Forecasting day-ahead electricity prices in Europe
The importance of considering market integration

Lago Garcia, Jesus; De Ridder, Fjo; Vrancx, Peter; De Schutter, Bart

DOI
10.1016/j.apenergy.2017.11.098

Publication date
2018

Document Version
Final published version

Published in
Applied Energy

Applied Energy 211 (2018) 890–903

Contents lists available at ScienceDirect

Applied Energy

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

Forecasting day-ahead electricity prices in Europe: The importance of considering market integration

Jesus Lago[a,b,*], Fjo De Ridder[b], Peter Vrancx[c], Bart De Schutter[a]

[a] Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628CD Delft, The Netherlands
[b] Energy Technology, VITO-Energyville, ThorPark, 3600 Genk, Belgium
[c] AI Lab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium

HIGHLIGHTS

- Models to include market integration in electricity price forecasting are proposed.
- The forecasters lead to accuracy improvements that are statistically significant.
- Deep neural networks are used as based models of the larger modeling framework.
- A forecasters that predicts prices in various markets leads to the best results.
- A novel feature selection algorithm based on functional ANOVA is proposed.

ARTICLE INFO

Keywords:
Electricity price forecasting
Electricity market integration
Deep neural networks
Functional ANOVA
Bayesian optimization

ABSTRACT

Motivated by the increasing integration among electricity markets, in this paper we propose two different methods to incorporate market integration in electricity price forecasting and to improve the predictive performance. First, we propose a deep neural network that considers features from connected markets to improve the predictive accuracy in a local market. To measure the importance of these features, we propose a novel feature selection algorithm that, by using Bayesian optimization and functional analysis of variance, evaluates the effect of the features on the algorithm performance. In addition, using market integration, we propose a second model that, by simultaneously predicting prices from two markets, improves the forecasting accuracy even further. As a case study, we consider the electricity market in Belgium and the improvements in forecasting accuracy when using various French electricity features. We show that the two proposed models lead to improvements that are statistically significant. Particularly, due to market integration, the predictive accuracy is improved from 15.7% to 12.5% sMAPE (symmetric mean absolute percentage error). In addition, we show that the proposed feature selection algorithm is able to perform a correct assessment, i.e. to discard the irrelevant features.

1. Introduction

As a result of the liberalization and deregulation of the electricity markets in the last two decades, the dynamics of electricity trade have been completely reshaped. In particular, electricity has become a commodity that displays a set of characteristics that are uncommon to other markets: a constant balance between production and consumption, load and generation that are influenced by external weather conditions, and dependence of the consumption on the hour of the day, day of the week, and time of the year [1]. Due to these facts, the dynamics of electricity prices exhibit behavior unseen in other markets, e.g. sudden and unexpected price peaks or seasonality of prices at three different levels (daily, weekly, and yearly) [1].

As a result of this unique behavior, electricity markets have become a central point of research in the energy sector and accurate electricity price forecasting has emerged as one of the biggest challenges faced by the different market entities. The usual motivation behind these efforts is a purely economic one: as forecasting accuracy increases, the negative economic effects of price uncertainty are mitigated and the market players make an economic profit. In addition, another important fact to consider is that electricity markets are established to keep the grid stable. In particular, as prices become more volatile, the balance of the grid is compromised, strategic reserves may have to be used, and the risk of a blackout increases. Therefore, by accurately forecasting

* Corresponding author at: Energy Technology-VITO, Energyville, Thorpark, 3600 Genk, Belgium.
E-mail address: j.lagogarcia@tudelft.nl (J. Lago).

https://doi.org/10.1016/j.apenergy.2017.11.098
Received 19 July 2017; Received in revised form 24 November 2017; Accepted 26 November 2017
Available online 01 December 2017

electricity prices, not only economic profits can be made, but also the system stability is improved.

Due to the above motivations, electricity price forecasting has been continuously developed and improved for the last decades, and as a result, the literature comprises a large variety of distinctive approaches, e.g. see the literature review [1]. Nevertheless, to the best of our knowledge, a topic that has been not yet addressed is the influence of neighboring and connected markets, i.e. market integration, on the forecast accuracy. In particular, as different areas in the world, e.g. the European Union [2], are enforcing a larger level of integration across national electricity markets, it is sensible to assume that neighboring markets might play a role in the forecasting efficiency. To address this scientific gap, this paper proposes a modeling framework that is able to improve predictive accuracy by exploiting the relations across electricity markets. In particular, by modeling market integration in two different ways, the proposed framework is shown to obtain statistically significant improvements.

The paper is organized as follows: Section 2 starts by presenting the literature review, motivation, and contributions. Next, Sections 3 and 4 respectively describe the methods and data that are used in the research. Then, Section 5 defines the proposed modeling framework. Next, Section 6 derives a novel approach for feature selection and uses it to select the optimal features in the case study. Finally, Section 7 evaluates the proposed modeling framework by means of predictive accuracy, and Section 8 summarizes and concludes the paper.

2. Literature survey and contributions

In this section, we present the literature review of three topics that are relevant for the research: electricity price forecasting, market integration, and feature selection. Based on that, we motivate our work and explain our contributions.

2.1. Electricity price forecasting

The price forecasting literature is typically divided into five areas: (1) multi-agent or game theory models simulating the operation of market agents, (2) fundamental methods employing physical and economic factors, (3) reduced-form models using statistical properties of electricity trade for risk and derivatives evaluation, (4) statistical models comprising time series and econometric models, and (5) artificial intelligence methods [1]. For forecasting day-ahead prices, or in general any other type of electricity spot prices, statistical and artificial intelligence methods have showed to yield the best results [1]. As a result, they are the main focus of this review.

Typical statistical methods are: AR and ARX models [3], ARIMA models [4,5], dynamic regression [6], transfer functions [6], double seasonal Holtz-Winter model [7], TARX model [8], semi/non-parametric models [3], or GARCH-based models [9]. In addition, within the same class of methods, different hybrid models have been also applied, e.g. wavelet-based models [5,10,11].

Statistical models are usually linear forecasters, and as such, they are successful in the areas where the frequency of the data is low, e.g. for weekly patterns. However, for hourly values, the nonlinear behavior of the data might be too complicated to predict [12]. As a result, motivated by the need for forecasters that are able to predict the nonlinear behavior of hourly prices, several artificial intelligence methods have been proposed. Among these methods, artificial neural networks [13–16], support vector regressors [17], radial basis function networks [18], and fuzzy networks [19] are among the most commonly used. A recent study [20] showed that *Deep Neural Networks (DNNs)* can also be a successful alternative.

The results comparing the accuracy of the mentioned models have however produced unclear conclusions [14]. In general, the effectiveness of each model seems to depend on the market under study and on the period considered.

2.2. Market integration

In the last decades, the EU has passed several laws trying to achieve a single and integrated European electricity market [2,21]. At the moment, while a single market is far from existing, there is evidence suggesting that the level of integration across the different regional markets has been increasing over time [22]. In particular, evidence suggests that in the case of Belgium and France, the spot prices share strong common dynamics [23].

While some researchers have evaluated the level of integration of the European markets [22–24], and others have proposed statistical models to evaluate the probability of spike transmissions across EU markets [25], the literature regarding market integration to improve forecasting accuracy is rather scarce. To the best of our knowledge, only two other works have taken into account some sort of market integration, namely [26,27].

In particular, [26] analyzes the effect of using the day-ahead prices of the *Energy Exchange Austria (EXAA)* on a given day to forecast the prices of other European markets on the same day. Using the fact that for the EXAA market the clearing prices are released before the closure of other European markets, [26] models the price dynamics of several European markets and considers the EXAA prices of the same day as part of these models. It is shown that, for certain European markets, using the available prices from the EXAA improves the forecasting accuracy in a statistically significant manner.

Similarly, [27] considers external price forecasts from other European markets as exogenous inputs of an artificial neural network to predict Italian day-ahead prices. [27] shows that using the given forecasts the accuracy of their network can be improved from 19.08% to 18.40% *mean absolute percentage error (MAPE)*.

2.3. Feature selection

Feature selection is defined as the process to select, for a given model, the subset of important and relevant input variables, i.e. features. Typically, three families of methods to perform feature selection exist: *filter*, *wrapper*, and *embedded methods* [28]. Filter methods apply some statistical measure to assess the importance of features [29]. Their main disadvantage is that, as the specific model performance is not evaluated and the relations between features are not considered, they may select redundant information or avoid selecting some important features. Their main advantage is that, as a model does not have to be estimated, they are very fast. By contrast, wrapper methods perform a search across several feature sets, evaluating the performance of a given set by first estimating the prediction model and then using the predictive accuracy of the model as the performance measure of the set [29]. Their main advantage is that they consider a more realistic evaluation of the performance and interrelations of the features; their drawback is a long computation time. Finally, embedded methods, e.g. regularization [30, Chapter 7], learn the feature selection at the same time the model is estimated. Their advantage is that, while being less computationally expensive than wrapper methods, they still consider the underlying model. However, as a drawback, they are specific to a learning algorithm, and thus, they cannot always be applied.

Approaches for feature selection in the electricity price forecasting literature vary according to the prediction model used. For time series methods using only prices, e.g. ARIMA, autocorrelation plots [10] or the Akaike information criterion [31] have been commonly used. In the case of forecasters with explanatory variables, e.g. neural networks, most researchers have used trial and error or filter methods based on linear analysis techniques: statistical sensitivity analysis [7,13], correlation analysis [32], or principal component analysis [33]. Since prices display nonlinear dynamics, the mentioned techniques might be limited [34]; to address this, nonlinear filter methods such as the relief algorithm [35] or techniques based on mutual information [34,36,37] have been proposed. More recently, a hybrid nonlinear filter-wrapper

method, which uses mutual information and information content as a first filter step and a real-coded genetic algorithm as a second wrapper step, has been proposed [38].

2.4. Motivation and contributions

While the effects of market integration can dramatically modify the dynamics of electricity prices, there is a lack of a general modeling framework that could model this effect and analyze its impact on the electricity market. To address this gap, in this paper we provide general models to identify these relations and a technique to quantify the importance of market integration. As we will show, understanding these relations is key to improve the accuracy of forecasting models, and thus, to obtain energy systems that are economically more efficient.

The two available papers on market integration in price forecasting, [26,27], are both limited to the case where the day-ahead prices of neighboring markets are known in advance. While these papers provide a first modeling approach for market integration, the methodologies are very specific and can only be applied in limited situations. In particular, most European electricity markets release their day-ahead prices at the same time, and thus, the prices of neighboring markets cannot be obtained in advance. The only exception to this rule is the EXAA market, which was the object of study of [26]. In addition to this limitation, neither [26] nor [27] analyzed the relevance of market integration.

