Electrical Power and Energy Systems 160 (2024) 110106

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

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

Explaining the solutions of the unit commitment with interpretable machine learning

Sara Lumbreras [a,*], Diego Tejada [b], Daniel Elechiguerra [c]

[a] *Universidad Pontificia Comillas, IIT, Spain*
[b] *TNO, Netherlands*
[c] *MIT Sloan, United States*

A R T I C L E I N F O

Keywords:
Power systems
Unit commitment
Explainable machine learning
Interpretability

A B S T R A C T

The energy transition needs mathematical models to address the complexity of shifting towards sustainable energy sources. In addition to providing accurate solutions, these models must be explainable and available for discussion among stakeholders to facilitate informed decision-making and ensure a successful transition. This paper contributes to the explainability of power systems models by applying interpretable machine learning techniques to improve understanding of the solutions to the unit commitment problem. It applies them to a case study based on the IEEE 118N system. The developed methodology aims at describing the optimal commitment solutions as a function of the conditions of the system in a compact manner that is understandable by a human being. This type of information takes the form of 'which plants are needed under which conditions' and is routinely learned by experience by system operators and other agents participating in the system. This experiential knowledge is realized in an approximate form that is simple enough to help make or justify decisions. By applying interpretable machine learning techniques, our methodology can automatically extract what was previously only available through human experience and reflection. Our approach involves model trees and node clustering to find a concise description of the different situations where the system can be found. Our results show that the methodology can explain these modes of operation for the 118N system in a sufficiently simple manner to be understood by a human unfamiliar with the system. This shows that interpretable machine learning can provide valuable insights into real solutions of the unit commitment and help improve decision-making in this area.

1. Introduction

The global energy landscape is undergoing a profound transformation, driven by the imperative to mitigate climate change and transition towards sustainable and renewable energy sources. As this energy transition gains momentum, there is an escalating demand for accurate, reliable, and comprehensive mathematical models to guide decision-making processes. These models play a pivotal role in assessing the complex interdependencies of various energy systems, optimizing resource allocation, and projecting future energy scenarios. However, merely providing mathematical solutions is no longer sufficient; it is equally crucial to ensure that these models are explainable and readily available for discussion among stakeholders. This paper explores the increasing need for explainability and stakeholder engagement in the context of mathematical modeling, highlighting their essential contributions to effective decision-making and the successful implementation of sustainable energy solutions.

The Unit Commitment (UC) problem aims to plan the scheduling of the system units over a given time horizon, minimizing the total operating costs, respecting certain physical and temporal constraints from generators and transmission lines, and guaranteeing system security requirements [1]. Given its importance, the UC is one of the most studied problems in the electricity sector. One promising recent line of research in this context is the application of Machine Learning (ML) techniques to the UC problem, such as the prediction of electricity generation from non-dispatchable renewable sources, which allows for reducing the uncertainty of the problem and, thus, obtaining more efficient results [2,3]. The existing literature on the topic comprises a wide variety of approaches. The majority of these research papers address the UC problem as a supervised learning task, in which precomputed solutions of the optimization problem are used to train an ML

* Corresponding author.
E-mail address: sara.lumbreras@iit.upcomillas.es (S. Lumbreras).

https://doi.org/10.1016/j.ijepes.2024.110106
Received 29 June 2023; Received in revised form 22 May 2024; Accepted 20 June 2024
Available online 25 June 2024

Nomenclature

Sets:

n	hours from 1 to N
t	thermal unit running from 1 to T
$t1(t)$	thermal units with $TU_t = 1$
i	transmission node from 1 to I
l	transmission lines from 1 to L
s	scenarios from 1 to S

Parameters:

D_{ni}	Power demand [M.W.]
DU_{ni}, DD_{ni}	Up and down reserve requirements [M.W.]
$\overline{P}_t, \underline{P}_t$	Maximum and minimum thermal output [M.W.]
SU_t, SD_t	Maximum start-up and shutdown capacities
CF_t	Fixed cost [\$/h]
CV_t	Variable cost [\$/MWh]
CSU_t, CSD_t	Start-up and shutdown cost [\$]
$CPNS$	cost of not satisfying part of the demand [\$/MWh]
RU_t, RD_t	Up and down ramp limits [MW/h]
TSU_t, TSD_t	Start-up and shutdown time [h]
TU_t, TD_t	Minimum up and down times [h]
NC_t	thermal unit connection to node [-]
$\overline{F}_l$	Maximum transmission flow capacity [M.W.]
NF_l	Connection 'from node' in line l [-]
NT_l	Connection 'to node' in line l [-]
IG_{sni}	Intermitent generation production (i.e., renewable resources) [M.W.]

Variables:

uc_{nt}	Commitment decision {0,1}
$\widetilde{uc}_{nt}$	Commitment decision in the state transition (S.T.) formulation {0,1}
su_{nt}, sd_{nt}	Start-up and shutdown decisions {0,1}
p^T_{nt}	Total thermal output [M.W.]
p_{nt}	Power output above minimum output [M.W.]
ru_{nt}, rd_{nt}	Up and down reserve of thermal units [M.W.]
f_{nl}	transmission line flow [M.W.]
ens_{ni}	energy non served [M.W.]

model to replicate them and solve the problem faster. With few exceptions attempting to develop strictly surrogate models [4], the general purpose of these works is not to create ML models that can be used as a solver themselves but rather to provide tools that can assist the optimization problem [5].

Most of the proposed methods rely on developing ML models trained to generate complete or partial solutions to the UC problem, which are then used as starting points for the optimization problem. The result in all the proposed approaches is a substantial reduction of the computation without significantly moving away from the optimal solution. For instance, a genetic-based ANN model is developed in [6] to compute a set of preliminary solutions that are subsequently optimized to obtain the optimal generation planning for a thermal power system. In [7], ANNs are used to predict the commitment of generating units. If the outcome is not considered certain enough, dynamic programming is employed to narrow down the outcome. Similarly, an ANN is developed in [8] to predict the discrete variables of the problem, e.g., the commitment of the thermal generators, and an optimization approach is applied to obtain the continuous ones, e.g., the production of generating units.

Alternative proposals not based on ANN appear in [9] and [10], which develop a multi-target random forest and a k-nearest neighbors regression to obtain warm-start solutions for the UC problem. The latter study also proposes an alternative approach, where another k-nearest neighbor algorithm is developed to predict the transmission constraints that should be included in the optimization problem, leaving out the non-critical ones. A similar method is explored in [5], extending it to a broader variety of constraints.

Besides the supervised learning approach, different models have been developed addressing the UC problem from other perspectives. On the one hand, unsupervised learning can be used to reduce the dimension of the problem and thus save execution time. For instance, the model proposed in [11] addresses the UC problem under uncertainty by clustering the given scenarios and applying the UC optimization model to these clusters. The result is an intermediate solution between stochastic and scenario-based unit commitment, which can significantly reduce the size of the problem. In [12], the clustering of decision variables, such as the commitment of multiple units, is performed without significantly increasing the total cost of the generation planning compared to the base formulation. On the other hand, there are multiple studies, such as those developed in [13,14,15,16], that address the resolution of the UC using reinforcement learning algorithms, widely used in the field of optimization. Besides reducing execution time, the main advantage of these approaches is that they can generate feasible and efficient solutions without the need for a precise UC model definition and, in any case, without having to develop the precomputed scenarios that are necessary for other approaches [16].

Interpretable machine learning (IML) is a field of study that focuses on developing accurate and transparent models in how they compute their outcomes. IML aims to create models that humans can easily understand and interpret, allowing them to gain insights into how the model produced its predictions.

Traditional machine learning models, such as neural networks, are often called black-box models because it is difficult to understand how they arrive at their predictions. In contrast, interpretable machine learning models such as decision trees and rule-based models are transparent in the way they make decisions, making them easier for humans to understand [17].

Interpretable machine learning has become increasingly important in recent years as machine learning algorithms have become more widespread in various industries. In many cases, it is not enough for a model to make accurate predictions; it is also essential to understand how those predictions were made. This is particularly true in healthcare and finance, where decisions based on machine learning algorithms can have significant consequences.

Despite the abundance of existing studies related to the unit commitment problem and, in particular, to the application of ML for its resolution, they have yet to target the development of models that can be used to explain the solutions generated by UC models. The ML models that replace or assist the UC model and the optimization problem are conceived as black boxes whose only purpose is to obtain the optimal generation schedules, regardless of their underlying behavior. Consequently, applying interpretable machine learning to this subject constitutes a virtually unexplored horizon so far.

Despite the large amount of research related to the UC problem and, mainly, the application of machine learning to its resolution, no works are dedicated to developing models that can be used to understand the solutions generated by UC models. The interpretation of such results still requires a deep knowledge of the specific system since the optimization algorithm does not transparently explain how it obtained the solution found. This paper fills this gap and develops a methodology based on interpretable machine learning that can estimate the values of the variables and dual variables of the optimal solutions of the UC problem in a human-understandable way.

