Applied Energy 306 (2022) 118049

Contents lists available at ScienceDirect

Applied Energy

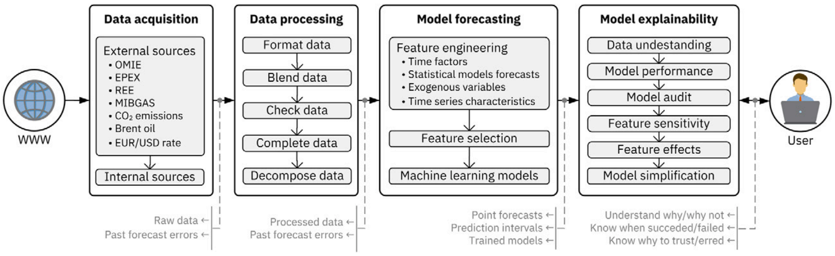
journal homepage: www.elsevier.com/locate/apenergy

Framework for collaborative intelligence in forecasting day-ahead electricity price

Sergio Beltrán [a,b,*], Alain Castro [a,b], Ion Irizar [a,b], Gorka Naveran [c], Imanol Yeregui [d]

[a] *Ceit - Basque Research and Technology Alliance (BRTA), Manuel de Lardizábal 15, 20018 Donostia - San Sebastián, Spain*
[b] *Universidad de Navarra, Tecnun, Manuel de Lardizábal 13, 20018, Donostia - San Sebastián, Spain*
[c] *Giroa - Veolia Servicios Norte SA, Laida Bidea Edificio 407, 48170 Zamudio, Spain*
[d] *Genelek Sistemas SL, Plaza Urola, 20750 Zumaia, Spain*

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Augmented analytics
Automated machine learning
Ensemble models
Explainable artificial intelligence
Time series decomposition
Time series hybrid models

ABSTRACT

Electricity price forecasting in wholesale markets is an essential asset for deciding bidding strategies and operational schedules. The decision making process is limited if no understanding is given on how and why such electricity price points have been forecast. The present article proposes a novel framework that promotes human–machine collaboration in forecasting day-ahead electricity price in wholesale markets. The framework is based on a new model architecture that uses a plethora of statistical and machine learning models, a wide range of exogenous features, a combination of several time series decomposition methods and a collection of time series characteristics based on signal processing and time series analysis methods. The model architecture is supported by open-source automated machine learning platforms that provide a baseline reference used for comparison purposes. The objective of the framework is not only to provide forecasts, but to promote a human-in-the-loop approach by providing a data story based on a collection of model-agnostic methods aimed at interpreting the mechanisms and behavior of the new model architecture and its predictions. The framework has been applied to the Spanish wholesale market. The forecasting results show good accuracy on mean absolute error (1.859, 95% HDI [0.575, 3.924] EUR $(\mathrm{MWh})^{-1}$) and mean absolute scaled error (0.378, 95% HDI [0.091, 0.934]). Moreover, the framework demonstrates its human-centric capabilities by providing graphical and numeric explanations that augments understanding on the model and its electricity price point forecasts.

1. Introduction

Electricity price forecasting in wholesale markets has become an essential asset for the energy sector. Since the early 1990s, vertically-integrated monopoly structures that have traditionally regulated electrical generation, transport and distribution have been replaced by deregulated, liberalized markets. Electricity is now commonly traded in competitive auctions (called pools and power exchanges), where generating companies submit energy offers and their corresponding price, and consumption companies bid for them. A single-round auction is performed for every hour of the next day to determine the market

[*] Corresponding author at: Ceit - Basque Research and Technology Alliance (BRTA), Manuel de Lardizábal 15, 20018 Donostia - San Sebastián, Spain.
E-mail address: sbeltran@ceit.es (S. Beltrán).
URL: https://ceit.es (S. Beltrán).

https://doi.org/10.1016/j.apenergy.2021.118049
Received 8 April 2021; Received in revised form 22 September 2021; Accepted 9 October 2021
Available online 23 October 2021

Acronyms

ACF	Auto-correlation function
ALM	Advanced linear model
ARIMA	Autoregressive integrated moving average model
AutoML	Automated machine learning
AT	Aiolfi and Timmermann method
BART	Bayesian additive regression trees
BaLM	Bayesian linear model
BG	The Bates and Granger method
BoLM	Gradient boosting linear model
CEEMDAN	Complete ensemble empirical mode decomposition with adaptive noise
CLS	Constrained least squares
DWT	Discrete wavelet transformation
EIG_1	Standard eigenvector method
EIG_2	Bias-corrected eigenvector method
EMD	Empirical mode decomposition
FFNN	Feed-forward neural network
GBDT	Gradient boosting decision tree
GP	Gaussian process
HDI	Highest density interval
HDV	Highest density value
ICE	Individual conditional expectation
IM	Interquartile mean
KNN	k-nearest neighbors algorithm
LAD	Least absolute deviation
LC	Linear combination
LM	Linear model
LOESS	Locally estimated scatterplot smoothing
MAE	Mean absolute error
MASE	Mean absolute scaled error
MARS	Multivariate adaptive regression splines
MED	Median
MODWT	Maximal overlap discrete wavelet transformation
NG	Newbold and Granger method
OLS	Ordinary least squares
PACF	Partial auto-correlation function
PD	Partial dependence
PLM	Penalized linear model
PLS	Partial least squares
RF	Random forests and extremely randomized trees
RFE	Recursive feature elimination
RIBM	Rule- and instance-based model
RLM	Regularized linear model
SA	Simple average
SRC	Standardized regression coefficients
STL	Seasonal and trend decomposition using LOESS
SVM	Support vector machine
VMD	Variational mode decomposition

clearing price that results when energy supply bids match predicted demand [1]. This market liberalization has promoted significant efficiency improvements, stimulated technical innovation and led to investments in generation [2].

Nonetheless, competitive markets have also brought in price uncertainty. As electricity cannot be economically stored, complex price dynamics have arisen from the different market participants' strategies, including expected energy supply. The latter is especially relevant in recent years due to the increasing expansion rates of renewable energy sources, whose market offers clearly depend on changing weather conditions. For example, Baldick [3], Ketterer [4] and Martínez-Anido et al. [5] concluded that price volatility was aggravated by increasing wind penetration for the markets in Texas, Germany and New England, respectively. Looking at the trends on renewable energy source expansion, prices are expected to be more volatile than at present [6,7].

Price uncertainty leads to financial distress for market participants. Producers and consumers rely on price forecasts to prepare their corresponding bidding strategies to maximize profits. As the energy amount bid is usually substantial, the financial penalties for forecast errors can be very high. For example, Zareipour et al. [8] report that a 1% improvement in the forecast error would result in cost reductions of 0.1% to 0.35% for industrial consumers in Ontario's electricity market, which results to circa 1.5 USD M year^{-1} for a medium-size utility [9].

The forecasting errors also impact the economic efficiency of energy production and consumption schedule. Operational risks have been verified in industrial load scheduling [10], battery energy storage systems [11], thermal-based plants [12], and combined-cycle, coal-fired, cascade hydro and pumped-storage power plants [13]. Moreover, renewable energy sources particularly suffer from price uncertainty as errors have a direct impact on the economic efficiency of the resulting allocation [14]. These losses have been verified in several hydro-based generation sources [15,16].

The minimization of these financial distress and operational risks have made day-ahead electricity price forecasting increasingly important in today's energy sector. As consequence, it is currently one of the major topics of research in energy economics and finance [17].

1.1. Electricity price forecasting

Day-ahead electricity price forecasting focuses on predicting the next 24 clearing prices in wholesale markets.[1] Review and survey publications usually classify the forecasting techniques into five model groups: fundamental, multi-agent, reduced-form, statistical or econometric, and machine learning or computational intelligence [18]. As statistical and machine learning models have been shown to yield the best results [19], they are the focus of this section, and in turn, of the base methods that will be applied in this article.

Statistical time series models commonly include similar-day (or *naïve*) methods, which show good performance for stable market periods; exponential smoothing methods are robust against outliers; regression methods are good at handling linear relationships; auto-regressive-type methods are accurate for short range response; and generalized autoregressive conditional heteroskedastic methods are aimed at modeling price volatility [20].

Machine learning models are better at dealing with complexity and non-linearity. These characteristics arise mostly from the multiple regressors' influence on the electricity price — *e.g.*, expected energy load or generation capacity. These methods typically include artificial neural networks, which have shown high performance, and support-vector machines, which are efficient when dealing with regressors [21]. A more detailed review of the different models applicable to electricity price, including their benefits and weaknesses, can be found in [22–25].

Such a variety of models is evidence that no universal forecasting individual method works best for all markets and situations [26,27]. Nonetheless, the present authors find that efficiently using each model where it excels, and then combine them in an effective way is a

[1] Due to changes on daylight saving time, the number of forecast hours is 23, 24 or 25.

promising field of research. Thus, the first major contribution in the present article is not about developing a new forecasting model but rather proposing an appropriate architecture to combine existing ones. To that end, four methodologies seem to be promising directions for working beyond the state of the art: (i) addition of exogenous features, (ii) time series decomposition, (iii) time series feature extraction, and (iv) combining forecasts.

(i) Adding exogenous features (or regressors) is the process of acquiring and using contextual information to enhance forecasting accuracy. Electricity price is sometimes forecast based only on its own historical patterns. Nonetheless, the clearing prices strongly depend on external factors [28]. Several publications have demonstrated an increase in forecasting performance by using exogenous factors, such as system load [29], ambient temperature [30], wind generation [31] and market integration [32]. Still, there is progress to be made in the addition of more exogenous factors that might affect electricity price, such as CO_2 emission allowances, the fuel price of natural gas and oil, currency exchange rates and electrical generation capacity.

(ii) Time series decomposition aims at deconstructing the series into several components, each representing one of the underlying patterns. Electricity price is a complex non-linear and non-stationary time series that suffers from abrupt spikes and multiple frequencies. A divide-and-conquer strategy can improve price forecast accuracy by predicting the (more distinctive and identifiable) individual components, and then combining their forecasts. This has been demonstrated with three decomposition methods: discrete wavelet transformation (DWT, [33]), empirical mode decomposition (EMD, [34,35]) and variational mode decomposition (VMD, [36]). Seasonal and trend decomposition using LOESS (STL) has not yet been applied to forecasting electricity price, although it has shown good performance in predicting other commodities [37,38]. In addition, the possibility of combining these four decomposition methods remains to be explored. The objective would be to highlight the advantages of each approach, so that the characteristics of the electricity price series could be completely individualized.

(iii) Time series feature extraction reduces each series section into structural characteristics by means of methods in the domain of signal processing and time series analysis. The methods range from basic statistical equations, such as mean or maximum value, to more sophisticated measures, such as entropy or non-linearity. This methodology has been successfully applied for time series classification [39], clustering [40] and anomaly detection [41]. In the area of electricity price forecasting, the effects of time series feature extraction are currently unknown.

(iv) Combining forecasts into a single forecast aims to reduce the risk associated with selecting an individual forecasting model [42]. The reason lies in the lack of an individual model that captures all patterns in the data, concurrently. Thus, combining forecasts created from different models usually improves accuracy due to more comprehensive pattern recognition [43]. The key step in constructing an ensemble of forecasts is choosing a linear or non-linear combination function and appropriate weights for each base forecast. In this sense, combining forecasts approaches are usually classified into linear and non-linear forms.

The linear combination of forecasts in electricity price forecasting is a mature methodology that has been shown to reduce forecast uncertainty [19]. In spite of the frequently used linear ensembles, few studies have addressed non-linear combination approaches. However, it has been demonstrated in other fields that more satisfactory results can be obtained with non-linear approaches than with linear ones [44]. Given the increase of computational power and advances in machine learning algorithms, this article addresses the potential of using machine learning models for non-linear combination of forecasts. In addition, both linear and non-linear approaches will be combined using the stacked generalization methodology with the primary objective of increasing overall model performance [45,46].

Stacked generalization has been theoretically proven to represent an asymptotically optimal system for learning [47]. Moreover, under most conditions, the theory also guarantees a better combined forecast performance than can be achieved by any single forecast alone. This has been practically demonstrated in multiple fields [48]. In spite of this, surprisingly, aside from [49,50], we are not aware of any use of stacking approaches in electricity price forecasting. Moreover, diversity – that is, the difference among the individual machine learning models – is a fundamental key in stacking generalization [51]. However, [49,50] limit the base learners to relevance vector machines and decision trees. Therefore, in addition to stacking, this article also contributes by applying a rich library of machine learning algorithms for electricity price forecasting.

Stacking generalization needs the appropriate model architecture and tuning of the models' hyper-parameters. To assure the proposed model architecture and chosen hyper-parameters achieve good results, open-source automated machine learning platforms (AutoML) will be applied here. To the best of our knowledge, this is the first time the performance of AutoML systems is shown in the field of electricity price forecasting. This constitutes the second major contribution of the present article.

1.2. Automated machine learning platforms

In recent years, an active field of research has developed around the progressive automation of machine learning. AutoML platforms initially emerged so that novice users could create useful models, while experts could use them to speed up their tasks. Nonetheless, as machine learning pipelines are growing in complexity and computational cost, AutoML is becoming a complementary tool that leverages humans' combined domain and technical knowledge [52]. AutoML is quickly gaining ground in a wide range of industrial applications. Some examples can be seen in the fields of medical image classification [53], online travel mode detection [54] and customer delivery satisfaction [55].

Throughout the years, several off-the-shelf open source packages have been developed to provide automated machine learning [56–59]. Among the most well-known, the ones that will be applied in the present article are two (i and ii). The description that follows focus on model architecture and hyper-parameter tuning.

(i) H2O AutoML [60] performs a random search to tune the hyper-parameters of four machine learning model families: feed-forward neural networks, gradient boosting decision trees, penalized linear models and random forests. In a second stage, it builds a stacked ensemble on all previously trained models and another one on the best model of each family. H2O AutoML is programmed in Java.

(ii) TPOT [61] constructs machine learning pipelines of arbitrary length using Python scikit-learn algorithms and the XGBoost model [62]. It performs features pre-processing, construction and selection, followed by hyper-parameter optimization. TPOT supports ensembling, sparse matrices and multiprocessing.

These AutoML platforms will help comparing the performance of the proposed stacked ensemble architecture with benchmark values. Using these baseline references, the stacked ensemble complexity will be increased until an asymptotic error is achieved. Nonetheless, providing better accuracy by increasing model complexity also decreases model interpretability in that humans do not understand their predictions as easily.

1.3. Collaborative intelligence

Interpretable models can aid explaining *why* point forecasts have been predicted to that specific numeric value. Understandable forecasts can then be trusted more, which could guide users in making better decisions. In the context of day-ahead electricity price forecasting, improving the decision making process is crucial for market participants since it means minimizing financial distress and operational risks. The current approach is limited to understanding the limitations of point forecasts by plotting prediction intervals and the densities around them [63]. Only traditional, usually simple, models are easier to explain, but at the expense of limiting performance accuracy [64]. The ultimate goal, then, is to be able to increase model complexity in order to enhance forecasting accuracy while keeping the forecasts understandable. For this reason, the present article proposes a human-centered collaborative intelligence framework for electricity price forecasting as its third and last major contribution.

Explainable machine learning (or explainable artificial intelligence) has generated a new flurry of research that aims to interpret the behavior of models and their outcomes [65]. It has emerged as a method for facilitating effective and efficient human–machine collaboration in order to enhance cognitive performance and, ultimately, improve decision-making [66]. The benefits of such human intelligence augmentation have appeared through diverse domains such as medicine, policy-making and science [67]. The fact that explainable machine learning is currently highly embedded in the financial services industry [68] and mandatory in the insurance sector [69] is evidence of these benefits.

The state-of-the-art literature commonly classifies explainable machine learning techniques into (i) model-specific methods and (ii) model-agnostic methods [64,70,71]:

(i) Model-specific methods are based on using intrinsically interpretable models. They include traditional, simple models such as linear and logistic regression models, generalized linear and additive models, decision trees and rule-based models. Novel types of models designed to be directly interpretable include explainable neural networks [72], generalized additive models plus interactions [73], explainable boosting machines [74], monotonically constrained gradient boosting machines [62], scalable Bayesian rule lists [75], and super-sparse linear integer models [76].

(ii) Model-agnostic methods use *post hoc* interpretation techniques to understand the predictions of a previously trained, non-directly interpretable model. They form a collection of visual artifacts that describe model behavior by providing specific insights into the mechanisms of the model and detailed information about why such answers were generated [77]. Their scope of interpretation can be classified as global when they help understanding the entire relationship modeled by the trained response function, and as local when they promote understanding of a single instance.

Up to now, intrinsically interpretable models can only learn some patterns of electricity price time series. To increase forecasting accuracy, a more complex, non-directly interpretable model has been proposed in the present article. Thus, hereinafter explainable machine learning will focus on *post-hoc* model-agnostic methods.

2. Contributions

The present article proposes a novel framework that promotes human–machine collaboration in forecasting day-ahead electricity price in wholesale markets. The framework is a human-centric solution that aims at augmenting market participants' decision-making capabilities by providing not only point forecasts, but above all explanations of the behavior of a new model architecture and its forecasts. In particular, the article makes three major contributions (i to iii) beyond the current state-of-the art in the electricity price forecasting sector:

(i) A model architecture that includes (i.i) a plethora of statistical models in order to learn different linear patterns; (i.ii) exogenous features that could possibly affect clearing prices; (i.iii) a combination of several time series decomposition methods; (i.iv) a collection of time series characteristics based on signal processing and time series analysis methods; (i.v) a stack ensemble of a diverse set of machine learning models for recognizing non-linear, complex patterns. The ensemble is fed by an efficient selection of a comprehensible feature engineering carried out in (i.i) to (i.iv).

(ii) The use of open-source AutoML platforms that provide a baseline reference for the proposed model architecture.

(iii) A collection of state-of-the-art model-agnostic methods aimed at interpreting the behavior of the forecasting models and their outcomes.

The proposed framework is applied to the case study of the Spanish wholesale market. Nonetheless, the framework has not been developed specifically for the Spanish market. Based on a transversal methodology, the framework can be applied to other wholesale markets of electricity price. The implementation of the proposed framework is empathized by putting it into production on an in-house proprietary server. In addition to promoting human–machine collaboration, the server includes the tools necessary for efficient data and model governance.

This article is structured in five sections. Section 3 describes the methodology, implementation and deployment of the proposed framework. Section 4 discusses the results achieved. To finish, Section 5 outlines the conclusions and highlights the key points.

3. Materials and methods

The proposed framework is divided onto four sequential phases (see Fig. 1). First, the Data Acquisition phase automatically captures all data needed to forecast day-ahead electricity price. Secondly, these raw data is duly processed and cleaned by the Data Processing phase. Thirdly, the Model Forecasting phase creates and selects new features. These variables feed a machine learning architecture. The result is a day-ahead forecast of electricity price points and their prediction intervals. Finally, the Model Explainability phase includes model-agnostic interpretability tools to provide human understanding. A user can know why such outcomes were obtained, when the model succeeded, and why it erred. The following sections describe in detail each phase.

3.1. Data acquisition

The first framework phase captures and stores all data needed to forecast day-ahead electricity price. The World Wide Web can be used as a source of data that could affect electricity prices. These data will be called features based on exogenous factors as they relate to external factors independent from the electricity price. Since the proposed methodology uses the Spanish wholesale market as the case study, the following features (i to xviii) are refer to this market. Nonetheless, they could easily be obtained in case other market is considered.

(i) Electricity price in the Spanish wholesale market (*Operador del Mercado Ibérico de Energía — Polo Español*, OMIE; omie.es). Data summary (average [min, max]): 50.02 [2.06, 101.99] EUR $(\mathrm{MWh})^{-1}$; period: 1 h.

(ii) Electricity price in the French wholesale market (European Power Exchange SE, EPEX; epexspot.com). Data summary: 43.55 [−31.82, 874.01] EUR $(\mathrm{MWh})^{-1}$; period: 1 h.

(iii–xiii) Electric power generated in the Iberian Peninsula (*Red Eléctrica Española*, REE; demanda.ree.es; period: 10 min). The generated power is broken down into the following power sources: wind (5551 [212, 17 499] MW), nuclear (6307 [3721, 7127] MW), coal (3951 [166, 8715] MW), combined cycle (3464 [278, 17 159] MW), hydraulic (3060 [−3522, 11 348] MW), international interchanges (1062 [−5033, 5850] MW), Balearic Islands interchange (−144 [−318, 284] MW), photovoltaic solar (883 [−65, 3821] MW), solar thermal (574 [0, 2219] MW), renewable thermal (414 [241, 629] MW) and cogeneration (3542 [1958, 4247] MW).

(xiv) Electric power demand forecast in the Iberian Peninsula (*Red Eléctrica Española*, REE; demanda.ree.es). Data summary: 28 771 [18 075, 41 215] MW; period: 10 min. Intrinsically, it includes future atmospheric conditions that could affect electricity prices — *e.g.*, ambient temperature.

