# Explaining Learned Patterns in Deep Learning by Association Rules Mining

M. J. Jiménez-Navarro[1(✉)], M. Martínez-Ballesteros[1], F. Martínez-Álvarez[2], and G. Asencio-Cortés[2]

[1] Department of Computer Languages and Systems, University of Seville, Seville, Spain
{mjimenez3,mariamartinez}@us.es
[2] Data Science and Big Data Lab, Pablo de Olavide University, 41013, Seville, Spain
{fmaralv,guaasecor}@upo.es

**Abstract.** This paper proposes a novel approach that combines an association rule algorithm with a deep learning model to enhance the interpretability of prediction outcomes. The study aims to gain insights into the patterns that were learned correctly or incorrectly by the model. To identify these scenarios, an association rule algorithm is applied to extract the patterns learned by the deep learning model. The rules are then analyzed and classified based on specific metrics to draw conclusions about the behavior of the model. We applied this approach to a well-known dataset in various scenarios, such as underfitting and overfitting. The results demonstrate that the combination of the two techniques is highly effective in identifying the patterns learned by the model and analyzing its performance in different scenarios, through error analysis. We suggest that this methodology can enhance the transparency and interpretability of black-box models, thus improving their reliability for real-world applications.

**Keywords:** association rules · Apriori · deep learning · interpretability · explainable AI

## 1 Introduction

Deep learning has become a popular tool for solving complex problems in various domains, such as finance, healthcare, and engineering. However, their lack of interpretability and black-box nature pose significant challenges to trust and understanding their predictions. For example, a model that predicts a certain disease in medical diagnosis without providing a clear explanation for its prediction may not be trusted by physicians or patients, even if the model has high accuracy.

To address this issue, explainability approaches have been developed, such as global/local model-agnostic methods [18] and example-based methods. However, these approaches can be challenging to understand for non-experts in the field

https://doi.org/10.1007/978-3-031-42536-3_13

[9] and may be too general or too specific to identify particular scenarios where the model performed poorly or behaved like an outlier.

Association rules (AR) [2] are a powerful tool for enhancing interpretability in machine learning by identifying meaningful relationships between variables [16].

In this paper, we propose a model-agnostic approach that combines an AR mining with a deep learning model to enhance the interpretability of its predictions. By using the AR algorithm to extract the patterns learned by the deep learning model, the behavior of the model can be better understood using an intuitive cause-and-effect structure similar to a decision tree. Additionally, the rules identify generalizable scenarios without relying on global explanations. This approach can enhance trust and reliability in model predictions.

To demonstrate the effectiveness of our proposed methodology, we applied it to various scenarios within a well-known dataset using the Apriori [2] algorithm to discover AR. The results show that the AR algorithm is an effective and simple approach to identifying the learned patterns and analyzing the performance of the model. Thus, this approach has the potential to enhance the transparency and interpretability of black-box models across various domains, making them more reliable for real-world applications.

The main contributions of this paper can be summarized as follows:

- Development of a methodology to evaluate the behavior of black-box models.
- A simple and easy-to-understand methodology that can identify the strengths and weaknesses of a model.
- Analysis of several models in various scenarios, including overfitting and underfitting, on a well-known dataset.

The remainder of this paper is structured as follows. Section 2 discusses related work that focuses on interpretability approaches. In Sect. 3, we provide a detailed overview of the proposed methodology. Section 4.1 presents the experimental setting for different scenarios. In Sect. 4, we present the results and analysis of the proposed methodology applied to a public dataset. Finally, Sect. 5 summarizes the main conclusions of this work.

## 2 Related Works

Numerous studies have been conducted to understand the behavior learned by models, particularly in the context of deep learning due to its importance and black-box nature.

Some studies use a heatmap approach [10,12,17] to explain the behavior learned by the model in image classification by illustrating the areas where the model focused to make its prediction. However, these methods require expert knowledge to understand the learned behavior and are not applicable to tabular data, which is the scope of this work.

Other approaches use feature attribution methodologies, such as SHAP [14], to assign a numerical value that represents the importance of a feature [1,3,8].

However, these methods rely on expert knowledge to analyze feature attribution and provide overly general explanations.