These insights are usually learned by experience and reflection of the humans that participate in the operation of the system, and can be expressed in the structure of 'which plants are needed under which conditions'. Some examples, which guide the proposed methodology, can be found below:

The insights obtained through this process can be verbalized in natural language with the following questions:

- What are the variables in system operation that show variation across scenarios and are therefore worth studying? (Question 0)
- What are the most important features of a scenario, i.e. the ones that have the largest impact on the variables of interest? (Question 1)
- What are the specific dynamics of the variables of interest with respect to the important scenario features? (Question 2)
- What are the intrinsic links among variables in the problem? (Question 3)

This experiential knowledge is realized in an approximate form that is simple enough to be helpful in making or justifying decisions. By applying interpretable machine learning techniques, our methodology allows for the explicit extraction and articulation of operational insights typically gained through prolonged experience. The model adeptly identifies active generators under varying conditions and determines trigger levels of net demand that might necessitate costlier generation resources. This extracted knowledge is versatile and can be leveraged in several ways. Firstly, it can accelerate decision-making processes, particularly in unexpected scenarios, thereby enhancing adaptability. Furthermore, it serves as an invaluable tool for strategic planning. By acting as a reliable proxy for system operation, the model can inform expansion planning, assist generators in making informed investment decisions, and guide Transmission System Operators (TSOs) in the strategic expansion of transmission infrastructure. An additional significant application of this model is in regulatory oversight. By revealing operational patterns, the model can help regulators detect and investigate potential market abuses. This aspect of the model adds a layer of security and trust to market operations. We have incorporated a detailed description of these applications into the manuscript to clearly articulate the multifaceted uses and advantages of our proposed methodology, emphasizing its role in enhancing both the efficiency and transparency of power system management. This increase in transparency means that there are no potentially harmful implications of our model; on the contrary, it supports the ethical principle of transparency.

2. Formulation

The nomenclature used for the formulation of the paper is included below.

The UC model we used in this paper is the same as in the base case in paper [18], by the same authors of this article, which is based on the Tight and Compact formulation. The optimization problem involves determining the optimal dispatch of power generation to meet a given demand while also satisfying reserve requirements and considering the technical limitations of generators, all at the lowest possible operational cost. The objective function in equation (1) minimizes the variable costs associated with system production, including start-up and shutdown costs. The UC problem must satisfy the following constraints: balance between generation and demand (equation (2), which can also account for intermittent generation for a given scenario 's'), as well as up and down secondary reserve requirements (equations (3) and (4) and the technical limitations of generators, such as minimum and maximum production limits, start-up and shutdown limits, ramping constraints, and minimum up and down times.

$$min\sum_{nt} CF_t uc_{nt} + \sum_{nt} CV_t p^T_{nt} + \sum_{nt} CSU_t su_{nt} + \sum_{nt} CSD_t sd_{nt} + + \sum_{ni} CPNS ens_{ni} \tag{1}$$

s.t.

$$\sum_{t|NC_t=i} p^T_{nt} + \sum_{l|NT_l=i} f_{nl} - \sum_{l|NF_l=i} f_{nl} + ens_{ni} + IG_{ni} = D_{ni} \qquad \forall n, i \tag{2}$$

$$\sum_{t|NC_t=i} ru_{nt} \geq DU_{ni} \qquad \forall n, i \tag{3}$$

$$\sum_{t|NC_t=i} rd_{nt} \geq DD_{ni} \qquad \forall n, i \tag{4}$$

Equations (5)-(7) place restrictions on the maximum capacity of thermal units. It is important to note that (6) and (7) only apply to a specific subset of thermal units, namely those with a minimum up-time of 1 $t1(t)$, which is defined as the thermal units with minimum up-time $TU_t = 1$. For thermal units with $TU_t \geq 2$, , both constraints are replaced by a more concise equation using (5). Equation (8) ensures that production above the minimum minus the down reserve is always positive. The minimum thermal output plus the production above the minimum output constitutes the total thermal output as stated in (9). Finally, equation (10) outlines thermal units' commitment, startup, and shutdown logic.

$$p_{nt} + ru_{nt} \leq uc_{nt}\left(\overline{P}_t - \underline{P}_t\right) - sd_{n+1,t}(\overline{P}_t - SD_t) - su_{nt}(\overline{P}_t - SU_t) \qquad \forall n,\ t \notin t1 \tag{5}$$

$$\begin{aligned} &p_{nt} + ru_{nt} \leq uc_{nt}\left(\overline{P}_t - \underline{P}_t\right) - \\ &sd_{n+1,t}(\overline{P}_t - SD_t) su_{nt} \max(SD_t - SU_t, 0) \qquad \forall n,\ t \in t1 \end{aligned} \tag{6}$$

$$\begin{aligned} &p_{nt} + ru_{nt} \leq uc_{nt}\left(\overline{P}_t - \underline{P}_t\right) \\ &-sd_{n+1,t} \max(SU_t - SD_t, 0) su_{nt}(\overline{P}_t - SU_t) \qquad \forall n,\ t \in t1 \end{aligned} \tag{7}$$

$$p_{nt} - rd_{nt} \geq 0 \qquad \forall n, t \tag{8}$$

$$p^T_{nt} = p_{nt} + \underline{P}_t uc_{nt} \qquad \forall n, t \tag{9}$$

$$uc_{nt} - uc_{n-1,t} = su_{nt} - sd_{nt} \qquad \forall n, t \tag{10}$$

The ramping constraints ensure that the unit operates within its ramp-rate limits. An essential representation of these ramping constraints appears in [4], such as (11) and (12). Moreover, imposing more stringent ramping constraints is possible by utilizing the binary variables from the UC problem. A more comprehensive and robust ramping constraint appears in [11]. Equations (13) and (14) establish these restrictions.

$$p_{nt} - p_{n-1,t} + ru_{nt} \leq RU_t \qquad \forall n, t \tag{11}$$

$$p_{nt} - p_{n-1,t} - rd_{nt} \geq -RD_t \qquad \forall n, t \tag{12}$$

$$p_{nt} - p_{n-1,t} + ru_{nt} \leq RU_t uc_{nt} + \left(SU_t - \underline{P}_t - RU_t\right) su_{nt} \qquad \forall n, t \tag{13}$$

$$p_{nt} - p_{n-1,t} - rd_{nt} \geq -RD_t uc_{n-1,t} - \left(SD_t - \underline{P}_t - RD_t\right) sd_{nt} \qquad \forall n, t \tag{14}$$

The minimum number of periods that the unit must be online and offline is imposed in (15) and (16):

$$\sum_{n'=n+1-TU_t}^{n} su_{n't} \leq uc_{nt} \qquad \forall n, t \tag{15}$$

$$\sum_{n'=n+1-TD_t}^{n} sd_{n't} \leq 1 - uc_{nt} \qquad \forall n, t \tag{16}$$

Finally, equation (15) defines the power flow limit in the electrical lines of the system:

$$|f_{nl}| \leq \overline{F}_l \qquad \forall n, l \tag{17}$$

3. Case study description

The power system analyzed in this paper consists of a modified IEEE 118-bus test system, which includes multiple wind plants to account for the effect of intermittent generation. This case study has been widely used in UC studies, for example, [19,20,21]. The primary data source used to model the power system is [22]. This modified IEEE 118-bus system comprises 118 buses, 186 transmission lines, 54 thermal units, 91 loads, and three wind units (base case). All detailed parameters, such as generator characteristics, transmission network, load distribution profile, system-wide power demand, and wind power scenarios, are available at [23]. A single-node analysis is also included.

Scheduling is carried out for a 24-hour horizon divided into 24 h. A typical load profile has been defined for all scenarios within this time horizon, with an average and maximum level of 3991 MW and 5592 MW, respectively. This aggregate system demand is distributed among the 91 power system loads according to the load factor assigned to each node. Furthermore, it is worth noting that the cost of not satisfying part of the demand (*CPNS*) has been set to 10000 €/MWh.

A total of 365 wind generation scenarios have been considered, using the hourly wind capacity factors for different Spanish regions, based on MERRA-2. These scenarios simulate the present-day fleet of wind farms, the near-term and long-term future fleets, as described in [24]. The differentiation between the case study's scenarios lies in the generation profile of wind units.

Consequently, the base case, in which three wind units are spread over the system, presents an aggregated average and maximum production of 584 MW and 1522 MW, respectively. In addition to this base case, a high wind penetration case with another 365 scenarios and ten units has been elaborated to analyze the impact of wind penetration on model performance. This alternative case's aggregated average and maximum production are 1713 MW and 4848 MW, respectively. Wind production distributions are shown in Fig. 1.

4. Methodology

Our methodology involves several steps that extend across data processing, feature engineering, model selection and training, and model evaluation. The first step is data preparation. The original data used to train ML models often requires preprocessing to ensure an appropriate format. This stage is critical as it can significantly impact the performance of the models. The next step is feature engineering, which involves selecting and transforming the relevant features from the preprocessed data. This stage is crucial, as it helps to reduce noise and improve model accuracy. The third step is model selection, which involves choosing an appropriate interpretable machine-learning algorithm for solving the unit commitment problem. We propose using model trees and node clustering for enhanced transparency and interpretability. The final step is model evaluation, which involves testing the performance of the selected model on a separate test dataset. This stage helps ensure that the model performs well on new data and can be applied in practical settings.