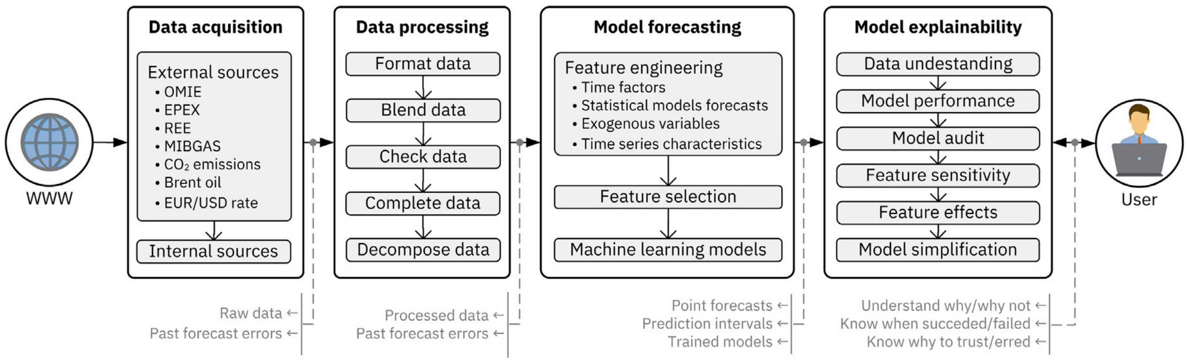


Fig. 1. Schematic representation of the proposed framework for collaborative intelligence in forecasting day-ahead electricity price.

(xv) Natural gas price in the Iberian market (*Mercado Ibérico del Gas*, MIBGAS; mibgas.es). Data summary: 19.15 [12.32, 41.69] EUR $(\mathrm{MWh})^{-1}$; period: 1 h.

(xvi) CO_2 European emission allowances (*Sistema Europeo de Negociación de* CO_2, SENDECO2; sendeco2.com). Data summary: 11.13 [3.96, 27.42] EUR $(\mathrm{t\ CO_2eq})^{-1}$; period: 1 day.

(xvii) Brent oil price (Markets Insider; url). Data summary: 58.47 [28.79, 86.29] USD barrel^{-1}; period: 1 day.

(xviii) EUR/USD currency rate (Markets Insider; url). Data summary: 1.138 [1.039, 1.251] EUR USD^{-1}; period: 1 day.

Note that interchanged markets have been considered here. Spain is electrically interconnected with France, Morocco and Portugal. Spain shares with Portugal the same electricity price most of the time. For example, in 2018 the shared ratio was 95 %. In contrast, only 25 % of French and Spanish prices were the same [78]. This means that, in principle, the energy interchanged between these two countries could affect Spanish prices. In addition, the energy transferred with France (5248 GWh) is seventeen times greater than that of Morocco (298 GWh, REE2019). From this point of view, the electrical energy exchanged with Morocco is negligible compared to the French interconnection. These are the reasons why only the French market has been taken into account for the interchanged markets.

These raw data are joined into a whole data set together with the forecast errors committed in the past by the framework. This data set starts on the 1st of January 2016, and is daily updated. It comprises the input of the Data Processing phase.

3.2. Data processing

The second framework phase duly prepares the raw data for modeling. The data is firstly formatted to the correct data type — *e.g.*, numerical, date or time. The data suffers from several time intervals since it comes from many sources. In particular, the different time intervals are the following: one data point each 10 min, 1 hour and 1 day. This multiple-interval data is blended into a single reference, the one-hour interval of the electricity price. For this, 10 min-period data is hourly averaged; and 1 day-period data is hourly interpolated carrying forward the last observation. Then, sanity checks are carried out to look for missing and non-valid values. If detected, they are imputed using Schumaker's algorithm, a univariate interpolation method based on shape-preserving splines [79].

Lastly, the electricity price time series is decomposed into several components, each representing an underlying pattern category. For this, the STL method [80] is used as the first algorithm to decompose the electricity price time series. The reason lies on the fact that the STL method has exhibited better results over classical [81] and more advanced decomposition methods [82]. The STL method isolates and extracts the following six components from the electricity price time series Y (EUR $(\mathrm{MWh})^{-1}$). (i) The trend component T (EUR $(\mathrm{MWh})^{-1}$) indicates a long-term change in data. (ii–v) The seasonal components S_s (EUR $(\mathrm{MWh})^{-1}$) describe specific patterns that reoccur after fixed s (h) time periods. In order to obtain frequency information about the price time series, the fast Fourier transform is applied. In the case of the Spanish electricity price time series, four fundamental periods are observed. Ordered from highest to lowest power spectral density, the s time periods are the following: 12 h, 24 h, 168 h (1 week) and 84 h ($^1/_2$ week). (vi) The last term is the remainder (or residual) component R (EUR $(\mathrm{MWh})^{-1}$). It represents the original time series when it has been detrended and deseasonalized. Thus, it does not exhibit any clear behavior or pattern. Eq. (1) gives the additive decomposition of the electricity price time series.

$$Y = T + S_{12} + S_{24} + S_{84} + S_{168} + R \tag{1}$$

The STL method requires tuning the seasonal window width of each seasonal component S_s. They control how rapidly each seasonal component S_s can change. The tuning procedure consists on a grid search over odd window widths greater or equal to 7 h [80]. The best seasonal window widths obtained with this procedure are the following: 21 h, 19 h, 17 h and 9 h for the seasonal components S_{12}, S_{24}, S_{84} and S_{168}, respectively.

3.3. Model forecasting

The third framework phase predicts day-ahead electricity price points. For this, a divide-and-conquer strategy is followed. Each electricity price time series component is predicted independently (Eq. (1)). Then, their forecasts are added up into a single time series. This gives the electricity price forecast. Finally, the prediction intervals are calculated. The following sections describe in detail the methods used to forecast each component.

The accuracy of forecasts will be measured by the mean absolute scaled error (MASE, Eq. (2) [83]). It is based on the mean absolute error (MAE), which measures the average of the absolute difference between observations (x) and forecasts ($\hat{x}$) over the forecast horizon (h). MAE is divided by the benchmark (*naïve*) mean absolute error ($\mathrm{MAE}_{naïve}$), which compares the observations (x) with *naïve* forecasts ($\hat{x}_{naïve}$). For non-seasonal time series (*i.e.*, trend T and remainder R) the *naïve* forecasts are equal to the last observed value (Eq. (3)). For seasonal time series (*i.e.*, seasonal components S_s) the *naïve* forecasts are equal to the observed value from the prior seasonal period s (Eq. (4)). The *naïve* forecasts of the general time series Y follow a similar-day strategy (Eq. (5)). It proceeds as follows: the forecast for hour i of a Monday is set equal to the price for the same hour a week ago; the forecast for hour i on the remaining days is set equal to the price for the same hour the day before. Note that three other similar-day techniques were considered: (i) $x_{i-7\,\mathrm{days}}$; (ii) $x_{i-1\,\mathrm{day}}$; and (iii) in addition to Monday, it considers Saturday and Sunday when lagging one week. Nonetheless, their predictive performance were worse so are not reported in this study.

$$\mathrm{MASE} = \frac{\mathrm{MAE}}{\mathrm{MAE}_{naïve}} = \frac{\sum_{i=1}^{h} |x_i - \hat{x}_i|}{\sum_{i=1}^{h} |x_i - \hat{x}_{i,naïve}|} \tag{2}$$

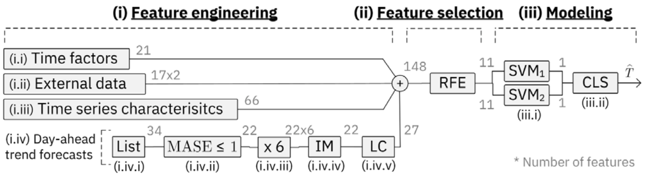


Fig. 2. Pipeline proposed to forecast the day-ahead trend of the electricity price time series.

$$\hat{x}_{i,naïve}\big|_{T,R} = x_0 \tag{3}$$

$$\hat{x}_{i,naïve}\big|_{S_s} = x_{i-s} \qquad s \in \{12, 24, 84, 168\} \tag{4}$$

$$\hat{x}_{i,naïve}\big|_{Y} = \begin{cases} x_{i-7\,\text{days}} & \text{Monday} \\ x_{i-1\,\text{day}} & \text{otherwise} \end{cases} \tag{5}$$

The MASE is interpreted as follows. Values greater than 1 indicate that benchmark (*naïve*) forecasts perform better than the ones under consideration. And it tends to 0 when forecasts approximate to observations. The MASE has favorable properties compared to other accuracy measures [81]. For these reasons, several authors have suggested MASE as a standard measure for time series forecasting [84].

3.3.1. *Trend forecasting*

The trend forecasting process consists of three sequential tasks: (i) feature engineering, (ii) feature selection and (iii) modeling. Fig. 2 depicts the sequential process as a pipeline (or workflow) of these three tasks. The process leads to the forecast of the day-ahead trend component of the electricity price ($\hat{T}$). The three tasks are subsequently described.

(i) The trend forecasting process starts with feature engineering. It aims to create features (or predictors) that could provide predictable information in the modeling process. These features can be classified into four groups: (i.i) time factors, (i.ii) exogenous data, (i.iii) time series characteristics and (i.iv) statistical forecasts. The Appendix contains the features added into the pipeline. They are described in more detailed in the following.

(i.i) A number of 21 time factors are firstly considered (Table 5). A binary categorization $\{0, 1\}$ is chosen to differentiate between weekday/weekend, AM/PM and working day/holiday. An integer number representation is utilized for consecutive time factors, such as for "hour of day" (ranging from 0 to 23) or "day of week" (from 0, Monday, to 6, Sunday). Integer time factors are transformed into Cartesian coordinates to capture the continuation between the last and first unit — *e.g.*, the hours 23 and 0, or Sunday and Monday. Eq. (6) transforms the integer number i into the Cartesian coordinates x and y, where l is the length of the time factor — *e.g.*, 24 for "hour of day", or 7 for "day of week":

$$\begin{cases} x = \sin\left(2\pi \cdot {}^{i}\!/_{l}\right) \\ y = \cos\left(2\pi \cdot {}^{i}\!/_{l}\right) \end{cases} \tag{6}$$

(i.ii) A number of 34 exogenous features are added into the pipeline (Table 6). These are based on the external data acquired in the Data Acquisition phase. Should the data exhibit seasonal patterns, an additive decomposition is carried out by the STL method. The underlying signal patterns can then be isolated into trend, seasonal components and remainder. The objective is to find better correlations between exogenous predictors and electricity price. This would eventually enhance forecast performance by providing better information into the modeling step. For example, the trend of the national electrical demand may provide more information to forecast the electricity price trend than the power demand itself, which contains seasonal patterns and non-specific behaviors. The exogenous data decomposition gives 11 trends. These are added together with 6 original (not-decomposed) signals. In addition, these variables are divided by the electricity price trend to create 17 ratios. The day previous to the forecasting day is the time window selected to compute these 34 exogenous features.

(i.iii) A number of 66 features are considered by extracting structural characteristics from the electricity price trend (Table 7). We use methods in the domain of signal processing and time series analysis. The methods range from basic statistical equations, such as mean or maximum value, to more sophisticated measures, such as entropy or non-linearity. The day previous to the forecasting day is the time window selected for computing these features. The objective is then helping the forecasting process by using the time-series characteristics of the day previous to the forecasting day.

(i.iv) Finally, 27 day-ahead forecasts of the electricity price trend are added into the pipeline. These forecasts are obtained by following five consecutive steps, from (i.iv.i) to (i.iv.v).

(i.iv.i) A list of 34 statistical models are initially selected (Table 8). A wide range of model types are included so that different information can be learned from the trend. The selected models cover smoothing methods, Theta models, auto-regressive-type methods and artificial neural networks.

(i.iv.ii) Models achieving a monthly average MASE higher than 1 – *i.e.*, the benchmark performance – are dropped. The initial list is reduced to 22 models (nº 1, 3–8, 15–22, 26–30, 33 and 34, Table 8). Their hyper-parameters are tuned by a grid search over a time-series cross-validation procedure: a day-length rolling window is forecast over a month period. The training data extents to six months previous to the rolling-window forecasting day. The best hyper-parameters minimize the average MASE obtained on the month period used for validation.

(i.iv.iii) Each one of the 22 tuned models is trained with six different time windows of the electricity price trend: the first training window covers the previous 31 days of the forecasting day (one month), the second training window uses the 62 previous days (two months), and so forth until six months are used for training. This procedure gives six forecasts for each one of the 22 statistical models tuned in step (i.iv.ii).

(i.iv.iv) The previous six forecasts are linearly combined. The combined forecast is the interquartile mean (IM) — *i.e.*, the 25% truncated mean. This combined forecast is preferred over the one resulted from step (i.iv.ii). The reason is that combining forecasts obtained from different training time windows reduces the risk associated with selecting an individual time window. Fig. 3 shows an example of such statement. The MASE quartile spread is plotted against the number of previous months used for training, where IM stands for interquartile mean. The learning statistical model chosen for the example is a double-seasonal (12 h, 24 h) Holt–Winters method (nº 2, Table 8). The figure shows the results in July 2016 (left) and August 2016 (right). In July, the training time window that achieved the lowest average MASE — $\min(\mu)$ — was 3 months, and the highest — $\max(\mu)$ — was 4 months. In August the results were different: the best training time window was 4 months, and the worst was 2 months. Thus, the best training time window changes over time. Nonetheless, the interquartile mean achieved a low average MASE on both months. Although it did not always obtain the lowest MASE, on the long run it was the most robust approach. In addition, combining forecasts avoids finding the best training time window for each model type selected in step (i.iv.ii).

Note that, besides the interquartile mean, other linear combination methods were tested. Nonetheless, their predictive performance were worse or statistically equal on average. As example, Fig. 4 shows the MASE quartile spread resulted from August to December 2016. The tested methods were the following: simple average (SA), median (MED), the Bates and Granger method (BG, [85]), the Newbold and Granger method (NG, [86]), the Aiolfi and Timmermann method (AT, [87]), ordinary least squares (OLS, [88]), constrained least squares

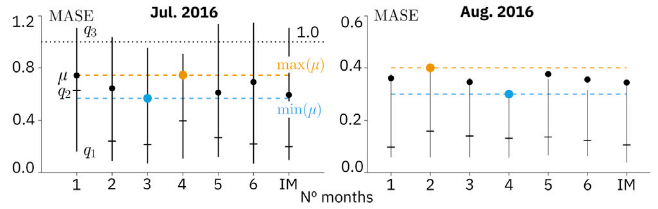


Fig. 3. MASE quartile spread obtained with six different time windows (number of months of training data) and the interquartile mean (IM).

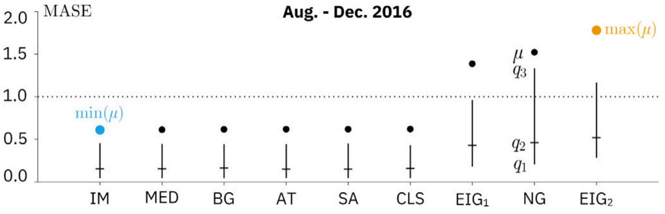


Fig. 4. MASE quartile spread obtained by several linear combination methods of forecasts. Learning statistical model: double-seasonal (12 h, 24 h) Holt–Winters method.

(CLS, [88]), least absolute deviation (LAD, [88]), the standard eigenvector method (EIG$_1$, [89]) and the bias-corrected eigenvector method (EIG$_2$, [89]). Based on the results, no statistical mean differences were found among the methods IM, MED, BG, AT, SA and CLS. Higher MASE values were found for the Newbold and Granger method (NG) and the eigenvector-based methods (EIG$_1$, EIG$_2$). The reason was the unstable behavior of the estimated weights (or slopes β) used in the linear combination – *i.e.*, weighted summation – of the input forecasts: minor fluctuations in the input forecasts induced major shifts of their corresponding weights. This caused poor out-of-sample performance. For this reason, extreme MASE results were obtained for the regression-based methods OLS and LAD. These two are not represented in the figure.

(i.iv.v) Finally, the 22 forecasts obtained in step (i.iv.iv) are linearly combined (LC). In step (i.iv.iv), forecasts from the same model type were combined. Here, forecasts from different model types are combined. Due to their stability behavior, five linear combination methods are used: simple average, interquartile mean, median, the Bates and Granger method and the Aiolfi and Timmermann method. This gives a total of 27 (22 + 5) day-ahead forecasts of the electricity price trend that are added into the pipeline.

(ii) The next step in the trend forecasting pipeline is feature selection. The aforementioned feature engineering task has provided a vast number of predictors. The hope was that some of the predictors capture a predictive relationship with the outcome. But some may not be relevant if they do not contain predictive information. For a number of machine learning models, notably support vector machines, predictive performance is degraded as the number of uninformative predictors increases. Although other models are more insensitive to irrelevant predictors, including the minimum number of features can help to reduce complexity. Considering that the number of features is 148, obtaining the best subset will imply evaluating $2^{148} - 1$ combinations of features. For evident computational reasons, a stepwise selection method is applied.

The feature subset selection is carried out by means of recursive feature elimination (RFE, [90]). RFE is a sequential backward selection method that recursively considers smaller sets of features following three steps. (i) It begins by building a model on the entire set of predictors, and then computes a score of the relevance that each predictor has on the outcome. (ii) The least relevant predictor is then removed from the current set of features. (iii) The model is re-built, and relevance scores are computed again. Steps (ii) and (iii) are recursively repeated until the number of features is depleted. The best feature subset is the one that gives the lowest MASE.

Table 1
Best features subset selected by the RFE-SVM for day-ahead forecasting the electricity price trend.

Nº	Description
1–2	Average and median values of the previous day
3–4	First and third values of the previous day
5–7	Last, second last and third last values of the previous day
8	Double-seasonal (12 h, 24 h) Holt–Winters method
9	Advanced linear model. Lags from 24 h to 96 h
10	Advanced linear model. Lags selected from 24 h to 169 h
11	MARIMA ($p = 30, d = 2, q = 30$)

Prior to applying RFE, zero variance features are removed. Perfect multicollinearity – *i.e.*, exact linear relationship – is also checked among features. It is identified by the rank of the matrix formed by the features. If detected, the perfect correlated feature is dropped. This procedure is repeated until no multicollinearity is detected.

The model used in the RFE method is a support vector machine (SVM) with a linear kernel. Feature relevance is computed by squaring the fitted SVM weights [90]. The feature with the lowest squared weight is dropped. Each time the model is re-built, the best model hyper-parameters are chosen based on a grid search computed on a time series cross-validation procedure. The hyper-parameters are C — *i.e.*, the cost of constraints violation —, and the ϵ parameter of the insensitive-loss function. This ensures that the model is properly tuned for each subset of features.

The best subset includes 11 features (Table 1). No features based on exogenous data (Table 6) or time series characteristics (Table 7) have been selected.

(iii) The trend forecasting process finishes with the modeling step. The previous steps have tried to capture the patterns that the trend exhibits with itself. Here the aim is to improve the forecasting accuracy by capturing the relation between the trend and exogenous features. (iii.i) To accomplish this aim, first two implementations of a linear-kernel support vector machine are used (SVM$_1$ [91] and SVM$_2$ [92]). Different implementations of a SVM algorithm can lead to different results, so combining their results could reduce uncertainty. The hyper-parameter C is tuned to 1000 and 200, and ϵ to $5 \cdot 10^{-5}$ and $7 \cdot 10^{-5}$, for SVM$_1$ and SVM$_2$, respectively. (iii.ii) Finally, the two outcomes from the SVM implementations are linearly combined by constrained least squares (CLS). The weights of this linear fit (w_i, $i \in \{\text{SVM}_1, \text{SVM}_2\}$) are subject to $0 \leq w_i \leq 1$ and $\sum w_i = 1$. CLS has been preferred over ordinary least squares due to the unstable weights of the latter. The tuned weights are $w_{\text{SVM}_1} = 0.68$ and $w_{\text{SVM}_1} = 0.32$.

3.3.2. Seasonal components forecasting

The forecasting process of the seasonal components S_{12}, S_{24}, S_{84} and S_{168} follows a procedure similar to the one used for the trend. It consists of two tasks carried out independently for each seasonal component: (i) feature engineering and (ii) feature selection. Fig. 5 depicts the sequential process as a pipeline of these two tasks. The process leads to the forecast of the day-ahead seasonal components of the electricity price ($\hat{S}_{12}$, $\hat{S}_{24}$, $\hat{S}_{84}$ and $\hat{S}_{168}$).