Finally, rule-based methods have been proposed for various applications due to their simplicity [15,19]. Bernardi et al. [5] use a non-parametric method to determine the range for out-of-distribution samples, Ferreira et al. [7] apply a generic algorithm to build a tree representation of the operations needed to obtain the outputs for a local example, Lal et al. [13] develop a methodology to extract rules from an ensemble of tree models, and Barbiero et al. [4] use an entropy-based model to generate first-order logic explanations in a deep learning model. Although these approaches are similar to our methodology, most of them do not have the ability to adapt to local/global explanations or be applied to any model.

## 3 Methodology

In this section, we present the model-agnostic method used to extract the patterns learned by a model and the subsequent analysis.

In our methodology, we assume a traditional modeling approach in which the dataset is split into at least a training and testing set. An arbitrary model is then trained on the training set and evaluated on the testing set.

The input to our methodology is a dataset consisting of the features $X$, the true target variable $y$, and the predictions made by the model $\hat{y}$. Note that the dataset with the predictions corresponds to the training set as our goal is to obtain the learned behavior during the training process using only the training set.

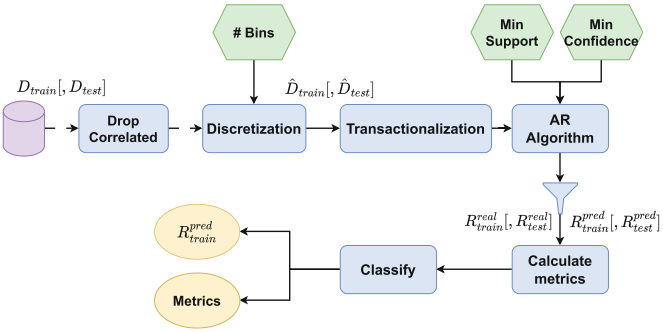


**Fig. 1.** Workflow representation of the methodology. Note that dashed lines represent optional steps. Note that $D_{train}$ and $D_{test}$ represent the training and test datasets, while $\hat{D}_{train}$ and $\hat{D}_{test}$ refer to the discretized version. In addition, $R^{real}_{train}$ and $R^{real}_{test}$ represent the rules obtained for the training and test dataset from the real target values, while $R^{pred}_{train}$ and $R^{pred}_{test}$ are the rules for the predictions made by the model.

### 3.1 Preprocessing

This step includes dropping correlated features and discretizing the remaining features shown in Fig. 1.

Optionally, the features $X$ can be preprocessed to remove highly correlated features. This is necessary because the Apriori algorithm may obtain similar rules for two correlated features, using them interchangeably because of their similarity in frequency.

Once the training dataset $D_{train}$ contains its predictions, all continuous columns are discretized because the Apriori algorithm cannot use continuous values and the explanations would be more generalizable using ranges instead of specific values. Discretization requires a number of bins, which is specified by the user. A K-means [6] method is used to determine the bin ranges for each feature, where the K-means algorithm uses the centroid of the cluster to determine the range of the bins. To discretize the target, the predictions $\hat{y}$ are used to determine the bins, which are then applied to $y$. Note that the goal of the methodology is to extract the patterns learned by the model, so the discretization must use the predictions as a reference for the target discretization. Therefore, the result of the discretization process is the discretized dataset $\hat{D}_{train}$.

### 3.2 Rules Mining

This step involves both the transactionalization and AR algorithm phases shown in Fig. 1. Once the dataset has been properly discretized, the next step is to apply the AR algorithm to obtain the rules, which has been the well-known Apriori algorithm for this study. However, before that, the Apriori algorithm needs to transform the discretized dataset $\hat{D}_{tr}$ into a set of transactions. These transactions contain information about the items presented in each instance. An item refers to the bin in which the value of each feature and target value is contained. With the transactions, the Apriori algorithm is used to obtain the rules that satisfy the minimum confidence and support thresholds. Typically, we are interested in rules with high confidence, which indicate the probability that the pattern represented in the rule has been learned by the model. The support represents the generality/specificity of the pattern. A low minimum support allows specific patterns to be represented, while high support only represents general rules with a high frequency. Therefore, the result of the Apriori algorithm is a set of rules $R_{train}$.

We are only interested in the patterns that have an impact on the target. For that reason, we select only those rules from the set $R_{train}$ that contain one of the target items/bins. Additionally, the Apriori algorithm may generate redundant rules. A redundant rule is one in which a subset of the antecedents obtains a greater confidence. These rules do not provide useful information, and for that reason, they are removed, considering more general rules. Note that there are two targets in our context: the real target $y$ and the predicted target $\hat{y}$. Therefore, we are interested in two sets of rules: $R^{true}_{train}$ and $R^{pred}_{train}$.

