Information Fusion 94 (2023) 169–180

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

A new approach based on association rules to add explainability to time series forecasting models

A.R. Troncoso-García [a], M. Martínez-Ballesteros [b], F. Martínez-Álvarez [a,*], A. Troncoso [a]

[a] Data Science and Big Data Lab, Pablo de Olavide University, ES-41013 Seville, Spain
[b] Department of Computer Science, University of Seville, ES-41012 Seville, Spain

ARTICLE INFO

Keywords:
Explainable AI
Machine learning
Time series forecasting
Interpretability
Association rules

ABSTRACT

Machine learning and deep learning have become the most useful and powerful tools in the last years to mine information from large datasets. Despite the successful application to many research fields, it is widely known that some of these solutions based on artificial intelligence are considered black-box models, meaning that most experts find difficult to explain and interpret the models and why they generate such outputs. In this context, explainable artificial intelligence is emerging with the aim of providing black-box models with sufficient interpretability. Thus, models could be easily understood and further applied. This work proposes a novel method to explain black-box models, by using numeric association rules to explain and interpret multi-step time series forecasting models. Thus, a multi-objective algorithm is used to discover quantitative association rules from the target model. Then, visual explanation techniques are applied to make the rules more interpretable. Data from Spanish electricity energy consumption has been used to assess the suitability of the proposal.

1. Introduction

Machine learning (ML) and deep learning (DL) algorithms are essential tools that are used to make predictions and classify large and heterogeneous data in different fields, such as medicine [1]. They are the technology behind artificial intelligence applications in industries such as object recognition, natural language processing, or self-driving cars. One of their most serious disadvantages is that they are considered black-box models, meaning that it is *impossible* to know how the model obtains the outputs by applying inner nonlinear operations to the inputs.

Explainability for artificial intelligence (XAI) is focused on *explaining* artificial intelligence (AI) models themselves. Explainability also concerns predictions generated by the AI models, clarifying the model's behavior to reticent humans.

First, what a good explanation is must be defined. The spotlight is on the human ability to understand the model. XAI is a human-model interaction, where the model is producing a certain output and humans want to know *how* and *why* these outputs are computed [2]. In that way, XAI is an extensive research field, becoming crucial as AI models are used today to make high-stakes decisions in essential sectors, namely health, security, or economy [3].

Interpretability could be defined as the transparency of the model, specifically related to humans' ability to understand it [4]. Some ML models are considered interpretable, such as decision trees or rule-based models, whereas others, such as neural networks, are not [5]. In addition, explainability could be seen as a relation between input data and the prediction of a model, in such a way that the model's decisions can be easily understood by humans [6]. There is also a huge set of associated concepts such as comprehension, trustability or transparency. Comprehension could be defined as the action of grasping with the intellect, in other words, the ability of understanding, whereas trustability is about the reliance on the truth or ability of the model. Finally, transparency is defined as the state of being transparent, having the property of showing through [7].

Time series are a special type of data where there is a sequence of data points indexed in time order. Data have been collected from the same source at different points in time, usually over a time interval. They differ from other kinds such as tabular data, images or textual description. Time series forecasting is a kind of regression problem where numerical data are predicted. Predictions are made based on historical time-stamped data. Traditional ML and DL models are usually specifically adapted for time series forecasting [8,9].

* Corresponding author.
E-mail addresses: artrogar@upo.es (A.R. Troncoso-García), mariamartinez@us.es (M. Martínez-Ballesteros), fmaralv@upo.es (F. Martínez-Álvarez), atrolor@upo.es (A. Troncoso).

https://doi.org/10.1016/j.inffus.2023.01.021
Received 11 November 2022; Received in revised form 11 January 2023; Accepted 24 January 2023
Available online 26 January 2023

This paper carries out a study to explore the explainability of ML and DL methods specifically for time series forecasting. Preexisting methods are used for data prediction. Then association rules (ARs) are extracted to find connections between input data and predicted values. Finally, explainability is added towards visual representations of the rules. Summarizing, the main contributions are:

1. A new methodology focused on visual explainability for time series forecasting is created.
2. The use of association rules as an agnostic approach for adding explainability to ML and DL methods is proposed. The performance of the rules is tested by three different methods using the same time series dataset.
3. Visual representation of the rules is presented to clarify the explanation of the models' predictions.

The remainder of the article is structured as follows. In Section 2, recent advances in interpretability through ARs are reviewed. Section 3 describes and details the experiments carried out and Section 4 presents the results that have been obtained. Section 5 concludes the paper.

2. Related work

Interpretability techniques are grouped with respect to the phase of experimentation and prediction in which they are used [10]. Pre-model interpretability is related to data interpretability and exploratory and data visualization. Then, in-model interpretability is focused on creating models that could explain themselves, for example, decision tree algorithms. More complex models are specifically designed for this purpose. In-model interpretability is created by adding explainable layers to make models transparent. Finally, post-model interpretability refers to the interpretation of the outputs according to the inputs, and it is independent of the model that has been used to obtain the outputs or predictions [11].

In the literature, explainability is generally reached by two different approaches: on the one hand, in-model, in other words, by design. The concept of creating an interpretable model is supported by the concept of transparency. On the other hand, explainability could be reached in a post-model scenario by applying XAI techniques to existent ML models. This is also called post-hoc explainability [12].

Traditionally, certain ML models have been considered interpretable, such as linear or logistic regression, decision trees, k-nearest neighbors, Bayesian models and rule-based models whereas DL approaches like neural networks are not [3]. In this paper, the focus is on adding explainability to existing ML and DL models for time series forecasting.

2.1. Post-hoc XAI

Post-hoc techniques are focused on increasing the interpretability of preexisting ML and DL solutions by adding certain explainable layers to the models. One of their main advantages is the fact that they are independent of the model; in other words, they are model-agnostic methods. Explainable techniques are applied to the results or predictions that have been obtained after training the model. In the case of time series forecasting, where specific algorithms are used, post-hoc XAI techniques have high level of interest [13].

There are several techniques for post-hoc explainability and they are usually classified in the following categories: text, visual, local, and feature relevance explanations. In general, local explanations and feature relevance techniques are the most representative in the literature [3]. There is no better technique than the other. One or more of these techniques could be more suitable depending on the nature of data that the model is predicting. Here, as the goal is time series forecasting, the most essential aspects are discussed as follows.

- *Text explanations*. Explanations are generated as textual description. Although time series data could be translated into a set of numerical intervals, textual description could not be really understandable in time series scope [14].
- *Visual explanations*. Explanations are presented in a certain graphical representation, which makes it easier to have a mental image of the predictions [15]. Concerning time series data, which could be easily represented in two-dimensional axes, a graphical explanation could be particularly useful.
- *Local explanations*. These explanations refer not to obtain explanations of the complete dataset but for a specific instance of the data. It could be useful to explain concrete examples of the data [16].
- *Feature relevance explanations*. It aims to get information about which attributes or features are more important for the predictions [17]. In time series data the attributes are past events. This kind of explanation could lead to misunderstanding. However, it could be complemented with alternative representations of the set of attributes, as it is proposed in this paper.

In conclusion, the most useful explanations for the explainability of time series forecasting are, at first sight, visual and feature relevance explanations.

Several approaches for adding post-hoc interpretability, achieving an explainable result for an initial non-explainable model, can be found in the literature.

First, Local Interpretable Model-Agnostic (LIME) technique is revised. LIME is one popular technique for adding explainability to preexisting ML models. LIME can be used as a library for Python. LIME is widely used due to simplicity. LIME explains the behavior of black box methods based on linear models around one instance of interest [18], meaning that LIME is giving local explanations. Explanations are generated by perturbing a chosen point in the input data and making new predictions with the perturbed data as input. Thus, the critical attributes to make predictions are detected. LIME explainable ability is evaluated in [19]. A LIME use case is presented in [20], where LIME is applied as a way of adding explainability to a sleep apnea prediction method. A similar approach is SHapely Additive exPlanation (SHAP) [21]. This method calculates the contribution of each aspect for a concrete prediction based on game theory. This contribution is called SHAP value. SHAP also calculates the global influence of each feature. For example, in [22] SHAP is applied to understand and verify AI models for concrete fire-induced spalling. However, both LIME and SHAP have a limitation: LIME and SHAP introduce perturbation to data and recalculate the predictions. Because of that, predictions need to be made by pre-existing models (such as Scikit-Learn library in Python). LIME and SHAP application to big data or streaming systems is problematic.

Concerning XAI for demand forecasting, which is the initial problem in this paper, there are some examples in literature. The work presented in [23] adds an explainable layer to several machine learning regression models in electric vehicle load demand forecasting problem. In addition, an agnostic architecture based on semantic technologies is proposed in the field of manufacturing demand forecasting [24].

Furthermore, in [25] an interactive web browser system, Summit, is put on. The system can create understandable explanations for neural networks' internal behavior, showing activation levels and relationships between neurons inside a neural network. It summarized and visualized the data for human comprehension. In [26], graph neural networks are explored as a tool for increase explainability in an automated medical decision pipeline. Additionally, the research to create a correspondence between clinical text with diagnoses and the corresponding graphic output of the ICD-10 code is presented in [27]. They achieved the goal of an interpretable result through various visualizations that show the correspondence between each code and the piece of text. It has been tested in a real medical setting and clinicians found it very advantageous.

The estimation of the performance of the tools that add interpretability is ambiguous. A model is considered interpretable when final users could understand in certain way how the model is working or when users could explain how the predictions are made. For example, the research in [28] is to create a bias for evaluating XAI in computer vision field. However, post-hoc XAI methods in other scopes must be evaluated by people. If the end users are able to understand the computations made by ML and DL algorithms, the model is *explainable* and post-hoc XAI techniques are achieving their goal [29].

2.2. Association rules for XAI

Association rules learning is a kind of unsupervised machine learning method. An association rule is a 'if-them' statement where a combination of conditions (called antecedent) of the input data maps into a certain aspect of the output values (called consequent). A set of ARs is generated to explain the relationship between the main aspects of the input and output data. Traditional rule-based models are considered explainable themselves [30].