(i) The forecasting process starts with feature engineering. (i.i) Time factors (Table 5) add 21 features to the forecasting process. (i.ii) The exogenous data decomposed by the STL method gives several seasonal components (Table 6). For each seasonal component, the features are selected with regard to their corresponding seasonal period. For example, the external factors' time series decomposed with a period 12 h are selected to forecast the seasonal component S_{12}. This varies the total number of features used to forecast each seasonal component. The periods 12 and 24 decompose exogenous factors into 15 new features, and the periods 84 and 168 into 11 and 12 new features, respectively. These features are divided by the seasonal components of the electricity

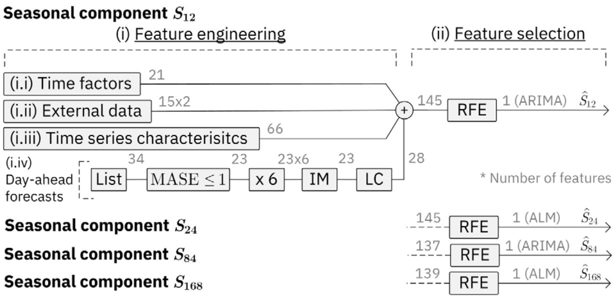


Fig. 5. Pipeline proposed to forecast the day-ahead seasonal components of the electricity price time series.

price to give 15, 15, 11 and 12 ratios. These are included as new features. (i.iii) In addition, extracting time series characteristics from the seasonal component gives 66 more features (Table 7). (i.iv) Finally, a total of 28 day-ahead forecasts of the seasonal components ($\hat{S}_S$) are added as new features to the pipeline. The procedure to obtain the 28 forecasts is as follows. The total of 34 statistical models listed on Table 8 are pruned when tested against an out-of-sample MASE equal to or less than 1. Then, the selected model types are trained over six different-length training time windows. The interquartile mean (IM) is calculated from their six predictions. All model forecasts are linearly combined (LC) with the methods IM, MED, BG, AT and SA. Added to time factors, exogenous features and time series characteristics, the total number of features amounts to 145 for the seasonal components S_{12} and S_{24}, 137 for S_{84} and 139 for S_{168}.

(ii) The best feature subset is chosen by means of the recursive feature elimination method. The learning algorithm is a gradient boosting decision tree model (XGBoost, [62]). Feature relevance is computed by means of a performance-based method [93]. It measures the increase in the prediction error of the model after the feature's values are permuted. This breaks the relationship between the feature and the true outcome. The feature that is dropped in each iteration is the one that, when shuffling its values, increases the model error the least. The reason is that the model ignored that feature the most for the prediction.

Each time the model is re-built the best hyper-parameters are chosen based on a Bayesian optimization of the time-series cross-validated MASE. These hyper-parameters are the learning rate (range of search: [0.001, 0.1]), the maximum depth of a tree [2, 16], the subsample ratio of the training instances [0.5, 0.9] and the subsample ratio of columns when constructing each tree [0.5, 0.9]. The number of learning trees is stopped when the average out-of-sample MASE begins to rise.

The aforementioned procedure gives a single feature as the best subset for each seasonal component S_s ($s \in \{12, 24, 84, 168\}$). The feature is the interquartile mean of the six forecasts predicted by a specific statistical model type trained on six different-length time windows. These models are listed in Table 2: an autoregressive integrated moving average model (ARIMA) for seasonal components S_{12} and S_{84} (nº 3), and an advanced linear model (ALM) for seasonal components S_{24} and S_{168} (nº 16). Note that no linear or non-linear combination has been found to perform better than the models listed in Table 2, including machine learning models. Thereof, the modeling step that was performed in the trend forecasting process has been excluded from Fig. 5.

Table 2
List of tuned statistical models chosen for forecasting the seasonal components S_s of the electricity price time series ($s \in \{12, 24, 84, 168\}$).

S_s	Statistical model
S_{12}	ARIMA ($p = 0, d = 0, q = 25$) ($P = 25, D = 2, Q = 24$)(12)
S_{24}	ALM with lags from 24 to 99
S_{84}	ARIMA ($p = 5, d = 0, q = 5$) ($P = 0, D = 2, Q = 0$)(84)
S_{168}	ALM with a stepwise selection of lags from 24 to 341

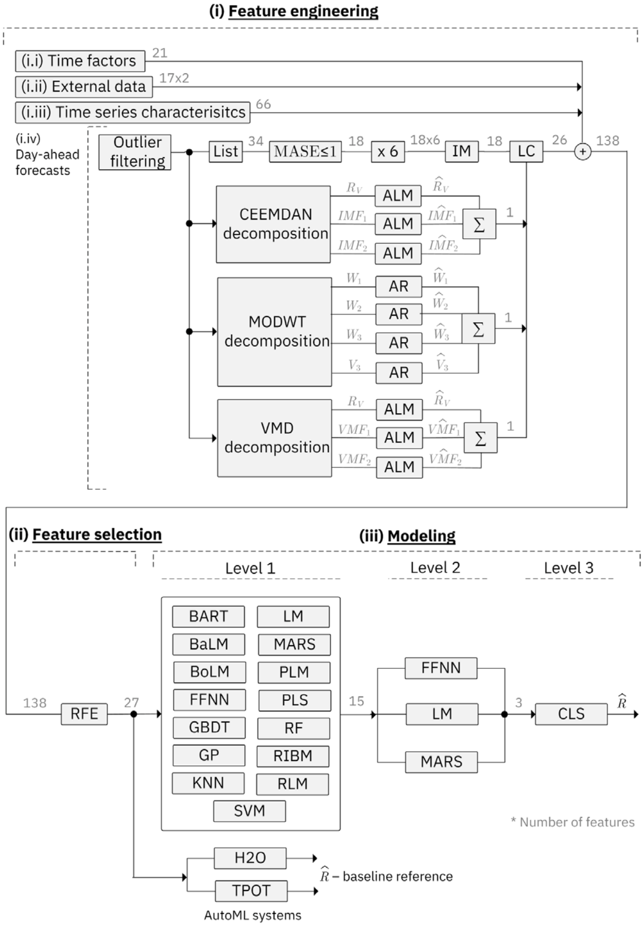


Fig. 6. Pipeline proposed to forecast the day-ahead remainder component of the electricity price time series.

3.3.3. Remainder forecasting

The forecasting process of the remainder component R follows a procedure similar to the one used in forecasting the trend and seasonal components. Nevertheless, due to the complexity of the term, several augmentations have been implemented. The forecasting process consists of three sequential tasks: (i) feature engineering, (ii) feature selection and (iii) modeling. Fig. 6 depicts the sequential process as a pipeline of these three tasks. The process leads to the forecast of the day-ahead remainder component of the electricity price ($\hat{R}$).

(i) The forecasting process starts with feature engineering. (i.i) Time factors (Table 5) add 21 features to the pipeline. (i.ii) The STL decomposition of the exogenous data gives 17 remainder components that are used as features (Table 6). These features are divided by the remainder of the electricity price to give another 17 new features. (i.iii) In addition, extracting time series characteristics from the remainder component gives 66 more features (Table 7).

(i.iv) A total of 26 day-ahead forecasts of the remainder component ($\hat{R}$) are added as new features to the pipeline. For that, outliers are firstly detected and replaced by suitable values. Electricity prices suffer from short-lived, generally unanticipated abrupt changes known as spikes or jumps [1]. If not filtered, they could hinder the forecasting performance [94–96]. Here, spikes are considered to be additive outliers. They appear as a surprisingly large or small value occurring for a single observation. Subsequent observations are unaffected by these outliers. Their localization and correction is carried out following the

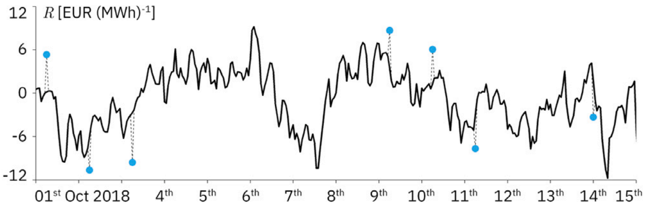


Fig. 7. Visual example of the outlier filtering carried out in the remainder component of the electricity price time series.

method proposed by Chen et al. [97]. As an example, Fig. 7 shows the outlier filtering carried out for the first days of October 2018. Dots are identified as additive outliers in the original time series, represented by a dashed line. Outliers are replaced by suitable values, leading to a smoother time series represented by a continuous line.

The total of 34 statistical models listed on Table 8 are pruned to 18 when tested against an out-of-sample MASE equal to or less than 1. Then, the selected model types are trained over six different-length training time windows. The interquartile mean (IM) is calculated from their six predictions. This leads to 18 day-ahead forecasts of the remainder component ($\hat{R}$).

In parallel, the outlier filtered remainder component R is further decomposed into a collection of elemental time series with more meaningful instantaneous frequencies. A statistical model is then used to predict each elemental signal individually. Finally, the corresponding prediction results of each elemental signal are aggregated as the final forecasting results. Predicting more meaningful time series has the aim of providing more useful information to the modeling step (iii).

Three methods are applied in parallel to decompose the filtered remainder component R: complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN, [98]), maximal overlap discrete wavelet transformation (MODWT, [99]) and variational mode decomposition (VMD, [100]). The number of decomposed elemental time series and their underlying forecasting model are chosen by minimizing the time-series cross-validated MASE. The CEEMDAN method decomposes the filtered R into two intrinsic mode functions ($IMF_i, i \in \{1,2\}$) and a residue (R_C) that captures the lowest frequency. These signals are forecast by a linear model with a stepwise selection of 341 lags used as predictors (ALM, [101]). The MODWT method decomposes the filtered R into the first, second and third level wavelet coefficients ($W_i, i \in \{1,2,3\}$) and the third level scaling coefficients (V_3). The elemental signals are forecast by an auto-regressive model AR(p = 53) [102]. The Haar wavelet transform filter is adopted here. The VMD method decomposes the filtered R into three variational mode functions ($VMF_i, i \in \{1,2,3\}$) and a residue (R_V). These signals are forecast by a linear model with a stepwise selection of 341 lags used as predictors (ALM, [101]).

Fig. 8 depicts an example of the results obtained from the three aforementioned decomposition methods. The decomposed signal is the filtered remainder component R shown on Fig. 7. The top plot shows the three signals resulted from the CEEMDAN decomposition; the two middle plots show the MODWT elemental signals; and the bottom plot shows the time series obtained from the VMD method. The low number of elemental time series decomposed by each method implies that the methods could not extract much meaningful information. The intrinsic mode functions (IMF_1 and IMF_2), the wavelet coefficients (W_1, W_2 and W_3), and the residue R_V manifest a high frequency, random noise. This implies that a first STL decomposition followed by this second decomposition has achieved a complete separation of all the underlying patterns of the original electricity price time series. All these patterns have been forecast and their aggregation have been added to the workflow.

The last step of the feature engineering process is to linearly combine (LC) all day-ahead forecasts of the filtered remainder term. The number of combined forecasts are 18 coming from the direct prediction, and 3 resulted from the decomposition techniques. These forecasts are linearly combined using five methods: AT, BG, IM, MED and SA. This involves adding a total of 26 features to the pipeline. Added to 21 time factors, 30 exogenous data features and 64 time series characteristics, the total number of features amounts to 138.

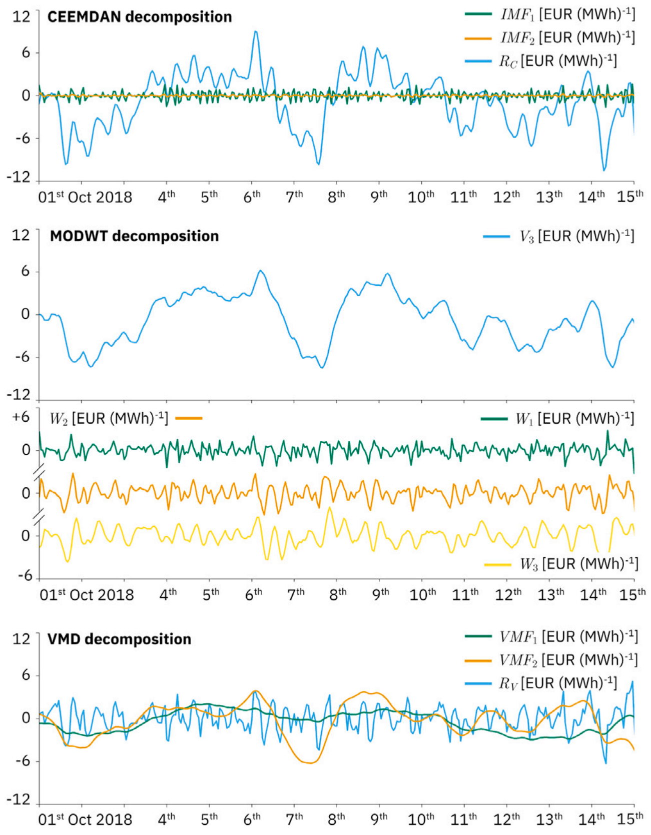


Fig. 8. Visual example of the signals decomposed from the outlier filtered remainder component of the electricity price time series.

(ii) The second step of the forecasting process is feature selection. The best feature subset is chosen by carrying out the same procedure followed in the seasonal components section: a recursive feature elimination methodology (RFE) using a gradient boosting decision tree model (XGBoost, [62]) tuned by Bayesian optimization. This procedure gives the best features listed in Table 3. It is interesting to note that three exogenous factors have been selected: the remainder term of the electrical power estimated for the Iberian Peninsula, the natural gas price of the previous day, and the EUR/USD currency rate of the previous day. In addition, several time series characteristics obtained from the previous day can help to predict the evolution of R for the next day. No forecasts from the three decomposition methods are included on the best feature subset *per se*. Nevertheless, they are included on the linear combination of forecasts by means of the simple average, median and interquartile mean.

(iii) The modeling step uses a stacked ensemble model architecture. The forecasts of R obtained in the feature engineering step have tried to capture the patterns that the remainder component exhibits with itself. Here the aim is to improve robustness and generalizability over a single predictor by combining the forecasts in several lineal and non-lineal fashions. At the same time, the forecasts are also combined with exogenous data to capture the relation that the electricity price has with them.

The stacked ensemble model architecture consists on three levels or layers. Ii contains 19 learners in total: 15 on the first level, 3 on the

Table 3
Best features subset selected by a recursive feature elimination method to forecast the remainder term (R) of the electricity price time series.

Nº	Description
Time factors – Table 5	
1	Hour of day
Exogenous factors – Table 6	
2	Remainder term of the electric power forecast
3	Natural gas price (−1 day)
4	EUR/USD currency rate (−1 day)
Time series characteristics (−1 day) – Table 7	
5–7	2º, max and median values
8	Spectral Shannon entropy
9	PACF features: diff2-pacf5
10	Holt's linear trend method: β
11	STL linearity
12	Non linearity
13	Symbolic transformations: MotifTwo
14	Correlation: trev
15–17	ACF features: e-acf10, (diff1, diff2)–acf10
Forecasts of models – Table 8	
18	ARIMA $(p = 1, d = 0, q = 1)$ $(P = 4, D = 0, Q = 1)(12)$
19	*Naïve* model
20	TBATS model
21	Linear model with stepwise selection of lags
22	Linear model with combined lags
23	State-space ARIMA
24	Regularized linear model with lagged regressors
25–27	Linear combination of forecasts: SA, MED, IM

second and 1 on the third and last level. The prediction of Level 3 is the final day-ahead forecast of the remainder term ($\hat{R}$) of the electricity price time series. Added to the forecasts of the trend ($\hat{T}$) and seasonal terms ($\hat{S}_{12}$, $\hat{S}_{24}$, $\hat{S}_{84}$ and $\hat{S}_{168}$) it will make the final day-ahead forecast of the electricity price ($\hat{Y}$, Eq. (1)).

Level 1 comprises fifteen learners: Bayesian additive regression trees (BART, [103]), Bayesian linear model (BaLM, [104]), gradient boosting linear model (BoLM, [105]), feed-forward neural network (FFNN, [60]), gradient boosting decision trees (GBDT, [62]), Gaussian process model (GP, [91]), k–nearest neighbors algorithm (KNN, [106]), linear model with an exhaustive search of the best predictors (LM, [107]), a bagged of multivariate adaptive regression splines (MARS, [108]), penalized linear model (PLM, [109]), partial least square algorithm (PLS, [110]), random decision forests and extremely randomized trees (RF, [111]), rule- and instance-based model (RIBM, [112]), regularized linear model (RLM, [62]) and support vector machines (SVM, [92]). These learners have been chosen based on a trade-off between individual accuracy and diversity. The reason of diversity is boosting the gain by combination. Here the overall ensemble diversity is measured in two steps. First, the Pearson correlation coefficient (r_{Person}) is calculated between each pair of out-of-bag errors. Then, all the pairwise metrics are averaged into a single metric ($\bar{r}_{\text{Person}}$). The closer the value of $\bar{r}_{\text{Person}}$ is to zero, the larger is the diversity.

Level 2 includes three learners: a feed-forward neural network (FFNN, [60]), a linear model with an exhaustive search for the best predictors (LM, [107]) and a bagged ensemble of multivariate adaptive regression splines (MARS, [108]).

Level 3 consists of a constrained least square fit (CLS) on the predictions of Level 2. Applying a simple weighted average is enough on Level 3. The reason lies on the high correlation of the out-of-bag cross-validation errors of the learners of Level 2. This is the reason why no more than three levels are implemented on the stacked model ensemble architecture. In other words, it seems that not much more information can be squeezed from the input feature set (Table 3).

The learners of Level 2 have been chosen among the fifteen machine learning models mentioned on Level 1. The selected learners of Level 2 are not the ones that give the best out-of-fold MASE. Combining two good-performance learners (LM and MARS) with a poor-performance learner (FFNN) obtains the best accuracy when combined with a CLS fit. The learners have been chosen by evaluating the MASE of the entire permutation space given by selecting 1, 2 and 3 learners from the fifteen machine learning models mentioned on Level 1. Note that the weights of the CLS fit are not chosen based on this procedure.

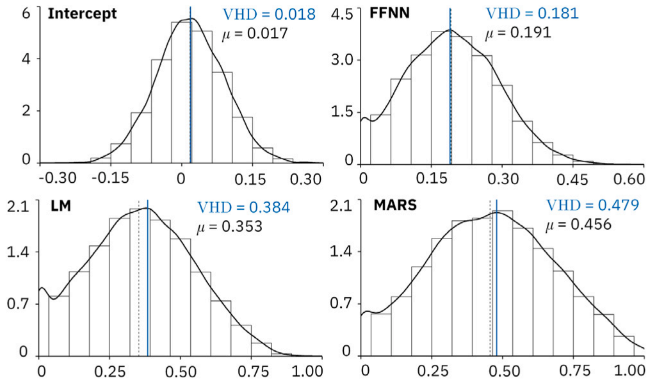


Fig. 9. Distribution of the intercept and weights of the constrained least square fit in Level 3.

The stacked model ensemble training procedure follows six consecutive steps. First, Level 1 learners are tuned individually by Bayesian optimization or Cartesian grid search. The tuned hyper-parameters are listed on Table 4. Secondly, the learners are trained on ten time-series folds – *i.e.*, data splits –. The number of ten folds have been chosen due to computational cost reasons and empirical evidence of superior performance [46]. The folds are selected so that each fold can only be predicted by training on previous folds. In addition, a data chunk belonging to a specific day is not split among different folds. These ten folds are the same across all estimators. Thirdly, Level 2 learners are tuned individually on the out-of-fold predictions of Level 1 learners (Table 4). Fourthly, these tuned learners are trained on the same fold indexes as Level 1. The result is a data set composed of out-of-fold predictions of Level 2 learners.

Fifthly, the weights of Level 3 are calculated by bootstrap aggregation (also known as bagging [113]). The out-of-fold predictions of Level 2 are uniformly sampled with replacement. Data is bootstrapped preserving daily values. CLS is then computed on each one of the new bootstrap data sets. The simple average is then calculated on the fitted weights. The aim of bagging is reducing variance, and by extension, preventing over-fitting. The resulting weights are 0.02 for the intercept, 0.19 for the FFNN, 0.35 for the LM and 0.46 for the MARS. Fig. 9 shows the histograms of the distribution of the bootstrap intercept and weights. The mean value is represented by a black dashed vertical line. The estimated value associated with the highest density (HDV) – *i.e.* the one that would appear more often for unseen data – is represented by a blue continuous vertical line. Both mean and HDV appear very next to each other. This implies that the mean value is a good choice for unseen data (if it follows the same distribution as the computed data).

The sixth and last step of the stacked ensemble training procedure consists on fitting all learners to the whole levels' data set. These models are conveniently saved and stored on disk. They will be used to forecast R during production (Section 3.5).

Finally, the remainder component forecasting pipeline finishes by applying automated machine learning platforms (AutoML). They are used as a baseline reference of the forecasting accuracy of the stacked ensemble architecture. The applied AutoML platforms are H2O AutoML [60] and TPOT [61]. Both platforms have been trained on the same ten folds as the stacked ensemble architecture. This improves the comparison of results. They have been run on a time budget of one hour per fold, resulting in twenty hours of total computational time. In this work, the AutoML results help to assure the performance of the proposed stacked ensemble architecture.

Table 4
List of tuned hyper-parameters' values of the stacked ensemble models for Level 1 (left and center) and Level 2 (right).