4.1. Data preparation

The data preparation stage involved included normalization and feature extraction. From all the information employed by the model, e.g., intermittent generation, demand, thermal generators' characteristics, and network parameters, only the inputs that vary across time periods and scenarios will provide helpful information to the model. Intermittent generation (IG_{sni}) and demand (D_{ni}) both satisfy this requirement and, thus, will be selected as the basis for our dataset. This means that the model will not be provided with information about the parameters of the network topology and its generators, even though they have a decisive influence on the resulting solutions. This implies that the model will be network-dependent, and its validity will be restricted to the original parameters employed to solve the scenarios from which the ML model will be trained.

We will then use these variables to compute the net demand at each of the nodes where there is a wind generation unit (ND^i), which constitutes a more suitable set of features for the model since it represents the energy that needs to be fulfilled by thermal generators (18). The total net demand of the system (ND^T) will also be included in the input features (19) since this is likely to become a significant variable of the model due to its relevance in the solution of the UC problem when no network constraints are active. Moreover, this feature will be employed to build the input matrix for the single-node UC problem, as the nodal decomposition of the net demand is irrelevant for this task.

$$ND^i_{sn} = D_{ni} - IG_{sni} \qquad \forall s, n, i \in \Omega_{IG} \tag{18}$$

$$ND^T_{sn} = \sum_i D_{ni} - \sum_i IG_{sni} \qquad \forall s, n \tag{19}$$

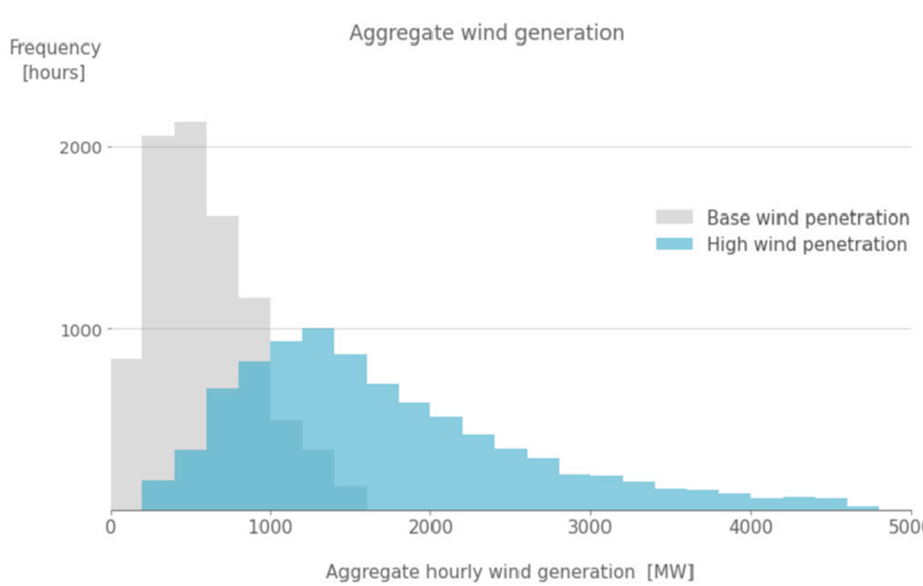

Fig. 1. Aggregate wind generation comparison between the low (base) and the high wind penetration cases.

Each resulting feature presents a different range of values, particularly the net system demand. This variety of feature scaling can negatively impact the performance of several machine learning algorithms and should be normalized [25]. In this case, only the scaling of the input features has been carried out by dividing by the maximum absolute value of the attribute since we are interested in preserving the sign of the original variable:

$$nd_{sn}^{i} = \frac{ND_{sn}^{i}}{\max_{s',n'}\left(\left|ND_{s',n'}^{i}\right|\right)} \qquad \forall s,n,i \in \{T,\Omega_{IG}\} \tag{20}$$

It should be noted that, although max-value scaling is arguably the simplest, and is highly intuitive for variables such as power flows and generations, it is problematic in the presence of outliers, which do not happen in our case study but would all for more sophisticated scaling (such as z-score).

With these scaled features, we build a simple input matrix to train our machine-learning model. However, the temporal interdependence between periods is one of the main challenges of UC problems, especially given the ramps and trajectory constraints. This is, in fact, the most significant difficulty faced by surrogate models that need to predict each hour separately as if they were not related to each other.

To mitigate this problem, information from adjacent hours can be provided to the model to capture intertemporal relationships. More precisely, we will include the variation of the net demand at the three previous and three following periods relative to the net demand at the period for which the output will be predicted (21). This will be carried out for both wind generation nodes ($\Delta ND^{i,j}$) and the whole system ($\Delta ND^{T,j}$). The selection of three adjacent hours (rather than four, or five), was chosen because of interpretability and ease of computation, and was supported by the good results obtained by the model. If poor results had been observed, this definition would have been revised to include a larger number of lags.

The problem with shifting the net demand features is that there is no information to perform these calculations for the first and last periods of the scheduling horizon. This is not an issue for some specific machine learning algorithms such as decision trees and other tree-based ensembles. Still, most algorithms cannot handle missing data, e.g., linear regression and logistic regression.

Consequently, depending on the model used to predict the desired outcomes, we will follow different imputing strategies: if the model can deal with missing data, we will assign a *NaN* value, representing that the true value is missing. Please note that this does not mean any fault in the data, just that, if we want to include a variable that describes net demand the hour before, there is just no value for the first hour because there is no previous hour. Otherwise, we will impute 0 –this is equivalent to assuming that the net demand is kept equal to the corresponding first or last period in the periods outside the optimization horizon, which is the best assumption we could make.

$$\begin{aligned}
&if(n+j \geq 1) and (n+j \leq 24): \\
&\Delta ND_{s,n}^{i,j} = ND_{s,n+j}^{i} - ND_{s,n}^{i} \qquad \forall i \in \{T,\Omega_{IG}\} \\
&else: \\
&\quad \Delta ND_{s,n}^{i,j} = IMP \qquad \forall i \in \{T,\Omega_{IG}\} \\
&\forall s,n,j \in \{\pm 1, \pm 2, \pm 3\}
\end{aligned} \tag{21}$$

After constructing these new features, we have built an input matrix (X^{NC}) where rows are identified by the corresponding scenario and time period, and columns correspond to all the net demand features that have been created. Only system-wide features will be needed for the single-node problem to build the input matrix (X^{SN}).

$$\begin{aligned}
X^{NC} = &\underset{i\in\{T,\Omega_{IG}\}}{concat}\left[\Delta ND^{i,-3}, \Delta ND^{i,-2}, \Delta ND^{i,-1}, \right. \\
&\left. ND^{i}, \Delta ND^{i,1}, \Delta ND^{i,2}, \Delta ND^{i,3}\right] \\
X^{SN} = &\\
&\left[\Delta ND^{T,-3}, \Delta ND^{T,-2}, \Delta ND^{T,-1}, ND^{T}, \Delta ND^{T,1}, \Delta ND^{T,2}, \Delta ND^{T,3}\right]
\end{aligned} \tag{22}$$

As a result, X^{SN} will be composed of 7 variables, whereas the number of input features obtained for the network-constrained UC problem (X^{NC}) will range from 28 in the low wind penetration scenarios (3 wind generation nodes) to 77 in the high wind penetration scenarios (10 wind generation nodes). To avoid the loss of interpretability of the IML model that this large number of features could imply, an initial feature selection was included as part of the model to restrict its dimensionality. The insights obtained through this process can be verbalized as an answer to the question "What are the variables in system operation that show variation across scenarios and are therefore worth studying?" (Question 0).

4.2. Feature selection

As mentioned above, the number of input features will range from 28 to 77, which can substantially hamper the interpretability of the model. The goal of this step will be to reduce this number to only 15 to improve interpretability, as it is often understood that human cognitive processing capacity cannot go much more beyond than 9 variables approximately. To select these variables, we will use the feature importance of a random forest model, which is usually able to achieve robust results without the need to optimize its hyperparameters. We will train the random forest on the set of outputs we want to predict, introducing all the available input variables. The algorithm will infer relationships between these inputs and the outputs, assigning importance values to each feature.

We will choose the 15 most relevant features using these values and discard the others. Since this process is carried out for each type of output to be predicted, the resulting feature importances will differ, and a different input matrix for each model will be obtained.

4.3. Model selection and training

Since the UC problem consists of continuous and binary variables, it will be necessary to develop two different classes of models, one for regression and another for classification tasks. Both classes will be based on multi-output decision trees, and we will use them to predict each group of variables of the same type together. This joint prediction has a significant advantage from the interpretability point of view since it facilitates the representation and understanding of the results and allows us to respect, to a greater extent, the relationships between output variables.

As we want to guarantee the interpretability of the models, we will not carry out the optimization of the hyperparameters globally. The only parameter that we fix, for the sake of interpretability, is tree depth. The branching of the decision tree can be based on any variable (continuous or discrete) or combinations of variables. Tree depth defers to the maximum number of branchings that can happen sequentially. The deeper the tree (the more nodes and branches) the more accurate it can be, but the more difficult it is to interpret. The depth chosen for the models presented in this paper was 5 and 6 for the regression and classification models, respectively. We subsequently improve interpretability by clustering the terminal nodes, so that the resulting tree is as compact as possible.

