Politechnika Wrocławska

FIELD OF SCIENCE: Social Sciences

DISCIPLINE OF SCIENCE: Management and Quality Studies

DOCTORAL DISSERTATION

Deep learning in point, probabilistic and ensemble forecasting of electricity prices

Grzegorz Jarosław Marcjasz

Supervisor/Supervisors:
prof. dr hab. inż. Rafał Weron

Keywords: Decision-making, Electricity price forecasting, Machine learning, Probabilistic forecasting, Ensemble forecasting

WROCŁAW 2023

Abstract

This doctoral thesis explores the applicability of deep learning for various electricity price forecasting tasks, from the standpoint of decision-makers. It provides evidence that – when carefully calibrated – deep neural network models consistently demonstrate superior performance in point, probabilistic and ensemble forecasting compared to state-of-the-art benchmarks. Finally, the thesis also highlights the importance of following the best practices, as robust comparisons and replicability are key to research excellence.

Streszczenie

Niniejsza rozprawa doktorska bada możliwości wynikające ze stosowania uczenia głębokiego do prognozowania cen energii elektrycznej z punktu widzenia decydentów. Dostarcza dowodów na to, że – przy starannej kalibracji – modele głębokich sieci neuronowych konsekwentnie wykazują wyższą dokładność w prognozowaniu punktowym, probabilistycznym i trajektorii w porównaniu z wymagającymi benchmarkami. Rozprawa podkreśla również znaczenie przestrzegania najlepszych praktyk, ponieważ wiarygodne porównania i powtarzalność są kluczem do doskonałości badań naukowych.

Contents

Chapter 1

Introduction

1.1 Background

Electricity price forecasting (EPF)[1] is a branch of predictive analytics on the interface of computer and information sciences, electrical engineering, finance, and – above all – management science. It centers on predicting electricity prices in wholesale markets, with timeframes ranging from minutes- and hours-ahead for real-time/intraday auctions and continuous trading, through day-ahead auctions, to medium- and long-term horizons, spanning weeks, months, or even years, entailing exchange-traded and over-the-counter futures and forward contracts (Jędrzejewski et al., 2022).

As Hong et al. (2020) emphasize, the energy industry relies on forecasters to predict load, generation, and prices. These forecasts are then being employed for planning and operations by all business entities involved in the generation, distribution and transmission of electrical energy. While long-term predictions have been used for more than a century, modern energy forecasting literature focuses on *intraday* (ID) and *day-ahead* (DA) markets (Petropoulos et al., 2024). So does this thesis.

The emergence of EFP as a dynamic and interdisciplinary research field traces back to the early 1990s, coinciding with the deregulation of traditionally monopolistic and government-controlled power systems (Mayer and Trück, 2018). Until a decade ago, the models primarily included relatively parsimonious (linear) regression models and (artificial) neural networks, like those illustrated in Fig. 1.1. Such models were built on expert knowledge and predicted the price $P_{d,h}$ on day d and hour h using such *explanatory variables* (*inputs*, *predictors*) as past electricity prices, day-ahead predictions $X_{d,h}$ of exogenous variable(s), e.g., load forecasts, and calendar effects, e.g., dummies to represent weekly seasonality and holidays (Dudek, 2016; Keles et al., 2016; Weron,

[1] In this thesis, 'EPF' will denote both *electricity price forecasting* and *electricity price forecast*, while the plural form, *forecasts*, will be succinctly represented as 'EPFs'.