Parameter	Value	Parameter	Value	Parameter	Value
BART — Bayesian additive regression trees		*KNN — k-nearest neighbors algorithm*		*FFNN – Feed-forward neural network*	
alpha	0.967	Distance	0.5	Activation	RectifierWithDropout
beta	3.387	k	111	epochs	80
k	4.4	Kernel	Gaussian	Epsilon	10^{-10}
nu	1.198	*LM — Linear model*		input_dropout_ratio	0
num_trees	23	n_features	16	Hidden	200, 200
BaLM — Bayesian linear model		*MARS — Mult. adaptive reg. splines*		hidden_dropout_ratios	0.1, 0.1
–	–	degree	1	L_2	10^{-5}
BoLM — Boosting linear model		nprune	11	rho	0.99
alpha	0.058	*PLM — Penalized linear models*		*LM — Linear model*	
nrounds	69	alpha	0.998	n_features	12
Lambda	0.007	Lambda	0.016	*MARS — Mult. adaptive regression splines*	
FFNN — Feed-forward neural network		*PLS — Partial least squares*		degree	1
activation	RectifierWithDropout	Method	kernelpls	nprune	13
epochs	1000	ncomp	18		
epsilon	10^{-6}	*RF — Random forests*			
input_dropout_ratio	0	min.node.size	3		
hidden	50	mtry	28		
hidden_dropout_ratios	0.5	num.trees	550		
L_2	0	Splitrule	Extratrees		
rho	0.99	*RIBM — Rule- and instance-based model*			
GBDT — Gradient boosting decision trees		Committees	70		
colsample_bytree	0.73	Neighbors	0		
eta	0.005	*SVM — Support vector machine*			
max_depth	7	Cost	0.027		
nrounds	1453	Epsilon	0.622		
subsample	0.46	Kernel	Linear		
GP — Gaussian process					
kernel	Vanilladot				

3.4. Model explainability

The Model Explainability phase is the fourth (and last) framework stage. It is placed between the Model Forecasting phase and the framework user. As such, it allows the user to interpret the behavior of the developed model and its outcome. The ultimate goal is to facilitate effective and efficient human–machine collaboration in order to enhance the user's cognitive performance and, ultimately, improve decision-making. This is done by providing a data story based on a collection of *post-hoc* model-agnostic methods and visual artifacts. They describe model behavior by providing specific insights into the mechanisms of the model and detailed information about why such answers are generated.

The Model Explainability phase consists on five consecutive modules (i to v). They have been ordered in such a way that the framework user can be easily accompanied through model understanding. (i) The Data Understanding module analyzes the input data so that their main characteristics can be summarized. The goal is to maximize the user's insight into the data set by uncovering underlying patterns and structure. (ii) The Model Performance module implements techniques that facilitate assessing model quality and goodness of fit. (iii) The Model Audit module assesses on residuals' diagnostics. (iv) The Feature Sensitivity module aims at studying how the uncertainty in the model outcome can be apportioned to different sources of uncertainty in the input data. (iv) The Features Effects module indicate the direction and magnitude of change in the electricity price due to changes in the input feature values. (v) Finally, the Model Simplification module draws a summary of the model by explaining a intrinsically interpretable model that approximate the predictions. These five modules are supported by a variety of quantitative and graphical techniques that enhance model explainability.

The color palettes used on the visual representations have been carefully selected. The qualitative color scale, designed for coding categorical information, is the one proposed by Okabe and Ito [114]. The sequential color palette, designed for coding ordered information, is Viridis [115]. The diverging color scale, designed for coding ordered information around a central neutral value, is Scico [116]. The three color palettes are designed to be perceived by color vision deficiency readers. Both Viridis and Scico's colors span as wide as possible so that differences are easy to see. They are also perceptually uniform, meaning that values close to each other have similar appearance. These two properties hold true for the colors regular form and when converted to black and white.

The font typeface selected for the visual representations is IBM Plex® Sans [117]. It is a highly legible typeface due to its open-angled terminals, uniformed stems, clear crossbars with consistent thickness and open counter forms. It was designed to work with user interface environments.

The uncertainty throughout the Model explainability phase is summarized by calculating the span of values that are most probable and cover 95% of the distribution. More probable values have higher probability density. This way, the span is called the 95% highest density interval (HDI, [118]). Here, it is computed by the non-parametric bootstrap method. The computation randomly draws 20,000 independent samples from the original data set.

3.4.1. Data understanding

The aim of the Data Understanding module is to open-mindedly explore and analyze the input data for summarizing their main characteristics. The main goal is to maximize the user's insight into the data set by uncovering underlying patterns and structure. This is carried out by quantitative techniques and key visual representations organized into a coherent structure.

The visual representations are divided into five categories (i to v). (i) The first category allows the user to visualize the individual components obtained by STL decomposition (trend, seasonal and remainder components). The time series that can be represented include the electricity price and the exogenous factors of Table 6. (ii) The percentile values of the STL decomposition components are plotted against time. The objective is to discover time patterns. (iii) The individual STL components are plotted against each other to analyze their possible correlation. Data points are colored by the number of neighboring points so that the overall distribution can be analyzed. This density scatter plot is supported by the Pearson correlation coefficient. (iv) The STL individual components are ranked by their correlation with the electricity price. This allows hypothesizing about their possible degree of influence on the electricity price. (v) The fifth (and last) category consists on displaying the wavelet power spectrum of the individual components of the electricity price. The wavelet power spectrum is computed by applying the Morlet wavelet [119]. The vertical axis shows the fundamental periods observed when the fast Fourier transform is applied. The horizontal axis shows dates. The average (over time) wavelet power in the frequency domain is also displayed. Both plots aim at analyzing the time–frequency distribution – *i.e.*, the periodic phenomena in the presence of potential frequency changes across time.

3.4.2. *Model performance*

The aim of the Model Performance module is to assess on model quality and goodness of fit. The examination is carried out by diagnostic scores and visual verification. The visual representations are divided into four categories (i to iv). (i) The first category allows the user to graphically compare the temporal evolution of the electricity price time series, its STL decomposition components and their forecasts. (ii) The second category quantifies the forecasting error of the electricity price time series and its STL decomposition components. The error is measured by the mean absolute scaled error (MASE) and the mean absolute error (MAE, Eq. (2)). (iii) The third category shows the MASE performance of the stack ensemble models, individually and grouped by level. (iv) The fourth category represents models' diversity using a correlogram of their residuals. The residuals are equal to the difference between the observations (R) and the corresponding predicted values ($\hat{R}$). The correlogram is a graphical display of the correlation matrix. It holds the Pearson correlation coefficients for all possible combinations of models' residuals. The correlation coefficients are ordered according to the degree of association. It is obtained by a complete-linkage hierarchical cluster method.

3.4.3. *Model audit*

The aim of the Model Audit module is to assess the validity of the model by residuals' diagnostics. Residuals are diagnosed with five complementary analysis (i to v) that use visual inspection and a number of formal statistical hypothesis tests.

(i) Linearity assesses on whether the relationship between the electricity price observations (Y) and the corresponding forecasts ($\hat{Y}$) is linear. Linearity is graphically inspected with a scatter plot (Y *vs.* $\hat{Y}$). And it is numerically quantified by the coefficient of determination (r^2_{Person}) and the fitted parameters (β_0, β_1) of a simple linear regression fit ($Y = \beta_0 + \beta_1 \cdot \hat{Y}$).

(ii) Normality assesses if the residuals are normally distributed. Normality is graphically inspected with a histogram, a quantile–quantile plot and a box plot. For comparison reasons, the ideal normal distribution is superposed on the plots. Several normality tests support the visual diagnosis. The Student's t-test determines if the mean of the residuals is significantly different from zero. The D'Agostino test and the Anscombe–Glynn test aim to establish whether the residuals' distribution skewness and kurtosis are zero and three, respectively. The Shapiro–Wilk test and the Anderson–Darling test assess on general normality. These two tests have been selected among others because they provide the best power [120].

(iii) Homoscedasticity assesses on the homogeneity of variance of the residuals. It is visually inspected by plotting the residuals *versus* the predicted values. The studentized Breusch–Pagan test, the Goldfeld–Quandt test and the Harrison–McCabe test numerically support the homoscedasticity analysis [121].

(iv) Independence assesses on whether residuals are not linearly related with residuals at prior time steps. The strength of the relationship is measured by the Pearson correlation coefficient. It is graphically inspected by the auto-correlation function (ACF). In order to remove the effect of indirect correlations, the partial auto-correlation function is also displayed (PACF). Both plots show confidence intervals at a 5% significance level. The Box–Pierce test examines the null hypothesis of independence.

(v) Outliers in residuals are detected and displayed over the time evolution. They are considered to be additive outliers. They appear as a surprisingly large or small value occurring for a single observation. Their detection is carried out following the method proposed by Chen et al. [97].

3.4.4. *Feature sensitivity*

The aim of the Feature Sensitivity module is to study how the uncertainty in the electricity price forecast $\hat{Y}$ can be apportioned to different sources of uncertainty in the input variables. By definition, the remainder component $\hat{R}$ does not exhibit any clear behavior or pattern. This means that, generally, the uncertainty in $\hat{Y}$ will be mostly produced by the uncertainty in the forecast remainder component $\hat{R}$. The objective is then to identify how the forecast remainder component $\hat{R}$ depends on the uncertainty in the model input features. In case the uncertainty in $\hat{Y}$ is produced by the trend or seasonal components, the same methods hereinafter described could be applied following the same fashion. The Feature sensitivity analysis is carried out on Level 1 (Table 3), Level 2 and Level 3 features of the stack ensemble architecture presented on Fig. 6.

The analysis is divided into two groups: (i) local sensitivity analysis, (ii) and global sensitivity analysis. Local sensitivity analysis evaluates how the model output is influenced by the model inputs when predicting for one specific observation. In contrast, global sensitivity analysis assesses the influence over the entire variation range of the model inputs.

(i) Local sensitivity analysis is conveyed by Shapley values [122]. A Shapley value is the contribution of a feature value to the difference between the actual prediction and the mean prediction. To calculate this contribution, the classical method requires retraining the model on all feature subsets $S \subseteq F$, where F is the set of all features [123]. It assigns an importance value to each feature that represents the effect on the model prediction of including that feature. To compute this effect, a model $f_{S\cup\{i\}}$ is trained with that feature present, and another model f_S is trained with the feature withheld. Then, predictions from the two models are compared on the current input $f_{S\cup\{i\}}\left(x_{S\cup\{i\}} - f_S(x_S)\right)$ where x_S represents the values of the input features in the set S. Since the effect of withholding a feature depends on other features in the model, the preceding differences are computed for all possible subsets $S \subseteq F \setminus \{i\}$. The Shapley values are then computed and used as feature attributions. They are a weighted average of all possible differences:

$$\phi_i = \sum_{S\subseteq F\setminus\{i\}} \frac{|S|!\,(|F|-|S|-1)!}{|F|!} \cdot \left[f_{S\cup\{i\}}\left(x_{S\cup\{i\}}\right) - f_S(x_S)\right] \tag{7}$$

The value of the ith feature contributed ϕ_i to the prediction of this particular observation compared to the average prediction for the data set. This classical procedure is considered theoretically optimal in the sense that it is the only set of additive values that satisfies important properties [124]. Nonetheless, it requires retrain the model on $2^{|F|}$ possible subsets of the feature value. To avoid this high computational cost, Štrumbelj et al. [125] proposed an approximation of Eq. (7). It uses Monte Carlo sampling and approximates the effect of removing a feature from the model by integrating over samples from the training data set. The number of Monte Carlo samples for estimating the Shapley value $\hat{\phi}_i$ is selected here using a convergence analysis.

The convergence analysis is performed by computing the results of the estimated Shapley values using different k number ($k \in \mathbb{N}$) of Monte Carlo simulations (n_{MC}). Convergence is first analyzed visually by examining the stability of the Shapley values with increasing n_{MC}. Further, a quantitative convergence analysis is performed by evaluating

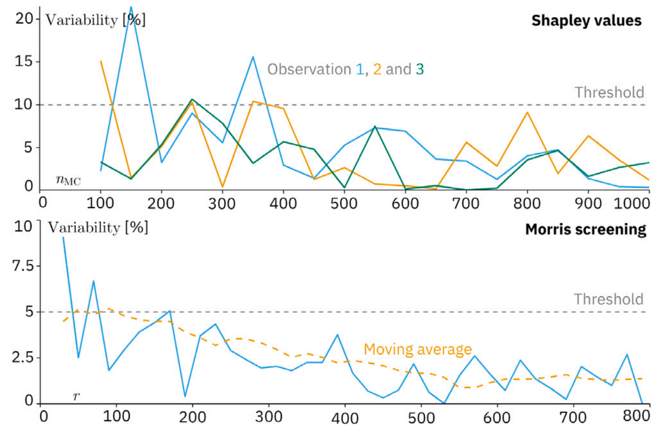


Fig. 10. Visual convergence analysis carried out for the local sensitivity analysis based on Shapley values (upper plot), and the global sensitivity analysis based on the Morris's elementary effects screening method (lower plot).

a total estimated Shapley value $\hat{\phi}_T$ as the sum of the estimated absolute Shapley value $|\hat{\phi}_i|$ of all input features (n_{F}):

$$\hat{\phi}_T = \sum_{i=1}^{n_{\mathrm{F}}} |\hat{\phi}_i| \tag{8}$$

The variability of $\hat{\phi}_T$ is expressed as the relative percentage of change of $\hat{\phi}_T$ from $n_{\mathrm{MC,k-1}}$ to $n_{\mathrm{MC,k}}$ (Eq. (9)). The convergence is considered achieved with $n_{\mathrm{MC,k}}$ if the variability stays lower than a threshold value of 10%.

$$\text{Variability} = \left| \frac{\hat{\phi}_T(n_{\mathrm{MC,k-1}}) - \hat{\phi}_T(n_{\mathrm{MC,k}})}{\hat{\phi}_T(n_{\mathrm{MC,k-1}})} \right| \cdot 100 \tag{9}$$

Fig. 10 shows an example of the convergence analysis. The x-axis represents the number of Monte Carlo samples $n_{\mathrm{MC,k}}$, and the y-axis the variability (Eq. (9)) for three different observations. The minimum number of Monte Carlo samples that achieves a variability lower than the threshold 10% is $n_{\mathrm{MC}} = 400$. In order to keep the computational cost as low as possible, 400 is the number of Monte Carlo samples selected for estimating Shapley values.

(ii) Global sensitivity analysis is conveyed by three complementary methods: (ii.i) Morris's elementary effects screening method [126]; (ii.ii) performance-based feature sensitivity [93]; (ii.iii) and Shapley values [125]. Global sensitivity analysis based on high linear assumptions cannot been implemented for this regression problem – *e.g.* Standardized Regression Coefficients [127]. The coefficient of determination R^2 computed by the multivariate linear regression between input features (Table 3) and output ($\hat{R}$) is 0.60, 95% HDI [0.58, 0.61]. This value is smaller than the minimum threshold of 0.7 that is recommended to obtain effective results [127].

(ii.i) Morris's elementary effects screening method is based on a one-at-a-time perturbation of the model inputs under investigation. The model input space is firstly discretized by transforming the input factors into dimensionless variables in the interval (0,1). Then, each input interval is divided into a number of p levels. This grid is sampled at a random starting point, and the next samples differ only in one coordinate from the preceding one. For each sample a perturbation Δ of the factor value is considered as a multiple of $^{1}/_{(p-1)}$. The sequence of $k+1$ points is called a trajectory. One point in this trajectory represents one prediction of the model. The magnitude of variation in the model output Y due to the predefined variation of one input X is called elementary effect (EE):

$$EE_i = \frac{Y(X + e_i \cdot \Delta) - Y(X)}{\Delta} \tag{10}$$

where e_i is a vector of zeros, except for the ith component that equals ±1. It represents an incremental change in input i. While one trajectory allows the evaluation of one elementary effect for each input i, a set of r trajectories enables statistical evaluation of the finite distribution of the elementary effects. The elementary effects are evaluated by the mean of their absolute value (μ^*) and their standard deviation (σ):

$$\mu_i^* = \frac{1}{r} \cdot \sum_{j=1}^{r} |EE_i^{(j)}| \tag{11}$$

$$\sigma_i = \sqrt{\frac{1}{r} \cdot \sum_{j=1}^{r} \left(EE_i^{(j)} - \frac{1}{r} \cdot \sum_{j=1}^{r} EE_i^{(j)} \right)} \tag{12}$$

The number of r trajectories is selected by a convergence analysis. The analysis is carried out following a similar fashion than with the Shapley values method. In this case, the variability is computed using the variables μ_i^* and σ_i. Due to the stability of the variability, the threshold value is selected to 5%. The bottom plot of Fig. 10 shows the convergence analysis as the evolution of the variability against the number of r trajectories. The selected number of r trajectories is 200 with $p = 8$ grid levels. This leads to a total of 5600 model predictions.

The parameter μ_i^* is a measure of influence of the ith input on the output. The larger μ_i^* is, the more the ith input contributes to the dispersion of the output. The parameter σ_i is a measure of non-linear and/or interaction effects of the ith input. If σ_i is small, elementary effects have low variations on the support of the input. Thus, the effect of a perturbation is the same all along the support, suggesting a linear relationship between the studied input and the output. On the other hand, the larger σ_i is, the less likely the linearity hypothesis is. Hence, a variable with a large σ_i will be considered having non-linear effects, or being implied in an interaction with at least one other variable. Non-linearity and feature interaction are linked together by the standard deviation (σ). In order to support the Morris screening method by assessing specifically on feature interaction, the Friedman's H-statistic method is applied [128].

Friedman's H-statistic method measures two-way feature interaction effects via the decomposition of the prediction function. If a feature i has no interaction with any other feature, the prediction function can be expressed as the sum of the partial dependence function [129] that depends only on i and the partial dependence function that only depends on features other than i. If the variance of the full function is completely explained by the sum of the partial dependence functions, there is no interaction between feature i and the other features. Any variance that is not explained can be attributed to the interaction and is used as a measure of interaction strength. The interaction is measured by Friedman's H-statistic (square root of the H-squared test statistic) and takes on values between 0 (no interaction) to 1 (100% of standard deviation of the full function due to interaction).

Morris screening method allows ranking the input features in order of importance. Nonetheless, it does not measure how much of the model's output varies for a feature considering what it means for prediction accuracy. To complement the Morris screening method on this matter, a performance-based feature sensitivity analysis is carried out.

(ii.ii) Performance-based feature sensitivity is calculated by the increase in the model's prediction error after permuting its values [93]. A feature is sensitive if shuffling its values increases the model error because the model relied on the feature for the prediction. A feature is not sensitive if shuffling its values leaves the model error unchanged because the model ignored the feature for the prediction. This procedure breaks the relationship between the feature and the true outcome, and also the interaction effects with other features.

Permutation of feature values adds randomness to the measurement. When the permutation is repeated, the results might vary greatly. Repeating the permutation and averaging the importance measures over repetitions stabilizes the measure. The maximum number of repetitions is $n \cdot (n-1)$, which is computationally expensive when the data set has a high number of n observations. Therefore, the number of repetitions is selected by a convergence analysis. The analysis is carried out following a similar fashion than with the previous convergence analyses. In this case, the variability is computed using the performance metric MASE. Based on the results, the selected number of repetitions is 15.

(ii.iii) The local sensitivity analysis carried out previously with Shapley values is extended here with a global scope. Shapley values are computed for each feature i of every observation. Then, the average is computed on the absolute values for every feature. The higher the mean, the greater the impact of that feature i on the model's outcome.

3.4.5. *Feature effects*

The aim of the Feature Effects module is to study the direction and magnitude of change in the predicted outcome due to changes in feature values. Since almost all uncertainty in the predicted outcome $\hat{Y}$ is given by the uncertainty in the remainder component $\hat{R}$, the objective is then to identify how the remainder component $\hat{R}$ behaves under changes of input features. The Feature effects are analyzed for Level 1 (Table 3), Level 2 and Level 3 of the stack ensemble architecture presented on Fig. 6.

Feature effects are analyzed on a local and global scope. Local feature effects allow understanding how the model response $\hat{R}$ changes if a selected input feature is changed from a specific observation, while keeping all other features fixed. They are graphically inspected by individual conditional expectations plots (ICE, [130]). ICE plots show a conditional expectation of the dependent variable ($\hat{R}$) for a particular explanatory feature. The values for a specific feature and instance are computed by following two steps. First, while all other features are kept the same, the feature's instance value is replaced with values from a grid taken from the feature entire range of values. Secondly, the model makes predictions for these newly created instances. The result is a set of predicted new $\hat{R}$ points corresponding to the set of feature grid point values.

Global feature effects show the way a feature impacts the model response $\hat{R}$ on the entire range of instances. In order to summary the ICE profiles obtained for every observation, they are graphically displayed by percentiles. For a specific feature, an ICE profile is firstly computed for each observation of the data set. Then, the percentiles 1%, 5%, 25%, 75%, 95% and 99% are calculated from the set of ICE profiles. Plotting the percentiles of the ICE curves uncover possible heterogeneous effects that can be hidden on the entire range of instances. The average of the ICE profiles is called partial dependence curve (PD, [129]). It shows how the average prediction $\hat{R}$ changes when a specific feature is changed.

The global relationship between features and predicted outcome is supported by Shapley values. They help uncover the curvature relationship between the predicted response ($\hat{R}$) and the individual feature. Plots can show a rug – *i.e.* indicators for data points – on the x-axis. It represents the feature distribution so that regions with almost no data cannot be over-interpreted.

3.4.6. *Model simplification*

The Model Simplification module is the last model analysis carried out by the proposed framework. It aims at drawing summary conclusions about the model. This is done by providing an intrinsically interpretable model that approximates the predictions.

The selected intrinsically interpretable model is a decision tree [131]. Decision trees are directed graphs in which each interior node corresponds to an input feature. The terminal nodes (or leaf nodes) represent a value of the target variable given the values of the input variables represented by the path from the root to the leaf. To predict the outcome in each leaf node, the average outcome of the training data in this node is used. The paths can be visualized with simple if-then rules. In short, decision trees are data-derived flowcharts that follow a boolean-like logic. As such, they are displayed graphically in a natural way that is easy to interpret.