In contrast to [26,27], we propose a general modeling framework that is able to model and analyze market integration for any given market. In particular, we propose a modeling framework based on DNNs that considers market integration features that are available beforehand in all European markets. Using past prices and publicly available load/generation forecasts in neighboring markets, we propose a first forecaster that models market integration effects on price dynamics. Next, we propose a second forecaster that further generalizes market integration: besides modeling market integration using input features, the second forecaster also includes the effect in the output space. By simultaneously predicting prices in multiple markets, the proposed forecaster is able to improve the predictive accuracy.

Finally, we also contribute to the field of feature selection algorithms. More specifically, while the feature selection methods for electricity price forecasting proposed in the literature provide good and fast algorithms, they suffer from three main drawbacks: (1) They all [7,10,13,32–36,38] perform a filter step where the model performance is not directly considered; therefore, the resulting selected features might be redundant or incomplete. (2) In the case of the algorithms for nonlinear models [34–36,38], the inputs have to be transformed to lower-dimensional spaces; as a result, feature information might be lost. (3) While they provide a selection of features, none of these methods computes the relative importance of each feature.

To address these issues, we propose a wrapper selection algorithm based on functional ANOVA that directly selects features using nonlinear models and without any feature transformation. While the proposed approach is computationally more expensive than previously proposed methods, it can perform a more accurate feature selection as it avoids transformations, selects the features based on the original model, and computes the individual performance of each feature.

3. Preliminaries

In this section we introduce the theoretical concepts and algorithms that are used and/or modified later on in the paper.

3.1. Day-ahead forecasting

The day-ahead electricity market is a type of power exchange widely used in several regions of the world. In its most general format, producers and consumers have to submit bids for the 24 hours of day d before some deadline on day $d-1$ (in most European markets, this deadline occurs at 11:00 am or 12:00 am). Except for some markets, these bids are typically defined per hour, i.e. every market player has to submit 24 bids.

After the deadline, the market operator takes into account all the bids and computes the market clearing price for each of the 24 hours. Then, consumer/producer bids larger/lower or equal than the market clearing prices are approved, and a contract is established.

A useful forecaster of the day ahead market should thus be able to predict the set of 24 market clearing prices of day d based on the information available before the deadline of day $d-1$.

3.2. Deep learning and DNNs

During the last decade, the field of neural networks has gone trough some major innovations that have lead to what nowadays is known as deep learning [30]. Specifically, the term deep refers to the fact that, thanks to the novel developments of recent years, we can now train different neural network configurations whose depth is not just limited to a single hidden layer (as in the traditional multilayer perceptron), and which have systemically showed better generalization capabilities [30].

While there are different DNN architectures, e.g. convolutional networks or recurrent networks, in this paper we consider a standard DNN, i.e. a multilayer perceptron with more than a single hidden layer.

3.2.1. Representation

Defining by $\mathbf{X} = [x_1,\dots,x_n]^\top \in \mathbb{R}^n$ the input of the network, by $\mathbf{Y} = [y_1,y_2,\dots,y_m]^\top \in \mathbb{R}^m$ the output of the network, by n_k the number of neurons of the hidden layer, and by $\mathbf{z}_k = [z_{k1},\dots,z_{kn_k}]^\top$ the state vector in the hidden layer, a general DNN with two hidden layers can be represented as in Fig. 1.

In this representation, the parameters of the model are represented by the set of weights $\mathbf{W}$ that establish the mapping connections between the different neurons of the network [30].

3.2.2. Training

The process of estimating the model weights $\mathbf{W}$ is usually called training. In particular, given a training set $\mathscr{S}_\mathscr{T} = \{(\mathbf{X}_k,\mathbf{Y}_k)\}_{k=1}^N$ with N data points, the network training is done by solving a general optimization problem with the following structure:

$$\underset{\mathbf{W}}{\text{minimize}} \quad \sum_{k=1}^{N} g_k(\mathbf{Y}_k, F(\mathbf{X}_k,\mathbf{W})), \tag{1}$$

where F: $\mathbb{R}^n \to \mathbb{R}^m$ is the neural network map, and g_k is the problem-specific cost function, e.g. the Euclidean norm or the average cross-entropy. Traditional methods to solve (1) include *gradient descent* or the *Levenberg-Marquardt* algorithm [1]. However, while these methods work well for small sized-networks, they display computational and

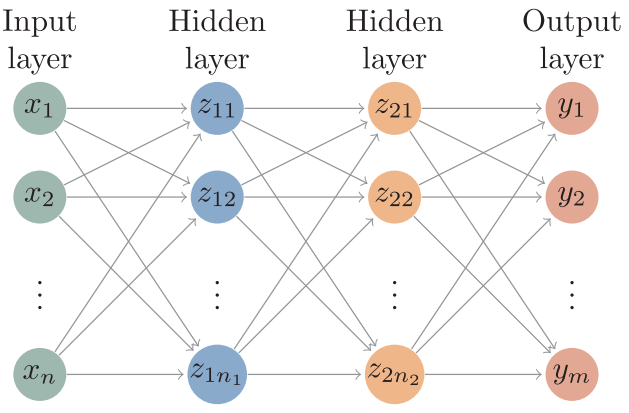


Fig. 1. Example of a DNN.

scalability issues for DNNs. In particular, better alternatives for DNNs are the *stochastic gradient descent* and all its variants [39].

It is important to note that (1) is an approximation of the real problem we wish to minimize. Particularly, in an ideal situation, we would minimize the cost function w.r.t. to the underlying data distribution; however, as the distribution is unknown, the problem has to be approximated by minimizing the cost function over the finite training set. This is especially relevant for neural networks, where a model could be overfitted and have a good performance in the training set, but perform badly in the test set, i.e. a set with a different data distribution. To avoid this situation, the network is usually trained in combination with regularization techniques, e.g. early stopping, and using out-of-sample data to evaluate the performance [30].

3.2.3. Network hyperparameters

In addition to the weights, the network has several parameters that need to be selected before the training process. Typical parameters include the number of neurons of the hidden layers, the number of hidden layers, the type of activation functions, or the learning rate of the stochastic gradient descent method. To distinguish them from the main parameters, i.e. the network weights, they are referred to as the network hyperparameters.

3.3. Hyperparameter selection

In order to perform the selection of model hyperparameters, papers in the field of electricity price forecasting have traditionally defined a number of configurations and chosen the one with the best performance [7,14,27,32,34]. Another approach, yet less usual, has been the use of evolutionary optimization algorithms in order to select the best network configuration [40]. However, while these approaches might work under some conditions, they have some flaws. In particular, while the first method implements fast decision-making, it does not provide an optimal selection of hyperparameters. Similarly, while the second method optimizes the selection, it evaluates a very large number of points in the hyperparameter space. As a result, if the function to be evaluated is costly, e.g. when training a DNN, the second method requires a large computation time.

An alternative to tackle these issues is *Bayesian optimization* [41], a family of algorithms for optimizing black-box functions that require a lower number of function evaluations than evolutionary optimization techniques. In particular, their working principle is to sequentially evaluate new samples in the function space, drawing new samples by using the information obtained in the previously explored samples as a prior belief. Based on that, they reduce the number of evaluated sample points and lead to a more efficient optimization.

3.3.1. Hyperparameter optimization

We consider a Bayesian optimization algorithm that has been widely used in the machine learning community. In particular, we use the *Tree-Structured Parzen Estimator (TPE)* [42], an optimization algorithm within the family of *sequential model-based optimization* methods [43]. The basic principle of a sequential model-based optimization algorithm is to optimize a black-box function, e.g. the performance of a neural network as a function of the hyperparameters, by iteratively estimating an approximation of the function and exploring the function space using the local minima of the approximation. At any given iteration i, the algorithm evaluates the black-box function at a new point θ_i. Next, it estimates an approximation $\mathscr{M}$ of the black-box function by fitting the previously sampled points to the obtained function evaluations. Then, it selects the next sample point θ_{i+1} by numerically optimizing $\mathscr{M}$ and starts the next iteration. Finally, after a maximum number of iterations T have been performed, the algorithm selects the best configuration. Algorithm 1 represents an example of a sequential model-based optimization algorithm for hyperparameter selection.

Algorithm 1. Hyperparameter Optimization

```
procedure SMBO(T,θ_0)
  θ_i ← θ_0
  ℋ ← ∅
  for i = 1,...,T do
    p_i ← TrainNetwork(θ_i)
    ℋ ← ℋ ∪ {(p_i,θ_i)}
    if i < T then
      ℳ_i(θ) ← EstimateModel(ℋ)
      θ_i ← argmax_θ ℳ_i(θ)
    end if
  end for
  θ* ← BestHyperparameters(ℋ)
  return θ*
end procedure
```

3.3.2. Hyperparameter analysis

An optional step after hyperparameter optimization is to perform an analysis of the hyperparameter importance. In particular, while the optimal hyperparameter configuration has been already obtained, it is unknown how much each hyperparameter contributes to the overall performance. Investigating this is specially relevant in order to avoid unnecessary model complexities; e.g. while the optimal number of neurons might be large, reducing the number of neurons might barely affect the performance.

Functional ANOVA. An approach for carrying on such an analysis is proposed in [44], where a novel method based on random forests and functional ANOVA is introduced. In particular, [44] considers the generic case of having z hyperparameters with domains $\Theta_1,...,\Theta_z$, and defines the following concepts:

- Hyperparameter set $Z = \{1,...,z\}$.
- Hyperparameter space Θ: $\Theta_1 \times ... \times \Theta_z$.
- Hyperparameter instantiation $\theta = [\theta_1,...,\theta_z]^\top$.
- Hyperparameter subset $U = \{u_1,...,u_q\} \subseteq Z$ and associated partial hyperparameter instantiation $\theta_U = [\theta_{u_1},...,\theta_{u_q}]^\top$.

Then, given a set $\mathscr{H} = \{(\theta_k,p_k)\}_{k=1}^T$ of hyperparameter realizations, the proposed method fits a random forest model $\mathscr{M}_{\mathrm{RF}}(\theta)$ to build a predictor of the performance p as a function of the hyperparameter vector θ.

Then, using $\mathscr{M}_{\mathrm{RF}}$, the method defines a *marginal performance predictor* $\hat{a}(\theta_U)$ as a forecaster of the performance of any partial hyperparameter instantiation θ_U. In particular, given a subset $U \subseteq Z$,$\hat{a}(\theta_U)$ provides an estimation of the average performance across the hyperparameter space $Z \backslash U$ when the hyperparameters of U are fixed at θ_U.

Finally, using the marginal performance predictor $\hat{a}(\theta_U)$, the algorithm carries out a functional ANOVA analysis to estimate the importance of each hyperparameter. Particularly, defining the total variance across the performance by $\mathbb{V}$, the algorithm partitions $\mathbb{V}$ as a sum of individual variance contributions of subsets $U \subseteq Z$ to $\mathbb{V}$:

$$\mathbb{V} = \sum_{U \subseteq Z} \mathbb{V}_U, \tag{2}$$

where $\mathbb{V}_U$ is the contribution of subset U to the total variance. Then, the importance $\mathbb{F}_U$ of each subset U is computed based on the subset contribution to the total performance variance:

$$\mathbb{F}_U = \frac{\mathbb{V}_U}{\mathbb{V}}. \tag{3}$$

For the particular case of the hyperparameter importance, the algorithm just evaluates $\mathbb{F}_U$ for each subset $U = \{i\}$ composed of a single hyperparameter. As in [44], we refer to the variance contributions $\mathbb{F}_U$ of single hyperparameters as *main effects* and to the rest as *interaction effects*.