Having said that, examples about the creation of explainable models by using ARs have been found in the literature. In the major cases, explainable models are applied to essential areas, such as disease detection in health care [11]. For example, in [31], a model based on rules and Bayesian analysis is built and tested in personalized medicine and health. In [32] a multi-objective optimization for multiple ARs is developed for interpretable classification. At first, they generated a set of ARs and classify them according to the *interestingness score* and *support* measure. Prioritized rules are studied using a two-layer neural network. Experiments showed that the model obtains better performance and better execution time than other AR mining models. Finally, interpretability is added to an existing model known as the *Takagi–Sugeno–Kang* fuzzy model in [33] by generating ARs.

On the other hand, rule-based approaches could be used as a post-hoc method for adding explainability to more complex and higher accuracy models such as neural networks. For example, on the basis of already commented LIME tool, the same authors have developed a new approach called Anchors. Anchors generates 'if-then' rules in order to increase the explainability of the results obtained by local perturbations [34]. In [35], ARs are extracted using a model based on the well-known *Apriori* algorithm for explainability in an omic-data neural network. ARs are evaluated regarding a set of quantitative quality measures such as *confidence*, *support*, *lift* or *conviction*. Then, the explainable model is validated by human experts. Another example is shown in [36], where ARs are extracted from a decision tree model with high accuracy values. In [37] ARs are used to explain the predictions produced using a tabular classification dataset. Experimentation is carried out by building a neighborhood of similar instances and making predictions for those perturbed instances. Then, ARs are generated and the *k*-optimal ones are selected. The focus is on rules that cover more instances rather than the highest predictive ones. Lastly, in the survey carried out in [38] several methods using ARs are also presented.

3. Methodology

The main goal of this work is to create visual explanations by exploring the ability of ARs to interpret the predictions of a time series made by ML and DL models. The general process is illustrated in Fig. 1.

First, the target time series is acquired and pre-processed so it can feed the time series forecasting model that is wanted to be explained. More specifically, several time windows w are used for predicting a certain number of future horizons h.

Later, the outputs generated by the model are used to feed the association rule extraction module, which is in charge of discovering rules. ARs are shown as a reliable way to understand the internal behavior of the model and the connections between input data and predictions. In particular, the target time series are used as antecedent and the forecasts by the model are used as consequent, thus that this module can extract ARs, meaning the module can find relationships between the actual time series and the predicted values.

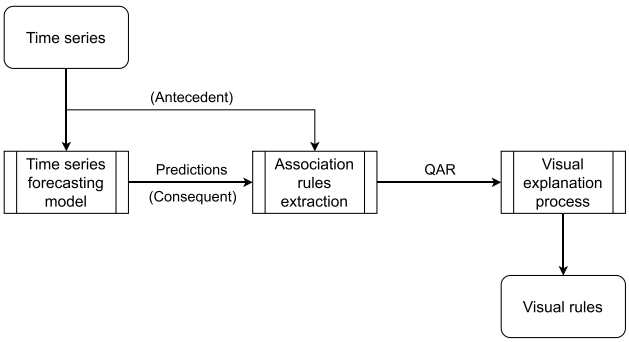


Fig. 1. Diagram showing the proposed methodology.

Such ARs are used as input of the visual explanation process, a technique of model-agnostic explainability. The visual explanation process is able to explain the time series forecasting model by visually depicting the values of the time series. Graphical representations are colorized according to the values of time series that influence the most when making predictions for each particular prediction horizon.

3.1. Main steps description

3.1.1. Time series forecasting formulation

The proposed methodology is tested using the predictions of three different ML methods, all using the same time series input data.

Given a time series with previous values up to time t, $[X_1, \ldots, X_t]$, the task is to predict the h next values of the time series, from a window of w past values. This multi-step forecasting problem can be formulated as follows, where f is the model to be learned by the machine learning model in the training phase:

$$\left[X_{t+1}, X_{t+2}, \ldots, X_{t+h}\right] = f\left(X_t, X_{t-1}, \ldots, X_{t-(w-1)}\right) \tag{1}$$

Therefore, the input data are used to train the ML and DL models, obtaining the prediction model $\hat{f}$. In an ordinary machine learning experiment, we would test the model with different input data and measure the performance of the algorithm. However, we aim at learning *how* and *why* the model $\hat{f}$ makes a prediction. Consequently, we use the model $\hat{f}$ to make predictions for the same data that have been used to train it. That is:

$$\left[\hat{X}_{t+1}, \ldots, \hat{X}_{t+h}\right] = \hat{f}\left(X_t, \ldots, X_{t-(w-1)}\right) \tag{2}$$

where $[\hat{X}_{t+1}, \ldots, \hat{X}_{t+h}]$ are the values predicted by the machine learning model $\hat{f}$.

3.1.2. Association rules

In the field of data mining, association rule learning is a popular and well-known method to discover interesting relations among variables in large databases. They are considered an interpretable ML method [3], because of their simplicity and similarities to the human way of reasoning. ARs are also providing high-accuracy results.

Let $A = \{a_1, a_2, \ldots, a_n\}$ be a set of features or attributes, with values in $\mathbb{R}$. Let S and T be two disjoint subsets of A, that is, $S \subset A$, $T \subset A$ and $S \cap T = \emptyset$. An AR is known as a quantitative association rule (hereinafter referred as QAR) when the domain is continuous. A QAR is a rule $X \Rightarrow Y$, in which features in S belong to the antecedent X, and features in T belong to the consequent Y, such that X and Y are formed by a conjunction of multiple Boolean expressions of the form $a_i \in [l, u]$, (with $l, u \in \mathbb{R}$). Thus, in a QAR, the features or attributes of the antecedent are related to the features of the consequent, establishing an interval of membership values for each attribute involved in the rule.

Many measures could be found in the literature to assess the quality of QARs. Definition and mathematical equations of the main quality measures can be found in [39]. In particular, *support* (Eq. (3)), *confidence* (Eq. (4)), and *gain* (Eq. (5)) have been the objectives to be optimized by the association rule extraction model in order to assess the generality, reliability and information gain of the rules, respectively.

The support of the rule $X \Longrightarrow Y$ is the percentage of records in the dataset that contain X and Y simultaneously. Note that $n(X \cap Y)$ is the number of instances that satisfy the conditions for the antecedent X and Y in the dataset simultaneously. N is the total number of instances in the dataset. Support values are ranged in the interval [0, 1].

$$Support(X \Longrightarrow Y) = \frac{n(X \cap Y)}{N} \tag{3}$$

The confidence is the probability that instances containing X, also contain Y. The confidence values range in the interval [0, 1].

$$Confidence(X \Longrightarrow Y) = \frac{support(X \Longrightarrow Y)}{support(X)} \tag{4}$$

Support and confidence are the most used measures for QAR optimization. However, optimization of the support may not be enough, since very general QARs could be obtained, and the amplitude of the intervals could be increased to reach the whole domain of each attribute. Additionally, if the confidence is only optimized may present some drawbacks because this measure does not consider the support of the consequent of the rule, therefore it is not able to detect negative dependence among items. To overcome these issues, other measures such as gain, can be optimized due to the antecedent and the consequent of the rule are considered.

Gain is calculated from the difference between the confidence of the rule and consequent support. It is also known as added value or change of support. Gain values range in the interval [−0.5, 1].

$$Gain(X \Longrightarrow Y) = confidence(X \Longrightarrow Y) - support(Y) \tag{5}$$

The input dataset D for the rule extraction algorithm is constructed as follows:

$$D = \{(X^{(i)}, Y^{(i)}) : i = 1, 2, \ldots, N\} \tag{6}$$

where N is the number of instances, $X^{(i)}$ and $Y^{(i)}$ are the characteristics that belong to the antecedent and the consequent of the rule, respectively. These features are defined as follows:

$$X^{(i)} = [X_{t-(w-1)}, \ldots, X_{t-1}, X_t] \tag{7}$$

$$Y^{(i)} = [\hat{X}_{t+1}, \hat{X}_{t+2}, \ldots, \hat{X}_{t+h}] \tag{8}$$

where $t = w + (i-1) * h$.

In order to ensure that rules with all prediction horizons in the consequent are obtained, the input dataset D is divided into subsets D_j with $j = 1, \ldots, h$.

$$D_j = \{(X^{(i)}, Y_j^{(i)}) : i = 1, 2, \ldots, N\} \tag{9}$$

where the attributes forming the consequent of the rule are made up of a single attribute:

$$Y_j^{(i)} = \hat{X}_{t+j} \tag{10}$$

3.1.3. Visual explanations

The set of QARs obtained previously are used as an input for the creation of graphical representations. The objective is to show visually the importance of each item in the input data to predict the target values.

Association rules are well known for being 'if-then' statements, where the *if* clause or antecedent is explaining the results that are on the *then* one, also called consequent. QARs are usually intuitive, and they are considered an interpretable ML method [40]. Here, the antecedent has data from the $X_{t-(i)}$ past values that have been used for making predictions whereas the consequent contains the X_{t+j} next values that the models are predicting.

However, regarding time series data, the understandability of QARs could be reduced. In the field of time series, data are quantitative variables concerning the area in which they are measured. Due to this, both the antecedent and the consequent of the QARs that are obtained using an association rule extraction algorithm show the variable and the numerical interval that makes true each certain association, that is:

$$X_{t-(w-1)} \in [l_{w-1}, u_{w-1}] \wedge \ldots \wedge \wedge X_{t-1} \in [l_1, u_1] \wedge X_t \in [l_0, u_0] \Rightarrow \hat{X}_{t+j} \in [l_j, u_j] \tag{11}$$

being $j = 1, \ldots, h$ and where w is the length of the windows composed of the past values used to predict the h next values.

The main idea here concerns gaining knowledge of the attributes that appeared more frequently in the antecedent. This means that these attributes are more important for the predictive model when making predictions. Thus, visual representations are created by computing the antecedent of each rule in the set of rules. This process is made for all prediction horizons. In this work, 24 prediction horizons are considered and there are 168 attributes (see Eqs. (7) and (8) in Section 3.1.2).

For each rule, the number of times that attributes appear is counted. This process is called occurrence calculation. The pseudocode is presented in Algorithm 1.