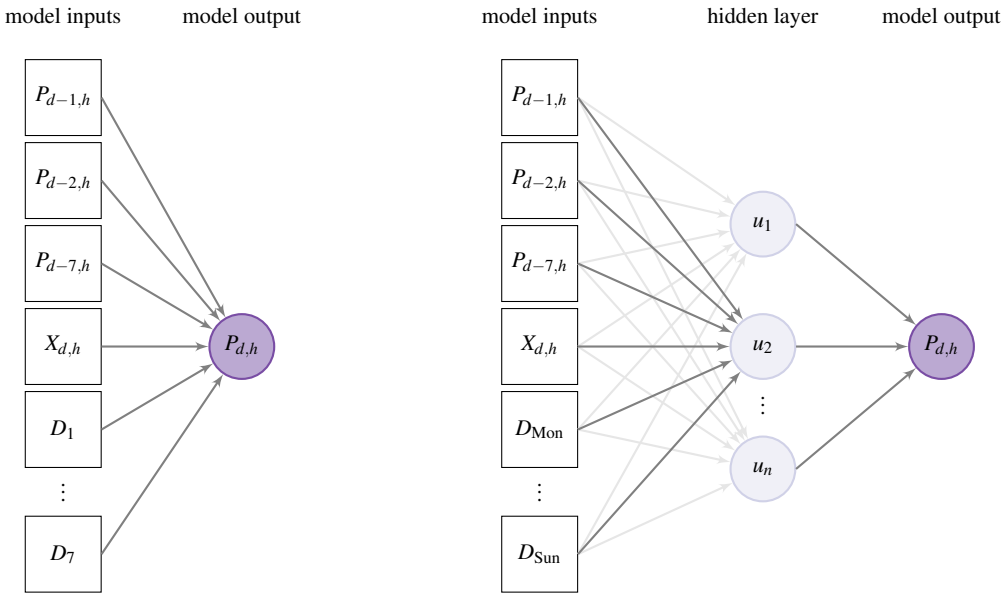


Figure 1.1: Illustration of a neural network equivalent to a parsimonious EPF linear regression (*left panel*) with arrows corresponding to the β coefficients in Eqn. (1.1), and a feedforward neural network with the same inputs and one hidden layer (*right panel*). White squares represent individual explanatory variables (inputs, predictors), while the purple circle is the output (predicted) variable. Based on Figure 1 in Jędrzejewski et al. (2022).

2014). If no hidden layers are present, such a model can be written in an equation form:

$$P_{d,h} = \underbrace{\beta_1 P_{d-1,h} + \beta_2 P_{d-2,h} + \beta_3 P_{d-7,h}}_{\text{autoregressive effects}} + \underbrace{\beta_4 X_{d,h}}_{\text{exog. variable}} + \underbrace{\sum_{j=1}^{7} \beta_{h,j+4} D_j}_{\text{weekday dummies}} + \varepsilon_{d,h}. \qquad (1.1)$$

In Figure 1.1, arrows indicate the flow of information, with each arrow having a corresponding coefficient (weight) assigned to it. The output is calculated as a weighted sum of all the inputs. It should also be noted that in such models the price $P_{d,h}$ was predicted independently for each hour h, sometimes using past prices from different hours as explanatory variables. Interestingly, the joint prediction of all 24 hourly prices, e.g., using vector autoregressive models, generally underperforms (Ziel and Weron, 2018), in contrast to the natural gas and crude oil markets (Rubaszek et al., 2020).

As more efficient algorithms, more computing power and more data became available, the models became more complex, eventually leading to the widespread use of *machine learning* (ML) techniques (Maciejowska et al., 2023).[2] Two main streams of EPF approaches have emerged (Lago et al., 2021): (linear) regression models estimated using

[2] As Januschowski et al. (2020) argue, *statistical learning* and *machine learning* are synonyms. These

shrinkage and selection operators and (nonlinear) deep learning techniques. A typical representative of the first group is the *LASSO-Estimated AutoRegressive* (LEAR) model proposed in Lago et al. (2021), i.e., **Paper 2**, as one of two challenging benchmarks for contemporary EPF. The LEAR model is a parameter-rich autoregressive structure with ca. 250 explanatory variables:

$$\begin{aligned} P_{d,h} = &\sum_{i=1}^{24} \left(\beta_{h,i} P_{d-1,i} + \beta_{h,i+24} P_{d-2,i} + \beta_{h,i+48} P_{d-3,i} + \beta_{h,i+72} P_{d-7,i}\right) \\ &+ \sum_{i=1}^{24} \left(\beta_{h,i+96} X^1_{d,i} + \beta_{h,i+120} X^1_{d-1,i} + \beta_{h,i+144} X^1_{d-7,i}\right) \\ &+ \sum_{i=1}^{24} \left(\beta_{h,i+168} X^2_{d,i} + \beta_{h,i+192} X^2_{d-1,i} + \beta_{h,i+216} X^2_{d-7,i}\right) \\ &+ \sum_{k=1}^{7} \beta_{h,240+k} D_k + \varepsilon_{d,h}, \end{aligned} \tag{1.2}$$