It is important to note that, in addition to the importance $\mathbb{F}_U$, the algorithm also provides, for each partial hyperparameter instantiation θ_U, the prediction of the marginal performance $\hat{a}(\theta_U)$ and an estimation of its standard deviation σ_{θ_U}.

3.4. Performance metrics

In order to evaluate the accuracy of the proposed models, we need a performance metric. In this paper, as motivated below, we use the *symmetric mean absolute percentage error (sMAPE)* [45]. Given a vector $\mathbf{Y} = [y_1,...,y_N]^\top$ of real outputs and a vector $\hat{\mathbf{Y}} = [\hat{y}_1,...,\hat{y}_N]^\top$ of predicted outputs, the sMAPE metric can be computed as:

$$\text{sMAPE} = \frac{100}{N}\sum_{k=1}^{N}\frac{|y_k - \hat{y}_k|}{(|y_k| + |\hat{y}_k|)/2}. \tag{4}$$

The reason for selecting the sMAPE instead of the more traditional MAPE is the fact that the MAPE is affected by different issues [45]. Particularly, for our application, the MAPE becomes sensitive to values close to zero. When an output y_i gets close to zero, the corresponding MAPE contribution becomes very large and it dominates the final value.

3.5. Diebold-Mariano (DM) Test

The sMAPE metric defined above only provides an assessment of which model has, for the data use, a better accuracy. While the accuracy of a model can be higher, the difference in performance might be not significant enough to establish that the model is really better. To assess the statistical significance in the difference of predictive accuracy performance, a commonly used tool is the Diebold-Mariano test [46].

Given a time series vector $\mathbf{Y} = [y_1,...,y_N]^\top$ to be forecasted, two prediction models M_1 and M_2, and the associated forecasting errors $\boldsymbol{\varepsilon}^{M_1} = [\varepsilon_1^{M_1},...,\varepsilon_N^{M_1}]^\top$ and $\boldsymbol{\varepsilon}^{M_2} = [\varepsilon_1^{M_2},...,\varepsilon_N^{M_2}]^\top$, the DM test evaluates whether there is a significant difference in performance accuracy based on an error loss function $L(\varepsilon_k^{M_i})$. In particular, the DM test builds a loss differential function as:

$$d_k^{M_1,M_2} = L(\varepsilon_k^{M_1}) - L(\varepsilon_k^{M_2}), \tag{5}$$

and then, it tests the null hypothesis H_0 of both models having equal accuracy, i.e. equal expected loss, against the alternative hypothesis H_1 of the models having different accuracy, i.e.:

$$\text{Two-sided DM test} \begin{cases} H_0\colon\ \mathbb{E}(d_k^{M_1,M_2}) = 0, \\ H_1\colon\ \mathbb{E}(d_k^{M_1,M_2}) \neq 0, \end{cases} \tag{6}$$

with $\mathbb{E}$ representing the expected value. Similar to the standard two-sided test, a one-sided DM test can be built by testing the null hypothesis that the accuracy of M_1 is equal or worse than the accuracy of M_2 versus the alternative hypothesis of the accuracy of M_1 being better:

$$\text{One-sided DM test} \begin{cases} H_0\colon\ \mathbb{E}(d_k^{M_1,M_2}) \geqslant 0, \\ H_1\colon\ \mathbb{E}(d_k^{M_1,M_2}) < 0. \end{cases} \tag{7}$$

While the loss function L can be freely chosen, it has to ensure that the resulting loss differential is covariance stationary. A loss function that is typically used is:

$$L(\varepsilon_k^{M_i}) = |\varepsilon_k^{M_i}|^p, \tag{8}$$

where usually $p \in \{1,2\}$.

4. Data

In this section, the data used for the research is introduced.

4.1. Data selection and motivation

In general, when looking at the day-ahead forecasting literature, many inputs have been proposed as meaningful explanatory variables, e.g. temperature, gas and coal prices, grid load, available generation, or weather [1].

To make our selection, we try to make sure that the selected data is not only related to the price dynamics, but also fulfills some minimum requirements. More specifically, we only choose data that is freely available for most European markets so that the proposed models can easily be exported to other EU markets. Moreover, we ensure that the data represents market integration, i.e. that comes from two connected markets. In particular, we select the period from 01/01/2010 to 31/11/2016 as the time range of study, and we consider the following data:

1. Day-ahead prices from the EPEX-Belgium and EPEX-France power exchanges. They are respectively denoted as p_B and p_F.
2. Day-ahead forecasts of the grid load and generation capacity in Belgium and France. Like in other European markets, these forecasts are available before the bid deadline on the website of the *transmission system operators (TSOs)*: ELIA for Belgium and RTE for France. They are respectively denoted as l_B and g_B for Belgium, and as l_F and g_F for France.
3. Calendar of public holidays H_F and H_B in France and Belgium in the defined time range.

While it could be argued that different weather data could also be easily accessible and important for the forecasting, for our research, we have decided to disregard them for two main reasons:

1. Weather factors are already indirectly taken into account in the grid load and generation forecasts provided by the TSO. In particular, the generation forecast has to consider weather information regarding wind speed and solar radiation. Likewise, load forecasts also need to consider temperature and other weather variables to obtain the electricity consumption.
2. Weather data are local phenomena, and as such, they can greatly vary from one part of a country to another. As a result, unlike the grid load or generation data, it is not possible to select a single value of the temperature or any other weather data for a given time interval.

4.2. Data processing

It is important to note that the data used is mostly unprocessed. In particular, as we intend to forecast and detect spikes, price outliers are not eliminated. The only data transformation is a price interpolation and elimination every year corresponding respectively to the missing and extra values due to the daylight saving. In addition, while all the metrics and tests are computed using the real prices, the training of the neural networks is done with data normalized to the interval $[-1,1]$. This last step is necessary because the input features have very different ranges; therefore, if the data is not normalized, the training time increases and the final result is a network that displays, in general, worse performance [47].

4.3. Data division

To perform the different experiments, we divide the data into three sets:

1. Training set (01/01/2010 to 31/11/2014): These data are used for

training and estimating the different models.
2. Validation set (01/11/2014 to 31/11/2015): A year of data is used to conduct early-stopping to ensure that the model does not overfit and to select optimal hyperparameters and features.
3. Test set (01/11/2015 to 31/11/2016): A year of data, which is not used at any step during the model estimation process, is employed as the out-of-sample dataset to compare and evaluate the models.

4.4. Data access

For the sake of reproducibility, we have only used publicly available data. In particular, the load and generation day-ahead forecasts are available on the webpages of RTE [48] and Elia [49], the respective TSOs in France and Belgium. In the case of the prices, they can be obtained from the ENTSO-E transparency platform [50].

5. Modeling framework

In this section, two different models are proposed to include market integration in day-ahead forecasting. The two models are similar to each other as both of them try to forecast the full set of day-ahead prices. However, they differ from each other in the number and type of prices that they predict; in particular, while the first model predicts the day-ahead prices of a single market, the second model combines a dual market prediction into a single model.

5.1. Single-market day-ahead forecaster

The basic model for predicting day-ahead prices uses a DNN in order to forecast the set of 24 day-ahead prices.

5.1.1. Conceptual idea

Based on the results of [20], we select a DNN with two hidden layers as forecasting model. Defining the input of the model as the relevant data $\mathbf{X} = [x_1,...,x_n]^\top \in \mathbb{R}^n$ available at day $d-1$ in the local and neighboring markets, and letting n_1 and n_2 be the number of neurons of the first and the second hidden layer respectively, and $\mathbf{p} = [p_1,p_2,...,p_{24}]^\top \in \mathbb{R}^{24}$ the set of 24 day-ahead prices to be forecasted, the proposed model can be represented as in Fig. 2.

5.1.2. Model parameters

The parameters of the DNN are represented by the set of weights that establish the mapping connections between the different neurons of the network:

- $\mathbf{W}_{i,i}$: the vector of weights between the input $\mathbf{X}$ and the neuron i of the first hidden layer.
- $\mathbf{W}_{h,i}$: the vector of weights between the first hidden layer and the neuron i of the second hidden layer.
- $\mathbf{W}_{o,i}$: the vector of weights between the second hidden layer and the output price vector $\mathbf{p}$.
- $\mathbf{b}_k = [b_{k1},...,b_{kn_k}]^\top$: the vector of bias weights in the k^{th} hidden layer, with $k = 1,2$.
- $\mathbf{b}_o = [b_{o,1},...,b_{o,24}]^\top$: the vector of bias weights in the output layer.

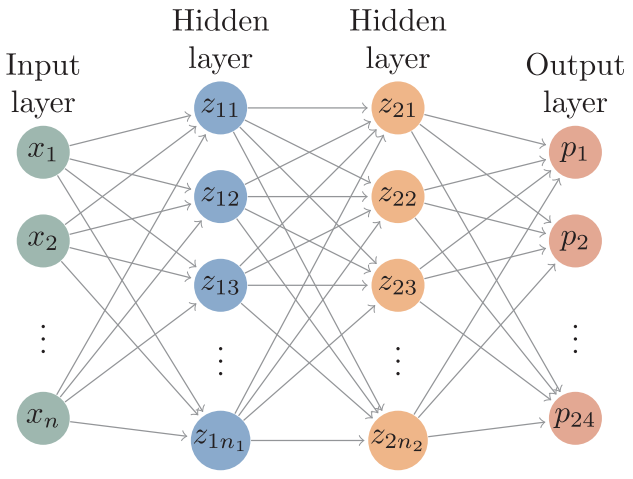


Fig. 2. DNN to forecast day-ahead prices.

5.1.3. Model equations

Using the above definitions, the equations of the DNN can be defined as:

$$z_{1i} = f_{1i}(\mathbf{W}_{i,i}^\top \cdot \mathbf{X} + b_{1i}), \quad \text{for } i = 1,...n_1, \tag{9a}$$

$$z_{2i} = f_{2i}(\mathbf{W}_{h,i}^\top \cdot \mathbf{z}_1 + b_{2i}), \quad \text{for } i = 1,...n_2, \tag{9b}$$

$$p_i = f_{o,i}(\mathbf{W}_{o,i}^\top \cdot \mathbf{z}_2 + b_{o,i}), \quad \text{for } i = 1,...24, \tag{9c}$$

where f_{1i} and f_{2i} respectively represent the activation function of neuron i in the first and second hidden layer, and where $f_{o,i}$ is the activation function of neuron i in the output layer.

5.1.4. Network structure

The rectified linear unit [51] is selected as the activation function of the two hidden layers. However, as the prices are real numbers, no activation function is used for the output layer.