Previously, given a set of QARs, the information contained in the antecedent of each of the rules is saved. This information is composed by both the name of the attribute (X_t) and the range that makes true the rule statement. Then, something similar to a counter is implemented. The amount of times that each attribute appears is computed. Thus, the summation of the amount of times that each attribute X_t appears in the set of QARs is made. As a result, a list of 168 elements is generated. Each item of the list contains the number of times that the corresponding attribute appears in the set of rules. This is executed for the 24 sets of rules, meaning that there is one list for each prediction horizon. Finally, a matrix with 168 columns and 24 rows is obtained. The size of the matrix depends on the number of attributes that could be in the antecedent of the QARs and the number of prediction horizons that are evaluated. As a final step, the resulting matrix is normalized meaning that all the possible values are between 0 and 1.

At the end, these calculations make possible a visual representation of the information shown by the QARs. For each of the 24 prediction horizons, the attributes are graded according to their influence on the predictions. The more a variable appears on the antecedent, the more they are used as an essential tool for time series forecasting. This information could be graphically represented by translating the numbers into a graded color range. For example, using a heatmap. It could also be used for coloring a graphical representation of time series, as it will be reported and shown in Section 4.

Algorithm 1 Algorithm for representing rules

Require: a list of antecedent elements A, number of features w

```
C ← []                                          ▷ List with size = w
for a in A do
    ▷ each element a contains the name of the X_{t-i} attribute, where i
      is a number between 0, and 167 and the associated interval [l_i, u_i]
    j ← i                          ▷ name of the variable that will work as index
    C[j] = C[j] + 1
end for
return C
```

3.1.4. Comparison with alternative methods

This methodology is tested and compared with an existing explainable method: LIME [18]. LIME has been previously described as a post-hoc explainable method.

LIME has a local behavior. Meaning that the explanations are obtained locally, specifically for one instance. The proposed methodology tries to explain the predictions in a general way. Thus, LIME has been here adapted to explain the whole of the predictions. Then, LIME has been executed randomly a certain number of times, identifying the 10 most relevant features to make predictions for each iteration. Then, these features are computed and the *important* features for LIME are obtained. Attributes counting is done in a similar way as the antecedent of the rules in Section 3.1.3. LIME results are used as a baseline to compare the explanations provided after applying the proposed methodology. Results are presented in Section 4.

3.2. Fundamentals of the methods chosen for the proposed methodology

In the previous section, the methodology was described in a general way. However, every step requires the selection of particular methods to perform each of the tasks involved in the flowchart. In this section, fundamentals of the selected methods are described.

3.2.1. Time series forecasting models description

Three specific ML or DL models have been used to test this methodology. The models are not tested or compared because they have already been widely applied to several time series datasets and have been proved to be accurate and powerful methods for time series forecasting by other authors [41–47]. The models used to make the predictions are commented as follows.

- **Wk-NN** (Weighted k-Nearest Neighbors algorithm) [41,42,44,45]. The algorithm is a generalization of the well-known k-nearest neighbors (kNN) method. In this case, the algorithm achieves more accuracy by adding weights according to the distance with the concrete point. That means that the closer elements are more important (translated into the fact that they have a higher weight) than the further ones. The prediction is computed by a weighted average of the h next values to the k-nearest neighbors of the w past values.
- **bigPSF** (Pattern Sequence based Forecasting algorithm for big data) [43]. This algorithm is the extension and adaption of the original PSF algorithm [48,49] which also has a version for handling with functional time series [50]. It is a multi-output approach specifically adequate for time series forecasting. It is scalable thanks to distributed computation using Apache Spark framework and it is also a flexible tool due to its multi-output nature. The bigPSF makes a clustering from data as an initial step identifying the past points belonging to the same cluster that the point to be predicted. Thus, the prediction is the average of the h next values to these past points.
- **LSTM** (Long Short-Term Memory network) [46,47]. It is a deep learning method widely used for time series forecasting. This network is a recurrent neural network. It is satisfactory for time series forecasting due to its ability to deal with sequential data.

The predictions obtained by these models are used to test the ability of proposed methodology to explain different algorithms.

3.2.2. Association rules extraction model

The approach presented in this paper uses an evolutionary-genetic algorithm for the extraction of QARs, hereafter referred to as MOQAR [39].

MOQAR mines QARs in datasets with continuous attributes without discretizing the attributes of the dataset trying to find the best trade-off among all the measures optimized. A detailed description of MOQAR can be found in [39]. The main features of the algorithm are described below:

- An individual in the population represents a rule that codes the membership of the attributes in the rules (antecedent or consequent) and their interval bounds. Let R be an individual of the population that represents a rule, let K^R be the subset of attributes of the dataset, $K^R \subset$ A, which are expressed in the rule R and let a be an attribute $a \in K^R$. Let I^R be a function, $I^R : K^R \longrightarrow \mathbb{R}$, which defines the relation between the attributes in K^R and the bounds of the intervals for such attributes. Thus, $I^R(a) = [l_a^R, u_a^R]$ represents the lower and upper bounds of the attribute a, which belongs to the rule R. Let T^R be a function, $T^R : K^R \longrightarrow 1, 2$, which defines the relation between the attributes belonging to K^R and the type of membership of the attributes. Therefore, $T^R(a)$ represents the membership type of the attribute a in rule R, that is, if a belongs to the antecedent or the consequent of R. Thus, $T^R(a) = 1$ if a belongs to the antecedent of the rule R or $T^R(a) = 2$ if a belongs to the consequent of the rule.
- MOQAR performs an evolutionary process to learn the most appropriate intervals of the attributes, so that the intervals are adjusted in a self-adaptive way to find QAR with high interpretability, interestingness, and precision.
- MOQAR tries to find rules that satisfy the coverage of instances that are still not covered. In this way, instances already covered by the previous rules are penalized. Therefore, the samples covered by few rules have a higher priority to be selected to generate the new population.
- The number of generations determines when the evolutionary process ends, which is repeated until the desired number of iterations is reached. Finally, MOQAR returns the set of QARs discovered that satisfies the defined minimum quality thresholds.

Then, MOQAR is applied to all subsets D_j defined in Section 3.1.2 separately for the extraction of QARs. Parameters are configured to only retrieve the rules with confidence and accuracy greater than 0.5 and support greater than 0.05. This step is just to minimize the number of valid rules for each iteration.

4. Results

This section analyzes the results obtained after carrying out the methodology detailed above. It is divided into three sections. Section 4.1 presents the dataset used in this work. The second one, Section 4.2 shows the QARs that have been extracted. The third one, Section 4.3 gives information about the visual explanations that have been created with this information.

4.1. Time series data

The input data used for this experiment are a time series of electrical energy consumption in Spain [44]. Data have been collected with 10-minute frequency during nine years and six months, specifically between January 1st 2007 and June 21st 2016. It is a window of X_{t-i} past values with $i = 0, \ldots, w-1$ that is used to estimate a future window of X_{t+j} with $j = 1, \ldots, h$ values. The value of w has been set to 168, representing 1 day and 4 h, whereas the value h is 24, that is, 4 h. That means that 24 h (1 day) has got here 144 elements.

An example of the time series data used for testing this methodology could be seen in Fig. 2. The blue line is the input data used for the antecedent of the QARs whereas the green line points (predictions) are the consequent. The real values are also represented.

4.2. Association rules

This section is presenting the QARs obtained by the rules mining algorithm MOQAR. Here the focus is not yet on explainability itself whereas it is on the QARs. The reason is that QARs are used as a way through explaining ML and DL models.

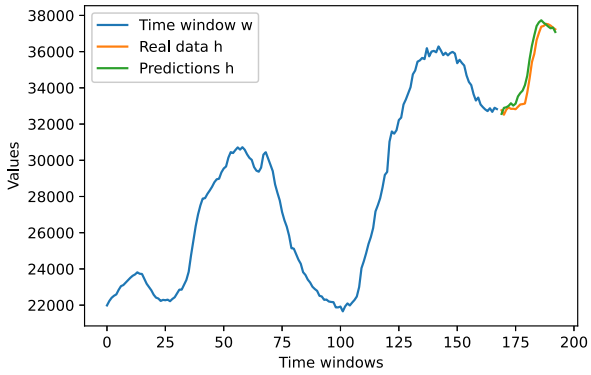


Fig. 2. Example of a time series instance. Predictions made with LSTM model.

4.2.1. QARs

The MOQAR algorithm is used for obtaining QARs. A set of rules is extracted for each of the 24 samples forming the prediction horizon. For each one, between 12 and 20 rules are obtained, resulting in a total of about 400 rules across all horizons (for each of the three predictions obtained by three different ML and DL models: WkNN, bigPSF, LSTM).

QARs have been filtered by a certain query. The selected QARs satisfy a minimum confidence threshold higher than 0.8, a minimum support threshold higher than 0.3 and a total sum of the width of all rule intervals less than 12000.

Then, the most representative QARs with reference to confidence and support are presented. Tables 1–3 present an example of QAR for each prediction horizon for each of the models. A set of rules have been generated for each prediction horizon X_{t+j} with $j = 1, \ldots, 24$ values. These tables contain only one example for each prediction horizon for the three models. Each rule is part of the set of QARs that have been obtained by the rules mining algorithm.

Tables 1–3 also show the quality measures of each corresponding set of rules. In particular, support (Eq. (3)), confidence (Eq. (4)) and gain (Eq. (5)) measures that assess the generality, reliability and gain information of QARs, respectively, are presented.

It could be observed a slight decrease in confidence and gain as time passes, above all in WkNN (Table 1) and LSTM (Table 3) results. A set of 24 future events are predicted here using ML and DL models. When the prediction is further from the present event (higher values of h), predictions are worse. The same is observed concerning the quality of the rules.

Comparing the three different methods, in general, the set of QARs obtained for bigPSF predictions, in Table 2, have worse quality in terms of generality, reliability and gain information than the other two methods. Rules for WkNN and LSTM models seem to be similar concerning the mentioned quality measures.