Variable importance and interactions displayed in the surrogate model are assumed to be indicative of the internal mechanisms of the complex model. Variables that are higher or used more frequently are more relevant. Variables that are above and below one another can have interactions. The decision tree surrogate model has a global focus of interpretation. Nonetheless, local behavior can also be visualized by highlighting the paths of specific instances through the internal nodes.

The decision tree surrogate model is trained on the original inputs and predictions of the stack ensemble model. A depth of four nodes is chosen as a trade-off between accuracy and interpretability. The decision tree is tuned by time-series cross-validation. The coefficient of determination (r^2_{Pearson}) between forecasts is calculated to ensure that the decision tree surrogate model approximates the stack ensemble model reasonably well.

3.5. *Deployment into production*

The framework has been deployed on a proprietary system with an Intel® Xeon™ Processor E5607 (4 cores, 2.26 GHz, 8 MiB of cache size). The system includes 8 GiB of DDR3 RAM and two 250 GiB SATA hard drives. Ubuntu 18.04.3 LTS runs as the operating system.

The system incorporates tools for efficient data and model governance. All data is automatically backed up on a frequent basis to prevent loss. Data integrity is checked to assure accuracy among backups. Code versioning is controlled through a private Git repository (v2.24.0). Data is secured by a high-quality software that protects against malware. Unauthorized data access is prevented by a firewall that blocks all ports. No personal data is allowed to be stored in the system. Finally, modeling and system errors are registered and archived along with their effects. This governance is formally documented and approved by the system administrators.

The framework is mostly implemented in R programming language v3.6.3 [132]. The framework data is stored on a relational database managed by PostgreSQL v12.1. The back-end system architecture is developed in Java v8. Trained machine learning models are efficiently stored to disk for increasing computing performance. R-programming models are saved via serialization into non-compressed files of *rds* format. No trained statistical time series models are stored in the system.

4. Results and discussions

This section discusses the results obtained by the proposed framework. It follows the sequence of results that the Model explainability phase can show based on its six modules (see Fig. 1). First, the main characteristics of the input data are summarized in the Data understanding module. Then, model quality and goodness of fit is assessed by the Model performance module. Thirdly, model validation is audited through residuals' diagnostics (Model audit module). The influence and effects of model inputs on the electricity price forecasts are studied on the fourth (Features sensitivity) and fifth modules (Features effects). Finally, the Model simplification module builds a surrogate model that helps provide general model characteristics.

4.0.1. *Data understanding*

Fig. 11 shows five charts, labeled from *i* to *v*. Plot *i* depicts the decomposition of the electricity price time series into six components of distinctive pattern. Here, one month of data is represented. The irregular electricity price time series (top plot, Y) is decomposed into a smooth trend that indicates the long-term change (T) and four seasonal components that describe specific patterns that reoccur periodically at 12 h (S_{12}), 24 h (S_{24}), 84 h (1/2 week, S_{84}) and 168 h (1 week, S_{168}). The remainder component (bottom plot, R) does not exhibit any clear behavior. It is interesting to note that the magnitude of the remainder term is approximately twice larger than that of the seasonal components. In addition, several abrupt changes can be observed in the remainder component. These observations indicate a high volatility of the electricity price.

Plot *ii* shows the daily (left) and weakly (right) patterns described by the detrended electricity price ($Y - T$). The relative amount of data is presented by a gradient color, from 10% (dark blue) to 90% (light blue). The electricity price shows a clear daily pattern, with one expected minimum at four in the morning, and two maximums at around nine in the morning and afternoon. Spanish weekdays (Monday to Friday) do not exhibit visual difference among them, whereas a price drop is seen for weekends (Saturday and Sunday).

Plot *iii* depicts the correlation that the electric power demand predicted for the Iberian Peninsula has on the electricity price. The correlation is visualized by scatter plots for the detrended time series ($Y - T$) on the left, and the remainder component time series (R) on the right. The detrended terms are moderately correlated ($r_{\text{Pearson}} = 0.72$, 95% HDI [0.71, 0.73]). The remainder terms are low correlated ($r_{\text{Pearson}} = 0.23$, 95% HDI [0.21, 0.25]). This can be caused by the influence that the seasonal components of the electric power demand have on the electricity price counterparts. The supply and demand relationship is emphasized by a positive correlation value, meaning that higher expected power demands tend to imply higher electricity prices.

Plot *iv* shows the correlation strength between the exogenous data (Table 6) and the electricity price. The correlation is calculated by the absolute value of the Pearson correlation coefficient ($|r_{\text{Pearson}}|$). Four categories are colored depending on the correlation strength: high (yellow), moderate (green), weak (blue) and no correlation (purple). If the exogenous data time series exhibit seasonal patterns, their trends (left) and seasonal components (right) are removed. Hydraulic power, cogeneration, combined cycle and coal power are the highest correlated with the electricity price, both on the detrended ($Y - T$) and the remainder components (R). It is interesting to note here that the hydraulic power generation in the Iberian Peninsula is usually the last to enter the wholesale market actions, fixing the final electricity price [78]. Coal-fired power stations usually follow hydraulic plants in the face of gas shortages in cogeneration and combined cycle plants [78]. In addition to fixing the price, due to the flexibility of operation of these power plants, it is logical that they sell their energy at high expected electricity prices, giving thus a high correlation. No correlation is observed for the EUR/USD currency rates, Brent oil prices and CO_2 European emission allowance prices.

Finally, Plot *v* displays the annual wavelet power spectrum of the detrended term ($Y - T$, top plot) and the remainder term (R, bottom plot) of the electricity price. The detrended series presents high wavelet power at distinctive time periods (12 h, 24 h, 168 h and, to a lesser extent, 84 h). This is not the case for the Spanish summer holiday period: a low wavelet power is observed starting from July to September. This means that the price of electricity behaves differently during the summer vacation season.

4.0.2. Model performance

Fig. 12 shows four charts, labeled from *i* to *iv*. Plot *i* represents a month comparison between actual electricity prices and forecasts. Actual and predicted values are shown in black and blue color, respectively. The top plot shows the whole electricity price series (Y). The bottom plot shows the detrended and deseasonalized electricity price series (R). Visually, forecasts overlap actual prices in trend and seasonality. The highest error variances are given by the uncertainty of the remainder component forecast. These results are quantitatively detailed in the following plot.

Plot *ii* shows the mean absolute value (MAE, right plot) and the mean absolute scaled error (MASE, left plot). These metrics are calculated for the year 2017. The average is marked with a cross. The 95% highest density interval of the metrics distribution (95% HDI) is plotted with a horizontal line. The average MAE is $1.859\,\text{EUR}\,(\text{MWh})^{-1}$, 95% HDI [$0.575\,\text{EUR}\,(\text{MWh})^{-1}$, $3.924\,\text{EUR}\,(\text{MWh})^{-1}$]. The largest proportion of this value is given by the remainder component forecast (R), whose average MAE is $1.867\,\text{EUR}\,(\text{MWh})^{-1}$. The MAE of the trend ($T$) and the seasonalities (S) is relatively small. The average MAE increases with the period of the seasonality. Nevertheless, the largest MAE is usually under $0.2\,\text{EUR}\,(\text{MWh})^{-1}$ for S_{12}, S_{24} and S_{84} seasonalities. The MAE of the seasonality S_{168} is usually under $0.5\,\text{EUR}\,(\text{MWh})^{-1}$.

When a *naïve* model is used as benchmark, the proposed model architecture achieves an average MASE of 0.378, 95% HDI [0.091, 0.934]. The 95% HDI does not cover the unity benchmark reference. This means that the proposed model obtains statistically significantly better forecasts than the benchmark model. It is not the case for the remainder component forecast (R). However, it achieves a relevant average reduction of 24% (MASE = 0.765). Due to the importance of an accurate forecast of the remainder component, plot *iii* presents its forecasting performance details.

Plot *iii.i* shows the individual models' performance for the pipeline proposed in Fig. 6. The models are ranked in descending order based on the average MASE obtained for the year 2017. Time series statistical models' performances (Level 0) are colored in purple, blue for Level 1, green for Level 2 and yellow for Level 3. The plot top shows the MASE performance of the model of Level 3. This model gives the final values of the remainder component forecast. The model achieves an average MASE of 0.765, as already presented in Plot *ii*. As levels decrease, the performance tend to decrease by obtaining a higher average MASE. The most accurate model of Level 0 is the simple average of the 18 forecasts obtained in the feature engineering step. Those forecasts were obtained by time series statistical models. The most accurate models of Level 1 and Level 2 are a boosting linear model and a bagged ensemble of multivariate additive regression splines, respectively.

Plot *iii.ii* presents the grouped levels' performance of the proposed model architecture. On average, the models of Level 1 give 11.3% better results than Level 0 models; 2.3% of Level 2 with respect to Level 1; and 1.3% for Level 3 with respect to Level 1. The resulting figure of 14.9% improvement has been achieved by stacking models. Moreover, the stack ensemble architecture has obtained a better combined forecast than any individual model alone.

The average MASE of each model (Plot *iii.i*) and level (Plot *iii.ii*) is joined together by a dotted gray line. This line converges to a vertical asymptote of around 0.765 MASE. This means that the stacked generalization technique has obtained an asymptotically *optimal* learning. That is to say, for the given input features, no significantly better results could have been obtained by introducing more levels in the ensemble stacked architecture. This is also true if more models had been placed in each level.

The AutoML frameworks' results help assure that no significantly better accuracy could have been obtained for the given input features. H2O AutoML and TPOT get a MAE of $1.886\,\text{EUR}\,(\text{MWh})^{-1}$ and $1.912\,\text{EUR}\,(\text{MWh})^{-1}$, respectively. That is to say, a MASE of 0.773 and 0.784, respectively. These results are 1.0% and 2.4% higher than the achieved MAE of $1.867\,\text{EUR}\,(\text{MWh})^{-1}$ and MASE of 0.765.

On other hand, several models of Level 1 (blue line) achieve on average better results than two models of Level 2 (a linear model and a feed-forward neural network). These two models, together with a bagged ensemble of multivariate adaptive regression splines have been selected as the ones that increase Level 3 performance the most. Thus, it is interesting to note that not placing the best individual models on Level 2 has improved overall performance more than if individual better models were selected. This is due to models's diversity. Diversity is presented in the following plot.

Plot *iv* shows correlograms of the residuals for Level 1 and Level 2 models. Here, low correlation coefficients mean higher diversity — *i.e.*, gain increase by combination —. The correlations of Level 1 span from 0.77 (yellow) to 1.00 (green). A high average value of 0.93 (95% HDI [0.92, 0.94]) stems from good models' accuracy. Correlations are ordered according to the degree of association. The lower part of the correlogram (MARS, SVM, LM, PLS, GP, BaLM, BoLM and PLM) is characterized for the linearity of the models' forecast. Their correlation

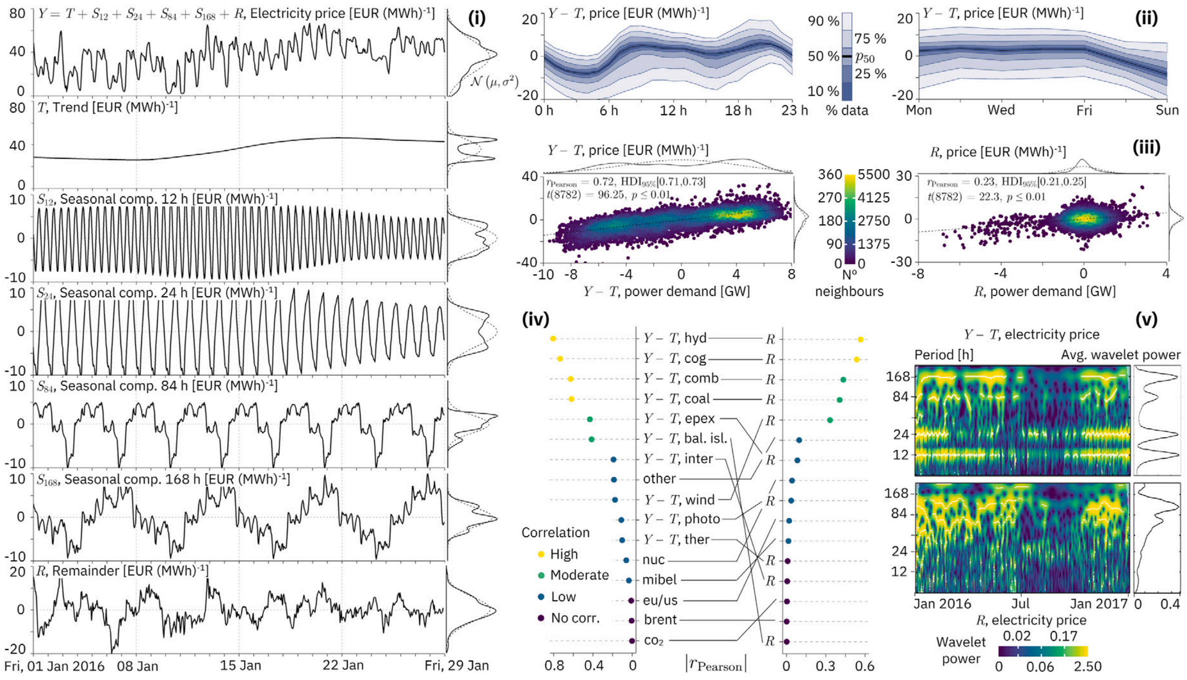


Fig. 11. Visual representations of the Data Understanding module. (i) STL decomposition. (ii) Daily (left) and weakly (right) patterns. (iii) Correlation with electricity price, detrended (left) and detrended and deseasonalized (right). (iv) Correlation strength rank for exogenous factors. (v) Wavelet power spectrum for the detrended (top), and detrended and deseasonalized (bottom) electricity price time series.

is around 1.0. This means that the residuals are very similar. It implies that one model would have contributed nearly the same than using the eight of them. Nevertheless, the use of all these "linear models" decreased a bit the overall performance. Regression tree-based models are also grouped together (GBDT, RF and BART). They are high correlated among them, and less with the "linear models". Three types of model residuals differ more from the rest: a feed-forward neural network (FFNN), a rule- and instance-based model (RIBM) and a *k*-nearest neighbor model (KNN). Although their individual performance is not among the best, their forecasts help increasing overall model performance. As for Level 2, the average correlation is increased to 0.95, 95% HDI [0.89, 1.00]. A higher average correlation comes from the fact that, as the level increases, models tend to give closer (and more accurate) forecasts. Again, although the individual performance of a FFNN is not among the best, it helps increasing overall performance by giving diversity.

4.0.3. Model audit

Fig. 13 shows five charts, labeled from *i* to *v*. The forecasts of the year 2017 are used here. Plot *i* presents a high linearity between observed (Y) and predicted ($\hat{Y}$) electricity price. The linearity is assured in three different ways: (i) a high coefficient of determination ($r^2_{\text{Pearson}} = 0.956$, 95% HDI [0.954, 0.957]); (ii) a near zero value of the intercept of a simple linear regression fit ($\beta_0 = -0.91$, 95% HDI [−1.17, −0.66]); (iii) and a slope almost equal to unity ($\beta_1 = 1.017$, 95% HDI [1.012, 1.022]). It can also be observed that electricity prices smaller than $25\,\text{EUR}\,(\text{MWh})^{-1}$ are scarce and more difficult to predict.

Plot *ii* shows that the residuals are normally distributed. Plot *ii.i* presents the visual comparison between the residuals' density histogram (in black color) and the normal distribution $\mathcal{N}(\mu = -0.045, \sigma^2 = 6.04)$ (in blue color). The highest density interval of the average μ (95% HDI [−0.099, 0.009]) contains the ideal (normal) zero value. It means that the average of the residuals is not significantly different from zero. It is important to consider here that the authors have not increased the model forecasts by $0.045\,\text{EUR}\,(\text{MWh})^{-1}$ to achieve a zero mean. The reason is that the model audit is carried out on test data set, not on training/validation data set. With respect to the shape of the actual distribution, it resembles a normal distribution. Nevertheless, it has heavier tails. In this sense, it could be considered as a logistic distribution with location parameter $\mu = -0.052$ and scale parameter $s = 1.339$.

The heavier tails are better visualized on the quantile–quantile Plot *ii.ii*. The tails imply a high kurtosis (Kurt = 4.8, 95% HDI [4.4, 5.3]). In this sense, the Anscombe–Glynn test finds evidence of a kurtosis different from the ideal (normal) value of three. In addition, the Shapiro–Wilk test and the Anderson–Darling test reject the normal distribution of residuals. However, the skewness is not significantly different from the ideal (normal) value of zero. This is confirmed by the D'Agostino test and a skewness highest density interval that includes zero (95% HDI [−0.15, 0.12]).

Plot *iii* displays the residuals' variance homogeneity. The residuals do not show a clear variance change across the predicted electricity prices. Moreover, the residuals form an approximate horizontal band around the zero line. This indicates homogeneity of the error variance. However, the studentized Breusch–Pagan test, the Goldfeld–Quandt test and the Harrison–McCabe test do not support the homoscedasticity claim.

Plot *iv* indicates a residual linear relation of errors with respect to errors at prior time steps. The top plot shows the auto-correlation function (ACF). Linear dependence decreases with lags up to lag 24. The bottom plot shows the partial auto-correlation function (PACF). When the effect of indirect correlations are removed, no high dependence is observed for any particular lag.

Finally, Plot *v* presents an example of outlier detection of residuals. The month of January 2017 is split into two plots. Outliers are presented by blue single dots. The plot is used as a first step for analyzing the cause of an outlier. As an example, the third outlier observed in January will be subsequently analyzed in the Feature sensitivity section.

4.0.4. Feature sensitivity

Fig. 14 shows an example of a local sensitivity plot. It analyzes the residual outlier observed at 22 h of the 12th of January 2017 (see Plot *v*, Fig. 13). It depicts the contributions of each feature to the

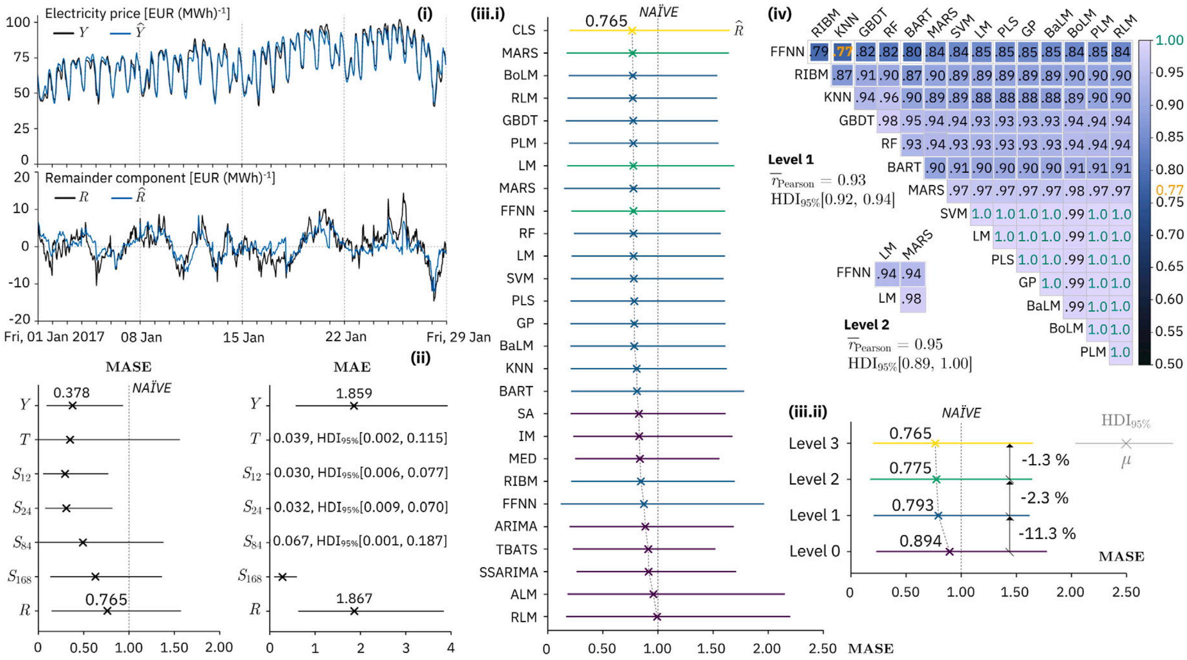


Fig. 12. Visual representations of the Model Performance module. (i) Forecasts of the electricity price time series. (ii) MASE and MAE metrics of the forecasts of the electricity price time series (iii) Models' and levels' performance in forecasting the remainder component R. (iv) Level 1 and Level 2 model diversity represented by the correlogram of the residuals.