### 3.3 Calculate Metrics

To obtain meaningful insights from the analysis, it is necessary to calculate a set of metrics for each rule obtained from the training set $\hat{D}_{train}$. Two types of metrics are typically used: AR metrics and performance metrics. The most commonly used AR metrics are confidence and support, while performance metrics can include any useful metric for analysis. In our case, we have chosen the mean squared error ($MSE$) as the error increases greatly when there are great differences between true and predictions. It is important to note that we calculate AR metrics for both $R^{true}_{train}$ and $R^{pred}_{train}$, as this helps to identify whether a learned pattern represents an actual pattern or not. Note that these metrics are detailed in Sect. 4.2

Optionally, metrics can also be calculated for other datasets, such as the test dataset $D_{test}$. To calculate these metrics, the entire process is repeated starting from the discretization step, but using the bins obtained from the train set $D_{train}$. Specifically, we are interested in calculating the performance metrics of the predicted rules $R^{pred}_{test}$, which can help to evaluate the generalization of the potential rules learned by the model.

### 3.4 Classify

Finally, to facilitate their comprehension, the rules are classified into four categories based on whether they represent a real pattern or not, and whether they were correctly learned (CL) by the model or not.

The rules that represent a real pattern (RP) are identified based on the confidence difference between $R^{pred}_{train}$ and $R^{true}_{train}$. A high difference between real and predicted rules may indicate that the pattern learned by the model (assuming high confidence) does not correspond to a real pattern (an Unreal Pattern or UP) that should have been learned if the predictions were closer to the actual values. On the contrary, a low difference may indicate that the model has learned a real pattern that exists in reality. To determine rules with large or low differences, the z-score is used, with those above 3 considered unreal patterns and those below 3 considered real patterns.

The rules that were incorrectly learned (IL) by the model represent those with high errors calculated from $y$ and $\hat{y}$ for each rule in $R^{pred}_{train}$. Rules with high errors are considered poorly learned patterns, whereas those with low errors are considered correctly learned rules. Again, to determine rules with high or low errors, the z-score is used, with rules above a z-score of 1 considered poorly learned and those below a z-score of 1 considered correctly learned.

## 4 Results and Discussion

In this section, we present the results obtained for the studied dataset. In Sect. 4.1 the experimental setting is described. In Sect. 4.2 the metrics used for the evaluation and analysis are shown. In Sect. 4.3 the results obtained after applying our methodology are presented.

### 4.1 Experimental Setting

To carry out our experiment, we selected a well-known dataset to test our methodology and analyze two different scenarios. In particular, the dataset used in our experiment is the California Housing dataset [11]. It contains information from the 1990 California census and includes eight real features, such as median income ($mi$), longitude ($l$), total rooms ($tr$), and more. The target variable is to predict the mean house value ($mhv$) using these features. The dataset also includes categorical data that was removed to use only real data.

To test the proposed methodology, we used two different scenarios: underfitting and overfitting the model over the data. To underfit the model, we selected a fully connected model with only 1 neuron, 1 hidden layer, and trained for 5 epochs as it is not enough to fit the data. For overfitting, we selected a model with 512 neurons in 10 hidden layers and trained for 100 epochs. We will identify the patterns obtained after applying our methodology and evaluate the strengths and weaknesses of the model.

### 4.2 Metrics

In this section, we describe the main metrics used in our methodology to evaluate the both the AR obtained and the performance of the model.

The support for a rule ($A \Longrightarrow C$), where $A$ and $C$ denote the antecedents and consequents respectively, is the percentage of instances in the dataset that satisfy both the antecedent and consequent conditions. $frq(A, C)$ represents the number of instances that satisfy both the antecedent and consequent conditions, while $N$ represents the total number of instances in the dataset. The support values range from 0 to 1.

$$Support(A \Longrightarrow C) = \frac{frq(A, C)}{N} \tag{1}$$

The confidence, is the probability that instances containing $A$ also contain $C$, and it also ranges from 0 to 1.

$$Confidence(A \Longrightarrow C) = \frac{frq(A, C)}{frq(A)} \tag{2}$$

The MSE is a commonly used metric in regression problems, where it measures the squared difference between the predicted and true values.