To select the dimension n of the network input and the dimensions n_1 and n_2 of the hidden layers, a feature selection and hyperparameter optimization are performed.

5.1.5. Training

The DNN is trained by minimizing the mean absolute error. In particular, given the training set $\mathscr{S}_\mathscr{F} = \{(\mathbf{X}_k,\mathbf{p}_k)\}_{k=1}^N$, the optimization problem that is solved to train the neural network is:

$$\underset{\mathbf{W}}{\text{minimize}} \quad \sum_{k=1}^{N} \|\mathbf{p}_k - F(\mathbf{X}_k,\mathbf{W})\|_1, \tag{10}$$

where F: $\mathbb{R}^n \to \mathbb{R}^{24}$ is the neural network map. The selection of the mean absolute error instead of the more traditional root mean square error is done for a simple reason: as the electricity prices have very large spikes, the Euclidean norm would put too much importance on the spiky prices.

The optimization problem is initialized via single-start with the Glorot initialization [52] and solved using Adam [53], a version of the stochastic gradient descent method that computes adaptive learning rates for each model parameter. Adam is selected for a clear reason: as the learning rate is automatically computed, the time needed to tune the learning rate is smaller in comparison with other optimization methods. Together with Adam, the forecaster also considers early stopping [54] to avoid overfitting.

5.2. Dual market day-ahead forecaster

A possible variant of the single-market model is a forecaster that predicts the prices of two markets in a single model. While this might seem counter-intuitive at first, i.e. adding extra outputs to the model could compromise its ability to forecast the set of 24 prices that we are really interested in, this approach can, in fact, lead to neural networks that are able to generalize better.

5.2.1. Conceptual idea

The general idea behind forecasting two markets together is that, as we expect prices in both markets to be interrelated and to have similar dynamics, by forecasting both time series in a single model we expect the neural network to learn more accurate relations. In particular, it has been empirically shown that DNNs can learn features that can, to some

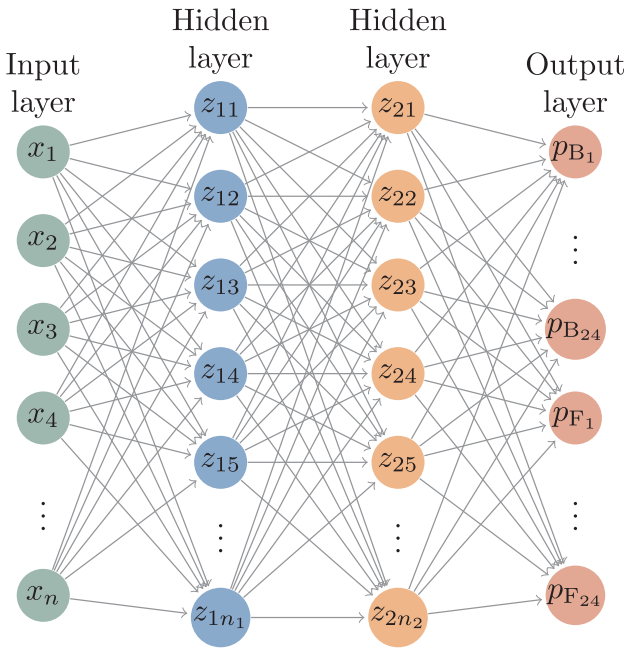


Fig. 3. DNN to simultaneously forecast day-ahead prices in two markets.

extent, generalize across tasks [55]. Similarly, it has also been shown that, by forcing DNNs to learn auxiliary related tasks, the performance and learning speed can be improved [56,57].

There are some possible hypotheses that can explain why training with multiple outputs can help to improve the performance:

1. The simplest explanation is the amount of data: as more data is available, the neural network can learn more relevant features. Moreover, as the tasks are related, the neural network has more data to learn features that are common to all tasks.
2. A second reason is regularization: By solving different tasks, the network is forced to learn features useful for all tasks and to not overfit to the data of a single task.

5.2.2. Model implementation

Consider an electricity market B and a second electricity market F that is connected to B. Then, defining the output of the network by $\mathbf{p} = [p_{B_1},...,p_{B_{24}},p_{F_1},...,p_{F_{24}}]^\intercal \in \mathbb{R}^{48}$, i.e. the set of 48 day-ahead prices from markets B and F, and keeping the rest of the DNN parameter definitions the same, the new DNN structure can be represented as in Fig. 3. In addition, as both models only differ in the output size, the implementation details are exactly the same as defined for the single-market model in Section 5.1.5.

6. Feature selection algorithm

As explained in the introduction, while the feature selection methods for electricity price forecasting proposed in the literature provide good and fast algorithms, they have two drawbacks:

1. They perform a filter step where the model performance is not considered.
2. For the nonlinear methods, the different inputs have to be transformed, i.e. the selection is not done over the original feature set, and thus, some feature information might be lost.

Therefore, we propose a nonlinear wrapper method that directly evaluates the features on the prediction model; in particular, while the approach is more computationally demanding, it can provide a better selection as it uses the real predictive performance without any data transformations.

6.1. Algorithm definition

In Section 3.3 we have introduced the TPE algorithm, a method for hyperparameter optimization, together with functional ANOVA, an approach for assessing hyperparameter importance. In this section, we combine both methods to build a feature selection algorithm that consists of four steps:

1. Model the features as hyperparameters.
2. Optimize the hyperparameters/features.
3. Analyze the results.
4. Select the important features.

6.1.1. Features as hyperparameters

The first step of the algorithm is to model the selection of features as model hyperparameters. In particular, we consider two types of features:

1. Binary features θ_B, whose selection can be done through a binary variable, i.e. $\theta_B \in \{0,1\}$, where $\theta_B = 0$ would represent feature exclusion and $\theta_B = 1$ feature inclusion. Binary features represent the type of features considered by traditional algorithms. An example would be whether to include holidays data or whether to select a specific lag in an ARIMA model.
2. Integer features θ_I, which not only can model the inclusion-exclusion of an input, but also select some associated size or length, i.e. $\theta_I \in \mathbb{Z}$, where $\theta_I = 0$ represents exclusion. Examples would be the number of past days of price data or the maximum lag of an ARIMA model.

Given these definitions, the binary features are modeled as hyperparameters using the hyperparameter space Θ_B and the hyperparameter set $B = \{1,...,n_B\}$. Likewise, the integer features are modeled by the hyperparameter space Θ_I and the hyperparameter set $I = \{n_B + 1,...,n_B + n_I\}$. Finally, the full hyperparameter space is defined by $\Theta = \Theta_B \cup \Theta_I$ and the hyperparameter set by $Z = B \cup I$.

6.1.2. Feature optimization

The second step of the algorithm is to perform a TPE optimization over the hyperparameter-feature space. The result of the algorithm is the optimal feature selection θ^* together with the set $\mathscr{H} = \{(\theta_k,p_k)\}_{k=1}^T$ of feature-performance pairs, where p_k represents the model predictive accuracy when using the feature selection θ_k.

The fact that a feature is part of θ^*, does not guarantee that the feature is relevant; specifically, a feature might have little or no effect in the performance, and still, as long as it does not have a negative effect, it might appear in the optimal configuration. As a result, if no further processing is considered, the algorithm might select redundant features, and in turn, lead to more computationally expensive models and increase the risk of overfit.

6.1.3. Feature analysis

To solve the problem of detecting unnecessary features, the algorithm comprises a third step where feature importance is analyzed. In particular, using the functional ANOVA methodology proposed in [44], the algorithm analyzes $\mathscr{H}$ and provides the importance of each feature i and each pairwise interaction $\{i,j\}$ as the percentage-wise contribution to the performance variance $\mathbb{V}$. Using the definitions given in Section 3.3.2 and (2) and (3), the algorithm computes the importance of feature Θ_i and each pairwise interaction $\Theta_i \times \Theta_j$ by:

$$\mathbb{F}_{\{i\}} = \frac{\mathbb{V}_{\{i\}}}{\mathbb{V}}, \qquad \mathbb{F}_{\{i,j\}} = \frac{\mathbb{V}_{\{i,j\}}}{\mathbb{V}}. \tag{11}$$

In addition, for each feature $i \in Z$ and feature instantiation $\theta_i \in \Theta_i$, the algorithm also provides the predicted marginal performance $\hat{a}(\theta_i)$.

6.1.4. Feature selection

The fourth and final algorithm step is the selection itself. In particular, making use of the obtained $\mathbb{F}_{\{i\}}$,$\mathbb{F}_{\{i,j\}}$ and $\hat{a}(\theta_i)$, the selection procedure performs the following steps:

1. Define a threshold parameter $\epsilon \in (0,1]$.
2. Make a pre-selection by discarding features that do not improve nor decrease the performance. In particular, regard features i whose importance $F_{\{i\}}$ is larger than ϵ:

$$U_1^* = \{i \in Z | F_{\{i\}} > \epsilon\}, \tag{12a}$$

or features i that have at least one pairwise contribution $F_{\{i,j\}}$ larger than ϵ:

$$U_2^* = \{i \in Z | \exists\, j \in Z \backslash \{i\}\colon F_{\{i,j\}} > \epsilon\}. \tag{12b}$$

3. With the remaining features in $U_1^* \cup U_2^*$, perform a second selection U^* by discarding those features whose predicted marginal performance $\hat{a}(\theta_i)$ is lower when being included than when being excluded, i.e.:

$$U^* = \{i \in U_1^* \cup U_2^* | \exists\, \theta_i \in \Theta_i\colon \mu_{\theta_{i,0}} < \hat{a}(\theta_i)\}, \tag{12c}$$

where $\mu_{\theta_{i,0}}$ represents the marginal performance $\hat{a}(\theta_i = 0)$ of excluding feature i.
4. Finally, the set of selected binary features can be obtained by:

$$U_B^* = U^* \cap B. \tag{12d}$$

Similarly, for the set of optimal integer features U_I^*, the selection is done in terms of the feature itself and the instantiation with the best performance:

$$U_I^* = \{\{i,\theta_i^*\} | i \in U^* \cap I, \theta_i^* = \underset{\theta_i}{\operatorname{argmax}}\, \hat{a}(\theta_i)\}. \tag{12e}$$

6.2. Case study

To evaluate the proposed algorithm, we use it to select the features for predicting Belgian prices and to obtain a first assessment of the effect of market integration, i.e. the effect of French features in forecasting Belgian prices. To perform the analysis, we consider the first and simpler DNN proposed in Section 5.

6.2.1. Feature definition

In order to perform the feature selection, we first need to model each possible input as either a binary or an integer feature. As described in Section 4, the available features are the day ahead prices p_B and p_F, the day-ahead forecasts l_B and l_F of the grid load, the day-ahead forecasts g_B and g_F of the available generation, and the calendar of public holidays H_B and H_F.