As for regression models, we will train single-variable linear regressions on each terminal node. This will allow the tree to capture the linear relationships in the resulting partitions of the feature space, something that a simple decision tree cannot achieve. It should be noted that the elicitation of these linear relationships allows to understand how any increase in demand will be covered by the units already committed and with available capacity. This results in the tree giving all the potentially interesting information about the UC problem: what units are on or off, and how will increases or decreases in demand be covered by the units.

In addition, the variable chosen for these regressions will not be unique, but each node will use the one that yields the best results. This method is similar to the one known as a model tree (Fig. 2). Still, its implementation has been simplified due to the high computational burden that its full implementation would entail.

A common problem with decision tree classifiers is the duplication of nodes and branches. Therefore, we will perform clustering (using the K-modes algorithm) of the terminal nodes to reduce the number of unique parameters of the model, substantially facilitating its interpretation. The nodes will be identified with labels corresponding to each cluster, and the outputs will be represented in an attached table. The exact process will be applied to the output variables since, in many cases, there is a high correlation between them.

5. Results

The interpretable machine learning models presented in the previous section were trained using the training set (80 % of the scenarios) and subsequently applied to predict the results corresponding to the test set (the remaining 20 % of scenarios). In this section, the resulting outcomes for some representative variables are p_{nt}^{T} (vProduct1),uc_{nt} (vCommit), f_{nl} (vCirPF), and the dual variable of the demand equation (the marginal price, eBalance). First, we will analyze and compare the feature importances obtained with the random forests for these variables.

Second, we will evaluate the validation performance of the regression model relative to its depth to select an appropriate balance between interpretability and accuracy. Then, the resulting model will be compared to other approaches with different levels of interpretability, allowing us to assess how good the predicted outcomes are for each variable. Additionally, the model's performance will be compared with the one obtained for the single-node and low-wind penetration scenarios to analyze the impact of these variations compared to the base case. The trained model will be represented together with specific terminal nodes to understand the model tree's behavior.

Finally, a similar process will be followed for the classification model. We will select a reasonable depth for an interpretable decision tree based on the validation performance. We will then analyze the impact of clustering terminal nodes and generators and represent the resulting models.

The insights obtained through this process can be verbalized in natural language with the following questions:

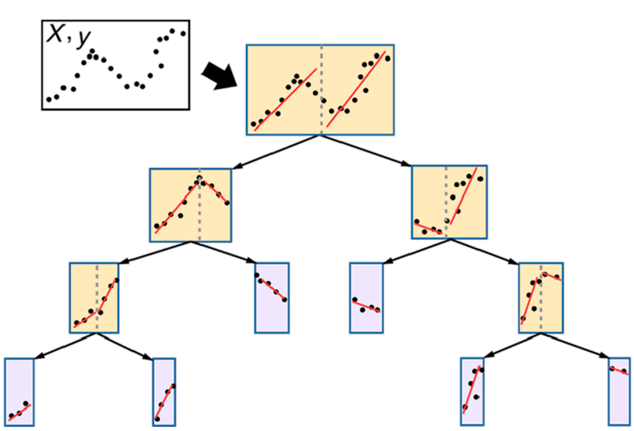

Fig. 2. Linear regression model tree.

- What are the variables in system operation that show variation across scenarios and are therefore worth studying? (Question 0) Humans participating in system operation will know, for instance, if a given generator is always on, or off, or whether it is subject to changes depending on the situation. The first step in the methodology, data preparation, addresses this question.
- What are the most important features of a scenario, i.e. the ones that have the largest impact on the variables of interest? (Question 1). This is addressed in the feature selection step.
- What are the specific dynamics of the variables of interest concerning the important scenario features? (Question 2) This description should be simple enough to be understood by a human but still accurate enough to give useful information. This is addressed in model development (regression for continuous and classification for binary variables).
- What are the intrinsinc links among variables in the problem? (Question 3) That is, are there joint dynamics for variable pairs or groups of variables? The node clustering step addresses this.

5.1. Feature selection

This step aims at the question "What are the most important features of a scenario, i.e. the ones that have the largest impact on the variables of interest? (Question 1)". As explained above, the number of input features for the network-constrained cases is too large to be handled by interpretable machine learning. Therefore, we have selected the 15 most relevant variables based on the feature importances extracted from random forests fitted on the training set. The following figure presents the most important variables for a following results will present the 20 most significant features of the high wind penetration scenarios for each studied variable.

Fig. 3 shows that there is essentially one primary variable in the model, which is the total net demand of the corresponding period (Total ND [h]). This is a reasonable result because when any constraint does not limit generators' output, it will directly depend on the total net demand of the system. Following the total net demand, the main features correspond to the increase of total net demand in adjacent periods, which provides helpful information for the model to account for the load gradient constraints of generators. We can also appreciate how the random forest algorithm considers node-related features. These will usually be related to areas of the network where power line congestion is likely to happen, which would constrain the generation of units located in those areas. However, the low relative importance of these variables makes it difficult to draw precise conclusions.

5.2. Regression models

After selecting the main features to be used by the model tree for each output, we must choose the combination of hyperparameters that allow us to obtain an optimal balance between accuracy and interpretability. For this purpose, the validation performance of the model tree has been analyzed as a function of the depth of the tree on which the algorithm is based. The performance of each variable in the regression models is evaluated using the root-mean-square error (RMSE), as it is one of the most common metrics to quantify the estimation error in continuous variables (23). Consequently, since each model has multiple outputs –as many as there are variables of each type–the global performance of each model will be analyzed by computing the mean RMSE obtained for each output.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{N}} \tag{23}$$

Applying this metric to the variables corresponding to vProduct1, we obtain the results presented in Fig. 4. In addition, the figure includes the

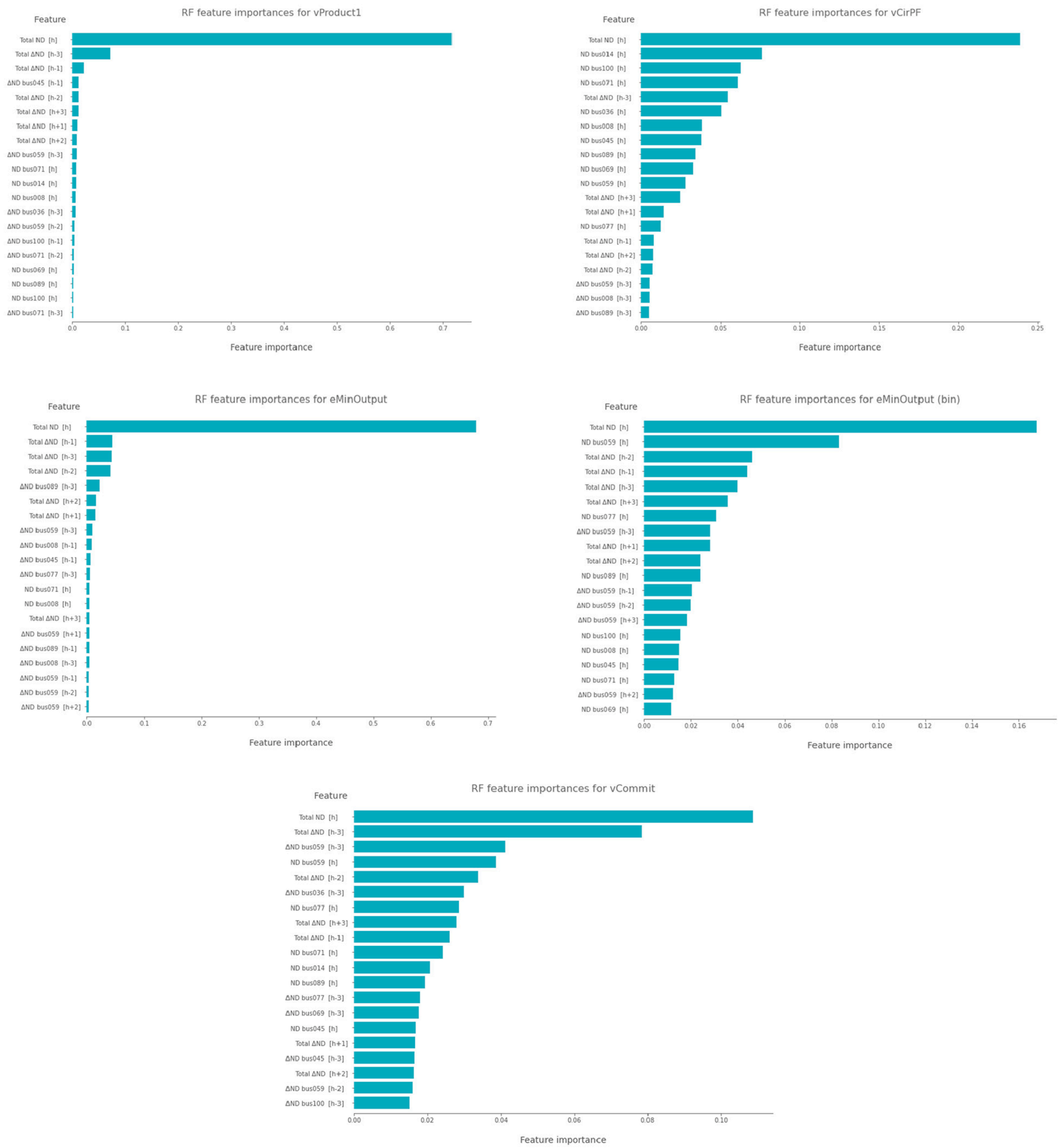

Fig. 3. Feature importance regarding optimization outputs.

mean RMSE obtained both by using a single decision tree that simultaneously predicts all outputs and a decision tree for each of the outputs (in this case, one tree for each of the generator's power production).