where $X^1_{d,h}$ and $X^2_{d,h}$ correspond to the first (e.g., load forecast) and second (e.g., wind generation forecast) exogenous forecast series available for the dataset, estimated using the *least absolute shrinkage and selection operator* (LASSO) of Tibshirani (1996). Although other regularization algorithms (e.g., ridge regression, elastic nets) have been considered for EPF, the LASSO has become the golden standard (Janke and Steinke, 2019; Marcjasz, 2020; Narajewski and Ziel, 2020; Özen and Yıldırım, 2021; Uniejewski et al., 2016; Ziel, 2016; Ziel and Weron, 2018).

Advances in computational resources, including massive usage of graphics processing units (GPUs) and more efficient optimization algorithms, have made it possible to train complex structures and paved the way for the second stream of EPF approaches (Jędrzejewski et al., 2022; Maciejowska et al., 2023). While this group is very diverse, its most prominent representative is a *deep neural network* (DNN) with two or more hidden layers, hundreds of nodes and thousands of links. A sample two hidden layer model – dubbed DNN – was proposed in Lago et al. (2021), i.e., **Paper 2**, as the second challenging benchmark for contemporary EPF. It uses the same inputs as the LEAR model in Eqn. (1.2), but admits a much more complex dependence structure between the inputs and the output(s). As a result, it outperforms the LEAR benchmark significantly.

Deep neural networks have consistently demonstrated superior forecasting results and generalization abilities, see Lago et al. (2018a) for a comprehensive comparison of point EPF performance. However, their widespread use has been hindered by bad practices, which lead to meaningless results that cannot be compared or compromise research reproducibility, problems with interpreting the estimated weights and network

terms originated in different environments, but there are no fundamental differences in the methods assigned to either class.

connections, as well as unproven performance in *probabilistic* (yielding the whole predictive distribution) and *trajectory* (also called *ensemble*) EPF.

1.2 Aim and objectives

It is exactly the aim of this thesis to address the above mentioned issues and develop robust, reliable and – when possible – interpretable DNN-based approaches for short-term point, probabilistic and ensemble forecasting of electricity prices. Approaches that significantly outperform regression-based predictions not only in terms of statistical error measures, but also in terms of economic benefits for business entities involved in intraday (ID) and day-ahead (DA) trading. To address this aim, four objectives are set:

- **Objective 1:** Identify the most common problems encountered in EPF machine learning research, present a set of best practices and publish open access codes for well-performing benchmark models.
- **Objective 2:** Develop an interpretable DNN model for point EPF that outperforms state-of-the-art benchmarks.
- **Objective 3:** Construct distributional DNNs that directly yield predictive distributions and are superior to state-of-the-art probabilistic models in terms of both statistical and economic measures.
- **Objective 4:** Develop a decision support method that uses distributional DNNs to generate trajectories of ID prices, then use it to construct profitable trading strategies.

The first objective focuses on improving the framework for point forecasting and lays the foundations for probabilistic and ensemble forecasting. The two papers that pursue this objective use automatic hyper-parameter optimisation schemes – **Paper 1** proposes a robust grid-search based procedure, while **Paper 2** uses Bayesian optimization. Both schemes have forecast averaging at their core, as it is no longer just a useful addition to the modelling pipeline, but can be seen as more of a necessity for deep learning models (Karabiber and Xydis, 2019). In addition, **Paper 2** provides a comprehensive literature review, identifies the most common issues and – most importantly – proposes an open-access benchmark comprising of two state-of-the-art models (LEAR, DNN) together with their predictions for 5 distinct datasets.