Considering that, given the market at time h, we aim at forecasting the time series vector $\mathbf{p}_{B_h} = [p_{B_{h+1}},\dots,p_{B_{h+24}}]^\top$ of Belgian day-ahead prices, the use of the day-ahead loads $\mathbf{l}_{B_h} = [l_{B_{h+1}},\dots,l_{B_{h+24}}]^\top$ and $\mathbf{l}_{F_h} = [l_{F_{h+1}},\dots,l_{F_{h+24}}]^\top$, and the use of the day-ahead capacity generations $\mathbf{g}_{B_h} = [g_{B_{h+1}},\dots,g_{B_{h+24}}]^\top$ and $\mathbf{g}_{F_h} = [g_{F_{h+1}},\dots,g_{F_{h+24}}]^\top$, should be modeled as binary features θ_{l_B},θ_{l_F},θ_{g_B}, and θ_{g_F}.

Similarly, for the public holidays, the features can also be modeled as binary variables θ_{H_B} and θ_{H_F}. In particular, as the set of 24 hours of a day is either a holiday or not, the holidays are defined as model inputs X_{H_B},$X_{H_F} \in \{0,1\}$, with 0 and 1 representing respectively no holiday and holiday.

To model the Belgian prices, we need to use an integer feature to select the number of the considered past values. In particular, as the prices display daily and weekly seasonality, we have to use two integer features: $\theta_{p_{B,d}} \in \{1,2,\dots,6\}$ as the feature modeling the number of past days during the last week (daily seasonality) and $\theta_{p_{B,w}} \in \{1,2,3\}$ as the feature modeling the number of days at weekly lags (weekly seasonality). Based on the selection of $\theta_{p_{B,d}}$ and $\theta_{p_{B,w}}$, the considered EPEX-Belgium past prices can be decomposed as the price inputs $\mathbf{X}^d_{p_{B,h}}$ at daily lags and the price inputs $\mathbf{X}^w_{p_{B,h}}$ at weekly lags:

$$\mathbf{X}^d_{p_{B,h}} = [p_{B_{h-i_1}},\dots,p_{B_{h-i_{N_d}}}]^\top, \tag{13a}$$

$$\mathbf{X}^w_{p_{B,h}} = [p_{B_{h-j_1}},\dots,p_{B_{h-j_{N_w}}}]^\top, \tag{13b}$$

where

$$\{i_1,\dots,i_{N_d}\} = \{i | 0 \leqslant i \leqslant 24\cdot\theta_{p_{B,d}} - 1\} \tag{13c}$$

$$\{j_1,\dots,j_{N_w}\} = \{j | 1 \leqslant k \leqslant \theta_{p_{B,w}},\ k\cdot 168\cdot\theta_{p_{B,d}} \leqslant j \leqslant k\cdot 192\cdot\theta_{p_{B,d}} - 1\}. \tag{13d}$$

It is important to note that, as this is the time series to be predicted, we disregard the cases where no daily nor weekly seasonality is used, i.e. $\theta_{p_{B,d}} = 0$ or $\theta_{p_{B,w}} = 0$.

Finally, for the EPEX-France prices we could use the same integer features as for EPEX-Belgium. However, for simplicity, we directly consider the same lags for both time series and model the French prices as a binary feature θ_{p_F}. It is important to note that, despite having the same length, the selection of both time series is still independent; particularly, the lags are only defined for Belgium, and the French prices are just excluded or included. The modeled input features are summarized in Table 1.

6.2.2. Hyperparameter optimization

In order to guarantee that the network is adapted according to the input size, we simultaneously optimize the hyperparameters of the DNN, i.e. the number of neurons n_1 and n_2. In particular, as the feature selection method is based on a hyperparameter optimization, we directly include the number of neurons as integer hyperparameters that are optimized together with the features. We set the domain of n_1 as the set of integers $\{100,101,\dots,400\}$ and the one of n_2 as $\{0\} \cup \{48,49,\dots,360\}$, where $n_2 = 0$ represents removing the second hidden layer and using a network of depth one.

6.2.3. Experimental setup

In order to use the proposed algorithm, we first need to define the threshold ϵ for the minimum variance contribution; in our case, we select $\epsilon = 0.5\%$. In addition, we also need to select the maximum number of iterations T of the TPE algorithm; we found $T = 1000$ to offer a good trade-off between performance and accuracy. Particularly, considering that training a single model takes 2 min, the full feature selection requires 30 h. While this might seem a long time, this step is only performed after some periodic time, e.g. a month, to reassess feature dependencies; therefore, the proposed approach and settings yield a feasible and accurate method for the time scale of day-ahead prices.

Table 1
Definition of the modeled input features.

Feature	Domain	Definition
$\theta_{p_{B,d}}$	{1,...,6}	Number of past days for input price sequence
$\theta_{p_{B,w}}$	{1,...,3}	Days at weekly lags for input price sequence
θ_{p_F}	{0,1}	Day-ahead price in France
θ_{l_B}	{0,1}	Load in Belgium
θ_{l_F}	{0,1}	Load in France
θ_{g_B}	{0,1}	Generation in Belgium
θ_{g_F}	{0,1}	Generation in France
θ_{H_B}	{0,1}	Holiday in Belgium
θ_{H_F}	{0,1}	Holiday in France

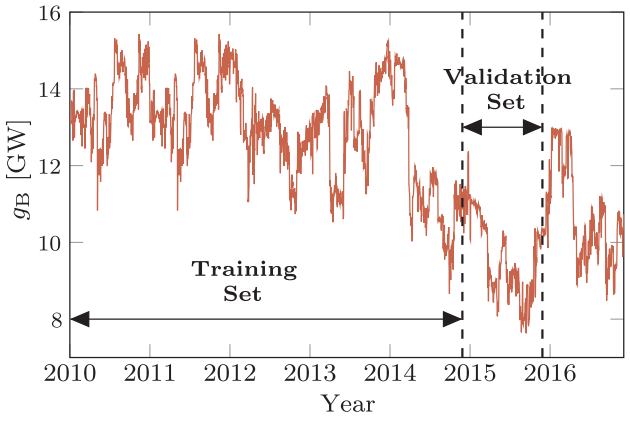


Fig. 4. Generation in Belgium in the considered period.

For implementing the functional analysis of variance, we use the python library fANOVA developed by the authors of [44]. Likewise, for implementing the TPE algorithm, we use the python library hyperopt [58].

6.2.4. Results

In a first conducted experiment, we obtained an unexpected result: inclusion/exclusion of the generation capacity in Belgium g_B accounts for roughly 75% of the performance variance $\mathbb{V}$, with inclusion of g_B dramatically decreasing the predictive accuracy. Since the generation capacity has been successfully used by other authors as a market driver [1], this result requires some explanation. From Fig. 4, which displays the time series of g_B, we can comprehend the result: right before the transition from the training to the validation set, the average g_B suffers a major change and drops from approximately 14 GW to 9 GW. Because of the drastic drop, it is likely that some relations that are learned based on the training set, do not hold in the validation set, and that as a result, the predictive performance in the validation set worsens when g_B is considered.

This regime change in g_B violates the assumption that conditions in the training, validation, and test sets are equal. Therefore, to perform a correct feature selection and to guarantee that the three datasets hold similar conditions, the experimental setup should disregard θ_{g_B}. It is important to note that, before taking this decision, we have considered shuffling the data to ensure homogeneous conditions between the three sets. However, this alternative was avoided for two reasons:

1. As the output prices in some samples are the input features in others, data has to be discarded in order to avoid data contamination between the three sets. As a result, since the larger the dataset the better the DNN can generalize, this implementation could potentially decrease the predictive accuracy of the model.
2. Since the end goal of the model is to forecast recent prices, it is meaningless to try to model an input-output relation that no longer holds.

Considering these facts, a correct feature selection is performed without θ_{g_B}. As depicted in Table 2, the first result to be noted from the new experimental results is that, as g_B is a big source of error, the variance $\hat{\mathbb{V}}$ of the sMAPE performance is reduced by a factor of 5. In addition, as it could be expected, the results obtained in this new experiment display a more distributed contribution among the different features. In particular, in the first experiment, g_B was responsible for 75% of the performance variance. Now, as depicted in Table 3, French prices and load account for roughly 50% of the total performance variance, and the available generation in France, the load in Belgium, and the number of past days play a minor role.

Table 2
Performance variance with and without g_B.

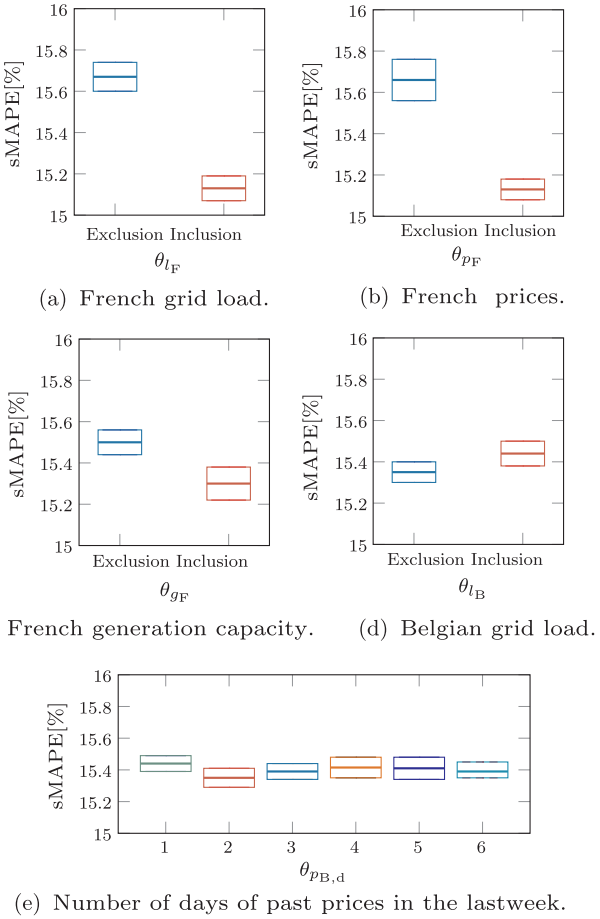
	$\hat{\mathbb{V}}$
Feature selection with g_B	0.58%^{2}
Feature selection without g_B	0.12%^{2}

Table 3
Variance contribution of single features for the second feature selection experiment.

	Contribution to $\mathbb{V}$
All main effects	64.9%
French load	28.4%
French prices	25.7%
French generation	4.78%
Belgium load	1.0%
Past days number	0.8%

Based on the above results, we can make a first selection and remove from the set of possible inputs the public holidays θ_{H_B} and θ_{H_F} as both seem not to be decisive. Similarly, we can select $\theta_{p_{B,w}} = 1$ as the number of days at weekly lags seems to be non-critical. Finally, to complete the feature selection, we should use the marginal performances of the five important features represented in Fig. 5; based on them, it is clear that we should select the price, load and generation in France, discard the grid load in Belgium, and use two days of past price data.

Together with the features, we have also optimized the hyperparameters of the model. The results show that the suitable numbers of neurons are $n_2 = 200$ and $n_1 = 320$.



(a) French grid load. (b) French prices.

(c) French generation capacity. (d) Belgian grid load.