The model tree significantly reduces the error compared to the case of a single decision tree, between 5 and 20 %, depending on the depth. As the maximum depth of the tree increases, this difference is progressively reduced. One of the main reasons for this effect is that the tree itself begins to capture some of the linear relationships in the data. This is one of the main disadvantages of training linear regression *a posteriori* rather than considering it during model training. If that were the case, better results could probably be achieved. Additionally, as the depth of the tree increases, the number of samples at the terminal nodes decreases, which reduces the information available for training the linear regressions, and their accuracy may be negatively affected. This effect can be solved simply by using a larger number of instances to train the model (if they are available or can be generated).

An interesting fact to note from this graph is that the model tree can obtain better results than a decision tree with a higher depth. This is important because, as indicated in section 5.2, training linear regressions with one feature and an intercept at the terminal nodes could double the number of model parameters at these nodes, which also occurs when the maximum depth of the tree is increased by one level.

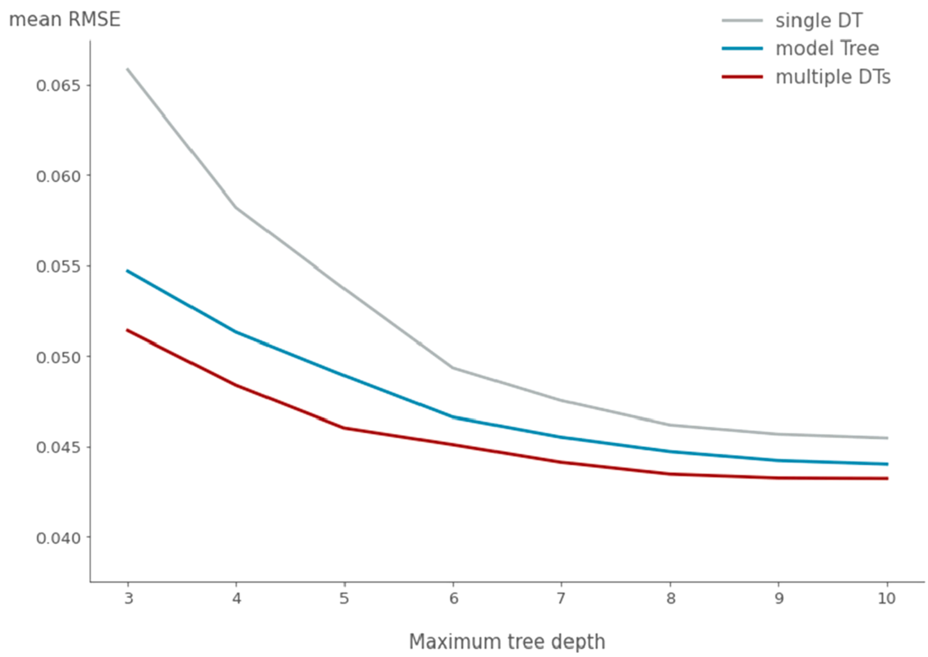

Fig. 4. Model tree and decision tree RMSE are relative to its maximum depth for vProduct1 in the network-constrained high wind penetration.

However, the interpretability of both models is not necessarily comparable. On the one hand, in the case of the model tree, it is not required to split the node, thus avoiding the addition of a new condition at each branch, and the linear regression at a terminal node provides more relevant information than just the mean value of the leaves into which the node would be split. On the other hand, not all variables at each terminal node will have a linear relationship with the feature selected at that node. In these cases, the model tree will maintain the mean value of the previous decision tree, thus having no negative impact on the interpretability of the model. Considering the above, it can be concluded that the model tree can achieve a better balance between interpretability and accuracy than the decision tree itself.

In addition, the performance obtained by the model tree is comparable to that achieved by training a decision tree with the same maximum depth for each output. This difference is initially below 6 % and is reduced as the depth increases, as the model tree has greater flexibility to capture the variability of the output. The comparison shown allows us to conclude that it would not be practical to train a decision tree for each variable (one model for each of the 54 generators) since this would imply an excessive and unmanageable complexity compared to the use of a single model tree with a greater level of depth, capable of obtaining results comparable to the previous ones, but with much lower complexity.

Finally, using the results shown in Fig. 4, the maximum depths of the model tree that would achieve the best balance between accuracy and interpretability would be either 5 or 6. Of these options, the second one obviously yields better results, but its representation and interpretation would be relatively complicated, so the depth selected for the final model tree will be 5.

Table 1 shows the set of hyperparameters selected for each of the regression models studied in this section. In all cases, the best results are obtained by training a fully-grown tree. On the contrary, the minimum samples per leaf and cost-complexity hyperparameters present a higher variability, depending on the generalization capacity needed in each case.

Table 1
Selected hyperparameters for the regression models in the network-constrained high wind penetration UC problem.

Hyperparameter	vProduct1	vCirPF	eMinOutput
Max. depth	5	5	5
Max. leaf nodes	32	32	32
Min. samples per leaf	10	21	12
Cost-complexity param.	6·10–7	2·10–7	3.5·10–7

Using these hyperparameters, we can train the model with the full training set and estimate the test outcomes. To properly evaluate the suitability of the obtained results, it is not enough to simply compute the error made by the model. This should be put in context. To do so, we will compare these results with those obtained through alternative approaches. First, we will calculate the maximum error that we should expect in each of the tasks. This error is obtained by using the mean value observed in the training set as the prediction for each variable. If a model cannot improve this value (the worst case), then the model is considered completely useless. Secondly, we will compare the results with other interpretable models, which will allow us to address, from another perspective, the balance between interpretability and performance. In this case, the comparison models will be a linear regression and a decision tree with the same depth as the model tree. Finally, we will compute the performance in the test set of a gradient-boosting decision tree (GBDT). GBDTs are not interpretable, but they are algorithms that usually achieve good results in a wide variety of tasks. Therefore, we will use it as a reference of what could be the minimum possible error for the given task. These results are summarized in Table 2.

As for vProduct1, the model tree yields the best result among the interpretable models, with a mean RMSE below the decision tree's and substantially lower than the linear regression's. Analyzing the results from a general perspective, we can observe that it achieves an error relatively close to that of the GBDT and almost three times lower than that of the worst case. Although the difference in performance relative to the GBDT is still notable, it is important to remember that this is a much more complex and non-interpretable model, so the result obtained by the model tree can be considered reasonably good. Therefore, this approach achieves an optimal balance between interpretability and performance. The case of eMinOutput is practically analogous to the previous one. The main difference is that both the decision tree and the model tree obtain results closer to the GBDT than with vProduct1, making this model even more appropriate.

However, the results obtained for vCirPF are less promising. Even

Table 2
Performance of the developed model tree compared to other approaches for the network-constrained high wind penetration UC problem.

	vProduct1		vCirPF		eMinOutput	
Variable	mean RMSE	std RMSE	mean RMSE	std RMSE	mean RMSE	std RMSE
Worst case	0.1353	0.1588	0.1269	0.0887	0.0910	0.0091
Linear regression	0.0720	0.0798	0.0645	0.0518	0.0628	0.0033
Decision tree (5)	0.0541	0.0659	0.0928	0.0686	0.0364	0.0021
Model tree (5)	0.0496	0.0641	0.0732	0.0587	0.0327	0.0017
GBDT	0.0401	0.0508	0.0510	0.0385	0.0295	0.0022

though the mean RMSE of the model tree is below the worst cases, it is far from that of the GBDT. The linear regression obtains a better result for this set of outputs. This fact may seem surprising, but we should remember that the linear regression model is fitted with the whole set of input features, while the model tree chooses a single variable for each terminal node. This characteristic is a major limitation of this model since the flow through the power lines is a much more complex variable than those associated with generators. The model tree does not have enough flexibility to capture all the existing relationships in the data. In any case, although linear regression achieves better performance, the size of the problem (15 features with their corresponding parameters and intercept for each of the 186 outputs) substantially complicates the interpretability of the model, so it would not necessarily be optimal. In addition, we can compare the above results with those obtained in the alternative UC problems defined above. As shown in Table 3, the mean RMSE obtained when the impact of the network is neglected is lower than in the base case. This is reasonable since, by eliminating the grid, the limitations that these may have on the operation of the generators are eliminated, and it will be easier to estimate these values. Similarly, reducing the penetration of wind generation reduces one of the main sources of variability in the problem, which can also lead to limitations on the generators, both due to the load ramp and their influence on grid constraints. Therefore, it is reasonable that the results also improve in this alternative case.

Subsequently, we will represent the model tree developed for vProduct1, selecting some representative generators to illustrate its interpretability. In line with what we observed in the feature selection stage, most of the partitions of the input space are based on the total net demand features since they have the most influence on the output of the generators. The variables associated with specific nodes appear at the bottom of the tree, where the tree has been able to segment more local relationships, which only take place in particular partitions of the input space.