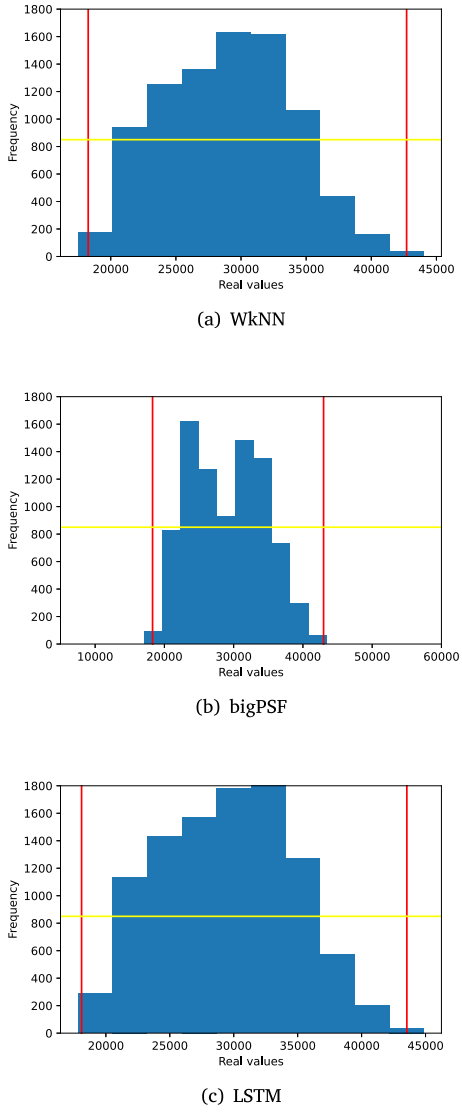
4.2.2. Real data ranges

After the study of the sets of rules, the range of real data covered by the QARs is also presented.

The real range is the interval formed by the minimum electrical consumption value and the maximum consumption value. Each timestamp has a real data interval.

Then, the amount of real data explained or covered by the set of rules obtained for each prediction horizon is computed. That means the interval covered by the set of rules for each timestamp, from the minimum value to the maximum value appearing in the antecedent of the QARs.

Range covered by QARs is calculated in two ways, concerning the real range of each prediction horizon and the most frequent values



(a) WkNN

(b) bigPSF

(c) LSTM

Fig. 3. Histogram for the best explained prediction horizon for each set of predictions.

inside that real range. Results for the three models are presented in Tables 4–6. The *Real range* shows the minimum and maximum real values for each prediction horizon. The *Range covered* means the range of the data explained by each set of rules. The consequent of each rule explains a certain interval of the prediction horizon. Thus, the range of the data explained for a particular prediction horizon is calculated as the union of all the intervals of the set of rules obtained for that horizon. Then, the *Percentage covered* is the relation between these two amplitudes, calculated as the covered range divided by the total real range.

Information about ranges is also presented graphically in Fig. 3. These histograms represent the frequency of the real values of the time series. Red lines show the interval covered by the rules, while the yellow lines identify the intervals with more than 50% of the frequency of the total number of samples. They are generated in order to analyze the distribution of the actual values of the time series in different intervals. The *Hist. range* means the range of values with more than 50% of the samples, in other words, the range of values of time series X_t with more than 50% of the frequency in the histogram. Then, the percentage of range covered from histograms (*Hist. range covered (%)* column) is about the relation of this interval and the *Range covered* by

Table 1
A selection of QARs obtained by MOQAR and quality measures for each prediction horizon for WkNN predictions.

h	Rule	Support	Confidence	Gain
1	IF $X_{t-1} \in [19144, 28171] \Longrightarrow \hat{X}_{t+1} \in [19017.01, 28913.1]$	0.44	0.99	0.49
2	IF $X_{t-2} \in [24681, 31386] \Longrightarrow \hat{X}_{t+2} \in [23969.40, 32495,10]$	0.44	0.94	0.43
3	IF $X_{t-1} \in [23712, 31206] \Longrightarrow \hat{X}_{t+3} \in [23061.80, 31713.90]$	0.48	0.94	0.38
4	IF $X_{t} \in [24278, 29659] \Longrightarrow \hat{X}_{t+4} \in [22952.34, 31432.35]$	0.50	0.91	0.40
5	IF $X_{t-7} \in [21853, 34799]$ AND $X_{t-3} \in [24217, 31889] \Longrightarrow \hat{X}_{t+5} \in [23578.02, 33340.73]$	0.49	0.91	0.34
6	IF $X_{t-137} \in [24123, 34045]$ AND $X_{t} \in [24228, 33258] \Longrightarrow \hat{X}_{t+6} \in [23522.44, 34127.99]$	0.55	0.94	0.29
7	IF $X_{t-1} \in [19479, 29894] \Longrightarrow \hat{X}_{t+7} \in [20182.95, 319619.64]$	0.55	0.94	0.32
8	IF $X_{t-139} \in [20047, 35442]$ AND $X_{t-3} \in [26766, 35500]$ AND $X_{t-1} \in [28240, 36184] \Longrightarrow \hat{X}_{t+8} \in [25700.28, 35917.49]$	0.53	0.93	0.29
9	IF $X_{t-132} \in [29726, 35683] \Longrightarrow \hat{X}_{t+9} \in [27901.31, 36201.71]$	0.38	0.82	0.37
10	IF $X_{t-1} \in [17833, 27315] \Longrightarrow \hat{X}_{t+10} \in [19250.12, 28036.68]$	0.49	0.85	0.30
11	IF $X_{t-1} \in [17314, 27711] \Longrightarrow \hat{X}_{t+11} \in [19441.25, 29911.07]$	0.41	0.91	0.40
12	IF $X_{t-131} \in [19310, 28828]$ AND $X_{t-2} \in [17828, 29448] \Longrightarrow \hat{X}_{t+12} \in [19542.52, 27435.63]$	0.43	0.90	0.35
13	IF $X_{t-126} \in [26008, 39504]$ AND $X_{t-2} \in [2699, 36825] \Longrightarrow \hat{X}_{t+13} \in [26766.30, 36299.52]$	0.41	0.91	0.40
14	IF $X_{t-128} \in [26085, 36854] \Longrightarrow \hat{X}_{t+14} \in [25842.66, 37906.29]$	0.59	0.86	0.26
15	IF $X_{t-138} \in [27742, 35781] \Longrightarrow \hat{X}_{t+15} \in [24720.23, 35510.30]$	0.41	0.90	0.29
16	IF $X_{t-133} \in [29348, 35433] \Longrightarrow \hat{X}_{t+16} \in [25145.03, 35422.21]$	0.38	0.94	0.36
17	IF $X_{t-133} \in [23483, 39795]$ AND $X_{t-13} \in [25760, 42015] \Longrightarrow \hat{X}_{t+17} \in [26774.85, 40080.67]$	0.36	0.92	0.42
18	IF $X_{t-1} \in [29939, 37660] \Longrightarrow \hat{X}_{t+18} \in [28619.09, 39431.19]$	0.48	0.85	0.26
19	IF $X_{t-128} \in [18910, 29497] \Longrightarrow \hat{X}_{t+19} \in [19705.19, 29289.99]$	0.51	0.82	0.30
20	IF $X_{t-1} \in [17314, 26914] \Longrightarrow \hat{X}_{t+20} \in [18925.39, 29780.93]$	0.31	0.87	0.32
21	IF $X_{t-117} \in [17690, 27283] \Longrightarrow \hat{X}_{t+21} \in [18672.80, 29454.99]$	0.45	0.87	0.20
22	IF $X_{t-127} \in [24899, 34971] \Longrightarrow \hat{X}_{t+22} \in [23731.33, 35017.20]$	0.55	0.90	0.20
23	IF $X_{t-121} \in [30108, 38394] \Longrightarrow \hat{X}_{t+23} \in [27811.31, 38833.49]$	0.40	0.89	0.33
24	IF $X_{t-121} \in [27204, 37072] \Longrightarrow \hat{X}_{t+24} \in [26002.81, 37814.75]$	0.50	0.88	0.22

Table 2
A selection of QARs obtained by MOQAR and quality measures for each prediction horizon for bigPSF predictions.