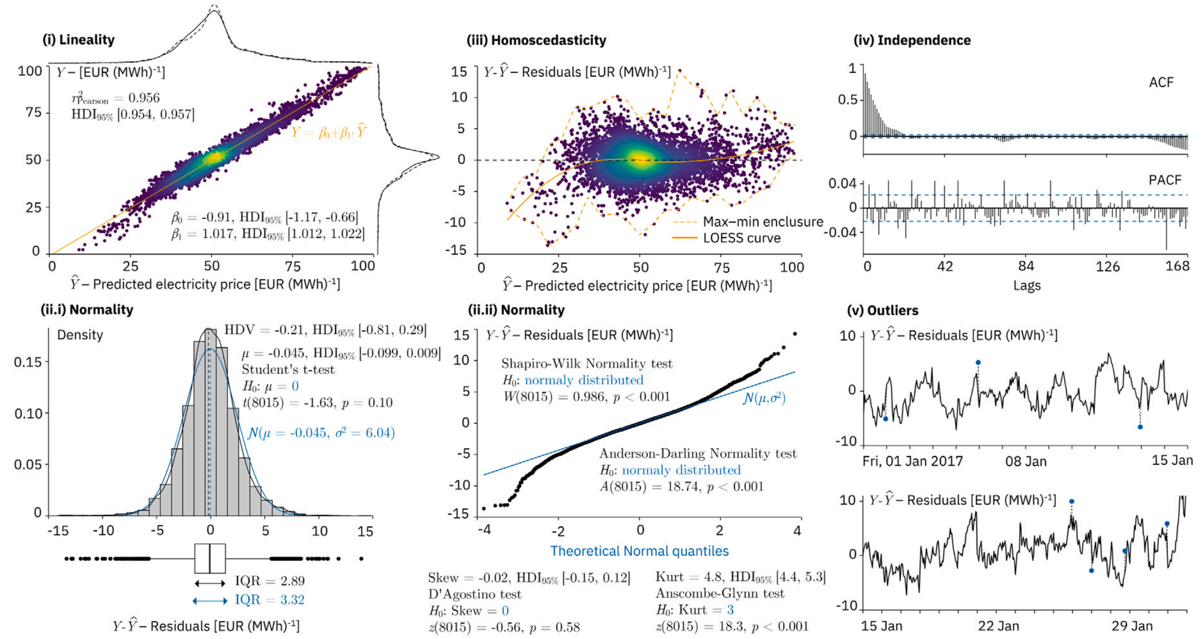


Fig. 13. Visual representations of the Model Audit module. (i) Linearity inspection between observed and predicted electricity price. (ii) Normality examination of the distribution of residuals by histogram (i) and quantile–quantile plot (ii). (iii) Homoscedasticity audit of residuals. (iv) Independence inspection of residuals. (v) Outlier detection of residuals.

difference between the prediction of the remainder component of the electricity price (1.17 EUR $(\mathrm{MWh})^{-1}$) and the mean prediction for that day (0.11 EUR $(\mathrm{MWh})^{-1}$). The actual value is -5.47 EUR $(\mathrm{MWh})^{-1}$. Contributions are ranked in descending order beginning from the largest, placed at the top. Positive contributions increase forecast values, and are colored in blue. Negative contributions are colored in yellow. The right plot shows the contributions for Level 1 features. The left plot shows the contributions for Level 2 features, *i.e.* the machine learning models used on the stacking ensemble architecture (see Fig. 6). The values of each feature at that specific instance is written in gray color. For example, the simple average (SA) forecast by the statistical time-series models is 1.84 EUR $(\mathrm{MWh})^{-1}$. Whereas for Level 2, the Gaussian Process model (GP) predicts 0.94 EUR $(\mathrm{MWh})^{-1}$.

Based on the presented results, the outlier has originated from the model giving to much importance to linear characteristics. The most influential feature of Level 1 is the linearity of the trend component of the STL decomposition (nº 33, Table 7). It is based on the coefficient of a linear regression applied to the trend component. The linearity seems to affect also Level 2 features. The most influential features of Level 2 give linear outcomes: the Gaussian Process model and the Bayesian Linear model. Both models gave the same forecast, 0.94 EUR $(\mathrm{MWh})^{-1}$.

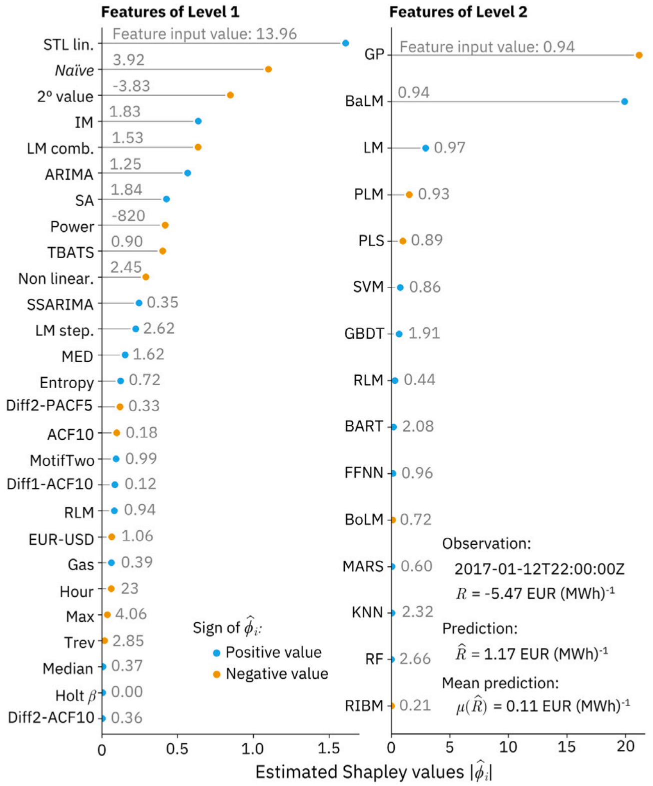


Fig. 14. Example of a local sensitivity analysis carried out on an instance with a residual outlier.

To sum up, a local sensitivity analysis has helped to evaluate how the model output is influenced by the model inputs when predicting for one specific observation. In contrast, the global sensitivity analysis shown on the following plots will assess the feature influence over the entire set of observations.

Fig. 15 shows three charts, labeled from *i* to *iii*. Plot *i* presents the global sensitivity results obtained by the Morris's elementary effects screening method. The X-axis represents the absolute value of the elementary effects (μ^*, Eq. (11)). The Y-axis represents their standard deviation (σ, Eq. (12)). Based on μ^*, the three input features that contribute the most to the dispersion of the model output are forecasts obtained by statistical time-series models: a state-space ARIMA model (feature n° 30, Table 8), a TBATS model (n° 8) and an ARIMA model (n° 3). In addition, based on σ, these three input features have the largest non-linear effects or the largest interactions with at least one other feature. In order to support the Morris screening method by assessing specifically on feature interaction, the results of Friedman's H-statistic method are presented in Plot *ii*.

The effects of two-way feature interactions are small. The maximum observed H-squared test statistic is limited to 0.12. This value is obtained by two features: the sum of squares of the first ten autocorrelation coefficients of the differenced time series (feature n° 30, Table 8) and a *naïve* model forecast based on carrying forward the last observation.

Plot *iii* shows the features' global sensitivity obtained by three different methods: the Morris's elementary effects screening method, the performance-based method, and the method based on Shapley values. The variables are ordered by their average degree of sensitivity (marked by blue dots), with the top variable having the greatest influence on the final model output. The three methods give a similar order, although the Morris's screening method tends to obtain a different order as the sensitivity decreases.

Forecasts of statistical time series models (Table 8) appear as the most sensitive features: a state-space ARIMA model (feature n° 30), a *naïve* model, a TBATS model (n° 8) and an ARIMA model (n° 3). Linear combinations of the forecasts of the statistical models have less sensitivity: interquartile mean (IM), simple average (SA) and median (MED). However, they give slightly better forecasts (see Plot *ii.i* of Fig. 12).

The features based on time series characteristics (Table 7) are in second position in terms of sensitivity. The three most sensitive input features are the following: (i) the linearity of the trend component of the STL decomposition (n° 33). It is based on the coefficient of a linear regression applied to the trend component. (ii) The second value of the previous day for the remainder component of the electricity price (n° 8). And (iii) the statistic of Teraesvirta's neural network test for neglected non-linearity (n° 42).

The features based on exogenous factors (Table 6) are in third position in terms of sensitivity. The remainder component of the electric power forecast for the Iberian Peninsula (n° 12) standouts from any other exogenous factor. This evidences a strong relationship between supply and demand. It is followed by the EUR/USD currency rate (n° 57) and the natural gas price in the Iberian market (n° 54). These two factors are related with coal-fired power stations and combined cycle plants. Both types of power plants usually enter the market in the last position following the hydraulic ones. They bid with high prices and very few hours of use (especially the combined cycle plants) because they need to recover their fixed costs. As all Spanish market sellers receive the same price regardless of the price they have bid, an influence on the electricity price is expected.

Based on the presented results, features based on time factors (Table 5) are not relevant to forecast the remainder component of the electricity price time series. Only the hour of the day was selected as input feature (n° 1), and it appears among the least influential ones. Note that the seasonality terms of the electricity price have been forecast previously, so theoretically speaking the remainder component should not have time patterns.

The aforementioned global sensitivity methods are also applied to the features of Level 2. The results are shown on Fig. 16. Again, the performance-based method (left plot) and the method based on Shapley values (right plot) give a similar order for the variables. Nevertheless, the 95% HDIs obtained by Shapley values are usually longer. The most influential features are the forecasts obtained by the Gaussian Process model and the Bayesian Lineal model. Nonetheless, these two models do not give the best average results (see Plot *iii.i* of Fig. 12). The bottom plot shows the Friedman's H-statistic. Again, the effects of two-way feature interactions are small among these features.

4.0.5. *Feature effects*

Fig. 17 shows three charts, labeled from *i* to *iii*. Plot *i* shows local feature effects as individual conditional expectation profiles (ICE). The dependent variable is the forecast of the remainder component of the electricity price. The observation selected for the local analysis has a predicted value of 0.42 EUR (MWh)$^{-1}$. The left plot displays local effects for features of Level 1 of the stack ensemble architecture (Fig. 6). Three features are here selected: the second value of the remainder component observed for the previous day (green), the *naïve* model forecast (blue) and the linearity of the trend component of the STL decomposition (yellow). For the selected instance, their values are −3.83 EUR (MWh)$^{-1}$, 3.92 EUR (MWh)$^{-1}$ and 14.0, respectively. The remainder term forecast increases linearly as the second value observed on the previous day increases. With respect to the *naïve* model forecast, the remainder term forecast decreases linearly. The slope of the STL linearity profile is steeper at lower values of the STL linearity. This means that it has higher sensitivity at lower values.

The center plot shows the local effects for features of Level 2. The most sensitive features are here selected. They correspond to the forecasts of a Bayesian linear model, a Gaussian process and a linear

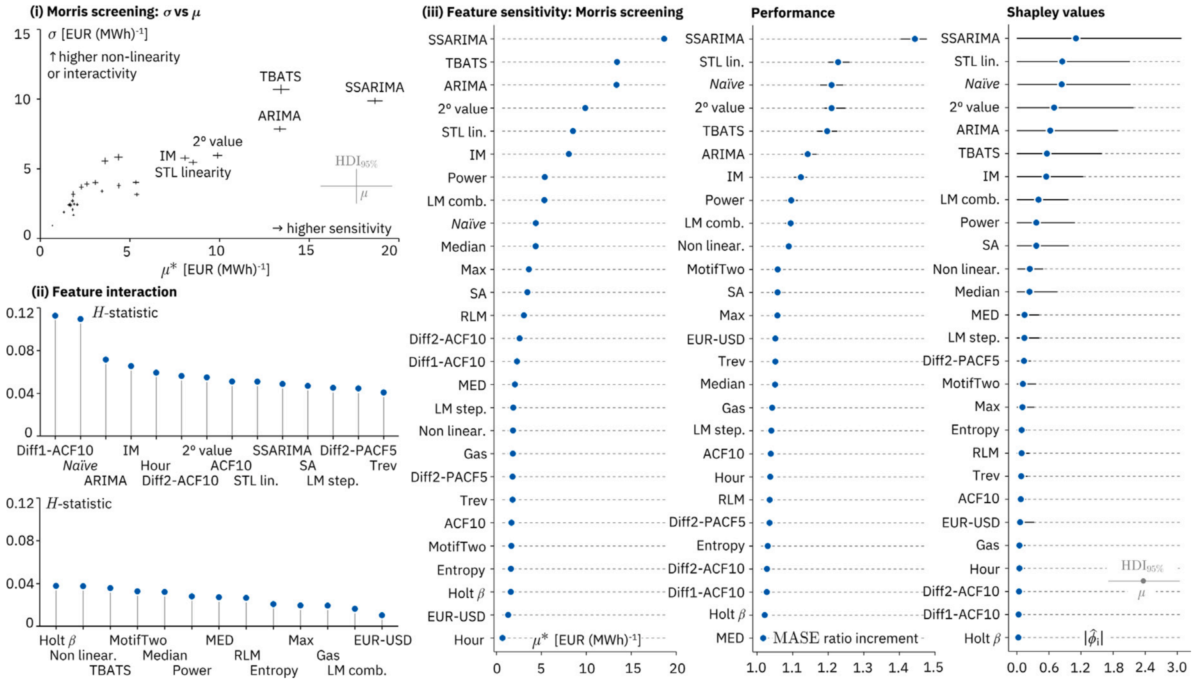


Fig. 15. Visual representations of the Feature sensitivity module. (i) Morris's elementary effects screening method: $\mu*$ *vs* σ. (ii) Overall feature interaction strength; (iii) Feature sensitivity obtained by the Morris's elementary effects screening method, performance-based feature importance method, and Shapley values.

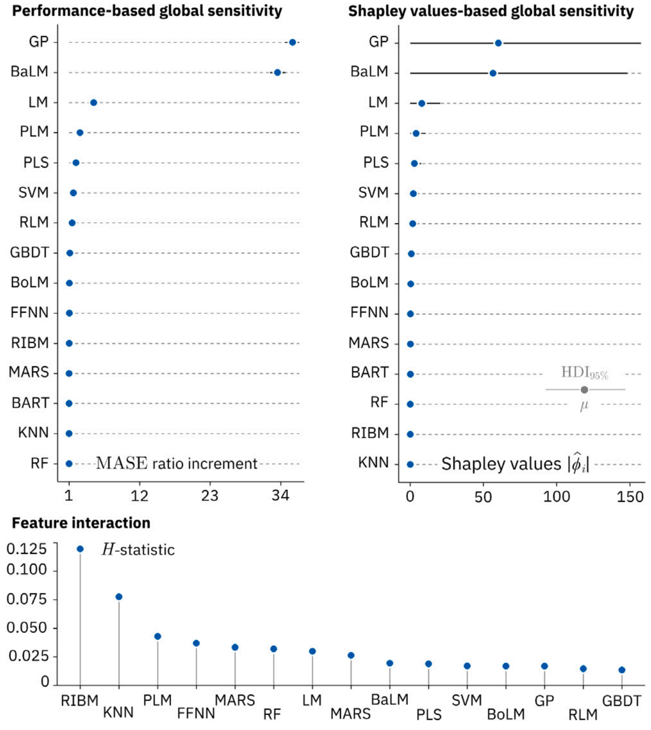


Fig. 16. Global sensitivity plots for the features of Level 2 of the stack ensemble architecture.

Model. For the selected instance, the three models give similar forecasts around $0.42\,\mathrm{EUR}\,(\mathrm{MWh})^{-1}$. They behave linearly with respect to the overall outcome. The Bayesian and Gaussian process models have the largest sensitivity, as denoted by their slope. In addition, the Gaussian process model tends to counter the Bayesian model by providing a similar but opposite slope. This is due to their high correlation.

The right plot shows the local effects for the three features of Level 3. They correspond to the forecasts of a bagged of multiple additive regression splines, a linear model and a feed-forward neural network. As expected, when the level of the stack ensemble architecture increases, the behavior of the individual model forecasts tend to be more linear. In addition, the slope tends to unity and the intercept to zero. Moreover, a similar behavior is observed for the three features. It means that the individual models give roughly the same forecast. Indeed, the three models obtain similar average performances (see Plot *iii.i* of Fig. 12).

Plot *ii* shows several examples of global feature effects. The analysis is carried out for all the observations of the year 2017. For this reason, percentiles of the ICE profiles are displayed instead of one curve for each observation. Three features based on exogenous factors (Table 6) are chosen here for the analysis: the remainder component of the electric power estimated for the Iberian Peninsula (nº 12), the EUR/USD currency rate (nº 57) and the natural gas price in the Iberian market (nº 54). The behavior of these features on the model outcome are displayed on the left, center and right plots, respectively. A positive linear behavior is observed for the three exogenous factors. A steeper slope is found for the electric power forecast. This is consequent with a higher sensitivity (see Plot *iii* of Fig. 15). At the lowest values of the remainder term forecast, the influence of the EUR/USD currency rate and the natural gas price tend to vary as the two factors increase. This is seen by a slight variation of the percentile p_{01} ICE profiles.

The global relationship between features and predicted outcome is supported by Shapley values. Plot *iii* shows the comparison between ICE profiles (left plot) and Shapley values (right plot). In this case, the most sensitive input feature is used for the example. It corresponds to the forecast of a state-space ARIMA model (feature nº 30, Table 8). A positive linear increase is observed on both plots. However, a slight curve relationship is presented at the lowest values of the final prediction. It means that the state-space model tends to overestimate on this range of values. The reason is that there are few observations whose final prediction is below $-10\,\mathrm{EUR}\,(\mathrm{MWh})^{-1}$. This is indicated by the isolated dark blue dots on the southwest part of the Shapley values plot.

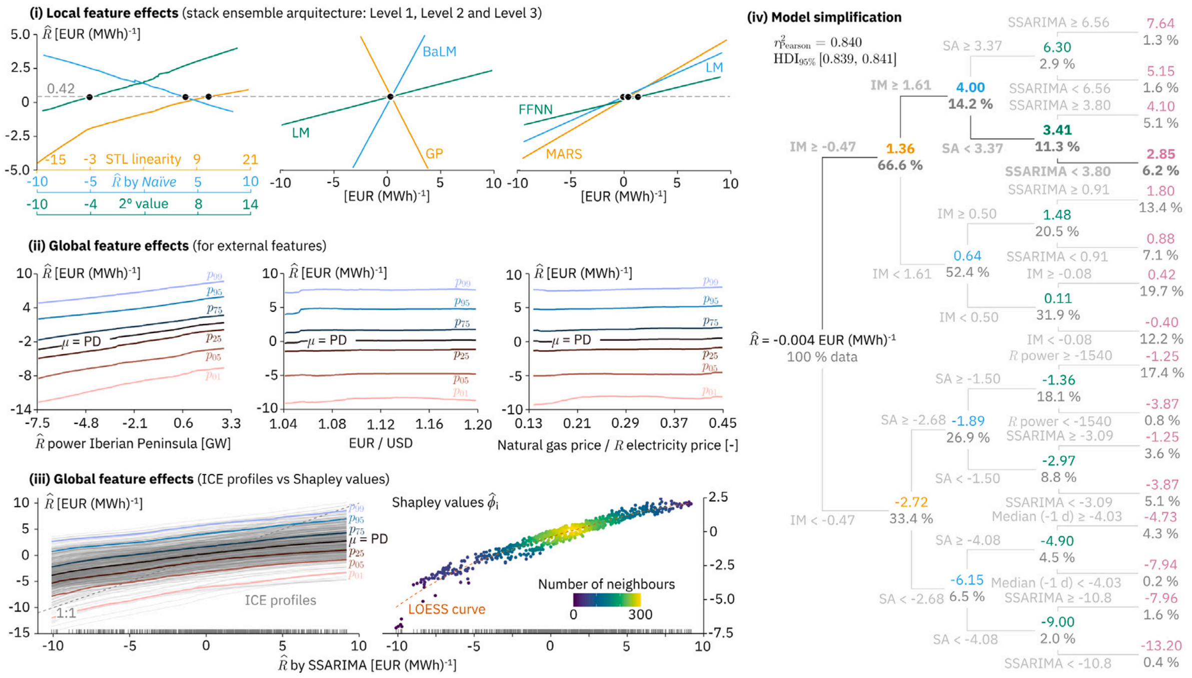


Fig. 17. Visual representations of the Feature effects module (left) and Model simplification module (right). (i) Local feature effects of the stack ensemble model (Fig. 6). (ii) Global feature effects for exogenous factors (Table 3). (iii) Comparison of global feature effects for the state-space ARIMA model. (iv) Model simplification visualization by a decision tree surrogate model.

Table 5
Features based on time factors.

Nº	Features	Range	No	Features	Range
1	Hour of day	$\{0, 1 \dots 23\}$	13	Month of year	$\{0, 1 \dots 11\}$
2–3	Hour of day. Cartesian coord.	$[-1, 1]$	14–15	Month of year. Cartesian coord.	$[-1, 1]$
4	Day of week	$\{0, 1 \dots 6\}$	16	Quarter of year	$\{0, 1, 2, 3\}$
5–6	Day of week. Cartesian coord.	$[-1, 1]$	17–18	Quarter of year. Cartesian coord.	$[-1, 1]$
7	Day of month	$\{0, 1 \dots 30\}$	19	AM/PM	$\{0, 1\}$
8–9	Day of month. Cartesian coord.	$[-1, 1]$	20	Weekday/Weekend	$\{0, 1\}$
10	Week of year	$\{0, 1 \dots 52\}$	21	National holidays	$\{0, 1\}$
11–12	Week of year. Cartesian coord.	$[-1, 1]$			

Table 6
Features based on exogenous factors.