Noteworthy in this figure is that not all outputs have a linear dependence on the variable associated with each terminal node. Consequently, the resulting model is relatively sparse, which facilitates its interpretation since the previous value of the decision tree is maintained.

Additionally, one can observe the expression $ord(h) > i$, with $i \in \{1, 2, 3\}$, as a criterion for tree partitioning. This is not because the period number is included as an input variable but because the model identifies that information is missing for certain variables and decides to segment the training samples when this occurs. Therefore, this logic is equivalent to that shown in the figure, with the cutoff time being dependent on the variable taken by the model.

Table 3
Performance of the developed model tree in the proposed UC problem alternatives.

	vProduct1		vCirPF		eMinOutput	
Variable	mean RMSE	std RMSE	mean RMSE	std RMSE	mean RMSE	std RMSE
Single node HWP	0.0423	0.0567	–	–	0.0295	0.0013
Network-const. HWP	0.0496	0.0641	0.0732	0.0587	0.0327	0.0017
Network-const. LWP	0.0409	0.0495	0.0536	0.0469	0.0308	0.0016

5.3. Classification models

In the same way that we have carried out the selection of hyperparameters for the regression model, we must repeat the process for the classification model. In this case, the metric used will be accuracy, which measures the proportion of samples that have been correctly classified. Therefore, we will evaluate the performance of the models by calculating the average accuracy of the set of output variables. Fig. 5 shows the average accuracy of a single decision tree for all variables versus that obtained by training one model per variable. For lower depths, the performance of the single decision tree is significantly worse than that of the multiple decision trees since the latter have much greater flexibility. However, this difference is drastically reduced as the maximum depth of the trees increases. Again, it can be concluded that a single decision tree is more effective in terms of interpretability and accuracy than the use of one tree for each variable. Still, in this case, the depth required to achieve comparable results is approximately two levels higher than that of the multiple trees.

Analyzing the previous graph, a maximum depth of 6 has been chosen for the decision tree classifiers. Although the representation of a tree of these dimensions can be complicated, the clustering of terminal nodes will allow compacting the result in an easier way to analyze.

The resulting hyperparameters for the models studied are shown in Table 4. The values obtained are equivalent to those corresponding to the regression model, although taking into account that, in this case, the depth has been increased by one level. One element to note is that, in this case, the maximum depths do not correspond to those of fully-grown trees. Instead, a smaller maximum number of terminal nodes has been selected.

Employing the hyperparameters shown in the previous table, we will train a decision tree for each output type. In this case, we do not intend to develop a model tree but to cluster both the terminal nodes and the output variables themselves. Fig. 6 illustrates the average accuracy of the model as a function of the number of unique terminal node output combinations. As can be seen, from 44 unique nodes onwards, the accuracy remains constant. This implies that there are 12 terminal nodes whose output combinations are precisely the same as those of other leaves in the tree. This kind of redundancy, more typical in classification tasks, does have an essential impact on the model and makes it difficult to understand.

Additionally, the number of unique terminal nodes can be further reduced without significantly affecting the model's performance since the outputs in some of them are virtually identical. For vCommit, the number of unique terminal nodes chosen will be 39. Interestingly, even though the depth of the tree is 6, the number of different nodes will be close to the 32 that a fully-grown decision tree of depth five would have, which has great benefits from an interpretation point of view.

Similarly, Fig. 7 shows the average model accuracy as a function of the number of generator clusters. From the beginning, it can be observed

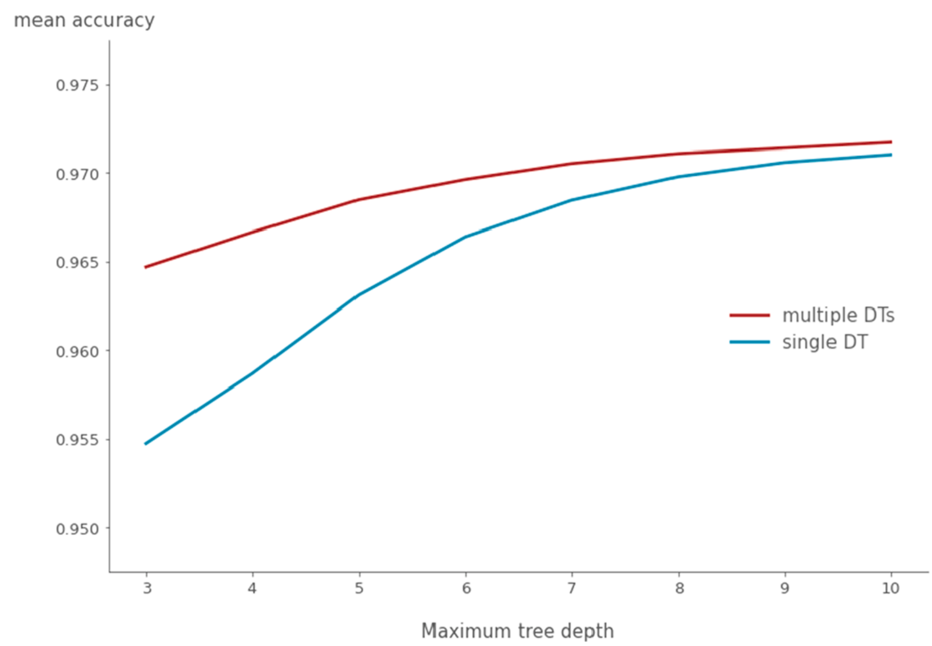

Fig. 5. Decision tree accuracy relative to its maximum depth for vCommit in the network-constrained high wind penetration UC problem.

Table 4
Selected hyperparameters for the classification models in the network-constrained high wind penetration UC problem.

Hyperparameter	vCommit	eMinOutput
Max. depth	6	6
Max. leaf nodes	56	60
Min. samples per leaf	12	6
Cost-complexity param.	5·10ˆ-7	1.5·10ˆ-6

that approximately 30 of these generators show a commitment behavior identical to that of other generators in the system. Most of these are generators that remain off in all the scenarios considered. In addition, we can find another five generators whose operation is very similar, so aggregating them will not imply a relevant loss of accuracy. Consequently, the number of generator clusters used for this model will be 19.

Considering both factors, we have reduced the dimension of the decision tree from 56 terminal nodes with 54 generators to only 39 unique terminal nodes with 19 generator clusters. This represents a reduction in the number of unique parameters at the terminal nodes of more than 75 % compared to the original case, with virtually no change in model performance (the mean accuracy has been reduced by an amount well below 1 %).

Using this methodology, we can evaluate the accuracy of the model using the test set. Additionally, we will compare these results with those obtained using alternative approaches (Table 5), as we did with the model tree.

The minimum expected mean accuracy for each task will be obtained by calculating the accuracy we would obtain by predicting the mode of each of the output variables in the training set.

As alternative interpretable models, we will include logistic regression, and the decision tree equivalent to the one elaborated, but without performing the clustering processes. Therefore, the latter results can be expected to be just slightly above those obtained by the proposed model since the impact of clustering is very small. Finally, as explained above, we will use a GBDT as a reference for the accuracy that could be achieved using a model without any limitation in terms of complexity.

Firstly, we can observe that the variability of vCommit is relatively low since the minimum expected accuracy is above 90 %. The three interpretable models presented achieve very similar performance and

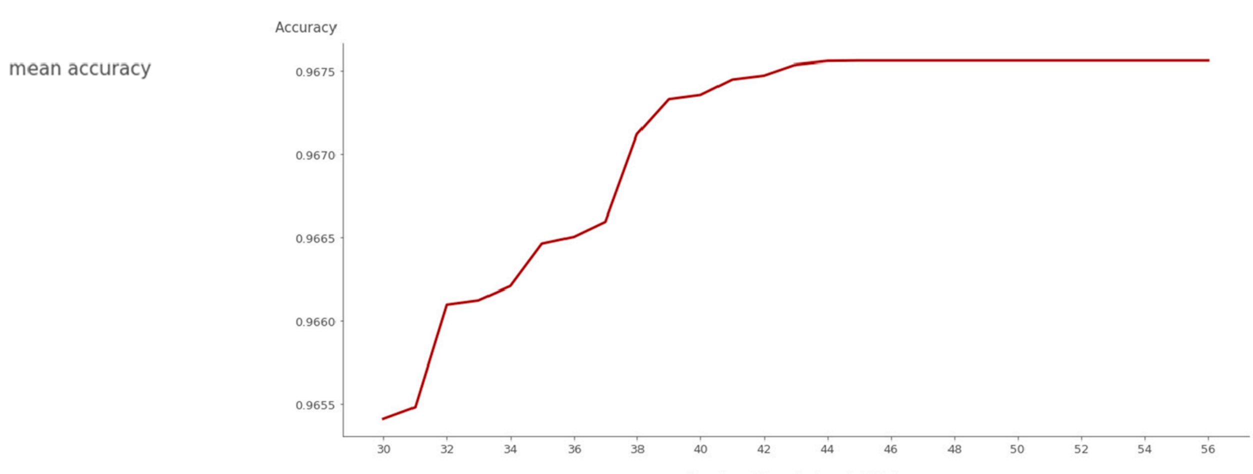

Fig. 6. Mean validation accuracy of the vCommit decision tree classifier relative to the number of unique terminal node output combinations.