h	Rule	Support	Confidence	Gain
1	IF $X_{t-3} \in [30257, 37092] \Longrightarrow \hat{X}_{t+1} \in [29453.74, 38204]$	0.42	0.98	0.29
2	IF $X_{t} \in [22620, 28163] \Longrightarrow \hat{X}_{t+2} \in [21481.78, 28856.24]$	0.29	0.92	0.51
3	IF $X_{t-2} \in [30543, 36093] \Longrightarrow \hat{X}_{t+3} \in [29769.89, 37712.91]$	0.29	0.91	0.49
4	IF $X_{t-1} \in [30442, 36810] \Longrightarrow \hat{X}_{t+4} \in [28625.66, 38039.09]$	0.33	0.97	0.47
5	IF $X_{t} \in [31609, 38062] \Longrightarrow \hat{X}_{t+5} \in [32062.35, 39097.34]$	0.25	0.88	0.58
6	IF $X_{t} \in [17580, 25676] \Longrightarrow \hat{X}_{t+6} \in [18689, 25649.63]$	0.27	0.92	0.58
7	IF $X_{t-142} \in [23692, 32399]$ AND $X_{t-135} \in [21140, 35781] \Longrightarrow \hat{X}_{t+7} \in [22308, 33175.12]$	0.32	0.93	0.45
8	IF $X_{t} \in [17979, 24963] \Longrightarrow \hat{X}_{t+8} \in [19057.73, 26007.90]$	0.23	0.93	0.57
9	IF $X_{t-135} \in [17328, 25184] \Longrightarrow \hat{X}_{t+9} \in [18313, 26017]$	0.26	0.84	0.48
10	IF $X_{t-3} \in [17470, 27274] \Longrightarrow \hat{X}_{t+10} \in [18761.80, 28424]$	0.32	0.85	0.38
11	IF $X_{t-136} \in [30946, 38959] \Longrightarrow \hat{X}_{t+11} \in [30906.26, 39571.81]$	0.31	0.79	0.36
12	IF $X_{t-137} \in [26772, 37134]$ AND $X_{t-133} \in [28909, 35795] \Longrightarrow \hat{X}_{t+12} \in [25732.37, 35733.31]$	0.37	0.92	0.38
13	IF $X_{t-134} \in [19183, 27744] \Longrightarrow \hat{X}_{t+13} \in [19079, 28437.19]$	0.37	0.83	0.37
14	IF $X_{t-133} \in [21014, 30336] \Longrightarrow \hat{X}_{t+14} \in [20218.29, 29710.86]$	0.39	0.78	0.28
15	IF $X_{t-131} \in [17753, 28053] \Longrightarrow \hat{X}_{t+15} \in [19140.54, 28240]$	0.30	0.74	0.29
16	IF $X_{t-7} \in [32758, 43561] \Longrightarrow \hat{X}_{t+16} \in [31543.33, 42054.65]$	0.19	0.76	0.40
17	IF $X_{t-1} \in [17701, 29841] \Longrightarrow \hat{X}_{t+17} \in [18916, 29526.07]$	0.45	0.81	0.88
18	IF $X_{t-128} \in [28894, 39700]$ AND $X_{t-126} \in [29426, 28979] \Longrightarrow \hat{X}_{t+18} \in [26997.84, 42409.79]$	0.43	0.94	0.34
19	IF $X_{t-123} \in [30900, 27691] \Longrightarrow \hat{X}_{t+19} \in [27691.11, 40420]$	0.33	0.92	0.36
20	IF $X_{t-122} \in [23533, 32592] \Longrightarrow \hat{X}_{t+20} \in [21582.51, 35484.23]$	0.55	0.96	0.11
21	IF $X_{t-126} \in [23401, 31375]$ AND $X_{t-125} \in [24196, 30367] \Longrightarrow \hat{X}_{t+21} \in [22382.49, 30945.97]$	0.32	0.83	0.29
22	IF $X_{t-124} \in [33308, 42534]$ AND $X_{t-1} \in [24675, 42736] \Longrightarrow \hat{X}_{t+22} \in [32400.65, 41603.20]$	0.16	0.84	0.58
23	IF $X_{t-119} \in [24894, 31978]$ AND $X_{t-6} \in [20090, 36256] \Longrightarrow \hat{X}_{t+23} \in [22874.77, 32427.15]$	0.37	0.89	0.28
24	IF $X_{t-124} \in [27764, 36510]$ AND $X_{t-118} \in [25175, 33831] \Longrightarrow \hat{X}_{t+24} \in [24446.19, 34432.56]$	0.37	0.93	0.28

the set of rules. With respect to the intervals defined by the maximum and minimum values of the real values of the time series, almost all the range (around 98%) of the prediction horizon is covered by the rules generated after applying the prediction model.

Overall, concerning the Tables mentioned above, each set of rules explains more than 95% on average for all the samples in the prediction horizon. Meaning that rules cover approx 95% of the total range of the data for each prediction horizon. However, when the range is defined by the histograms, the focus is on a reduction of the range where the majority number of samples are distributed. In that case it could be observed that the rules cover 100% of these intervals since the intervals are smaller.

The explainability reached by the three algorithms (WkNN, bigPSF and LSTM) could be compared using the information presented in this section.

In continuation of the comparison started in Section 4.2.1, some methods obtain better results than others. The percentage of real data range covered by the set of rules of bigPSF (Table 5) is lower than WkNN and LSTM data. Both WkNN and LSTM have a similar level of range covered by the generated QARs. However, the whole set of rules for the three models cover more than 90% of the real range of the attributes.

This section presents the QARs that have been obtained using a mining rules algorithm, their quality measures and the range of real data that rules are covering. These rules are then used as a way of *explaining* the ML and DL models' predictions.

Explainable results are presented in the following Section 4.3.

4.3. Visual explanations

The main goal of this paper is a rules-based approach for adding explainability to time series forecasting models. Then, QARs have been selected because they are widely known for being highly interpretable. However, due to the nature of the time series data, the information contained in the rules is better shown using graphical representations.

Table 3
A selection of QARs obtained by MOQAR and quality measures for each prediction horizon for LSTM predictions.

h	Rule	Support	Confidence	Gain
1	IF $X_{t-1} \in [25122, 32611]$ AND $X_t \in [25618, 33561] \Longrightarrow \hat{X}_{t+1} \in [25230.04, 33063.40]$	0.42	0.98	0.52
2	IF $X_{t-2} \in [29810, 39018] \Longrightarrow \hat{X}_{t+2} \in [28149.37, 38952.71]$	0.45	0.98	0.44
3	IF $X_t \in [26833, 32630] \Longrightarrow \hat{X}_{t+3} \in [25947.24, 33570.65]$	0.35	0.98	0.52
4	IF $X_{t-114} \in [23085, 39275]$ AND $X_t \in [26786, 36046] \Longrightarrow \hat{X}_{t+4} \in [26556.69, 36436.02]$	0.46	0.97	0.41
5	IF $X_t \in [29971, 38813] \Longrightarrow \hat{X}_{t+5} \in [28089.77, 38531.58]$	0.43	0.97	0.43
6	IF $X_{t-2} \in [22237, 29724]$ AND $X_{t-1} \in [22812, 29042] \Longrightarrow \hat{X}_{t+6} \in [22059.42, 30741.35]$	0.32	0.93	0.46
7	IF $X_t \in [23435, 29565] \Longrightarrow \hat{X}_{t+7} \in [23286.75, 31999.55]$	0.32	0.93	0.45
8	IF $X_{t-3} \in [19503, 29267] \Longrightarrow \hat{X}_{t+8} \in [19866.40, 31687.74]$	0.44	0.94	0.34
9	IF $X_{t-4} \in [21948, 33583] \Longrightarrow \hat{X}_{t+9} \in [21678.36, 33638.91]$	0.62	0.89	0.20
10	IF $X_{t-129} \in [29781, 40611]$ AND $X_{t-1} \in [28913, 42075] \Longrightarrow \hat{X}_{t+10} \in [29408.60, 40850.50]$	0.37	0.98	0.47
11	IF $X_{t-1} \in [17846, 26748] \Longrightarrow \hat{X}_{t+11} \in [18447.88, 29838.03]$	0.31	0.97	0.46
12	IF $X_{t-1} \in [20486, 28110] \Longrightarrow \hat{X}_{t+12} \in [20873.39, 31424.08]$	0.36	0.96	0.41
13	IF $X_{t-134} \in [29047, 35642]$ AND $X_{t-8} \in [19893, 35062] \Longrightarrow \hat{X}_{t+13} \in [25787.62, 36114.02]$	0.32	0.96	0.37
14	IF $X_{t-3} \in [19330, 29017] \Longrightarrow \hat{X}_{t+14} \in [19459.90, 31157.35]$	0.38	0.84	0.32
15	IF $X_{t-130} \in [19709, 28064] \Longrightarrow \hat{X}_{t+15} \in [19733.12, 32743.51]$	0.38	0.89	0.27
16	IF $X_{t-133} \in [19523, 28556]$ AND $X_{t-85} \in [25115, 41521] \Longrightarrow \hat{X}_{t+16} \in [19602.96, 30048.36]$	0.31	0.90	0.41
17	IF $X_{t-133} \in [29176, 39694]$ AND $X_{t-128} \in [28498, 37063] \Longrightarrow \hat{X}_{t+17} \in [26667.44, 38203.40]$	0.42	0.96	0.33
18	IF $X_{t-121} \in [26852, 36382]$ AND $X_{t-115} \in [24036, 36978] \Longrightarrow \hat{X}_{t+18} \in [26940.19, 37320.54]$	0.45	0.81	0.20
19	IF $X_t \in [31778, 41140] \Longrightarrow \hat{X}_{t+19} \in [28402.35, 42863.99]$	0.32	0.91	0.34
20	IF $X_{t-125} \in [29314, 38134]$ AND $X_{t-121} \in [30329, 37335] \Longrightarrow \hat{X}_{t+20} \in [24542.16, 39102.50]$	0.36	0.98	0.20
21	IF $X_{t-124} \in [28495, 35152] \Longrightarrow \hat{X}_{t+21} \in [26016.44, 35727.54]$	0.38	0.89	0.29
22	IF $X_{t-113} \in [18036, 26626]$ AND $X_{t-103} \in [18916, 30240] \Longrightarrow \hat{X}_{t+22} \in [18987.06, 31309.65]$	0.32	0.89	0.29
23	IF $X_{t-109} \in [18628, 27189] \Longrightarrow \hat{X}_{t+23} \in [18326.44, 30991.96]$	0.35	0.87	0.28
24	IF $X_{t-122} \in [31268, 41236] \Longrightarrow \hat{X}_{t+24} \in [29353.53, 42480.26]$	0.34	0.88	0.37

Table 4
WkNN real data explained by the set of rules obtained for each prediction horizon.