Nº	Description	Units	Features
1–6	Electricity price of the French wholesale market	EUR (MWh)$^{-1}$	$T, S_{12}, S_{24}, S_{84}, S_{168}, R$
7–12	Electric power forecast for the Iberian Peninsula	MW	$T, S_{12}, S_{24}, S_{84}, S_{168}, R$
13–15	Wind-generated electric power	MW	T, S_{24}, R
16	Nuclear-generated electric power	MW	Y
17–19	Coal-generated electric power	MW	T, S_{168}, R
20–25	Combine cycle-generated electric power	MW	$T, S_{12}, S_{24}, S_{84}, S_{168}, R$
26–31	Hydraulic-generated electric power	MW	$T, S_{12}, S_{24}, S_{84}, S_{168}, R$
32–34	International electric interchanges	MW	T, S_{12}, R
35–38	Balearic Islands electric interchanges	MW	T, S_{12}, S_{24}, R
39–42	Photovoltaic solar-generated electric power	MW	T, S_{12}, S_{24}, R
43–46	Solar thermal-generated electric power	MW	T, S_{12}, S_{24}, R
47	Renewable thermal-generated electric power	MW	Y
48–53	Cogeneration power	MW	$T, S_{12}, S_{24}, S_{84}, S_{168}, R$
54	Natural gas price in the Iberian Market	EUR (MWh)$^{-1}$	Y
55	CO_2 European emission allowances	EUR (t CO_2eq)$^{-1}$	Y
56	Brent oil price	USD barrel^{-1}	Y
57	EUR/USD currency rate	EUR USD^{-1}	Y

4.0.6. Model simplification

Plot *iv* of Fig. 17 shows the representation of a decision tree used as surrogate model to draw a summary of the forecasting model. The coefficient of determination r^2_{Pearson} between the forecasts of the surrogate model and the stack ensemble model is 0.840, 95% HDI [0.839, 0.841]. A high strength correlation means that the decision tree surrogate model approximates the stack ensemble model reasonably well.

The features used by the surrogate tree could be considered as relevant. These features are the following: the state-space ARIMA model forecasts (nº 30, Table 8), the simple average (SA) and interquartile mean (IM) of the statistical time-series models' forecasts (Table 3), the median value of the remainder component observed for the previous

Table 7
Features based on time series characteristics.

Nº	Description	Ref.	No	Description	Ref.
1–5	Average, Median, Std. dev., Max., Min.	–	35–37	Holt–Winter's seasonal method: α, β, γ	[133]
6	|Max. – Min.|	–	38–41	Heterogeneity: (ARCH, GARCH)–ACF, –R^2	[133]
7–9	First, Second, Third value	–	42	Non linearity	[133]
10–12	Last, Second last, Third last value	–	43	ARCH statistic	[134]
13	Spectral Shannon entropy	[133]	44–47	Correlation: Embed2 , AC9, FirstMin, trev	[135]
14–15	Stability, Lumpiness	[133]	48–49	Distri.: HistogramMode, OutlierInclude	[135]
16–18	Max level shift, Max var shift, Max kl shift	[133]	50	Entropy: SampEn	[135]
19	Crossing points	[133]	51–52	Forecasting: LocalSimple, LoopLocalSimple	[135]
20	Flat spots	[133]	53	Non-linear time-series analysis: FluctAnal	[135]
21	Hurst	[133]	54–55	Stationary: Std1thDer, SpreadRandomLocal	[135]
22–25	PACF features: (x, diff1, diff2, seas)-pacf5	[133]	56	Symbolic transformations: MotifTwo	[135]
26–27	Holt's linear trend method: α, β	[133]	57	Others: Walker	[135]
28–34	STL features: nperiods, seasonal period and strength, trend, spike, linearity, peak	[133]	58–66	ACF features: (e, x, diff1, diff2, seas)–acf1, (e, x, diff1, diff2)–acf10	[133]

day (nº 2, Table 7), and the remainder component of the electric power estimated for the Iberian Peninsula (nº 12, Table 6).

An example of a local behavior is highlighted on the plot by a specific path. The interpretation of the behavior is as follows: it starts with the average forecast of −0.004 EUR (MWh)$^{-1}$ for all predictions (100% of data). When the IM is above −0.47 EUR (MWh)$^{-1}$, the average forecast is corrected to 1.36 EUR (MWh)$^{-1}$. This is true for the 66.6% of all forecasts. From here and following the same path, the final forecast will be 2.85 EUR (MWh)$^{-1}$ when the IM is above 1.61 EUR (MWh)$^{-1}$, the SA is below 1.36 EUR (MWh)$^{-1}$ and the state-space ARIMA model forecasts a value below 3.80 EUR (MWh)$^{-1}$. This way, an intrinsically interpretable model as a decision tree can help indicating the internal mechanisms of the complex model when the final forecast is around 2.85 EUR (MWh)$^{-1}$.

5. Conclusions

The present article has proposed a novel framework that promotes human–machine collaboration in forecasting day-ahead electricity price in wholesale markets. The article has presented three major contributions (i to iii) to the current state of the art in the electricity price forecasting sector:

(i) A novel model architecture that includes a diverse set of predictors: (i.i) a plethora of statistical time-series models that learn different linear patterns; (i.ii) exogenous factors that affect the clearing prices; (i.iii) a combination of several time series decomposition methods; (i.iv) a collection of time series characteristics based on signal processing and time series analysis methods. These features are fed into a stack ensemble architecture that contains a diverse set of machine learning models that recognize non-linear, complex patterns.

(ii) The use of two open-source AutoML platforms that provide a baseline reference for the proposed model architecture.

(iii) A collection of state-of-the-art model-agnostic methods aimed at interpreting the behavior of the forecasting models and their outcomes. The framework has demonstrated the promotion of a human–machine collaboration by providing a data story. It is based on graphical and numeric explanations that augments understanding on the model and its electricity price point forecasts.

The framework has successfully been applied to the case study of the Spanish wholesale market. It has proven to not only provide accurate predictions, but above all to be a human-centric solution by providing explanations of the behavior of a new model architecture and its forecasts. In particular, the following results (i to xvi) can be highlighted for the Spanish wholesale market:

(i) The framework has successfully split the electricity price time series into four underlying periodical patterns of 12 h, 24 h, 168 h (1 week) and, to a lesser extent, 84 h (1/2 week). This patterns tend to fade in the summer vacation season. The magnitude of the remainder term is approximately twice larger than that of the seasonal components. This indicates a high volatility of the electricity price.

(ii) Analysis on the individual components of a time series has shown to be effective to extract main characteristics of the main series. Hourly patterns, correlation with exogenous factors and seasonal behavior have been identified for the Spanish electricity price.

(iii) The divide-and-conquer strategy of modeling (more distinctive and identifiable) individual components of the electricity price time series, and then combining their forecasts have demonstrated good results in terms of forecasting accuracy.

(iv) The forecasting results show good accuracy on mean absolute error (1.859 EUR (MWh)$^{-1}$, 95% HDI [0.575, 3.924]) and mean absolute scaled error (0.378, 95% HDI [0.091, 0.934]).

(v) The highest error variance is given by the uncertainty of the remainder component forecast. Its average mean absolute error and mean absolute scaled error is 1.867 EUR (MWh)$^{-1}$ and 0.378, respectively.

(vi) The seasonal components of the electricity price have been modeled by statistical time series models. No machine learning model has been found to perform better.

(vii) The AutoML frameworks have helped to assure the performance of the proposed model. H2O AutoML and TPOT get a MASE of 1.0% and 2.4% higher than the metrics achieved by the proposed model, respectively.

(viii) A first decomposition by the STL method followed by a second decomposition by the DWT, EMD and VMD methods has achieved a complete separation of all the underlying patterns of the electricity price time series. The forecasts of the individual components of the second decomposition (DWT, EMD and VMD) have not been selected as individual predictors for the proposed model. Their forecasts are only included by the linear combination methods of the statistical time series models: SA, IM and MED. This means that the underlying components of the first decomposition provides the most information.

(ix) Combining forecasts obtained from different training time windows has shown to reduce the risk associated with selecting an individual time window. The linear combination method that achieves the best accuracy is the interquartile mean (IM) — *i.e.*, the 25% truncated mean. Nevertheless, no statistical mean difference has been found among the methods: IM, MED, BG, AT, SA and CLS. The NG method, the eigenvector-based methods (EIG$_1$, EIG$_2$), OLS and LAD methods obtain poor out-of-sample performance.

(x) Convergence analysis has shown to be a requirement when applying global and local sensitivity methods. For this, a convergence method is proposed and successfully been applied to assess on the stability of the results. The number of Monte Carlo samples chosen to achieve a variability lower than a threshold is 400 for the Shapley values method; 5600 model predictions for the Morris's elementary effects screening method; and 15 permutations for the performance-based method.

(xi) A comparison between three global sensitivity methods (Morris's elementary effects screening method, performance-based method and Shapley values method) has been performed in order to sort the predictors based on their influence on the model outcome. The three methods give a similar order of features, although the Morris's screening method tends to obtain a different order as the sensitivity decreases. The work has demonstrated that the robustness of a global sensitivity analysis is significantly increased by using multiple methods.

(xii) A three-level stack ensemble model has achieved an improvement of 14.9% over the best statistical time series model for predicting the remainder component of the electricity price. The best forecast of Level 0 are given by a simple average of several statistical time series models. The best models of Level 1 and Level 2 is a gradient boosting linear model and a bagged of multiple additive regression splines, respectively.

(xiii) For the given input features, no significantly better results could have been obtained by the ensemble stacked architecture. The model has shown an asymptotically *optimal* learning. This has been concluded by reaching a performance asymptote on the MASE metric.

(xiv) The work has demonstrated the importance of a diverse set of machine learning models for the stack ensemble architecture. Here, diversity is presented by the absolute value of the Pearson correlation coefficient obtained between the residuals of two models.

(xv) An exhaustive model audit has validated the model by diagnosing its residuals. The analysis has shown a linear relation between electricity price observations and forecasts. Residuals can be said to be normally distributed. Their variance presents a homogeneous behavior across the predicted values. The residuals are not partially auto-correlated.

(xvi) The proposed framework has demonstrated to be a useful tool to study the direction and magnitude of change in the predicted outcome due to changes in feature values. In addition, it has successfully drawn summary characteristics about the model such as pointing the relevance of features and explaining local behaviors.

CRediT authorship contribution statement

Sergio Beltrán: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition. **Alain Castro:** Software, Resources. **Ion Irizar:** Project administration, Funding acquisition. **Gorka Naveran:** Project administration. **Imanol Yeregui:** Project administration.

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.

Acknowledgments

This work has been supported by the Basque Government by means of the "Hazitek Program for Research and Business Development", under OptiCogen project (grant agreement Nº ZL-2018/00488). It has also been supported by the Spanish Ministry of Science and Innovation by means of the "Cervera Grants Program for Technological Centres", under MIRAGED project (grant agreement Nº CER-20190001).

Appendix. Features

See Tables 5–8.

Table 8
List of statistical time series models whose forecasts are used as features.

Nº	R-package::function	Description	Ref.
1	fitAR::fitAR	AR(p) fitting	[102]
2	fGarch::garchFit	GARCH fitting	[136]
3	forecast::Arima	ARIMA(p, d, q)(P, D, Q)	[137]
4	forecast::dshw	Double-Seasonal Holt–Winters method	[137]
5	forecast::ets	Exponential smoothing state space model	[137]
6	forecast::(s)naive	(Seasonal) *naïve* model	[137]
7	forecast::nnetar	Feed-forward neural network with one hidden layer	[137]
8	forecast::tbats	TBATS model	[137]
9	forecast::thetaf	Theta method	[137]
10	forecTheta::dotm	Dynamic optimized Theta model	[138]
11	forecTheta::dstm	Dynamic standard Theta model	[138]
12	forecTheta::otm	Optimized Theta model	[138]
13	forecTheta::stheta	Standard Theta method	[138]
14	forecTheta::stm	Standard Theta model	[138]
15	glmnet::glmnet	Generalized linear model with lags	[109]
16	greybox::alm	Advanced linear model with lags	[101]
17	greybox::lmCombine	Linear model with combined lags	[101]
18	greybox::lmDynamic	Linear model with combined lags	[101]
19	greybox::stepwise	Linear model with stepwise selection of lags	[101]
20	MAPA::mapaest	Mutliple aggregation prediction algorithm	[139]
21	nnfor::elm	Extreme learning machine	[140]
22	nnfor::mlp	Multilayer perceptron	[140]
23	PSF::psf	Pattern sequence based forecasting	[141]
24	rugarch::arfimafit	ARFIMA fitting	[142]
25	rugarch::ugarchfit	GARCH fitting	[142]
26	smooth::ces	Complex exponential smoothing	[143]
27	smooth::es	Exponential smoothing in SSOE state-space form	[143]
28	smooth::gum	Generalized exponential smoothing	[143]
29	smooth::msarima	Multiple seasonal state-space ARIMA	[143]
30	smooth::sarima	State-space ARIMA	[143]
31	smooth::sma	Simple moving average in state space form	[143]
32	stats::HoltWinters	Holt–Winters filtering	[144]
33	TSPred::fittestMAS	Moving average smoothing	[145]
34	xgboost::(gblinear)	Regularized linear model with lags as regressors	[62]

References

[1] Weron R. Modeling and forecasting electricity loads and prices: a statistical approach. 1st ed.. England: John Wiley & Sons Ltd; 2006, http://dx.doi.org/10.1002/9781118673362.

[2] Borenstein S, Bushnell J. The us electricity industry after 20 years of restructuring. Annu Rev Econ 2015;7:437–63. http://dx.doi.org/10.1146/annurev-economics-080614-115630.

[3] Baldick R. Wind and energy markets: a case study of texas. IEEE Syst J 2012;6(1):27–34. http://dx.doi.org/10.1109/JSYST.2011.2162798.

[4] Ketterer JC. The impact of wind power generation on the electricity price in germany. Energy Econ 2014;44:270–80. http://dx.doi.org/10.1016/j.eneco.2014.04.003.

[5] Martínez-Anido CB, Brinkman G, Hodge B-M. The impact of wind power on electricity prices. Renew Energy 2016;94:474–87. http://dx.doi.org/10.1016/j.renene.2016.03.053.

[6] Green R, Vasilakos N. Market behaviour with large amounts of intermittent generation. Energy Policy 2010;38(7):3211–20. http://dx.doi.org/10.1016/j.enpol.2009.07.038.

[7] Milstein I, Tishler A. Can price volatility enhance market power? the case of renewable technologies in competitive electricity markets. Resour Energy Econ 2015;41:70–90. http://dx.doi.org/10.1016/j.reseneeco.2015.04.001.

[8] Zareipour H, Canizares CA, Bhattacharya K. Economic impact of electricity market price forecasting errors: a demand-side analysis. IEEE Trans Power Syst 2010;25(1):254–62. http://dx.doi.org/10.1109/TPWRS.2009.2030380.

[9] Hong T. Crystal ball lessons in predictive analytics. Energybiz Mag 2015;12(2):35–7.

[10] Mathaba T, Xia X, Zhang J. Analysing the economic benefit of electricity price forecast in industrial load scheduling. Electr Power Syst Res 2014;116:158–65. http://dx.doi.org/10.1016/j.epsr.2014.05.008.

[11] Ahlert KH, Block C. Assessing the impact of price forecast errors on the economics of distributed storage systems. Koloa, USA; 2010, http://dx.doi.org/10.1109/HICSS.2010.72.

[12] Mohammadi-Ivatloo B, Zareipour H, Ehsan M, Amjady N. Economic impact of price forecasting inaccuracies on self-scheduling of generation companies. Electr Power Syst Res 2011;81(2):617–24. http://dx.doi.org/10.1016/j.epsr.2010.10.022.

[13] Delarue E, Bosch PVanDen, D'haeseleer W. Effect of the accuracy of price forecasting on profit in a price based unit commitment. Electr Power Syst Res 2010;80(10):1306–13. http://dx.doi.org/10.1016/j.epsr.2010.05.001.

[14] Croonenbroeck C, Hüttel S. Quantifying the economic efficiency impact of inaccurate renewable energy price forecasts. Energy 2017;134(1):767–74. http://dx.doi.org/10.1016/j.energy.2017.06.077.

[15] Ugurlu U, Tas O, Kaya A, Oksuz I. The financial effect of the electricity price forecasts' inaccuracy on a hydro-based generation company. Energies 2018;11(8):1–19. http://dx.doi.org/10.3390/en11082093.

[16] Chazarra M, Pérez-Díaz JP, García-González J, Helseth A. Economic effects of forecasting inaccuracies in the automatic frequency restoration service for the day-ahead energy and reserve scheduling of pumped storage plants. Electr Power Syst Res 2019;174:105850. http://dx.doi.org/10.1016/j.epsr.2019.04.028.

[17] Ghoddusi H, Creamer GG, Rafizadeh N. Machine learning in energy economics and finance: a review. Energy Econ 2019;81:709–27. http://dx.doi.org/10.1016/j.eneco.2019.05.006.

[18] Weron R, Ziel F. Electricity price forecasting. In: Routledge handbook of energy economics. 1st ed.. Abingdon, UK: Routledge International Handbooks; 2019.

[19] Weron R. Electricity price forecasting: a review of the state-of-the-art with a look into the future. Int J Forecast 2014;30:1030–81. http://dx.doi.org/10.1016/j.ijforecast.2014.08.008.

[20] Gürtler M, Paulsen T. Forecasting performance of time series models on electricity spot markets: a quasi-meta-analysis. Int J Energy Sect Manag 2018;12(1):1750–6220. http://dx.doi.org/10.1108/IJESM-06-2017-0004.

[21] Jiang L, Hu G. A review on short-term electricity price forecasting techniques for energy markets. In: 15th international conference on control, automation, robotics and vision. Republic of Singapore; 2018.

[22] Niimura T. Forecasting techniques for deregulated electricity market prices – extended survey. In: IEEE PES power systems conference and exposition. Atlanta, USA; 2006, http://dx.doi.org/10.1109/PSCE.2006.296248.

[23] Haghi HV, Tafreshi SMM. Modeling and forecasting of energy prices using non-stationary markov models versus stationary hybrid models including a survey of all methods. In: IEEE canada electrical power conference. Montreal, Canada; 2007, http://dx.doi.org/10.1109/EPC.2007.4520370.

[24] Daneshi H, Daneshi A. Price forecasting in deregulated electricity markets – a bibliographical survey. In: 3rd international conference on electric utility deregulation and restructuring and power technologies. Nanjing; 2008, http://dx.doi.org/10.1109/DRPT.2008.4523487.

[25] Aggarwal SK, Saini LM, Kumar A. Electricity price forecasting in deregulated markets: a review and evaluation. Int J Electr Power Energy Syst 2009;31(1):13–22. http://dx.doi.org/10.1016/j.ijepes.2008.09.003.

[26] Aggarwal SK, Saini LM, Kumar A. Short term price forecasting in deregulated electricity markets. Int J Energy Sect Manag 2009;3(4):333–58. http://dx.doi.org/10.1108/17506220911005731.

[27] Maciejowska K, Weron R. Electricity price forecasting. HSC Research Reports HSC/19/01, Hugo Steinhaus Center, Wroclaw University of Technology.

[28] Waghmare M, Warkad SB. Review of price forecasting techniques in deregulated electricity market. J Interdiscip Res 2017;3(1):391–5.

[29] Nogales FJ, Contreras J, Conejo AJ, Espínola R. Forecasting next-day electricity prices by time series models. IEEE Trans Power Syst 2017;17(2):342–8. http://dx.doi.org/10.1109/TPWRS.2002.1007902.

[30] Knittel CR, Roberts MR. An empirical examination of restructured electricity prices. Energy Econ 2005;27(5):791–817. http://dx.doi.org/10.1016/j.eneco.2004.11.005.

[31] Cruz A, Muñoz A, Zamora JL, Espínola R. He effect of wind generation and weekday on spanish electricity spot price forecasting-. Electr Power Syst Res 2011;81(10):1924–35. http://dx.doi.org/10.1016/j.epsr.2011.06.002.

[32] Lago J, De Ridder F, Vrancx P, De Schutter B. Forecasting day-ahead electricity prices in europe: the importance of considering market integration. Appl Energy 2018;211(1):890–903. http://dx.doi.org/10.1016/j.apenergy.2017.11.098.

[33] Yang Z, Ce L, Lian L. Electricity price forecasting by a hybrid model, combining wavelet transform, arma and kernel-based extreme learning machine methods. Appl Energy 2017;190(15):291–305. http://dx.doi.org/10.1016/j.apenergy.2016.12.130.

[34] Qin Q, Xie K, He H, Li L, Chu X, Wei YM, Wu T. An effective and robust decomposition-ensemble energy price forecasting paradigm with local linear prediction. Energy Econ 2019;83:402–14. http://dx.doi.org/10.1016/j.eneco.2019.07.026.

[35] Dal Molin MH, Frizzo S, Donizetti J, Nied A, Cocco V, dos Santos L. Electricity price forecasting based on self-adaptive decomposition and heterogeneous ensemble learning. Energies 2020;13(19):5190. http://dx.doi.org/10.3390/en13195190.