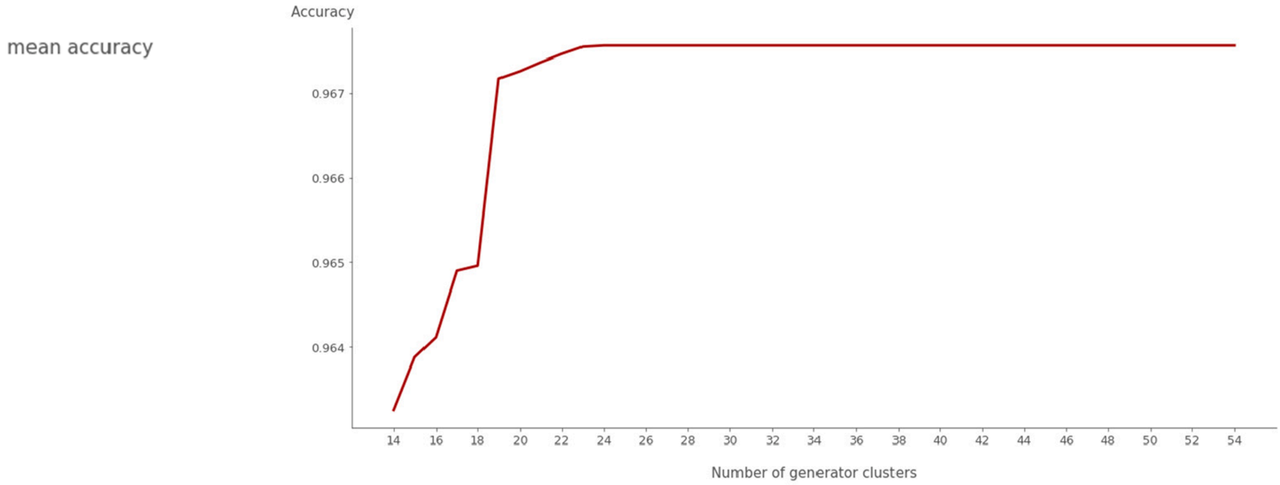

Fig. 7. Mean validation accuracy of the vCommit decision tree classifier relative to the number of generator clusters.

Table 5
Performance of the developed decision tree classifier compared to other approaches for the network-constrained high wind penetration UC problem.

Variable	vCommit mean accuracy	vCommit std accuracy	eMinOutput mean accuracy	eMinOutput std accuracy
Worst case	0.9102	0.1249	0.6376	0.0963
Logistic regression	0.9662	0.0549	0.8927	0.0434
Decision tree (6)	0.9677	0.0470	0.9219	0.0227
Clustered D.T. (6)	0.9673	0.0481	0.9207	0.0242
GBDT	0.9768	0.0334	0.9466	0.0176

are satisfactorily close to that obtained by the GBDT, which validates all the approaches. However, the interpretation of the logistic regressions is much less evident than that of the linear regressions and certainly less than that of the decision trees, so their presented performance would not justify their use. As for the two decision tree classifiers, it is clear that the former will achieve a slightly superior result. However, it can be seen that the difference between both models is practically negligible, so it can be concluded that the clustered decision tree presents the best balance between interpretability and accuracy.

Second, we analyze the results obtained for estimating eMinOutput in its binary form, indicating whether the associated constraint is active or not. In this case, the variability is much higher since the minimum accuracy is considerably low. Once again, the clustered decision tree greatly improves this result and stands out as the optimal interpretable algorithm among those presented here. However, even though the resulting accuracy is not excessively far from that obtained by the GBDT in relative terms, it is a remarkable difference in absolute terms.

Comparing the results obtained for the different cases studied, the conclusions drawn are analogous to those of the regression model. Either by omitting the impact of the network or by reducing the penetration of wind generation, the operation of generators is less limited by external constraints, which facilitates the modeling of their production. In particular, Table 6 seems to indicate that the constraints due to renewable generation in the base case have a greater impact or are more complicated to model than those of the power grid since reducing this factor improves the resulting accuracy to a greater extent than eliminating the network.

Finally, we will represent the clustered decision tree elaborated for the set of vCommit outputs. The mapping of the reference generators that are shown in the previous table allows identifying the rest of the generators in the same cluster is found in Table 7. Analyzing the representation of the decision tree in Fig. 8, it can also be seen that the system-wide input features are the ones that dominate the initial partitions of the input space, with the features related to specific nodes appearing at the bottom of the tree, where they play an essential role in identifying sets of scenarios that present similar behavior.

Another relevant observation involves the duplicity of terminal nodes, generally (but not exclusively) found in nearby branches according to the tree representation. This is one of the usual problems of decision tree classifiers, as mentioned above. While in certain contexts,

Table 6
Performance of the developed decision tree classifier in the proposed UC problem alternatives.

Variable	vCommit mean accuracy	vCommit std accuracy	eMinOutput mean accuracy	eMinOutput std accuracy
Single node HWP	0.9704	0.0493	0.9331	0.0245
Network-const. HWP	0.9673	0.0481	0.9207	0.0242
Network-const. LWP	0.9749	0.0437	0.9615	0.0261

Table 7
Mapping of the existing generators with their corresponding cluster reference generator.

Reference generator	Clustered generators
1	1, 2, 3, 8, 9
4	4
5	5, 27, 28
6	6, 30
7	7, 12, 13, 16, 17, 18, 19, 22, 23, 25, 26, 31, 32, 33, 38, 41, 42, 46, 47, 48, 49, 50, 51, 52, 53, 54
10	10
11	11, 21, 39
14	14
15	15
20	20
24	24
29	29
34	34
35	35
36	43
37	37
40	40
44	44
45	45

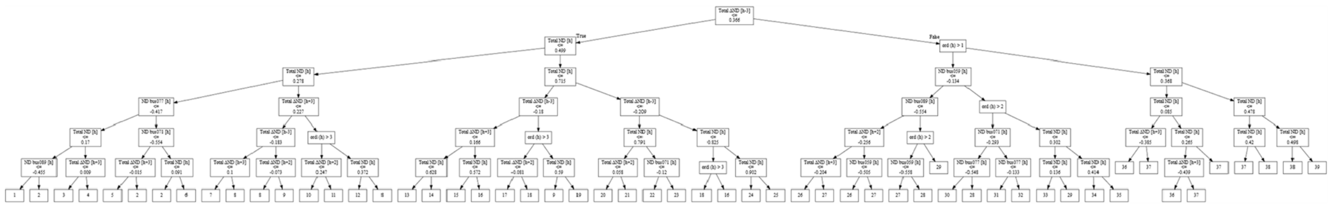

Fig. 8. Classification tree that explains the commitment variable.

decision lists can overcome this problem by obtaining more interpretable results than a decision tree itself, this problem still has many unique nodes. For that purpose, a set of conditions would have to be defined to replicate the space's corresponding partitions. This would result in an excessive number of complex conditions that would negatively impact the interpretability of the model. In this case, the exclusivity and exhaustivity of the decision three partitions are important advantages.

Finally, Table 7 allows us to understand how the generators have been grouped in each cluster. The most remarkable group is the one identified with generator 7, which gathers the set of generators that are not committed in any or practically none of the periods with which the model has been trained. Similarly, generator 5 represents the generators that will always be committed according to the decision tree classifier. The rest of the clusters correspond to either single generators or groups of generators that, due to their characteristics, present a very high correlation in their mode of operation. The clustering step addresses Question 3, "What are the intrinsic links among variables in the problem?".

6. Conclusions and discussion

In this paper, interpretable machine learning models have been shown to explain in a human-understandable way how the unit commitment problem applied to a specific system generates the optimal solutions and independently predicts the decision variables and dual variables of the problem. An example expression of the insights about the UC solutions that can be obtained through the proposed methodology can be seen in the proposed questions:

– What are the variables in system operation that show variation across scenarios and are therefore worth studying? (Question 0) Humans participating in system operation will know, for instance, if a given generator is always on or off or whether it is subject to changes depending on the situation. This question is addressed by the first step in the methodology, data preparation.
– What are the most important features of a scenario, i.e., the ones that have the largest impact on the variables of interest? (Question 1). This is addressed in the feature selection step.
– What are the specific dynamics of the variables of interest with respect to the important scenario features? (Question 2) This description should be simple enough to be understood by a human but still accurate enough to be able to give useful information. This is addressed in model development (regression for continuous and classification for binary variables).
– What are the intrinsic links among variables in the problem? (Question 3) That is, are there joint dynamics for variable pairs or groups of variables? This is addressed by the node clustering step.

Two models, a regression and a classification model, have been developed to estimate the continuous and binary variables of the UC, respectively. Both are based on multi-output decision trees since they offer the best balance between performance and interpretability among all intrinsically interpretable algorithms. On the one hand, the UC problem is complex and highly nonlinear, so decision trees have some advantage over other linear models in addressing the task. On the other hand, the possibility of jointly predicting variables of the same type makes interpreting the resulting model significantly easier than with any other approach. Training a single algorithm for each output would not be manageable; therefore, neither would they be properly interpretable. In addition, much of the interaction between the output variables would be lost.