Variable	Real range	Range covered	Percentage covered	Hist. range (+50%)	Hist. range covered (%)
X_{t+1}	[17353.74, 43439.52]	[18349.31, 43078.56]	94.80	[20052.2, 35863.4]	100.00
X_{t+2}	[17378.03, 42912.84]	[18515.72, 42905.94]	95.52	[19833.8, 35774.6]	100.00
X_{t+3}	[17405.38, 43056.83]	[18642.72, 42994.83]	94.93	[19941.2, 35776.4]	100.00
X_{t+4}	[17830.17, 42994.40]	[18873.59, 42860.35]	95.32	[19787.6, 35511.2]	100.00
X_{t+5}	[17266.97, 42924.54]	[18970.15, 42521.59]	91.80	[19466.6, 35322.2]	100.00
X_{t+6}	[18016.69, 42764.45]	[19014.41, 42557.55]	95.13	[19679.1, 35519.7]	100.00
X_{t+7}	[17778.78, 42686.84]	[18684.15, 42429.51]	95.33	[19824.0, 35424.0]	100.00
X_{t+8}	[17616.69, 42621.16]	[18910.71, 42642.87]	94.91	[19685.3, 35491.1]	100.00
X_{t+9}	[18148.24, 41955.98]	[19189.13, 42028.16]	95.93	[19906.1, 35374.7]	100.00
X_{t+10}	[18050.52, 41629.38]	[18992.12, 42119.70]	98.10	[19852.4, 35250.8]	100.00
X_{t+11}	[18087.15, 41318.29]	[18930.70, 41506.91]	97.18	[22402.8, 35139.8]	100.00
X_{t+12}	[18594.38, 41742.89]	[18876.67, 40992.48]	95.54	[22467.0, 35002.0]	100.00
X_{t+13}	[18268.31, 42541.82]	[19082.54, 41193.65]	91.10	[20052.2, 35305.4]	100.00
X_{t+14}	[18701.77, 43475.79]	[19095.86, 41315.21]	89.69	[20397.4, 35603.8]	100.00
X_{t+15}	[19180.59, 43548.42]	[19134.17, 42318.47]	95.14	[20457.0, 35853.0]	100.00
X_{t+16}	[18687.10, 44098.17]	[19032.63, 41962.34]	90.24	[20426.5, 36035.5]	100.00
X_{t+17}	[18656.08, 44227.79]	[19197.50, 43026.00]	93.18	[20533.1, 36175.7]	100.00
X_{t+18}	[18762.39, 44111.72]	[18529.88, 43278.63]	97.63	[20537.0, 36077.0]	100.00
X_{t+19}	[18303.98, 43753.61]	[18845.82, 43181.25]	95.62	[20610.0, 36288.0]	100.00
X_{t+20}	[18202.11, 43733.12]	[18510.75, 43103.91]	96.33	[20476.3, 36186.1]	100.00
X_{t+21}	[17908.19, 43593.15]	[18267.67, 42713.28]	95.18	[20128.7, 36080.9]	100.00
X_{t+22}	[18071.85, 43538.63]	[18081.27, 43309.40]	99.10	[20253.0, 36009.0]	100.00
X_{t+23}	[17722.50, 43503.10]	[18805.80, 43111.79]	94.28	[19968.9, 35898.3]	100.00
X_{t+24}	[18004.91, 43437.71]	[18392.98, 43140.33]	97.30	[20218.3, 36048.1]	100.00

Different models have been used for predicting 24 future values (4 h) using the time window as an input (168 past values, 1 day and 4 horus). For each of the 24 prediction horizons, a set of QAR is generated. QARs are commented in Section 4.2 and QARs are shown in Tables 1–3. A rule is composed by the antecedent (*if* clause) and the consequent (commonly *then* clause), after the arrow. Using the information that is stored in the consequence of the rules, a graphical representation is presented. The methodology used was previously presented in the appropriate Section 3.

Visual representations explaining the three models and a LIME baseline used for testing this methodology are shown in Fig. 4. Here heatmaps of the influence of each variable X_{t-i} for predicting the 24 horizons are presented. The 24 prediction horizons are labeling Y axe meanwhile the 168 features used as input are shown in X axe. The color code is giving purple for the less important variables and hot colors for the most frequent ones in the QAR's antecedent. Fig. 5 represents the importance of the attributes for all the 24 prediction horizons. The lineal representation is calculated by summing the incidence of each attribute.

Regarding both Figs. 4 and 5, it could be seen that the most important items are the last items, that is, the most recent time series values. In Fig. 4 it could also be seen a group of important values. The group that flows between X_{t-148} and X_{t-128} approximately. For the first predicted value X_{t+1}, on the top of the heatmap, it is about X_{t-148}. As 24 h is the same as 144 items in the time series data, that means exact the 24 before. For the last one time series value, X_{t+24}, the exact 24 h before is in X_{t-128}. This hot-colors pattern moves as the horizons are moving forward in time from X_{t+1} to X_{t+24}. This 'line' of attributes has a bigger impact in WkNN and bigPSF predictions, whereas slightly less in LSTM. The 'line' means that the elements of the day before are also important for calculating the predictions.

Then, regarding the baseline in (d) in both Figs. 4 and 5, LIME experiences difficulties to obtain a pattern in feature's importance. In Fig. 4, the heatmap is more or less completed colorized whereas the

Table 5
BigPSF real data explained by the set of rules obtained for each prediction horizon.

Variable	Real range	Range covered	Percentage covered	Hist. range (+50%)	Hist. range covered (%)
X_{t+1}	[18146.0, 43686.0]	[18434.764, 42514.098]	94.28	[20052.2, 35863.40]	100.00
X_{t+2}	[18385.68, 43586.0]	[18523.0, 42856.77]	96.56	[19833.8, 35774.60]	100.00
X_{t+3}	[18363.15, 43694.0]	[18543.91, 42666.41]	95.23	[19941.2, 35776.40]	100.00
X_{t+4}	[18288.97, 43373.0]	[18507.0, 42628.01]	96.16	[19787.6, 35511.2]	100.00
X_{t+5}	[18308.21, 43200.0]	[18454.0, 42574.99]	96.90	[19466.6, 37964.8]	100.00
X_{t+6}	[18215.0, 42997.0]	[18395.0, 42418.85]	96.95	[19679.1, 35519.7]	100.00
X_{t+7}	[18176.0, 42583.0]	[18417.93, 42117.74]	97.01	[19824.0, 35424.0]	100.00
X_{t+8}	[18291.66, 42983.0]	[18291.655, 42983.0]	100.00	[19685.30, 35491.10]	100.00
X_{t+9}	[18313.0, 42692.0]	[18313.0, 42290.33]	98.35	[19906.1, 35374.7]	100.00
X_{t+10}	[18247.0, 42208.0]	[18622.0, 41743.97]	96.50	[19852.4, 35250.8]	100.00
X_{t+11}	[18263.0, 41966.0]	[18263.0, 41530.38]	98.16	[22402.8, 35139.8]	100.00
X_{t+12}	[18195.0, 41579.52]	[18195.0, 41260.55]	98.64	[22467.0, 35002.0]	100.00
X_{t+13}	[18270.00, 41817.18]	[18500.0, 41008.49]	95.56	[20052.20, 35305.40]	100.00
X_{t+14}	[18370.00, 42434.06]	[18370.0, 42434.05]	97.63	[20397.4, 35603.8]	100.00
X_{t+15}	[18387.0, 42651.57]	[18627.0, 41713.47]	95.14	[20457.0, 35853.0]	100.00
X_{t+16}	[18414.0, 42922.93]	[18980.0, 42054.65]	94.15	[20426.5, 36035.5]	100.00
X_{t+17}	[18493.0, 43015.96]	[18916.0, 42288.44]	95.31	[20533.1, 36175.7]	100.00
X_{t+18}	[18528.26, 43089.86]	[19023.82, 42409.76]	95.21	[20537.0, 36077.0]	100.00
X_{t+19}	[18238.58, 43124.62]	[18652.38, 42137.85]	94.37	[20610.0, 36288.0]	100.00
X_{t+20}	[18319.21, 43323.32]	[18637.0, 43197.42]	98.23	[20476.30, 36186.10]	100.00
X_{t+21}	[18022.0, 43433.92]	[18272.0, 43284.04]	98.43	[20128.7, 36080.9]	100.00
X_{t+22}	[17860.0, 43486.55]	[18661.0, 42910.5]	94.63	[20253.0, 36009.0]	100.00
X_{t+23}	[17955.0, 43439.34]	[18231.32, 42939.35]	96.95	[19968.9, 35898.3]	100.00
X_{t+24}	[17894.0, 43540.49]	[18635.0, 43298.0]	96.17	[20218.3, 36048.1]	100.00

Table 6
LSTM real data explained by the set of rules obtained for each prediction horizon.

Variable	Real range	Range covered	Percentage covered	Hist. range (+50%)	Hist. range covered (%)
X_{t+1}	[17353.74, 43439.52]	[17591.27, 43439.52]	99.09	[20581.6, 36647.2]	100.00
X_{t+2}	[17378.03, 42912.84]	[17378.03, 42814.29]	99.61	[20591.8, 36340.6]	100.00
X_{t+3}	[17405.38, 43056.83]	[17643.35, 42960.25]	98.70	[20648.4, 36304.8]	100.00
X_{t+4}	[17830.17, 42994.40]	[17991.74, 42894.74]	98.96	[20715.9, 38915.2]	100.00
X_{t+5}	[17266.97, 42924.54]	[17403.38, 42800.99]	98.99	[20713.8, 38870.4]	100.00
X_{t+6}	[18016.69, 42764.45]	[18280.22, 42599.07]	98.27	[20637.9, 38627.2]	100.00
X_{t+7}	[17778.78, 42686.84]	[17921.82, 42483.08]	98.61	[20569.0, 38433.0]	100.00
X_{t+8}	[17616.69, 42621.16]	[17930.54, 42489.67]	98.22	[20673.3, 38567.4]	100.00
X_{t+9}	[18148.24, 41955.98]	[18296.10, 41799.98]	98.72	[20680.8, 38284.4]	100.00
X_{t+10}	[18050.52, 41629.38]	[18050.52, 41568.63]	99.74	[20558.6, 38069.8]	100.00
X_{t+11}	[18087.15, 41318.29]	[18087.15, 40789.47]	97.72	[20573.6, 37846.8]	100.00
X_{t+12}	[18594.38, 41742.89]	[18968.43, 41425.71]	97.01	[20591.1, 37993.8]	100.00
X_{t+13}	[18268.31, 42541.82]	[18377.28, 42366.73]	98.83	[20780.2, 36087.4]	100.00
X_{t+14}	[18701.77, 43475.79]	[18809.41, 43022.84]	97.74	[20880.6, 36448.2]	100.00
X_{t+15}	[19180.59, 43548.42]	[19184.77, 43487.88]	99.73	[20896.0, 36652.0]	100.00
X_{t+16}	[18687.10, 44098.17]	[19062.02, 44098.17]	98.52	[20970.9, 36984.3]	100.00
X_{t+17}	[18656.08, 44227.79]	[19097.35, 43654.17]	96.03	[20838.3, 37130.1]	100.00
X_{t+18}	[18762.39, 44111.72]	[19040.67, 43545.99]	96.67	[20689.9, 37093.3]	100.00
X_{t+19}	[18303.98, 43753.61]	[18623.58, 43156.52]	96.40	[20364.5, 36837.5]	100.00
X_{t+20}	[18202.11, 43733.12]	[18346.94, 43668.80]	99.18	[20426.5, 36755.5]	100.00
X_{t+21}	[17908.19, 43593.15]	[18207.39, 43384.63]	98.02	[20454.9, 36762.3]	100.00
X_{t+22}	[18071.85, 43538.63]	[18099.44, 43538.63]	99.89	[20513.4, 36763.8]	100.00
X_{t+23}	[17722.50, 43503.10]	[17935.50, 43223.04]	98.09	[20415.8, 36704.6]	100.00
X_{t+24}	[18004.91, 43437.71]	[18197.69, 43344.63]	98.88	[20479.6, 36635.2]	100.00

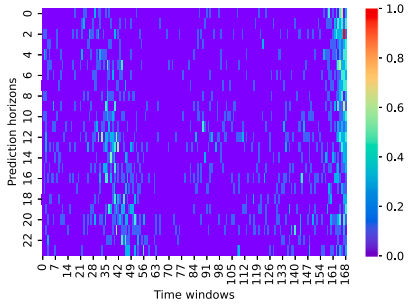
lineal representation in Fig. 5 is chaotic. Comparing the results with the baseline leads us to think that LIME is not really accurate with regard to time series data.