(e) Number of days of past prices in the lastweek.

Fig. 5. Marginal performance on the validation set of the five most important features.

6.3. Discussion

Based on the results of the feature selection algorithm, we should include the following features as model inputs:

1. Day-ahead load and generation in France:
2. Last two days of Belgian and French prices:
3. Belgian and French prices a week before:

In addition, while it seems that the different French market features, i.e. market integration features, play a large role in the forecasting accuracy, the results are only enough to have a general idea of the importance of French data; particularly, a statistical analysis is required before making any further conclusion.

Finally, while we have used the proposed algorithm to select the input features, we have not yet provided an evaluation of its accuracy. In particular, to assess its performance, we could compare models using only optimally selected features against models using also features that have been discarded; more specifically, we could evaluate the difference in predictive accuracy by means of hypothesis testing (see Section 7.2.4).

7. Evaluation of market integration and modeling framework

The analysis provided by the feature selection algorithm is based on the validation set; while this dataset is not used for training the network, it is employed for early stopping and hyperparameter optimization. Therefore, to have a fully fair and unbiased evaluation, we need an extra comparison using unseen data to the full training process. Moreover, as the feature selection results were obtained using the first proposed model, results for the second model are also required. Finally, to have a meaningful assessment, the statistical significance of the results should be computed. To fulfill the requirements, the goal of this section is twofold:

1. Provide statistical significance of the improvements of using French market data, i.e. market integration, by performing a DM test on the out-of-sample data represented by the test set.
2. Based on the same statistical test, demonstrate how a dual-market forecaster can provide significant improvements in predictive accuracy.

7.1. Diebold-Mariano test

To assess the statistical significance in the difference of predictive accuracy, we use the DM test as defined by (5)–(8). Since the neural network is trained using the absolute mean error, we choose to use also the absolute error to build the loss differential:

$$d_k^{M_1,M_2} = |\varepsilon_k^{M_1}| - |\varepsilon_k^{M_2}|. \tag{14}$$

In addition, we follow the same procedure as in [26] and we perform an independent DM test for each of the 24 time series representing the different hours of a day. The reason for this is that, as we use the same information to forecast the set of 24 prices, the forecast errors within the same day would exhibit a high correlation. Moreover, to have an assessment of the whole error sequence, we also perform the DM test considering serial correlation of order k in the error sequence. Particularly, recalling that optimal k-step-ahead forecast errors are at most $(k-1)$-dependent [46], we perform a DM test on the full loss differential considering serial correlation of order 23.

In the various experimental setups of this case study, we employ the one-sided DM test given by (7) at the 95% confidence level. This selection is done because we want to assess whether the performance of a forecaster A is statistically significantly better than a forecaster B, not whether the performances of forecasters A and B are significantly different (like it would be the case in the two-sided DM test). In more detail, for each hour $h = 1,...,24$ of the day, we test the null hypothesis of a model M_1 that uses French data having the same or worse accuracy than a model M_2 that uses no French data. More specifically, we perform the following tests:

$$\begin{cases} H_0: \mathbb{E}(d_{h_k}^{M_1,M_2}) \geqslant 0, \\ H_1: \mathbb{E}(d_{h_k}^{M_1,M_2}) < 0, \end{cases} \text{for } h = 1,...24, \tag{15}$$

where $[d_{h_1},...,d_{h_{N/24}}]^T$ represents the vector sequence of loss differentials of hour h. In addition, we perform the same test but considering the full loss differential sequence and assuming serial correlation:

$$\begin{cases} H_0: \mathbb{E}(d_k^{M_1,M_2}) \geqslant 0, \\ H_1: \mathbb{E}(d_k^{M_1,M_2}) < 0. \end{cases} \tag{16}$$

7.2. French market data: statistical significance

In Sections 6.2 and 6.3, we have showed that using market data from connected markets can help to improve the performance. In this section, we extend the analysis by directly comparing a model that includes this type of data against a model that excludes it, and then, performing a DM test to analyze the statistical significance.

7.2.1. Experimental setup

The model used to perform the evaluation is the single-market forecaster employed for the feature selection. In particular, based on the obtained hyperparameter results, we select $n_1 = 320$ and $n_2 = 200$; similarly, considering the optimized prices lags obtained in the feature selection, we consider, as input sequence for the model, the Belgium prices during the last two days and a week before. Then, we discard as input features the capacity generation in Belgium as well as the holidays in both countries. Then, in order to compare the effect of French data, we consider the remaining features as possible inputs for the model, i.e. we compare the first model excluding all the French data and only considering Belgian prices with respect to the second model including the French data. We respectively refer to these two models as M_{NoFR} and M_{FR}.

In addition, while the load in Belgium l_B appears to be non-relevant, we decided to repeat the previous experiment but including l_B in both models. The reason for this is twofold:

1. By adding the Belgian load, we ensure that the good results of using French data are not due to the fact that the model does not include specific Belgian regressors.
2. Furthermore, with this experiment, we can also validate the results of the feature selection algorithm. In particular, as the load does not seem to play a big role, we expect the performance difference between models with and without l_B to be insignificant.

Similar as before, we refer to these models by M_{NoFR,l_B} and M_{FR,l_B}.

7.2.2. Case 1: models without l_B

In this experiment, we compare M_{NoFR} against M_{FR} by evaluating their performance on the year of yet unused data represented by the test set. As in a real-world application, to account for the last available information, the two models are re-estimated after a number days/weeks. In our application, considering that a model takes around 2 minutes to be trained on the GPU, we decide to re-estimate them using the smallest possible period of a day.

A first comparison of the models is listed in Table 4 by means of sMAPE. From this first evaluation, we can see that including the French data seems to really enhance the performance of the forecaster.

To provide statistical significance to the above result, we perform a DM test as described in Section 7.1. The obtained results are depicted in

Table 4
Performance comparison between M_{NoFR} and M_{FR} in the out-of-sample data in terms of sMAPE.

Model	M_{NoFR}	M_{FR}
sMAPE	16.0%	13.2%

Table 5
Performance comparison between M_{NoFR,l_B} and M_{FR,l_B} in the out-of-sample data in terms of sMAPE.

Model	M_{NoFR,l_B}	M_{FR,l_B}
sMAPE	15.7%	13.1%

Fig. 6, where the test statistic is represented for each of the 24 hours of a day and where the points above the dashed line accept, with a 95% confidence level, the alternative hypothesis of M_{FR} having better performance accuracy. As we can see from the plot, the forecast improvements of the model M_{FR} including French data are statistically significant for each one of the 24 day-ahead prices.

When the DM test is performed on the full loss differential and taking into account serial correlation, the obtained metrics completely agree with the results obtained for the individual 24 hours. In particular, the obtained p-value is $1.2{\cdot}10^{-11}$, which confirms the strong statistical significance of using the French data in the prediction model.

7.2.3. Case 2: models with l_B

Using the same procedure, we compare M_{NoFR,l_B} against M_{FR,l_B}. From Table 5 we can see how, as before, the model including French data outperforms the alternative.

To provide statistical significance to the obtained accuracy difference we again perform the DM tests. The obtained results are illustrated in Fig. 7; as before, including French data leads to improvements in accuracy that are statistically significant for the 24 predicted values. As before, when we consider the DM test for the full loss differential with serial correlation, the p-value is $1.6{\cdot}10^{-12}$, a value that agrees with Fig. 7 and confirms once more that the improvements of using French data are statistically significant.

7.2.4. Accuracy of the feature selection

Using the results of the previous two sections, we can illustrate the accuracy of the proposed feature selection algorithm in Section 6. In particular, when performing the feature selection, we have observed that the contribution of the Belgian load l_B was rather insignificant and even slightly negative; this led to discard l_B as an input feature. In this section, to verify that the selection algorithm performed the right choice, we perform DM tests to compare M_{NoFR,l_B} against M_{NoFR} and M_{FR,l_B} against M_{FR}. In particular, we perform a two-sided DM test per model pair with the null hypothesis of the models having equal accuracy.

For the sake of simplicity, we avoid depicting the DM test results for each individual hour; instead we directly illustrate the p-values of the DM test when considering the whole loss differential sequence with serial correlation. As can be seen from Table 6, the obtained p-values for both tests are above 0.05, and as a result, the null hypothesis of equal accuracy cannot be rejected, i.e. there is no statistical evidence of the models using Belgian load having different accuracy than the models without it.

Based on the obtained results, it is clear that using l_B is not relevant, and thus, that the choice performed by the feature selection algorithm is correct. In particular, while this experiment does not analyze the performance of the feature selection on all the inputs, it does consider the most problematic feature. More specifically, as many researchers have successfully used the load as an explanatory variable [6–8,27,34] and as the load itself does not display any regime change in the considered time interval, it is rather striking to see its minimal effect on the performance. Therefore, by demonstrating that the algorithm is correct when discarding the load, we obtain an assessment of its general accuracy, and we can conclude that the algorithm performs a correct feature selection.

7.3. Evaluation of a dual-market forecaster

In this section, we evaluate the possible improvements of using the dual-market forecaster and multi-tasking by comparing the single-market model against the dual-market forecaster predicting the day-ahead prices in Belgium and France. The models are denoted by M_{Single} and M_{Dual} and they both use the optimal features and hyperparameters obtained for the single-market model in Section 6. It is important to note that, while in an ideal experiment the hyperparameters of the dual-market forecaster should be re-estimated, for simplicity we decided to directly use the hyperparameters obtained for the single-market forecaster.

The initial comparison is listed in Table 7. From this first evaluation it seems that using dual-market forecasts can improve the performance.

To provide statistical significance to these results, we again perform the DM test for each of the 24 hours of a day. The obtained statistics are depicted in Fig. 8; as before, the points above the upper dashed line accept, with a 95% confidence level, the alternative hypothesis of M_{Dual} having a better performance accuracy. In addition, as not every hourly forecast is statistically significant, we represent in the same figure the alternative DM test with the null hypothesis of M_{Single} having equal or lower accuracy than M_{Dual}. This test is characterized by the lower dashed line and any point below this line accepts, with a 95% confidence level, that M_{Single} has better performance accuracy.

As we can see from the plot, the forecast improvements of the dual-market forecaster are statistically significant in 7 of the 24 day-ahead prices. In addition, the single-market forecaster is not significantly better in any of the remaining 17 day-ahead prices. Therefore, as M_{Dual} is approximately better for a third of the day-ahead prices and not worse for the remaining two-thirds, we can conclude that the dual-market forecaster is a statistically significant better forecaster.