### 4.3 Results

The main results are presented in tables, where the antecedents and consequents follow the same structure: $F[lower, upper]$, where $F$ represents the feature abbreviation, and $lower$ and $upper$ represent the minimum and maximum bin values, respectively. For the hyperparameters of the methodology, we set the number of bins for each feature at five, the minimum support at 1%, and the minimum confidence at 75%.

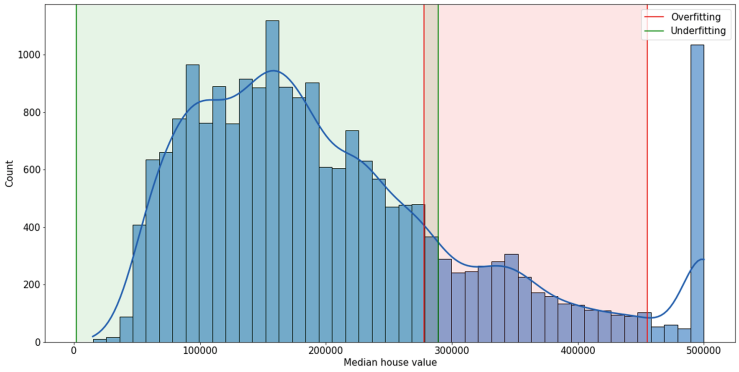


**Fig. 2.** Histogram of the target variable (blue) in $D_{train}$ and ranges covered by the rules obtained by the model in the overfitting (red) and underfitting (green) scenarios.

**Overfitting Scenario.** Table 1 displays the top five rules obtained by the Apriori algorithm, sorted by prediction confidence. It can be observed that the pattern learned by the model with the highest confidence suggests that when longitude (l) falls between $-118.954$ and $-118.147$ and the mean income (mi) ranges between 4.749 and 5.798, the mean house value is between 3.27e5 and 3.84e5. This indicates that houses in that area of the dataset are relatively expensive since this range is above the mean value of 2.07e5.

**Table 1.** Top five association rules discovered by the Apriori algorithm sorted by confidence in the overfitting scenario. Note that $S_{pred}$, $S_{true}$, $C_{pred}$ and $C_{true}$ denote the support ($S$) and confidence ($C$) for the predicted ($pred$) and true ($true$) values of the target. In addition, $MSE_{train}$ and $MSE_{test}$ denote the error in the training and test dataset, respectively. Finally, the type column corresponds to the assigned rule class.

| Antecedents | | Consequent | $S_{pred}$ | $S_{true}$ | $C_{pred}$ | $C_{true}$ | $MSE_{train}$ | $MSE_{test}$ | Type |
|---|---|---|---|---|---|---|---|---|---|
| $l \in [-118.95, -118.15] \wedge mi \in [4.8, 5.8]$ | $\Rightarrow$ | [3.27e5, 3.84e5] | 0.02 | 0 | 0.85 | 0.16 | 1.06e5 | 1.03e5 | IP CL |
| $l \in [-118.95, -118.15] \wedge mi \in [5.8, 7.2]$ | $\Rightarrow$ | [3.84e5, 4.55e5] | 0.01 | 0 | 0.82 | 0.16 | 9.18e4 | 9.84e4 | IP CL |
| $l \in [-118.954, -118.147] \wedge tr \in [1406, 2396] \wedge mi \in [3.8, 4.8]$ | $\Rightarrow$ | [2.78e5, 3.27e5] | 0.01 | 0 | 0.80 | 0.11 | 9.49e4 | 8.47e4 | IP CL |
| $l \in [-117.55, -116.60] \wedge mi \in [4.749, 5.798]$ | $\Rightarrow$ | [3.84e5, 4.55e5] | 0.01 | 0 | 0.78 | 0.04 | 1.96e5 | 2.00e5 | IP IL |
| $l \in [-118.95, -118.15] \wedge mi \in [3.8, 4.8]$ | $\Rightarrow$ | [2.78e5, 3.27e5] | 0.03 | 0 | 0.76 | 0.13 | 9.52e4 | 9.01e4 | IP CL |

When analyzing the antecedents of the top five rules obtained in Table 1, it becomes apparent that only three features were used: longitude ($l$), median income ($mi$), and total rooms ($tr$). This suggests that these features provide more information to the model compared to others, as the patterns obtained using other features lacked sufficient confidence. Furthermore, the consequent of the rules only shows patterns for house values above the mean (2.07e5) as shown in Fig. 2, which implies that the model could not learn patterns for cheaper houses with sufficient confidence.