The UC problem presents numerous full- or stepwise linear relationships, which cannot be modeled by a decision tree, given that in a tree, nodes relate to specific values for the variables. A linear regression tree can capture the linear relationships that apply in a particular context of the variables. This approach enables capturing a large amount of information that would escape a standard decision tree without compromising its interpretability.

Conversely, the problem faced by decision tree classifiers is related to the redundancy of sub-trees and terminal nodes, which repeat information unnecessarily. This hinders the interpretability of the model since it implies repeatedly representing the same parameters multiple times. Therefore, in this paper, we propose clustering these nodes, which will be compactly represented in an attached look-up table. In this way, nodes presenting an identical combination of outputs can be immediately identified. Furthermore, a significant redundancy has also been detected in the output variables themselves. Therefore, they have also been grouped by generator clusters, further reducing the number of parameters needed to describe the classification model.

The results obtained by applying both approaches to the UC problem studied in this work show that, in general, they achieve an optimal balance between performance and interpretability, usually outperforming the rest of the intrinsically interpretable algorithms in both aspects. In fact, the mean RMSE or accuracy of these models is relatively close to those obtained using black-box machine learning models, as is the case of GBDTs.

We should highlight that the models not only depend on the topology of the system with which the optimal solutions to the problem have been obtained but also, very importantly, on the scenarios that have been elaborated to obtain these solutions. These models, by default, try to extract the most general and valuable relationships they find in the data. Therefore, they will tend to learn the behavior of the UC model primarily in the most common sets of scenarios while potentially disregarding extreme or rare cases, which may not be found relevant either because of their low frequency or the error incurred when they are omitted.

If we want to understand the performance of the UC in both common and rare situations, the solution is to balance the training set so that the model perceives all of them as equally frequent. However, this solution would not be the most appropriate since we would be forcing the model to replicate specific relationships that, according to the dataset, are a minority without providing more helpful information for the model to carry out such a task. Therefore, the algorithm will try to learn these relationships, even if they are not actually general enough. Analyzing the results in such a context would be meaningless since, if we also balance the test set itself, we are modifying the variability of the test set in our favor. If, on the other hand, we evaluate the model with the original distribution, the results will necessarily be worse since we have focused the model on learning certain relationships, which are not the most general ones. While interpretability is paramount, even above the algorithm's own accuracy, it should not be ignored entirely. Suppose the model cannot replicate the data it intends to explain. In that case, it cannot be guaranteed to provide a reliable explanation of how the results were obtained [26]. Consequently, in order to carry out analyses of this type, it would be more appropriate to develop a set of scenarios that appropriately reflect the behaviors we seek to explain through the application of an IML model so that the model can suitably discern those cases.

Finally, it is essential to mention that even though the presented models shed light on the functioning of the UC problem and explain how it obtains the optimal solutions, the obtained results do not let us consider them surrogate models entirely equivalent to the optimization problem itself. Not even the models based on GBDTs, which have been employed as a reference in this paper, would be sufficiently accurate to consider such a possibility.

This conclusion is the same as that of numerous studies [5]. Therefore, the developed interpretable models should be understood as substitutes and complements to the UC problem. Our method complements the solution of the UC via optimization because it is able to present, in terms that are easily understood, how system conditions determine the generating units that need to be committed, and how any increments in load will be served using these units. The tree can be evaluated much quicker than solving the UC, which can be useful when integrating the UC solution in a higher-level planning solution. It should be stressed that this substitution can only be possible after the UC has been solved via optimization for each scenario considered, as the tree is built based on the solutions from optimization.

These trees are extremely useful tools to understand, in a general way, the operation of the optimization problem, to estimate which constraints will be active in the optimal solution of a scenario and for what reasons, or to predict approximate solutions, which can also be used as a starting point for the optimization itself.

One of the main challenges of intrinsically interpretable models, especially when dealing with such complex tasks as unit commitment, is achieving results that *compete* with other highly accurate but completely opaque models.

As indicated, interpretability has value, meaning it is unnecessary to aim for identical results. However, any effort to improve its accuracy will validate it as a reliable explanation of the underlying UC model.

Alternative models, clustering methods, or feature selection techniques could improve our future results. More developed visualization techniques could be useful in making insights more apparent and could be explored in further research. However, this paper has already shown that interpretable machine learning can be a useful tool to automatically extract insights about system operation that were previously only available through human experience.

CRediT authorship contribution statement

Sara Lumbreras: Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Diego Tejada:** Writing – review & editing, Formal analysis, Data curation. **Daniel Elechiguerra:** Writing – review & editing, Visualization, Methodology, Investigation, Formal analysis.

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.

References

[1] Bertsimas D, Litvinov E, Sun XA, Zhao J, Zheng T. Adaptive robust optimization for the security constrained unit commitment problem. IEEE Trans Power Syst 2013;28(1):52–63.
[2] Foley AM, Leahy PG, Marvuglia A, McKeogh EJ. Current methods and advances in the forecasting of wind power generation. Renew Energy 2012;37(1):1–8.
[3] Hossain M, Mekhilef S, Danesh M, Olatomiwa L, Shamshirband S. Application of extreme learning machine for short term output power forecasting of three grid-connected PV systems. J Clean Prod 2017;167:395–405.
[4] Sendaula MH, Biswas SK, Eltom A, Parten C, Kazibwe W. Application of artificial neural networks to unit commitment. In: *Proceedings of the first international forum on applications of neural networks to power systems,* pp. 256-260, 1991.
[5] Mohammadi F, Sahraei-Ardakani M, Trakas D, Hatziargyriou ND. Machine learning assisted stochastic unit commitment during hurricanes with predictable line outages. IEEE Trans Power Syst 2021.
[6] Huang SJ, Huang CL. Application of genetic-based neural networks to thermal unit commitment. IEEE Trans Power Syst 1997;12(2):654–60.
[7] Ouyang Z, Shahidehpour SM. A hybrid artificial neural network-dynamic programming approach to unit commitment. IEEE Trans Power Syst 1992;7(1):236–42.
[8] Nayak R, Sharma JD. A hybrid neural network and simulated annealing approach to the unit commitment problem. Comput Electr Eng 2000;26(6):461–77.
[9] Baker K. Learning warm-start points for ac optimal power flow. In: *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing,* pp. 1-6, 2019.
[10] Xavier AS, Qiu F, Ahmed S. Learning to solve large-scale security-constrained unit commitment problems. INFORMS J Comput 2020;33(2):419–835.
[11] Blanco I, Morales JM. An efficient robust solution to the two-stage stochastic unit commitment problem. IEEE Trans Power Syst 2017;32(6):4477–88.
[12] Meus J, Poncelet K, Delarue E. Applicability of a clustered unit commitment model in power system modeling. IEEE Trans Power Syst 2018;33(2):2195–204.
[13] Latha PG. An efficient machine learning algorithm for unit commitment problem. Int J Appl Eng Res 2018;13(3):135–41.
[14] J. E.A., I. A. T.P., y J. R. V.P., «Reinforcement Learning Solution for Unit Commitment Problem through Pursuit Method,» *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies,* pp. 324-327, 2009.
[15] Coronado CA, Figueroa MR, Roa-Sepulveda CA. A reinforcement learning solution for the unit commitment problem. In: *2012 47th International Universities Power Engineering Conference,* pp. 1-6, 2012.
[16] Dalal G, Mannor S. Reinforcement learning for the unit commitment problem. In: *2015 IEEE Eindhoven PowerTech,* pp. 1-6, 2015.
[17] Molnar C. Interpretable machine learning. A Guide for Making Black Box Models Explainable, Leanpub, 2019.
[18] Tejada-Arango DA, Lumbreras S, Sánchez-Martín P, Ramos A. Which unit-commitment formulation is best? A comparison framework. IEEE Trans Power Syst 2020;35(4):2926–36.
[19] Aminifar F, Fotuhi-Firuzabad M, Shahidehpour M. Unit commitment with probabilistic spinning reserve and interruptible load considerations. IEEE Trans Power Syst 2009;24(1):388–97.
[20] Hu B, Wu L, Marwali M. On the robust solution to SCUC with load and wind uncertainty correlations. IEEE Trans Power Syst 2014;29(6):2952–64.
[21] Jiang R, Wang J, Zhang M, Guan Y. Two-stage minimax regret robust unit commitment. IEEE Trans Power Syst 2013;28(3):2271–82.
[22] Sahin C, Shahidehpour M, Erkmen I. Allocation of hourly reserve versus demand response for security-constrained scheduling of stochastic wind energy. IEEE Trans Sustain Energy 2013;4(1):219–28.
[23] Tejada-Arango DA. «GitHub,» 11 March 2021. [En línea]. Available: https://github.com/datejada/modified-ieee-118-bus-system-data.
[24] Staffell I, Pfenninger S. Using bias-corrected reanalysis to simulate current and future wind power output. Energy 2016;114:1224–39.
[25] Singh D, Singh B. «Investigating the impact of data normalization on classification performance,» *Applied Soft Computing,* vol. 97, n° B, 2020.
[26] Murdoch WJ, Singh C, Kumbier K, Abbasi-Asl R, Yu B. Definitions, methods, and applications in interpretable machine learning. In: *Proceedings of tje National Academy of Sciences of the United States of America,* vol. 116, n° 44, pp. 22071-22080, 2019.