[36] Lahmiri S. Comparing variational and empirical mode decomposition in forecasting day-ahead energy prices. IEEE Syst J 2017;11(3):1907–10. http://dx.doi.org/10.1109/JSYST.2015.2487339.

[37] Gurnani M, Korke Y, Shah P, Udmale S, Sambhe V, Bhirud S. Forecasting of sales by using fusion of machine learning techniques. In: International conference on data management, analytics and innovation. Pune, India; 2017, http://dx.doi.org/10.1109/ICDMAI.2017.8073492.

[38] Xiong T, Lia, Bao Y. Seasonal forecasting of agricultural commodity price using a hybrid stl and elm method: evidence from the vegetable market in china. Neurocomputing 2018;275(31):2831–44. http://dx.doi.org/10.1016/j.neucom.2017.11.053.

[39] Fulcher BD, Little MA, Jones NS. Highly comparative time-series analysis: the empirical structure of time series and their methods. J R Soc Interface 2013;10:20130048. http://dx.doi.org/10.1098/rsif.2013.0048.

[40] Wang X, Smith K, Hyndman R. Characteristic-based clustering for time series data. Data Min Knowl Discov 2006;13:335. http://dx.doi.org/10.1007/s10618-005-0039-x.

[41] Hyndman R, Wang E, Laptev N. Large-scale unusual time series detection. In: IEEE international conference on data mining workshop. Atlantic City, USA; 2015, http://dx.doi.org/10.1109/ICDMW.2015.104.

[42] Hibon M, Evgeniou T. To combine or not to combine: selecting among forecasts and their combinations. Int J Forecast 2005;21(1):15–24. http://dx.doi.org/10.1016/j.ijforecast.2004.05.002.

[43] Wallis KF. Combining forecasts – forty years later. Appl Financ Econ 2011;21(1-2):33–41. http://dx.doi.org/10.1080/09603107.2011.523179.

[44] Hajirahimi Z, Khashei M. Hybrid structures in time series modeling and forecasting: a review. Eng Appl Artif Intell 2019;86:83–106. http://dx.doi.org/10.1016/j.engappai.2019.08.018.

[45] Wolpert DH. Stacked generalization. Neural Netw 1992;5(2):241–59. http://dx.doi.org/10.1016/S0893-6080(05)80023-1.

[46] Breiman L. Stacked regressions. Mach Learn 1996;24(1):49–64. http://dx.doi.org/10.2202/1544-6115.1309.

[47] van der Laan MJ, Polley EC, Hubbard AE. Super learner. Statist Appl Genet Mol Biol 2007;6(1):1544–6115. http://dx.doi.org/10.2202/1544-6115.1309.

[48] Ren Y, Zhang L, Suganthan PN. Ensemble classification and regression – recent developments, applications and future directions. IEEE Comput Intell Mag 2016;11(1):41–53. http://dx.doi.org/10.1109/MCI.2015.2471235.

[49] Agrawal RK, Muchahary F, Tripathi MM. Ensemble of relevance vector machines and boosted trees for electricity price forecasting. Appl Energy 2019;250(15):540–8. http://dx.doi.org/10.1016/j.apenergy.2019.05.062.

[50] Bhatia K, Mittal R, Varanasi J, Tripathi MM. An ensemble approach for electricity price forecasting in markets with renewable energy resources. Util Policy 2021;70:101185. http://dx.doi.org/10.1016/j.jup.2021.101185.

[51] Zhou ZH. Ensemble methods foundations and algorithms. Boca Raton: CRC Press Taylor & Francis Group; 2012.

[52] Hutter F, Kotthoff L, Vanschoren J. Automated machine learning: methods, systems, challenges. 1st ed.. USA: Springer International Publishing; 2019, http://dx.doi.org/10.1007/978-3-030-05318-5.

[53] Faes L, Wagner SK, Fu DJ, Liu X, Korot E, Ledsam JR, Back T, Chopra R, Pontikos N, Kern C, Moraes G, Schmid MK, Sim D, Balaskas K, Bachmann LM, Denniston AK, Keane PA. Automated deep learning design for medical image classification by health-care professionals with no coding experience: a feasibility study. Lancet Digit Health 2019;1(5):232–42. http://dx.doi.org/10.1016/S2589-7500(19)30108-6.

[54] Wang WM, Wang JW, Barenji AV, Li Z, Tsui E. Modeling of individual customer delivery satisfaction: an automl and multi-agent system approach. Ind Manag Data Syst 2019;119(4):840–66. http://dx.doi.org/10.1108/IMDS-07-2018-0279.

[55] Soares EF, Campos CAV, de Lucena SC. Online travel mode detection method using automated machine learning and feature engineering. Future Gener Comput Syst 2019;101:1201–12. http://dx.doi.org/10.1016/j.future.2019.07.056.

[56] Elshawi R, Maher M, Sakr S. Automated machine learning: state-of-the-art and open challenges. 2019, arXiv:1906.02287 arxiv.org/abs/1906.02287.

[57] He X, Zhao K, Chu X. Automl: a survey of the state-of-the-art. 2019, arXiv:1908.00709 arxiv.org/abs/1908.00709.

[58] Yao Q, Wang M, Chen Y, Dai W, Yi-Qi H, Yu-Feng L, Wei-Wei T, Qiang Y, Yang Y. Taking human out of learning applications: a survey on automated machine learning. 2019, arXiv:1810.13306 arxiv.org/abs/1810.13306.

[59] Zöller MA, Hubertitle MF. Survey on automated machine learning. 2019, arXiv:1904.12054 arxiv.org/abs/1904.12054.

[60] H2Oai. H2o, h2o version 3.26.0.9. 2019, github.com/h2oai/h2o-3.

[61] Le TT, Weixuan F, Moore JH. Scaling tree-based automated machine learning to biomedical big data with a feature set selector. Bioinformatics 2019;btz470. http://dx.doi.org/10.1093/bioinformatics/btz470.

[62] Chen T, Guestrin C. Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. San Francisco, USA; 2016, http://dx.doi.org/10.1145/2939672.2939785.

[63] Nowotarski J, Weron R. Recent advances in electricity price forecasting: a review of probabilistic forecasting. Renew Sustain Energy Rev 2018;81(1):1548–68. http://dx.doi.org/10.1016/j.rser.2017.05.234.

[64] Molnar C. In *Interpretable machine learning: a guide for making black box models explainable*, 1st ed. christophm.github.io/interpretable-ml-book.

[65] Lipton ZC. The mythos of model interpretability. 2017, arXiv:1606.03490 arxiv.org/abs/1606.03490.

[66] Doshi-Velez F, Kim B. Towards a rigorous science of interpretable machine learning. 2017, arXiv:1702.08608v2 arxiv.org/abs/1702.08608v2.

[67] Murdoch WJ, Singh C, Kumbier K, Abbasi-Asl R, Yu B. Definitions, methods, and applications in interpretable machine learning. Proc Natl Acad Sci 2019;116(44):22071–80. http://dx.doi.org/10.1073/pnas.1900654116.

[68] Hall P, Gill N, Schmidt N. Proposed guidelines for the responsible use of explainable machine learning. 2019, arXiv:1906.03533 arxiv.org/abs/1906.03533.

[69] Kahn J. Artificial intelligence has some explaining to do. 2019, bloomberg.com(Accessed: December 2019).

[70] Došilović FK, Brčić M, Hlupić N. Explainable artificial intelligence: a survey. In: 41st international convention on information and communication technology, electronics and microelectronics. Opatija, Croatia; 2018, http://dx.doi.org/10.23919/MIPRO.2018.8400040.

[71] Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D. A survey of methods for explaining black box models. ACM Comput Surv 2019;51(5):93. http://dx.doi.org/10.1145/3236009.

[72] Vaughan J, Sudjianto A, Brahimi E, Chen J, Nair VN. Explainable neural networks based on additive index models. 2018, arXiv:1806.01933 arxiv.org/abs/1806.01933.

[73] Lou Y, Caruana R, Gehrke J, Hooker G. Accurate intelligible models with pairwise interactions. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining. Chicago, USA; 2013, p. 623–31. http://dx.doi.org/10.1145/2487575.2487579.

[74] Caruana R, Lou Y, Gehrke J, Koch P, Sturm M, Elhadad N. Intelligible models for healthcare: predicting pneumonia risk and hospital 30-day readmission. In: Proceedings of the 21st ACM SIGKDD international conference on knowledge discovery and data mining. Sydney, Australia; 2015, p. 1721–30. http://dx.doi.org/10.1145/2783258.2788613.

[75] Scalable bayesian rule lists. 2017, arXiv:1602.08610 arxiv.org/abs/1602.08610.

[76] Ustun B, Tracà S, Rudin C. Supersparse linear integer models for interpretable classification. 2014, arXiv:1306.6677. arxiv.org/abs/1306.6677.

[77] Ribeiro MT, Singh S, Guestrin C. Model-agnostic interpretability of machine learning. 2016, arXiv:1606.05386 arxiv.org/abs/1606.05386.

[78] REE (Red Eléctrica Española). Informe del Sistema Eléctrico Español. (Spanish Electricity System Report 2019), 2019, www.ree.es.

[79] Schumaker LI. On shape preserving quadratic spline interpolation. SIAM J Numer Anal 1983;20(4):854–64. http://dx.doi.org/10.1137/0720057.

[80] Cleveland RB, Cleveland WS, McRae JE, Terpenning IJ. Stl: a seasonal-trend decomposition procedure based on loess. J Off Statist 1990;6(1):3–33.

[81] Hyndman RJ, Athanasopoulos G. Forecasting: principles and practice. 2nd ed.. Otexts; 2018, otexts.com/fpp2.

[82] Li F-F, Wang Z-Y, Zhao X, En Xie, Qiu J. Decomposition-ann methods for long-term discharge prediction based on fisher's ordered clustering with mesa. Water Resour Manag 2019;33:3095–110. http://dx.doi.org/10.1007/s11269-019-02295-8.

[83] Hyndman RJ, Koehler AB. Another look at measures of forecast accuracy. Int J Forecast 2006;22(4):679–88. http://dx.doi.org/10.1016/j.ijforecast.2006.03.001.

[84] Franses PH. A note on the mean absolute scaled error. Int J Forecast 2016;32:20–2. http://dx.doi.org/10.1016/j.ijforecast.2015.03.008.

[85] Bates JM, Granger CWJ. The combination of forecasts. J Oper Res Soc 1969;20(4):451–68. http://dx.doi.org/10.1057/jors.1969.103.

[86] Newbold P, Granger CWJ. Experience with forecasting univariate time series and the combination of forecasts. J R Statist Soc Ser A (Gen) 1974;137(2):131–65. http://dx.doi.org/10.2307/2344546.

[87] Aiolfi M. Timmermann A. Persistence in forecasting performance and conditional combination strategies. J Econometrics 2006;135(1):31–53. http://dx.doi.org/10.1016/j.jeconom.2005.07.015.

[88] Nowotarski J, Raviv E, Trück S, Weron R. An empirical comparison of alternative schemes for combining electricity spot price forecasts. Energy Econ 2014;46:395–412. http://dx.doi.org/10.1016/j.eneco.2014.07.014.

[89] Hsiao C, Wan SK. Is there an optimal forecast combination?. J Econometrics 2014;178(2):294–309. http://dx.doi.org/10.1016/j.jeconom.2013.11.003.

[90] Guyon I, Weston J, Barnhill S, Vapnik V. Gene selection for cancer classification using support vector machines. Mach Learn 2002;46(1–3):389–422. http://dx.doi.org/10.1023/A:1012487302797.

[91] Karatzoglou A, Smola A, Hornik K, Australia National ICT, Maniscalco MA, Teo CH. kernlab: kernel-based machine learning lab. R package version 0.9-29. cran.r-project.org/package=kernlab.

[92] Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F, Chang CC, Lin CC. e1071: misc functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7-3. cran.r-project.org/package=e1071.

[93] Fisher A, Rudin C, Dominici F. All models are wrong, but many are useful: learning a variable,s importance by studying an entire class of prediction models simultaneously. J Mach Learn Res 2019;20(177):1–81, jmlr.org/papers/v20/18-760.html.

[94] Janczura J, Trück S, Weron R, Wolff RC. Identifying spikes and seasonal components in electricity spot price data: a guide to robust modeling. Energy Econ 2013;38:96–110. http://dx.doi.org/10.1016/j.eneco.2013.03.013.

[95] Afanasyeva DO, Fedorova EA. On the impact of outlier filtering on the electricity price forecasting accuracy. Appl Energy 2019;236:196–210. http://dx.doi.org/10.1016/j.apenergy.2018.11.076.

[96] Fraunholz C, Kraft E, Keles D, Fichtner W. Advanced price forecasting in agent-based electricity market simulation. Appl Energy 2021;290(15):116688. http://dx.doi.org/10.1016/j.apenergy.2021.116688.

[97] Chen C, Liu LM. Joint estimation of model parameters and outlier effects in time series. J Amer Statist Assoc 1993;88(421):284–97. http://dx.doi.org/10.2307/2290724.

[98] Torres ME, Colominas MA, Schlotthauer G, Flandrin P. A complete ensemble empirical mode decomposition with adaptive noise. Acoust Speech Signal Process (ICASSP) 2011;4144–7. http://dx.doi.org/10.1109/ICASSP.2011.5947265.

[99] Percival DB, Walden AT. Wavelet methods for time series analysis. Cambridge University Press; 2000, http://dx.doi.org/10.1017/CBO9780511841040.

[100] Dragomiretskiy K, Zosso D. Variational mode decomposition. IEEE Trans Signal Process 2014;62(3):531–44. http://dx.doi.org/10.1109/TSP.2013.2288675.

[101] Svetunkov I. greybox: toolbox for model building and forecasting. R package version 0.5.7. cran.r-project.org/package=greybox.

[102] McLeod AI, Zhang Y. FitAR: Subset AR Model Fitting. R package version 1.94. cran.r-project.org/package=FitAR.

[103] Kapelner A, Bleich J. bartMachine: Bayesian additive regression trees. R package version 1.2.4.2. cran.r-project.org/package=bartMachine.

[104] Gelman A, Su YS, Yajima M, Hill J, Pittau MG, Kerman J, Zheng T, Dorie V. arm: data analysis using regression and multilevel/hierarchical models. R package version 1.10-1. cran.r-project.org/package=arm.

[105] Wang Z, T. Hothorn. bst: Gradient Boosting. R package version 0.3-17. cran.r-project.org/package=bst.

[106] Schliep K, Hechenbichler K, Lizee A. kknn: weighted k-nearest neighbors. R package version 1.3.1. cran.r-project.org/package=kknn.

[107] Lumley T, Miller A. leaps: regression subset selection. R package version 3.1. cran.r-project.org/package=leaps.

[108] Milborrow S. earth: multivariate adaptive regression splines. R package version 5.1.2. cran.r-project.org/package=earth.

[109] Friedman J, Hastie T, Tibshirani R, Narasimhan B, Simon N, Qian J. glmnet: lasso and elastic-net regularized generalized linear models. R package version 3.0-2. cran.r-project.org/package=glmnet.

[110] Mevik BH, Wehrens R, Liland KH, Hiemstra P. pls: partial least squares and principal component regression. R package version 2.7-2. cran.r-project.org/package=pls.

[111] Wright MN, Wager S, Probst P. ranger: a fast implementation of random forests. R package version 0.12.1. cran.r-project.org/package=ranger.

[112] Kuhn M, Weston S, Keefer C, Coulter N, Quinlan R. Cubist: rule- and instance-based regression modeling. R package version 0.2.3. cran.r-project.org/package=Cubist.

[113] Breiman L. Bagging predictors. Mach Learn 1996;24:123–40. http://dx.doi.org/10.1023/A:1018054314350.

[114] Okabe M, Ito K. Color universal design (cud): how to make figures and presentations that are friendly to colorblind people. 2008, jfly.iam.u-tokyo.ac.jp/color.

[115] Garnier S, Ross N, Rudis B, Sciaini M, Scherer C. Viridis: default color maps from matplotlib. R package version 0.5.1. cran.r-project.org/package=viridis.

[116] F. Crameri. Geodynamic diagnostics, scientific visualisation and staglab 3.0. Geosci Model Dev 2018;(11):2541–62. http://dx.doi.org/10.5194/gmd-11-2541-2018.

[117] IBM Plex® Sans version 4.0.2. github.com/IBM/plex.

[118] Hyndman RJ. Computing and graphing highest density regions. Amer Statist 1996;50(2):120–6. http://dx.doi.org/10.2307/2684423.

[119] Roesch A, Schmidbauer H. WaveletComp: computational wavelet analysis. R package version 1.1. cran.r-project.org/package=WaveletComp.

[120] Razali NM, Wah YB. Power comparisons of shapiro–wilk, kolmogorov–smirnov, lilliefors and anderson–darling tests. J Statist Model Anal 2011;2(1):21–33.

[121] Hothorn T, Zeileis A, Farebrother RW, Cummins C, Millo G, Mitchell D. lmtest: testing linear regression models. R package version 0.9–37. cran.r-project.org/package=lmtest.

[122] Shapley LS. A value for n-person games. In Kuhn HW, Tucker KW. In *Contributions to the theory of games ii*, Princeton, New Jersey, Princeton University Press, p. 307–317.

[123] Lipovetsky S, Conklin M. Analysis of regression in game theory approach. Appl Stoch Models Bus Ind 2001;17(4):319–30. http://dx.doi.org/10.1002/asmb.446.

[124] Lundberg SM, Lee SI. An unexpected unity among methods for interpreting model predictions. 2016, arXiv:1611.07478 arxiv.org/abs/1611.07478.

[125] Štrumbelj E, Kononenko I. Explaining prediction models and individual predictions with feature contributions. Knowl Inf Syst 2014;(41):647–65. http://dx.doi.org/10.1007/s10115-013-0679-x.

[126] Morris MD. Factorial sampling plans for preliminary computational experiments. Technometrics 1991;33(2):161–74. http://dx.doi.org/10.2307/1269043.

[127] Saltelli A, Tarantola S, Campolongo F, Ratto M. Sensitivity analysis in practice: a guide to assessing scientific models. John Wiley & Sons Ltd; 2004, http://dx.doi.org/10.1002/0470870958.

[128] Friedman JH, Popescu BE. Predictive learning via rule ensembles. Ann Appl Statist 2008;2(3):916–54, www.jstor.org/stable/30245114.

[129] Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Statist 2001;29(5):1189–232, www.jstor.org/stable/2699986.

[130] Goldstein A, Kapelner A, Bleich J, Pitkin E. Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. J Comput Graph Statist 2015;24(1):44–65. http://dx.doi.org/10.1080/10618600.2014.907095.

[131] Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and regression trees. Wadsworth statistics/probability, Chapman and Hall/CRC; 1984.

[132] R Core Team. R: a language and environment for statistical computing. www.r-project.org.

[133] Hyndman RJ, Kang Y, Montero-Manso P, Talagala T, Wang E, Yang Y, O'Hara-Wild M, Taieb SB, Hanqing C, Lake DK, Laptev N, Moorman JR. tsfeatures: time series feature extraction. R package version 1.0.1. cran.r-project.org/package=tsfeatures.

[134] Engle R. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. Econometrica 1982;50(4):987–1007. http://dx.doi.org/10.2307/1912773.

[135] Fulcher BD, Jones NS. Hctsa: a computational framework for automated time-series phenotyping using massive feature extraction. Cell Syst 2017;5:527. http://dx.doi.org/10.1016/j.cels.2017.10.001.

[136] Wuertz D, Setz T, Chalabi Y, Boudt C, Chausse P, Miklovac M. fGarch: rmetrics - autoregressive conditional heteroskedastic modelling. R package version 3042.83.2. cran.r-project.org/package=fGarch.

[137] Hyndman RJ, Athanasopoulos G, Bergmeir C, Caceres G, Chhay L, O'Hara-Wild M, Petropoulos F, Razbash S, Wang E, Yasmeen F. forecast: forecasting functions for time series and linear models. R package version 8.10. cran.r-project.org/package=forecast.

[138] Fiorucci JA, Louzada F, Yiqi B. forecTheta: forecasting time series by Theta models. R package version 2.2. cran.r-project.org/package=forecTheta.

[139] Kourentzes N, Petropoulos F. MAPA: multiple aggregation prediction algorithm. R package version 2.0.4. cran.r-project.org/package=MAPA.

[140] Kourentzes N. nnfor: time series forecasting with neural networks. R package version 0.9.6. cran.r-project.org/package=nnfor.

[141] Bokde N, Asencio-Cortes G, Martínez-Álvarez F. PSF: forecasting of univariate time series using the Pattern Sequence-based Forecasting (PSF) algorithm. R package version 0.4. cran.r-project.org/package=PSF.

[142] Ghalanos A, Kley T. rugarch: univariate GARCH models. cran.r-project.org/package=rugarch.

[143] Svetunkov I. smooth: forecasting using state space models. R package version 2.5.4. cran.r-project.org/package=smooth.

[144] R Core Team and contributors worldwide. The R stats package. R package version 3.6.2. stat.ethz.ch/R-manual/R-devel/library/stats/html/00Index.html.

[145] Salles RP, Ogasawara E. TSPred: functions for baseline-based time series prediction. R package version 4.0. cran.r-project.org/package=TSPred.