The second objective addresses one of the major drawbacks of neural network modeling, i.e., lack of interpretability. Typically, neural networks are so-called black-boxes – they do not "explain" the result, but only provide the expected outcome based on the input data. They are often shown to perform well, but – as Rudin (2019) argues – they are not safe to use in high-stakes decisions. Building on the NBEATS architecture of Oreshkin et al. (2020, 2021), **Paper 3** proposes a novel deep learning approach that admits (partial) interpretability and outputs the components of the electricity price –

trend, seasonality and exogenous factors. Such a property becomes especially important when considering the need for transparent and understandable models expressed in public discussions about machine learning (Carvalho et al., 2019).

The third objective concentrates on generating predictive distributions directly from DNNs, without the need for bootstrapping or quantile regression averaging that process point forecasts to produce probabilistic predictions (Nowotarski and Weron, 2018). Although distributional neural networks are not a new concept (Williams, 1996), they have seen very limited application in energy forecasting and only in combination with complex architectures such as convolutional neural networks and gated recurrent units (CNN, GRU; Afrasiabi et al., 2020) or recurrent neural networks (RNN; Mashlakov et al., 2021). The *distributional deep neural network* (DDNN) proposed in **Paper 4** is a multi-layer perceptron in which the information propagates only forward, i.e., it is far less complex than the CNNs, GRUs and RNNs, easier to interpret and less computationally demanding.

The fourth objective focuses on predicting price trajectories in continuous trading intraday markets and generating optimal buy/sell signals for managers of short-term energy portfolios. The time and price at which the trader enters the market plays a crucial role in generating profits. Trajectory (or ensemble) forecasting provides multiple instances of the future price trajectory, all modelled to reflect time dependence. **Paper 5** builds on the recent concept of Serafin et al. (2022) that uses a Gaussian copula to model temporal dependencies (Pinson et al., 2009). The proposed DDNN-based approach significantly improves upon the results of state-of-the-art ensemble methods with regards to almost every considered metric.

1.3 Contribution to the discipline of Management and Quality Sciences

Griffin (2021) defines forecasting as the process of developing assumptions or premises about the future. It can be used by managers in planning or decision-making. The decision-making is a data-driven procedure that involves selecting one option from a set of possibilities and consists of several steps: recognizing and defining the problem, identifying alternative possibilities, choosing the best option, and implementing it (Heizer et al., 2008). Even though forecasting is not mentioned directly in the six steps outlined by Heizer et al., the whole process requires the decision-maker to rely on forecasts and make an effort to improve their accuracy, which aligns with the assertion of Petropoulos et al. (2022) that forecasting has always been at the forefront of decision making and planning.

In fact, the field of forecasting can be placed in the realm of *predictive analytics*, which, together with descriptive and prescriptive analytics, forms so-called *business*

analytics (Lepenioti et al., 2020). This integration of diverse analytical approaches empowers organizations to gain insights from data, make informed decisions, and proactively plan for the future. The field of business analytics focuses on aiding the decision-making processes by providing insights and understanding of business performance based on data. In other words, it aims to prescribe a recommended action (through the use of optimization) for the predicted future events (Davenport and Harris, 2007; Delen and Ram, 2018) in order to achieve an optimal outcome according to some (financial) metric.

This thesis develops state-of-the-art deep neural network based approaches for point (**Paper 1** – **Paper 3**), probabilistic (**Paper 4**) and ensemble (**Paper 5**) electricity price forecasting tailored to the day-ahead and intraday markets. This is of practical importance. As Kraus et al. (2020) argue in a recent review article advocating the use of deep learning in operations research, customised architectures add value compared to the standard out-of-the-box approaches.

As the empirical studies in **Paper 4** and **Paper 5** show, the obtained forecasts can be used in decision-making processes where their accuracy is directly linked to operational profits. This is in line with recent trends in management science (Bertsimas and Kallus, 2020). Qi et al. (2020), who review the use of machine learning, note that it is being used extensively in prediction (e.g., of customer behavior), but also in the development of learning frameworks that integrate prediction and optimisation. By establishing a direct link between improvements in forecasting accuracy and the financial performance of the company using the forecasts, we see the forecasting models as decision support tools that help the company achieve its primary objective – generating profits.