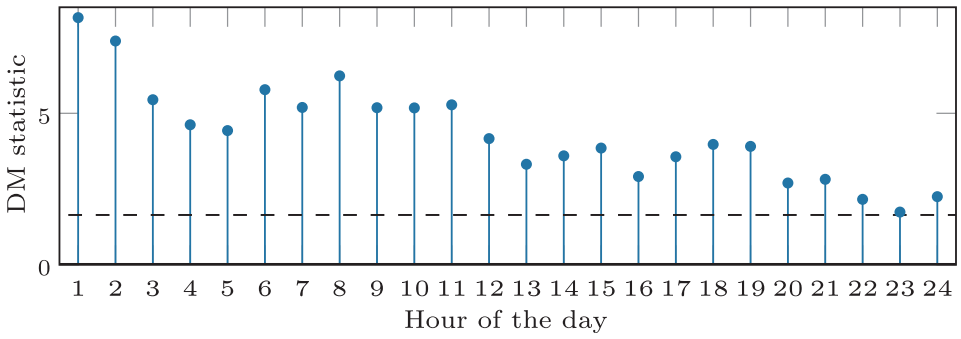


Fig. 6. DM test results when comparing M_{NoFR} and M_{FR}. Values above the dashed line reject the null hypothesis with a 95% confidence level, and in turn, represent cases where the accuracy of M_{FR} is significantly better.

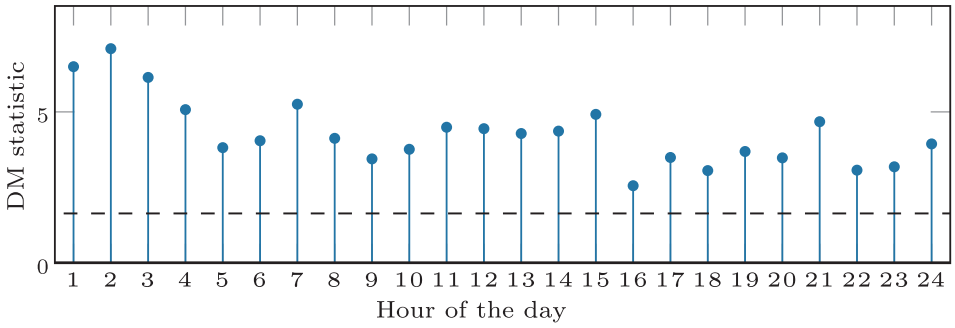


Fig. 7. DM test results when comparing M_{NoFR,l_B} and M_{FR,l_B}. Values above the dashed line reject the null hypothesis at a 5% significance level, and in turn, represent cases where the accuracy of M_{FR,l_B} is significantly better.

Table 6
p-values for DM test with the null hypothesis of models with l_B having equal accuracy as models without it.

Model Pair	*p*-value
M_{FR,l_B} vs M_{FR}	0.435
M_{NoFR,l_B} vs M_{NoFR}	0.275

Table 7
Performance comparison between the single and dual-market forecasters in terms of sMAPE.

Model	M_{Single}	M_{Dual}
sMAPE	13.2%	12.5%

Finally, we also perform the DM test on the full loss differential considering serial correlation. Once again, the obtained metrics agree with the results obtained for the individual 24 hours: with a *p*-value of $9.5{\cdot}10^{-03}$, the test results confirm the statistical significance of the difference in predictive accuracy when using the dual-market forecaster.

7.4. Analysis and discussion

To understand and explain the obtained results, we have to note that, as introduced in Section 2.4, market integration across European electricity markets has been increasing over the years due to EU regulations. This highly nonlinear and complex effect dramatically modifies the dynamics of electricity prices and is behind the obtained improvements of our models. In particular, our forecasters use this effect to outperform alternative techniques that have traditionally ignored it: the first forecaster proposed, which models market integration in the input space, obtains statistically significant improvements w.r.t. to model counterparts that disregard market integration. The second proposed forecaster, which goes one step further by modeling market integration in the output space, is shown to be crucial to obtain further significant improvements. For our case study, this translates to the following conclusions:

1. Using features from the French market significantly enhances the predictive accuracy of a model forecasting Belgian prices. The results are statistically significant and independent of whether Belgian features are considered or not.
2. A dual-market forecaster simultaneously predicting prices in France and Belgium can improve the predictive accuracy. In particular, by solving two related tasks, it is able to learn more useful features, to better generalize the price dynamics, and to obtain improvements that are statistically significant.
3. The proposed feature selection algorithm is able to perform a correct assessment of the importance of features.

In addition, it is interesting to see how explanatory variables from the EPEX-Belgium, e.g. load and generation, have almost no influence in the day-ahead prices. In fact, from the obtained results, it is surprising to observe how French factors play a larger role in Belgian prices than the local Belgian features.

As a final discussion, it is necessary to indicate why, while being neighboring countries of Belgium, The Netherlands and Germany and their respective markets have not been considered in the study. The reason for not considering The Netherlands is the fact that the amount of available online data is smaller than in France and Belgium, and thus, training the DNNs can be harder. In the case of Germany, the reason for not considering it is that, at the moment, there is not a direct interconnection of the electrical grid between Belgium and Germany.

7.5. Practical applications

As a last remark, it is important to point out the different practical applications that these results have. Particularly, there are two main obvious applications where this research can be highly beneficial. The first and most important application is its usage by utility companies to increase their economic profit. More specifically, a 1% improvement in the MAPE of the forecasting accuracy results in about 0.1–0.35% cost reduction [59]. For a medium-size utility company with a peak load of 5 GW, this translates to saving approximately $1.5 million per year [60,61].

In addition, improvements in forecasting accuracy are key to have a stable electrical grid. Particularly, as the integration of renewable

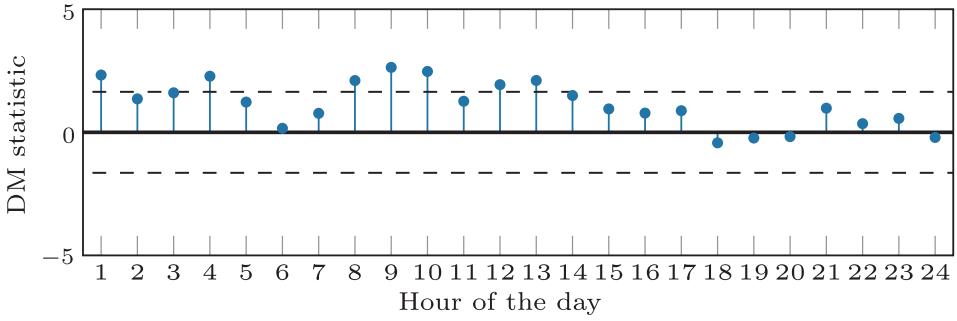


Fig. 8. DM test results when comparing M_{Single} and M_{Dual}. Values above the top dashed line represent cases where, with a 95% confidence level, M_{Dual} is significantly better. Similarly, values below the lower dashed line accept, with a 95% confidence level, that M_{Dual} is significantly worse.

energy sources increases, so do the imbalances in the electrical grid due to mismatches between generation and consumption. To tackle this issue demand response methods [62–64] have been traditionally applied. By accurate forecasting electricity prices it is also possible to improve the situation. In particular, prices are usually low (high) when generation is larger (lower) than consumption. Therefore, given the right forecasts, market agents have economic incentives to buy (sell) energy when prices are low (high), and in turn, to reduce the grid imbalances. Therefore, using accurate price forecasting, market agents can steered and motivated so that grid imbalances are reduced.

8. Conclusions

We have analyzed how market integration can be used to enhance the predictive accuracy of day-ahead price forecasting in electricity markets. In particular, we have proposed a first model that, by considering features from connected markets, improves the predictive performance. In addition, we have proposed a dual-market forecaster that, by multitasking and due to market integration, can further improve the predictive accuracy. As a case study, we have considered the electricity markets in Belgium and France. Then, we have showed how, considering market integration, the proposed forecasters lead to improvements that are statistically significant. Additionally, we have proposed a novel feature selection algorithm and using the same case study, we have shown how the algorithm correctly assesses feature importance.

In view of these results, it is clear that market integration can play a large role in electricity prices. In particular, the influence of neighboring markets seems to be important enough to build statistically significant differences in terms of forecasting accuracy. As a consequence, as the EU has implemented regulations to form an integrated EU market but there is still little insight in the outcome of such regulations, these results are important in terms of policy making. In particular, the fact that market integration largely modifies the price dynamics between Belgium and France is an indicator that the regulations that were put in place are working. As a result, using the proposed methodology, policy makers can benefit from a general tool to evaluate the market integration regulations in other EU regions.

In addition, these results are also of high importance in terms of grid stability and economic profit of market agents. In particular, as the knowledge of the dynamics of electricity prices increases, the grid operator might be able to better prevent some of the grid imbalances characterized by large price peaks. The increased knowledge is also economically beneficial for market agents: a 1% improvement in MAPE accuracy translates to savings of $1.5 million per year for a medium-size utility company.

As a first step to help policy markets, in future work the performed experiments will be expanded to the other European markets.

Acknowledgment

This research has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement no 675318 (INCITE).

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.apenergy.2017.11.098.