Finally, concrete examples of local explanations are also presented in Fig. 6. A random element of the input data have been chosen and graphically represented. The same element is represented for the three models (a), (b), (c) and the LIME baseline (d). The color code is obtained from the matrix previously used for the heatmap in Fig. 4. Thus, the purple points are less important in order to get prediction whereas the blue, green, yellow and red ones have an increasing importance. It could be seen again that the 24 h before and the most recent items are in hot colors, that is, most important. The LIME baseline shows the time series example completely colorized.

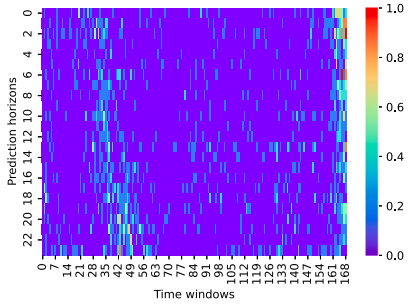
The conclusions that could be extracted for these representations are: the most important items to predict the present moment are the same moment the day before and the moment that have just happened. In Fig. 4, this importance is seen in green or light blue whereas less important values are in purple. In Fig. 5 the importance is seen in the top values of the lineal representation. Comparing the results of the methodology applied to the three methods and the LIME baseline in Figures 4–6, it could be observed that LIME does not find a pattern. LIME is chaotic when explains time series forecasting. By contrast, the proposed methodology based on QARs traces a clear pattern, providing more interpretable results. Previously discussed in this section, the pattern is that the most important features are the recent moments and the moment 24 before.

5. Conclusions

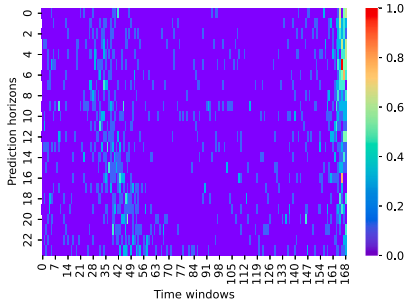
Achieving explainable models, high accurate but also transparent to the user, is a promising research scope in artificial intelligence and computer science nowadays. In that way, post-hoc explainability techniques are widely used as they are not generating new models but adding explainable layers or visual and local explanations. These



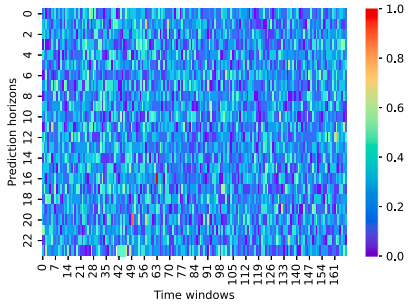
(a) WkNN



(b) bigPSF

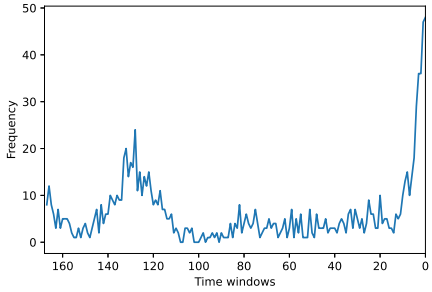


(c) LSTM

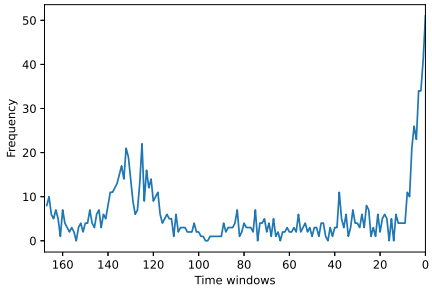


(d) LIME

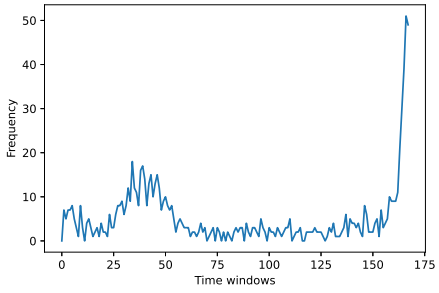
Fig. 4. Heatmap showing the importance of each time window item for each of the 24 prediction horizons.



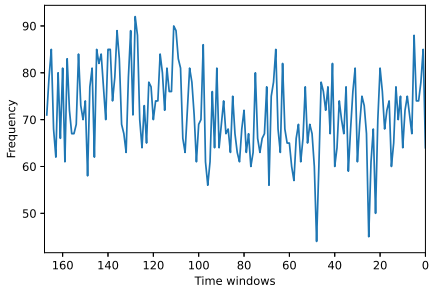
(a) WkNN



(b) bigPSF



(c) LSTM

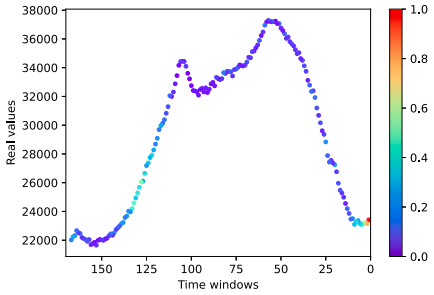


(d) LIME

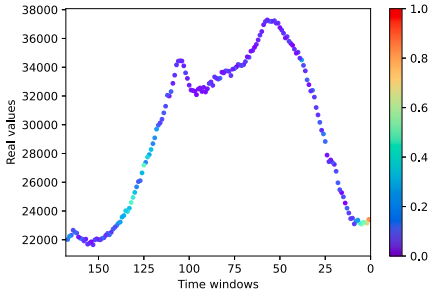
Fig. 5. Lineal graph showing the importance of each time window item for obtaining all the predictions.

explainability techniques are really useful as they are independent of the model. Post-hoc models could be used to explain the forecasts in a wide range of scopes. However, performance issues are found when general techniques as LIME are applied to big data or time series forecasting problems.

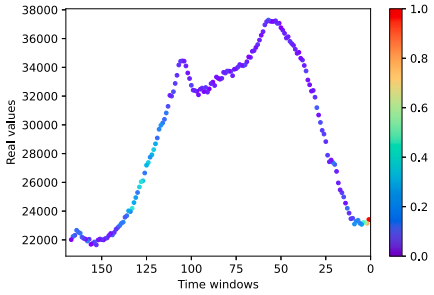
Here, a novel methodology specially designed to increase interpretability of time series forecasting is proposed. Visual representations have been used to explain how different predictive models (both deep learning and machine learning) are obtaining their forecasts. The main idea is to compute the importance of the input values in order to generate the predictions.



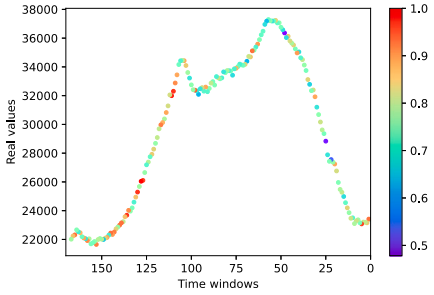
(a) WkNN



(b) bigPSF



(c) LSTM



(d) LIME

Fig. 6. A random instance colorized depending on the importance of the time window.

The initial idea is based on the well-known ability of association rules to predict and being interpretable themselves. Methodology is tested using a real-world time series data on electrical consumption in Spain, and three different predictive models. Trained and already tested models have been used to obtain predictions. As a result, three datasets with real data and predicted values are generated.

Then, QARs have been obtained using the evolutionary algorithm MOQAR. Rule's antecedent is formed by the input, that is, the past values of the time series, and whose consequent is formed by the output, i.e. the predictions obtained by the models.

The key layer of the methodology uses QARs to obtain explainable visual representations. Visual representations clarify and make the results *interpretable*. The diagrams are generated using the information of the QARs. The higher frequency of appearance in the antecedent of the rule, the more impact on predicting future values. That fact creates a color code. The color code shows graphically the importance of each variable or timestamp in making predictions.

In general, the explanations extracted after seeing the heatmap images and the local examples in Figs. 4 and 6 are logical concerning the kind of data that have been used: past values timestamp is 1 day and 4 h and future values are 4 h. The methodology finds that the most important items to predict are the present moment are the same moment the day before (24 h before) and the recent events.

Overall, the results obtained show that useful and interesting and interpretable visual representations could be generated from QARs. These diagrams could explain how predictions are obtained and which attributes have a great importance in time series forecasting.

This is an initial approach to this methodology applications. Future lines of research will continue in several ways. For instance, testing the methodology with different time series datasets and with more DL models. The objective is validate the methodology in wider scopes. In addition, comparisons with other model agnostic explainable techniques as is done here with LIME. The methodology could be extended, and more graphical representations could be generated.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank the Spanish Ministry of Science and Innovation and the Junta de Andalucía, Spain for their support within the projects PID2020-117954RB-C21 and TED2021-131311B-C22, PY20-00870 and UPO-138516, respectively.