Looking at the AR metrics, we can see that the support of the rules using the predictions is higher than when using the true target values. The confidence of the predictions ranges from 0.76 to 0.85, while the true confidence obtained from the target values ranges from 0.04 to 0.16. This suggests that the rules obtained may not represent real patterns, which could be present if the model performed better.

In terms of error, the model had an average error of 1.02e5 in both the training and test sets. The model does not seem to have any generalization problems for these specific patterns, even though it was configured to overfit the data, except for the third rule. Additionally, the error of the patterns seems to be around the average, except for the fourth rule, which represents a bar-learned pattern.

In general, the methodology found four correctly learned patterns (CL) and one rule with an incorrectly learned pattern (IL) whose error was above the mean.

**Underfitting Scenario.** Table 2 presents the top five rules obtained using our methodology and sorted by confidence. The first pattern indicates that if the housing mean age falls within the range of 17 to 21 years and the median income is between 0.5 and 2.1, then the median house value falls between the range of 1.92e3 and 1.24e5, which is a price range below the average.

**Table 2.** Top five association rules encountered by the Apriori algorithm sorted by confidence in the underfitting scenario. Note that $S_{pred}$, $S_{true}$, $C_{pred}$ and $C_{true}$ denote the support ($S$) and confidence ($C$) for the predicted ($pred$) and true ($true$) values of the target. In addition, $MSE_{train}$ and $MSE_{test}$ denote the error on the training and test dataset, respectively. Finally, the type column corresponds to the assigned rule class.

| Antecedents | Consequent | $S_{pred}$ | $S_{true}$ | $C_{pred}$ | $C_{true}$ | $MSE_{train}$ | $MSE_{test}$ | Type |
|---|---|---|---|---|---|---|---|---|
| $hma \in [17, 21] \wedge mi \in [0.5, 2.1]$ | $\Rightarrow$ [1.92e3, 1.24e5] | 0.02 | 0.01 | 0.99 | 0.73 | 4.71e4 | 3.93e5 | WFR CL |
| $hma \in [21, 27] \wedge mi \in [0.5, 2.1]$ | $\Rightarrow$ [1.92e3, 1.24e5] | 0.01 | 0.01 | 0.97 | 0.76 | 4.89e4 | 5.28e4 | WFR CL |
| $hma \in [48, 52] \wedge mi \in [3.8, 4.8]$ | $\Rightarrow$ [2.45e5, 2.89e5] | 0.01 | 0 | 0.95 | 0.17 | 9.27e4 | 7.96e4 | BFR IL |
| $hma \in [27, 32] \wedge mi \in [3.0, 3.8]$ | $\Rightarrow$ [1.66e5, 2.05e5] | 0.03 | 0.01 | 0.92 | 0.22 | 6.41e4 | 6.60e4 | BFR CL |
| $hma \in [27, 32] \wedge mi \in [2.1, 3.0]$ | $\Rightarrow$ [1.24e5, 1.66e5] | 0.03 | 0.01 | 0.92 | 0.21 | 5.43e4 | 5.07e4 | BFR CL |

As in the previous section, the relevant features for this model are housing median age ($hma$) and median income ($mi$) as shown in the antecedents. However, in contrast to the overfitted model, the consequents mostly consider rules above the mean (2.07e5) as shown in Fig. 2.

Analyzing the AR metrics, we observe that the support and confidence of the predictions are mostly greater than the real ones. However, the difference in the first two rules is not considerably greater compared to the other rules, and we consider them well-formed rules.

In terms of error metrics, the error is considerably better than the overfitted model, with an average of 6.01e04 in train and 5.98e04 in test. Additionally, generalization does not seem to be a problem in these patterns, as the training error is similar to or lower than the test error.

In summary, the methodology obtained two patterns that represent real patterns (WFR) with remarkable performance (CL), one rule that does not represent a real pattern (BFR) and has poor performance (IL), and two rules that do not represent a real pattern (BFR) but have good performance (CL).

## 5 Conclusions

In this work, we have developed a novel model-agnostic explainability methodology applied to a deep learning model. The method uses the well-known Apriori algorithm internally to obtain the patterns learned by the model in a simple format, which are then analyzed to draw conclusions about the learned behavior of the model. The results obtained provide a taxonomy of rules that can be classified based on the association and error metrics obtained.