1.4 Thesis structure

The remainder of the thesis is structured as follows. In Chapter 2, I introduce the day-ahead and intraday markets for electricity. Then, in Chapter 3 I review the basics of forecasting with neural networks, including the training process, network architecture, calibration windows, overfitting and hyperparameter optimization. In Chapter 4 I discuss the main results of the five papers that constitute the thesis (referred to as **Paper 1** – **Paper 5**; the full texts can be found in the Appendix). Next, in Chapter 5 I briefly describe the articles I have published during my undergraduate and graduate studies that are not part of the thesis. Finally, in Chapter 6 I summarize the key findings and conclude.

Chapter 2

The day-ahead and intraday markets for electricity

Since the deregulation of the electricity markets in the 1990s, a significant fraction of energy is traded at organized power exchanges utilizing *day-ahead* (DA) auctions (Mayer and Trück, 2018). Every morning participants place their supply and demand bids for each of the load periods (typically 24 hours) of the next day (Maciejowska et al., 2023). Around noon on the day preceding the delivery the market clearing prices are determined at the intersection of the supply and demand curves for each period, see Fig. 2.1.

The electricity prices are very volatile. The current technical limits for most European markets (e.g., markets operated by EPEX, including Germany and France, as well as the Iberian markets operated by OMIE) allow for the price to range from -500 to 4000 EUR. Interestingly, negative prices are sometimes observed (De Vos, 2015; Zhou et al., 2016). They reflect the technical limitations of ramping the generation down or up in coal-fired power plants. It may be less expensive for the plant operators to pay wholesale buyers to consume the electricity generated than to stop and restart the production a few hours later (Schneider, 2011). Negative prices are most commonly observed whenever there is a decrease in demand (e.g., weekends and holidays) or there is an abundance of electricity from *renewable energy sources* (RES); e.g., strong winds and high solar radiation. On the other hand, the price spikes occur when there is an atypically high demand, or the generation from renewable sources is scarce. Then, the most costly generation units are used to match demand. Additionally, the consumption patterns are seasonal (Dudek, 2023). The type of dependency is conditional on factors that vary from one market to another, i.e., the prevalence of air conditioned buildings, the climate, electric vehicle adoption, etc. For example, in the US-based Pennsylvania-New Jersey-Maryland (PJM) market the extreme prices can be observed at different loads depending on the season, see Fig. 2.2 – additionally, the magnitude of price spikes differs significantly between the winter and summer seasons.

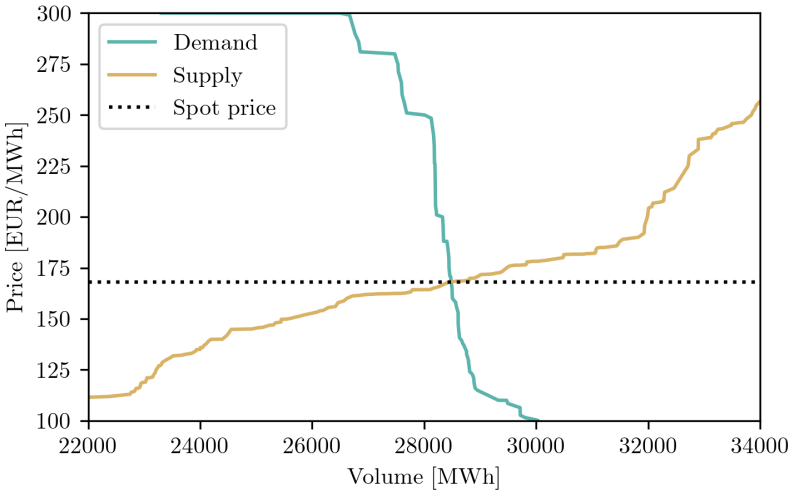


Figure 2.1: Aggregated demand and supply curves on 1 December 2022 for hour 18 in the Iberian electricity market; for better visibility plotted only for the 22000-34000 MWh range. The hourly spot price is at the intersection of the curves. Data source: OMIE.