References

[1] Weron R. Electricity price forecasting: a review of the state-of-the-art with a look into the future. Int J Forecast 2014;30(4):1030–81. http://dx.doi.org/10.1016/j.ijforecast.2014.08.008.
[2] Jamasb T, Pollitt M. Electricity market reform in the European union: review of progress toward liberalization & integration. Energy J 2005;26:11–41. http://dx.doi.org/10.5547/issn0195-6574-ej-vol26-nosi-2.
[3] Weron R, Misiorek A. Forecasting spot electricity prices: a comparison of parametric and semiparametric time series models. Int J Forecast 2008;24(4):744–63. http://dx.doi.org/10.1016/j.ijforecast.2008.08.004.
[4] Crespo Cuaresma J, Hlouskova J, Kossmeier S, Obersteiner M. Forecasting electricity spot-prices using linear univariate time-series models. Appl Energy 2004;77(1):87–106. http://dx.doi.org/10.1016/S0306-2619(03)00096-5.
[5] 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;190:291–305. http://dx.doi.org/10.1016/j.apenergy.2016.12.130.
[6] Nogales FJ, Contreras J, Conejo AJ, Espínola R. Forecasting next-day electricity prices by time series models. IEEE Trans Power Syst 2002;17(2):342–8. http://dx.doi.org/10.1109/MPER.2002.4312063.
[7] Cruz A, Muñoz A, Zamora J, Espínola R. The 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.
[8] Misiorek A, Trueck S, Weron R. Point and interval forecasting of spot electricity prices: linear vs. non-linear time series models. Stud Nonlinear Dyn Econometr 2006;10(3):1–36. http://dx.doi.org/10.2202/1558-3708.1362.
[9] Diongue AK, Guégan D, Vignal B. Forecasting electricity spot market prices with a k-factor GIGARCH process. Appl Energy 2009;86(4):505–10. http://dx.doi.org/10.1016/j.apenergy.2008.07.005.
[10] Conejo A, Plazas M, Espinola R, Molina A. Day-ahead electricity price forecasting using the wavelet transform and ARIMA models. IEEE Trans Power Syst 2005;20(2):1035–42. http://dx.doi.org/10.1109/TPWRS.2005.846054.
[11] Tan Z, Zhang J, Wang J, Xu J. Day-ahead electricity price forecasting using wavelet transform combined with ARIMA and GARCH models. Appl Energy 2010;87(11):3606–10. http://dx.doi.org/10.1016/j.apenergy.2010.05.012.
[12] Amjady N, Hemmati M. Energy price forecasting – problems and proposals for such predictions. IEEE Power Energy Mag 2006;4(2):20–9. http://dx.doi.org/10.1109/MPAE.2006.1597990.
[13] Szkuta B, Sanabria L, Dillon T. Electricity price short-term forecasting using artificial neural networks. IEEE Trans Power Syst 1999;14(3):851–7. http://dx.doi.org/10.1109/59.780895.
[14] Catalão JPS, Mariano SJPS, Mendes VMF, Ferreira LAFM. Short-term electricity prices forecasting in a competitive market: a neural network approach. Electr Power Syst Res 2007;77(10):1297–304. http://dx.doi.org/10.1016/j.epsr.2006.09.022.
[15] Xiao L, Shao W, Yu M, Ma J, Jin C. Research and application of a hybrid wavelet neural network model with the improved cuckoo search algorithm for electrical power system forecasting. Appl Energy 2017;198:203–22. http://dx.doi.org/10.1016/j.apenergy.2017.04.039.
[16] Wang D, Luo H, Grunder O, Lin Y, Guo H. Multi-step ahead electricity price forecasting using a hybrid model based on two-layer decomposition technique and BP neural network optimized by firefly algorithm. Appl Energy 2017;190:390–407. http://dx.doi.org/10.1016/j.apenergy.2016.12.134.
[17] Fan S, Mao C, Chen L. Next-day electricity-price forecasting using a hybrid network. IET Gener Transm Distrib 2007;1(1):176–82. http://dx.doi.org/10.1049/iet-gtd:20060006.
[18] Lin W-M, Gow H-J, Tsai M-T. An enhanced radial basis function network for short-term electricity price forecasting. Appl Energy 2010;87(10):3226–34. http://dx.doi.org/10.1016/j.apenergy.2010.04.006.
[19] Amjady N. Day-ahead price forecasting of electricity markets by a new fuzzy neural network. IEEE Trans Power Syst 2006;21(2):887–96. http://dx.doi.org/10.1109/tpwrs.2006.873409.
[20] Lago J, De Ridder F, De Schutter B. Forecasting spot electricity prices: deep learning approaches and empirical comparison of traditional algorithms. Appl Energy [submitted for publication].
[21] Meeus L, Belmans R. Electricity market integration in Europe. In: Proceedings of the 16th power systems computation conference; 2008.
[22] Bunn DW, Gianfreda A. Integration and shock transmissions across European electricity forward markets. Energy Econ 2010;32(2):278–91. http://dx.doi.org/10.1016/j.eneco.2009.09.005.
[23] de Menezes LM, Houllier MA. Reassessing the integration of European electricity markets: a fractional cointegration analysis. Energy Econ 2016;53:132–50. http://dx.doi.org/10.1016/j.eneco.2014.10.021.
[24] Zachmann G. Electricity wholesale market prices in Europe: convergence? Energy Econ 2008;30(4):1659–71. http://dx.doi.org/10.1016/j.eneco.2007.07.002.
[25] Lindström E, Regland F. Modeling extreme dependence between European electricity markets. Energy Econ 2012;34(4):899–904. http://dx.doi.org/10.1016/j.eneco.2012.04.006.
[26] Ziel F, Steinert R, Husmann S. Forecasting day ahead electricity spot prices: the impact of the EXAA to other European electricity markets. Energy Econ 2015;51:430–44. http://dx.doi.org/10.1016/j.eneco.2015.08.005.
[27] Panapakidis IP, Dagoumas AS. Day-ahead electricity price forecasting via the application of artificial neural network based models. Appl Energy 2016;172:132–51. http://dx.doi.org/10.1016/j.apenergy.2016.03.089.
[28] Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res 2003;3:1157–82.
[29] Carta JA, Cabrera P, Matías JM, Castellano F. Comparison of feature selection methods using ANNs in MCP-wind speed methods. A case study. Appl Energy 2015;158:490–507. http://dx.doi.org/10.1016/j.apenergy.2015.08.102.
[30] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016 <http://www.deeplearningbook.org/> .
[31] Stevenson M. Filtering and forecasting spot electricity prices in the increasingly

deregulated australian electricity market. In: QFRC research paper series, no. 63. Quantitative Finance Research Centre, University of Technology, Sydney; 2001. < http://www.qfrc.uts.edu.au/research/research_papers/rp63.pdf > .
[32] Rodriguez CP, Anders GJ. Energy price forecasting in the Ontario competitive power system market. IEEE Trans Power Syst 2004;19(1):366–74. http://dx.doi.org/10.1109/TPWRS.2003.821470.
[33] Hong Y, Wu C. Day-ahead electricity price forecasting using a hybrid principal component analysis network. Energies 2012;5(11):4711–25. http://dx.doi.org/10.3390/en5114711.
[34] Amjady N, Keynia F. Day-ahead price forecasting of electricity markets by mutual information technique and cascaded neuro-evolutionary algorithm. IEEE Trans Power Syst 2009;24(1):306–18. http://dx.doi.org/10.1109/tpwrs.2008.2006997.
[35] Amjady N, Daraeepour A, Keynia F. Day-ahead electricity price forecasting by modified relief algorithm and hybrid neural network. IET Gener Transm Distrib 2010;4(3):432–44. http://dx.doi.org/10.1049/iet-gtd.2009.0297.
[36] Keles D, Scelle J, Paraschiv F, Fichtner W. Extended forecast methods for day-ahead electricity spot prices applying artificial neural networks. Appl Energy 2016;162:218–30. http://dx.doi.org/10.1016/j.apenergy.2015.09.087.
[37] Ghasemi A, Shayeghi H, Moradzadeh M, Nooshyar M. A novel hybrid algorithm for electricity price and load forecasting in smart grids with demand-side management. Appl Energy 2016;177:40–59. http://dx.doi.org/10.1016/j.apenergy.2016.05.083.
[38] Abedinia O, Amjady N, Zareipour H. A new feature selection technique for load and price forecast of electrical power systems. IEEE Trans Power Syst 2017;32(1):62–74. http://dx.doi.org/10.1109/TPWRS.2016.2556620.
[39] Ruder S. An overview of gradient descent optimization algorithms; 2016. Available from: 1609.04747.
[40] Shafie-Khah M, Moghaddam MP, Sheikh-El-Eslami M. Price forecasting of day-ahead electricity markets using a hybrid forecast method. Energy Convers Manage 2011;52(5):2165–9. http://dx.doi.org/10.1016/j.enconman.2010.10.047.
[41] Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. J Global Optim 1998;13(4):455–92. http://dx.doi.org/10.1023/A:1008306431147.
[42] Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for hyper-parameter optimization. In: Advances in neural information processing systems; 2011. p. 2546–54. < http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization > .
[43] Hutter F, Hoos HH, Leyton-Brown K. Sequential model-based optimization for general algorithm configuration. International conference on learning and intelligent optimization Springer; 2011. p. 507–23. http://dx.doi.org/10.1007/978-3-642-25566-3_40.
[44] Hutter F, Hoos H, Leyton-Brown K. An efficient approach for assessing hyperparameter importance. In: Proceedings of the 31st international conference on machine learning. ICML'14, vol. 32; 2014. p. 754–62. < http://proceedings.mlr.press/v32/hutter14.pdf > .
[45] Makridakis S. Accuracy measures: theoretical and practical concerns. Int J Forecast 1993;9(4):527–9. http://dx.doi.org/10.1016/0169-2070(93)90079-3.
[46] Diebold FX, Mariano RS. Comparing predictive accuracy. J Bus Econ Stat 1995;13(3):253–63. http://dx.doi.org/10.1080/07350015.1995.10524599.
[47] LeCun Y, Bottou L, Orr GB, Müller K-R. Efficient BackProp. In: Orr GB, Müller K-R, editors. Neural networks: tricks of the trade Lecture Notes in Computer Science, vol. 1524. Berlin, Heidelberg: Springer; 1998. p. 9–50. http://dx.doi.org/10.1007/3-540-49430-8_2.
[48] RTE, Grid data. < https://data.rte-france.com/ > [accessed on 15.05.2017].
[49] Elia, Grid data. < http://www.elia.be/en/grid-data/dashboard > [accessed on 15.05.2017].
[50] ENTSO-E transparency platform. < https://transparency.entsoe.eu/ > [accessed on 15.05.2017].
[51] Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML); 2010. p. 807–14. < http://icml2010.haifa.il.ibm.com/papers/432.pdf > .
[52] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the international conference on artificial intelligence and statistics (AISTATS'10). Society for Artificial Intelligence and Statistics; 2010. p. 249–56.
[53] Kingma DP, Ba J. Adam: a method for stochastic optimization; 2014. Available from: 1412.6980.
[54] Yao Y, Rosasco L, Caponnetto A. On early stopping in gradient descent learning. Constr Approx 2007;26(2):289–315. http://dx.doi.org/10.1007/s00365-006-0663-2.
[55] Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks? In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, editors. Advances in neural information processing systems 27. Curran Associates, Inc.; 2014. p. 3320–8 <https://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks> .
[56] Jaderberg M, Mnih V, Czarnecki WM, Schaul T, Leibo JZ, Silver D, et al. Reinforcement learning with unsupervised auxiliary tasks; 2016. Available from: 1611.05397.
[57] Li X, Zhao L, Wei L, Yang M-H, Wu F, Zhuang Y, et al. DeepSaliency: multi-task deep neural network model for salient object detection. IEEE Trans Image Process 2016;25(8):3919–30. http://dx.doi.org/10.1109/TIP.2016.2579306.
[58] Bergstra J, Yamins D, Cox DD. Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In: Proceedings of the 30th international conference on machine learning; 2013. p. 115–23. < http://proceedings.mlr.press/v28/bergstra13.pdf > .
[59] 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.
[60] Hong T. Crystal ball lessons in predictive analytics. EnergyBiz 2015;12(2):35–7.
[61] Uniejewski B, Nowotarski J, Weron R. Automated variable selection and shrinkage for day-ahead electricity price forecasting. Energies 2016;9(8):621. http://dx.doi.org/10.3390/en9080621.
[62] Wang J, Zhong H, Ma Z, Xia Q, Kang C. Review and prospect of integrated demand response in the multi-energy system. Appl Energy 2017;202:772–82. http://dx.doi.org/10.1016/j.apenergy.2017.05.150.
[63] Nolan S, O'Malley M. Challenges and barriers to demand response deployment and evaluation. Appl Energy 2015;152:1–10. http://dx.doi.org/10.1016/j.apenergy.2015.04.083.
[64] Wang Q, Zhang C, Ding Y, Xydis G, Wang J, Østergaard J. Review of real-time electricity markets for integrating distributed energy resources and demand response. Appl Energy 2015;138:695–706. http://dx.doi.org/10.1016/j.apenergy.2014.10.048.