In the future, there are several issues that need to be addressed. First, the method does not consider the full coverage of the dataset in the rules. Secondly, the discretization process is critical, and it could be improved by using a metaheuristic. Thirdly, rules with significant overlap should be removed. Finally, the method should be studied with a large attribute space to evaluate its scalability.

**Acknowledgements.** The authors would like to thank the Spanish Ministry of Science and Innovation for the support under the projects PID2020-117954RB and TED2021-131311B, and the European Regional Development Fund and Junta de Andalucía for projects PY20-00870, PYC20 RE 078 USE and UPO-138516.

## References

1. Afchar, D., Guigue, V., Hennequin, R.: Towards rigorous interpretations: a formalisation of feature attribution. In: Proceedings of the 38th International Conference on Machine Learning, vol. 139, pp. 76–86. PMLR (2021)
2. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 207–216, May 1993
3. Albini, E., Long, J., Dervovic, D., Magazzeni, D.: Counterfactual shapley additive explanations. In: 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT 2022, pp. 1054–1070 (2022)
4. Barbiero, P., Ciravegna, G., Giannini, F., Lió, P., Gori, M., Melacci, S.: Entropy-based logic explanations of neural networks. arXiv (2021)
5. De Bernardi, G., Narteni, S., Cambiaso, E., Mongelli, M.: Rule-based out-of-distribution detection (2023)
6. Dash, R., Paramguru, R., Dash, R.: Comparative analysis of supervised and unsupervised discretization techniques. Int. J. Adv. Sci. Technol. **2**, 29–37 (2011)

7. Ferreira, L., Guimarães, F., Pedrosa-Silva, R.: Applying genetic programming to improve interpretability in machine learning models. In: Proceedings of: Congress on Evolutionary Computation, pp. 1–8 (2020)
8. Fumagalli, F., Muschalik, M., Kolpaczki, P., Hüllermeier, E., Hammer, B.: SHAP-IQ: unified approximation of any-order Shapley interactions (2023)
9. Gallardo-Gómez, J.A., Divina, F., Troncoso, A., Martínez-Álvarez, F.: Explainable artificial intelligence for the electric vehicle load demand forecasting problem. In: Proceedings of 17th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2022), pp. 413–422 (2023)
10. Hou, Y., Zheng, L., Gould, S.: Learning to structure an image with few colors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10116–10125 (2020)
11. Kelley Pace, R., Barry, R.: Sparse spatial autoregressions. Stat. Probab. Lett. **33**(3), 291–297 (1997)
12. Kohlbrenner, M., Bauer, A., Nakajima, S., Binder, A., Samek, W., Lapuschkin, S.: Towards best practice in explaining neural network decisions with LRP. In: Proceedings of: International Joint Conference on Neural Networks, pp. 1–7 (2020)
13. Lal, G.R., Chen, X., Mithal, V.: TE2Rules: extracting rule lists from tree ensembles (2022)
14. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS 2017, pp. 4768–4777. Curran Associates Inc. (2017)
15. Martín, D., Martínez-Ballesteros, M., García-Gil, D., Alcalá-Fdez, J., Herrera, F., Riquelme-Santos, J.C.: MRQAR: a generic MapReduce framework to discover quantitative association rules in big data problems. Knowl.-Based Syst. **153**, 176–192 (2018)
16. Martínez Ballesteros, M., Troncoso, A., Martínez-Álvarez, F., Riquelme, J.C.: Improving a multi-objective evolutionary algorithm to discover quantitative association rules. Knowl. Inf. Syst. **49**, 481–509 (2016)
17. Tjoa, E., Guan, C.: Quantifying explainability of saliency methods in deep neural networks with a synthetic dataset. IEEE Trans. Artif. Intell., 1–15 (2022)
18. Troncoso-García, A.R., Martínez-Ballesteros, M., Martínez-Álvarez, F., Troncoso, A.: Explainable machine learning for sleep apnea prediction. Procedia Comput. Sci. **207**, 2930–2939 (2022)
19. Troncoso-García, A.R., Martínez-Ballesteros, M., Martínez-Álvarez, F., Troncoso, A.: A new approach based on association rules to add explainability to time series forecasting models. Inf. Fusion **94**, 169–180 (2023)