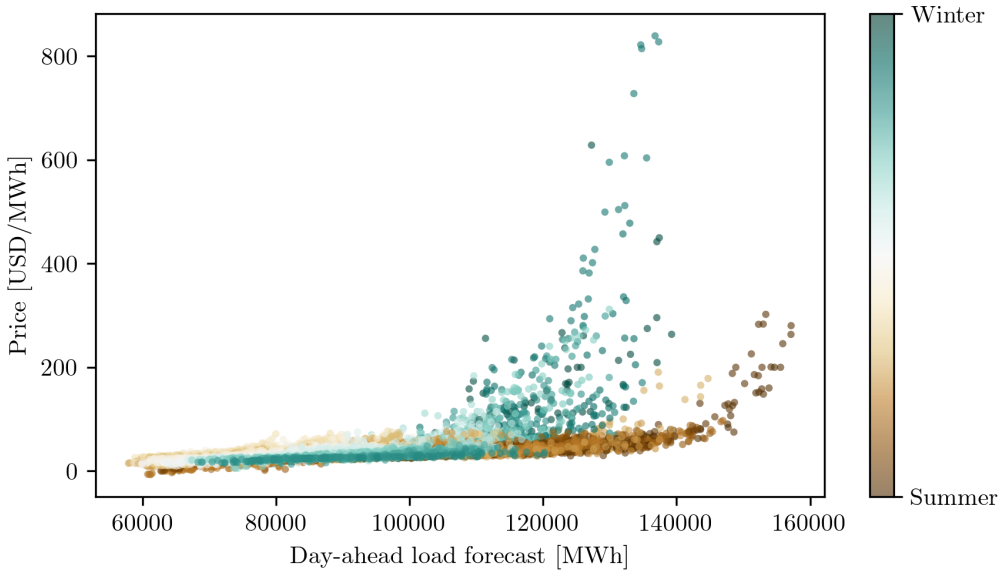


Figure 2.2: A scatter plot of hourly electricity price versus day-ahead system load forecast in the COMED zone (Northern Illinois, US) of the PJM market for years 2013 and 2014. Color marks the season (brown → summer, teal → winter, and lighter shades for the transitional seasons). Data source: PJM.

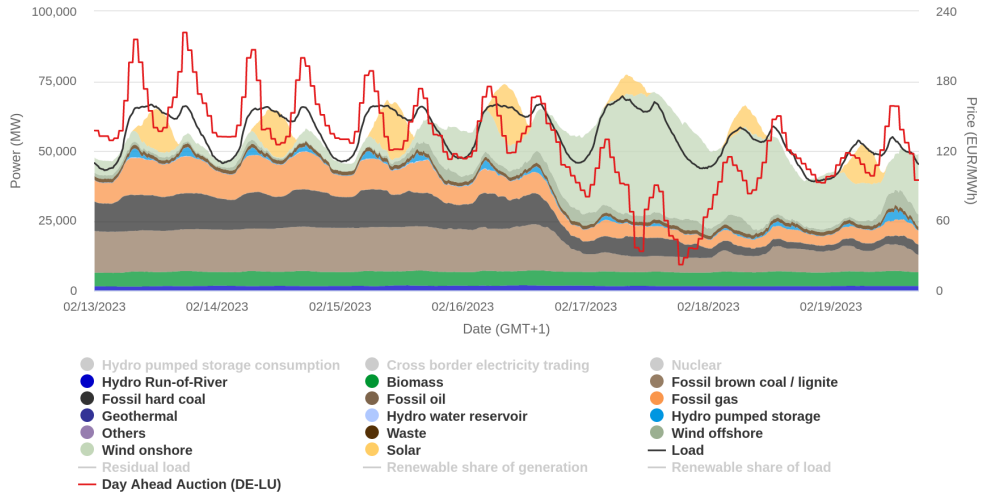


Figure 2.3: The generation profile and hourly day-ahead electricity prices (red line) in week 7 of 2023 in the German market. Source: Energy-Charts.

