallagownden P, Elamvazuthi I, Ibrahim T. A review on optimized control systems for building energy and comfort management of smart sustainable buildings. Renew Sust Energ Rev. 2014;34:409–29.
41. Teodosiu C, Hohota R, Rusaouën G, Woloszyn M. Numerical prediction of indoor air humidity and its effect on indoor environment. Build Environ. 2003;38(5):655–64.
42. Thomas B, Soleimani-Mohseni M. Artificial neural network models for indoor temperature prediction: investigations in two buildings. Neural Comput Appl. 2006;16(1):81–9.
43. White T. Hadoop: the definitive guide. O'Reilly Media, Inc. 2012.
44. Yaqoob I, Hashem IAT, Gani A, Mokhtar S, Ahmed E, Anuar NB, Vasilakos AV. Big data: from beginning to future. Int J Inf Manag. 2016;36(6):1231–47.
45. Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I. Spark: cluster computing with working sets. In: Proceedings of the 2nd USENIX conference on hot topics in cloud computing, vol 10, p 10; 2010.
46. Zhao HX, Magoulès F. A review on the prediction of building energy consumption. Renew Sust Energ Rev. 2012;16(6):3586–92.
47. Zhao J, Lasternas B, Lam KP, Yun R, Loftness V. Occupant behavior and schedule modeling for building energy simulation through office appliance power consumption data mining. Energy Build. 2014;82:341–55.
48. Zou H, Hastie T. Regularization and variable selection via the elastic net. J R Stat Soc. 2005;67(2):301–20.