References

[1] G. Yang, Q. Ye, J. Xia, Unbox the black-box for the medical explainable AI via multi-modal and multi-centre data fusion: A mini-review two showcases and beyond, Inf. Fusion 77 (2022) 29–52.
[2] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence 267 (2019) 1–38.
[3] A. Barredo-Arrieta, N. Díaz-Rodríguez, J. del Ser, et al., Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115.
[4] T. Rojat, R. Puget, D. Filliat, J. Del Ser, R. Gelin, N. Díaz-Rodríguez, Explainable Artificial Intelligence (XAI) on time series data: A survey, 2021, pp. 1–14, arXiv:2104.00950.
[5] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models, Nat. Mach. Intell. 1 (2018) 206–215.
[6] A. Abanda, Contributions to Time Series Classification: Meta-Learning and Explainability (Ph.D. thesis), University of the Basque Country, 2021.
[7] D. Doran, S. Schulz, T.R. Besold, What does explainable AI really mean? A new conceptualization of perspectives, 2017, arXiv preprint arXiv:1710.00794.
[8] J. Brownlee, Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python, in: Machine Learning Mastery, 2018.
[9] J.F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, A. Troncoso, Deep learning for time series forecasting: A survey, Big Data 9 (1) (2021) 3–21.

[10] V. Arya, R.K.E. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S.C. Hoffman, S. Houde, Q.V. Liao, R. Luss, A. Mojsilović, et al., One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques, 2019, arXiv preprint arXiv:1909.03012.

[11] D.V. Carvalho, E.M. Pereira, J.S. Cardoso, Machine learning interpretability: A survey on methods and metrics, Electronics 8 (8) (2019) 832.

[12] D. Minh, H.X. Wang, Y.F. Li, T.N. Nguyen, Explainable artificial intelligence: a comprehensive review, Artif. Intell. Rev. (2021) 1–66.

[13] A. Barredo Arrieta, S. Gil-Lopez, I. Laña, M.N. Bilbao, J. Del Ser, On the post-hoc explainability of deep echo state networks for time series forecasting, image and video classification, Neural Comput. Appl. 34 (13) (2022) 10257–10277.

[14] A. Preece, Asking 'Why' in AI: Explainability of intelligent systems–perspectives and challenges, Int. J. Intell. Syst. Account. Financ. Manage. 25 (2) (2018) 63–72.

[15] G. Vilone, L. Longo, Notions of explainability and evaluation approaches for explainable artificial intelligence, Inf. Fusion 76 (2021) 89–106.

[16] M.R. Zafar, N. Khan, Deterministic local interpretable model-agnostic explanations for stable explainability, Mach. Learn. Knowl. Extr. 3 (3) (2021) 525–541.

[17] S. Mishra, S. Dutta, J. Long, D. Magazzeni, A survey on the robustness of feature importance and counterfactual explanations, arXiv preprint arXiv:2111.00358.

[18] I. Palatnik de Sousa, M. Rebuzzi Vellasco, E. Costa da Silva, Local interpretable model-agnostic explanations for classification of lymph node metastases, Sensors 19 (13) (2019) 2969.

[19] J. Dieber, S. Kirrane, A novel model usability evaluation framework (MUsE) for explainable artificial intelligence, Inf. Fusion 81 (2022) 143–153.

[20] A.R. Troncoso-García, M. Martínez-Ballesteros, F. Martínez-Álvarez, A. Troncoso, Explainable machine learning for sleep apnea prediction, Procedia Comput. Sci. 207 (2022) 2930–2939.

[21] Y. Nohara, K. Matsumoto, H. Soejima, N. Nakashima, Explanation of machine learning models using improved Shapley additive explanation, in: Proceedings of the ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, 2019, pp. 546–546.

[22] M.K. al Bashiti, M.Z. Naser, Verifying domain knowledge and theories on fire-induced spalling of concrete through explainable artificial intelligence, Constr. Build. Mater. 348 (2022) 128648.

[23] J.A. Gallardo-Gómez, F. Divina, A. Troncoso, F. Martínez-Álvarez, Explainable artificial intelligence for the electric vehicle load demand forecasting problem, in: Proceedings of the International Workshop on Soft Computing Models in Industrial and Environmental Applications, 2023, pp. 413–422.

[24] J.M. Rožanec, B. Fortuna, D. Mladenić, Knowledge graph-based rich and confidentiality preserving Explainable Artificial Intelligence (XAI), Inf. Fusion 81 (2022) 91–102.

[25] F. Hohman, H. Park, C. Robinson, D.H.P. Chau, Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations, IEEE Trans. Vis. Comput. Graphics 26 (1) (2019) 1096–1106.

[26] A. Holzinger, B. Malle, A. Saranti, B. Pfeifer, Towards multi-modal causability with graph neural networks enabling information fusion for explainable AI, Inf. Fusion 71 (2021) 28–37.

[27] A. Atutxa, A. de Ilarraza, K. Gojenola, M. Oronoz, O. Perez-de Viñaspre, Interpretable deep learning to map diagnostic texts to ICD-10 codes, Int. J. Med. Inform. 129 (2019) 49–59.

[28] L. Arras, A. Osman, W. Samek, CLEVR-XAI: a benchmark dataset for the ground truth evaluation of neural network explanations, Inf. Fusion 81 (2022) 14–40.

[29] J. Zhu, A. Liapis, S. Risi, R. Bidarra, G.M. Youngblood, Explainable AI for designers: A human-centered perspective on mixed-initiative co-creation, in: Proceedings of the IEEE Conference on Computational Intelligence and Games, 2018, pp. 1–8.

[30] Q. Zhao, S.S. Bhowmick, Association Rule Mining: A Survey, Nanyang Technological University, Singapore, 2003, p. 135.

[31] B. Letham, C. Rudin, T.H. McCormick, D. Madigan, Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model, Ann. Appl. Stat. 9 (3) (2015) 1350–1371.

[32] D. Thi, P. Meysman, K. Laukens, MoMAC: Multi-objective optimization to combine multiple association rules into an interpretable classification, Appl. Intell. 52 (2022) 3090–3102.

[33] S. Nemet, D. Kukolj, G. Ostojić, et al., Aggregation framework for TSK fuzzy and association rules: interpretability improvement on a traffic accidents case, Appl. Intell. 49 (11) (2019) 3909–3922.

[34] M.T. Ribeiro, S. Singh, C. Guestrin, Anchors: High-precision model-agnostic explanations, Proc. AAAI Conf. Artif. Intell. 32 (1) (2018) 1–9.

[35] A. Anguita-Ruiz, A. Segura-Delgado, R. Alcalá, C.M. Aguilera, J. Alcalá-Fdez, eXplainable Artificial Intelligence (XAI) for the identification of biologically relevant gene expression patterns in longitudinal human studies, insights from obesity research, PLoS Comput. Biol. 16 (4) (2020) e1007792.

[36] B. Mahbooba, M. Timilsina, R. Sahal, M. Serrano, Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model, Complexity 2021 (2021).

[37] D. Rajapaksha, C. Bergmeir, W. Buntine, LoRMIkA: Local rule-based model interpretability with k-optimal associations, Inform. Sci. 540 (2020) 221–241.

[38] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (5) (2018) 1–42.

[39] M. Martínez Ballesteros, A. Troncoso, F. Martínez-Álvarez, J.C. Riquelme, Improving a multi-objective evolutionary algorithm to discover quantitative association rules, Knowl. Inf. Syst. 49 (2016) 11.

[40] F. Moleshi, A. Haeri, F. Martínez-Álvarez, A novel hybrid GA–PSO framework for mining quantitative association rules, Soft Comput. 24 (2020) 4645–4666.

[41] P. Jiménez-Herrera, L. Melgar-García, G. Asencio-Cortés, A. Troncoso, Streaming big time series forecasting based on nearest similar patterns with application to energy consumption, Log. J. IGPL (2022) 1–16.

[42] L. Melgar-García, D. Gutiérrez-Avilés, C. Rubio-Escudero, A. Troncoso, Nearest neighbors-based forecasting for electricity demand time series in streaming, in: Proceedings of the Conference of the Spanish Association for Artificial Intelligence, 2021, pp. 185–195.

[43] R. Pérez-Chacón, G. Asencio-Cortés, F. Martínez-Álvarez, A. Troncoso, Big data time series forecasting based on pattern sequence similarity and its application to the electricity demand, Inform. Sci. 540 (2020) 160–174.

[44] R. Talavera, R. Pérez-Chacón, M. Martínez-Ballesteros, A. Troncoso, F. Martínez-Álvarez, A Nearest Neighbours-Based Algorithm for Big Time Series Data Forecasting, in: Lecture Notes in Computer Science, vol. 5391, 2016, pp. 674–679.

[45] R. Talavera-Llames, R. Pérez-Chacón, A. Troncoso, F. Martínez-Álvarez, Mv-kwnn: A novel multivariate and multi-output weighted nearest neighbors algorithm for big data time series forecasting, Neurocomputing 353 (2019) 56–73.

[46] J.F. Torres, M.J. Jiménez-Navarro, F. Martínez-Álvarez, A. Troncoso, Electricity consumption time series forecasting using temporal convolutional networks, in: Proceedings of the Conference of the Spanish Association for Artificial Intelligence, 2021, pp. 216–225.

[47] J.F. Torres, F. Martínez-Álvarez, A. Troncoso, A deep LSTM network for the spanish electricity consumption forecasting, Neural Comput. Appl. 34 (2022) 10533–10545.

[48] N. Bokde, G. Asencio-Cortés, F. Martínez-Álvarez, K. Kulat, PSF: Introduction to R package for pattern sequence based forecasting algorithm, R J. 9 (1) (2017) 324–333.

[49] F. Martínez-Álvarez, A. Troncoso, J.C. Riquelme, J.S. Aguilar-Ruiz, Energy time series forecasting based on pattern sequence similarity, IEEE Trans. Knowl. Data Eng. 23 (8) (2011) 1230–1243.

[50] F. Martínez-Álvarez, A. Schmutz, G. Asencio-Cortés, J. Jacques, A novel hybrid algorithm to forecast functional time series based on pattern sequence similarity with application to electricity demand, Energies 12 (1) (2018) 94.