However, the DA auction is not sufficient for a smooth operation of a power system (Kraft et al., 2023). With the growing installed capacity of RES, the uncertainty of the actual output for the day-ahead delivery is very high, especially in periods of transition from low to high (or high to low) wind (Maciejowska, 2020). Based on the data publicly available on the ENTSO-E Transparency Platform at the time of writing the thesis, the installed capacity (the highest possible generation, under ideal conditions) of German wind power plants exceeds 60 GW, while the typical consumption (load) in peak hours is in the 60-80 GW range. On top of the wind power plants, photovoltaics provide almost another 60 GW of installed capacity – so RES can cover over 100% of the demand if the weather allows for that. However, there are also days with almost no electricity generated by RES, and the forecasts of the production for the next day are often highly uncertain (Gianfreda et al., 2020; Maciejowska et al., 2023). For example, Figure 2.3 depicts the generation mix throughout week 7 of year 2023 in Germany with the hourly DA prices. Note the very low renewable generation in the beginning of the week, followed by a high wind power production on Friday and its effect on the price.

This creates a need for market participants to adjust their day-ahead bids whenever new, more accurate weather information is available. *Intraday* (ID; or *real-time*) markets enable to trade electricity much closer to the delivery, even just minutes before (Narajewski and Ziel, 2020; Uniejewski et al., 2019a). For instance, in the EPEX market for Germany the DA auction results are available at ca. 12:45 on the day preceding the delivery (i.e., day $d-1$), then at 15:00 the ID auction ends, and at 16:00 the so-called ID continuous trading starts and runs until a few minutes before delivery on day d. As the trading is no longer auction-based, each bid must be matched with the other side of the

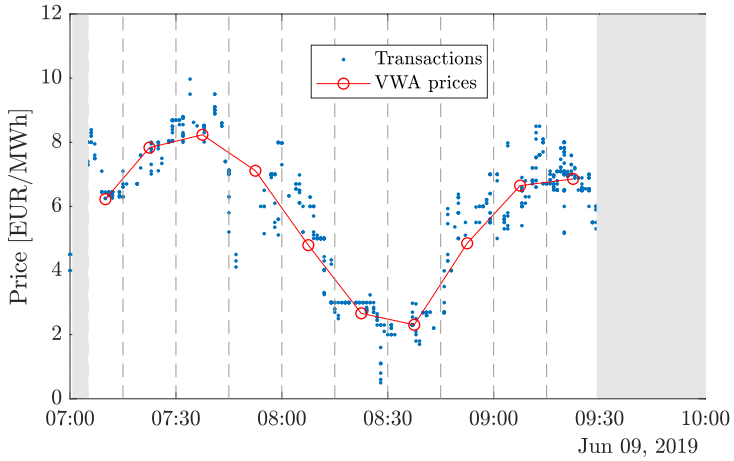


Figure 2.4: Illustration of the volume weighed average price trajectory (VWA; red circles) on the German intraday market, constructed from individual transaction prices (blue dots) for the hourly product with delivery starting at 10:00 on 9 June 2019. Source: Serafin et al. (2022).

trade – and there no longer is one price for the electricity delivered over a given period. A trajectory of prices can be observed for each hourly, half-hourly or quarter-hourly (depending on the product traded) load period (see Figure 2.4 for an illustration). The trajectory reflects the changing expectation of the price – based on the changes in the demand, weather conditions and unplanned power plant outages. In a scenario where a generator, a utility or an energy trader needs to balance out the day-ahead bids in the ID market, profit maximization (or cost minimization) turns into a problem of finding the optimal time to enter or exit the market.

The ID market, however, is not the closest to delivery – there is also a balancing market, which is a technical "safety net" that operates nearly on-line (Kraft et al., 2020; Weron, 2014). It exists to ensure that the demand is met by the supply at all times. The trading in the balancing market is very limited (and the fees are prohibitive), as it prioritizes energy system stability over economic incentives. Moreover, on the demand side there is also a mechanism of *demand response*, which can take the form of a contract that pays the energy consumer (e.g., a factory) for the reduction of the consumption on demand and for the readiness to do so. However, as these systems are targeting technical issues, they are of lesser interest to the decision makers from the perspective of scheduling and planning operations. Hence, the thesis focuses on the day-ahead and intraday markets only.

Chapter 3

A primer on neural networks

This section presents an overview of the challenges faced when using neural networks (NN) for forecasting electricity prices. It briefly covers the topics of training (or parameter estimation), network architecture, typical models for the day-ahead electricity markets, rolling calibration windows, overfitting and hyperparameter optimization.

3.1 Training

Let us assume that we have S samples of V independent variables (matrix $X_{S\times V}$) and the corresponding dependent variables (matrix $Y_{S\times 1}$). Then, a linear model can be formulated as:

$$y = \beta_0 + \sum_{i=1}^{V} \beta_i x_i + \varepsilon, \tag{3.1}$$

where x_i corresponds to the i-th independent variable, β_0 is the intercept, β_i is the coefficient for the i-th variable and ε is the error term. Typically, the β's are derived by ordinary least squares (OLS). However, without changing the formulation of the model, one could also construct a neural network to estimate or *train* them; the latter term is typically used when referring to the process of updating the weights in a neural network. This is illustrated in Fig. 3.1, where each of the arrows between the input and the output layer corresponds to a coefficient. In such a case, the β's would be iteratively updated based on the numerically computed gradients of the errors (Haykin, 1994); to mimic OLS, the mean squared error would be a good choice, but NN models are flexible with regards to the loss functions. An optimization algorithm, such as ADAM (Kingma and Ba, 2014) is used to govern the coefficient updates. Due to an iterative approach and randomized initial coefficients, the result may be different from that of the OLS estimation. It may even differ from one estimation of the network's parameters to the next.

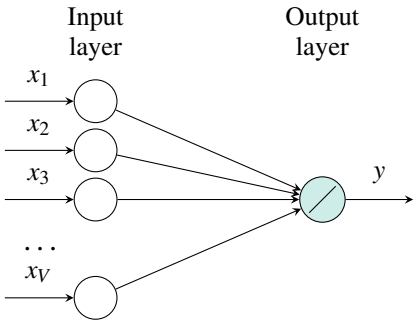


Figure 3.1: An illustration of a neural network equivalent to linear regression.

Since the training process is iterative, it can be divided into epochs that consist of steps. In each iteration, the network updates its weights once, based on the gradients computed for a subset (also called *batch*) of data. One full pass of the data sample is called an *epoch*. For example, if we set the training process to be 10 epochs long, we have $S = 360$ samples and use batches of 32 data points, we would update the weights $\lceil 360/32 \rceil = 12$ times per epoch and 120 times in total. However, determining the optimal number of epochs is a problem in itself. Typically an early stopping criterion is used to avoid overfitting (Goodfellow et al., 2016), see also Section 3.5.

3.2 Network architecture

The linear example depicted in Fig. 3.1 shows a network in which the computation takes place only in one neuron: the output one. Mathematically, given a set of inputs x_i, a trained network with weights w_i, i.e., coefficients equivalent to β_i in Eqn. (3.1), computes the following:

$$y = \sum_{i=1}^{V} w_i x_i, \tag{3.2}$$

with the intercept – or *bias* in NN nomenclature – omitted here for the sake of simplicity. Note, that the neuron in general computes a function of the sum of weighted inputs – in this example an identity. This is also true for more complex structures – a neural network with more layers (or with "wider" ones, i.e., consisting of more neurons) operates with the same principal operations, what changes is the number of model parameters and the dependencies between neuron outputs.

For example, we can add another layer to the network, between the inputs and the outputs; such layers are called *hidden*, as they perform computations that are not directly visible to the user. Let us assume that the hidden layer has 3 (hidden) neurons and that the *activation* function, i.e., the function applied to the sum of weighted inputs in a

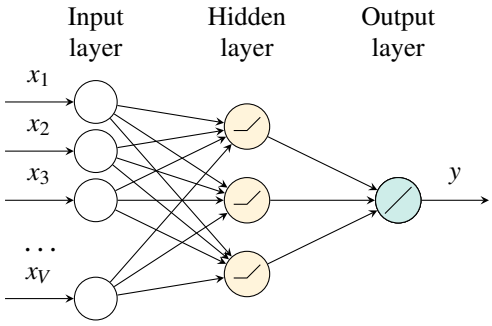


Figure 3.2: An illustration of a neural network with a single hidden layer.

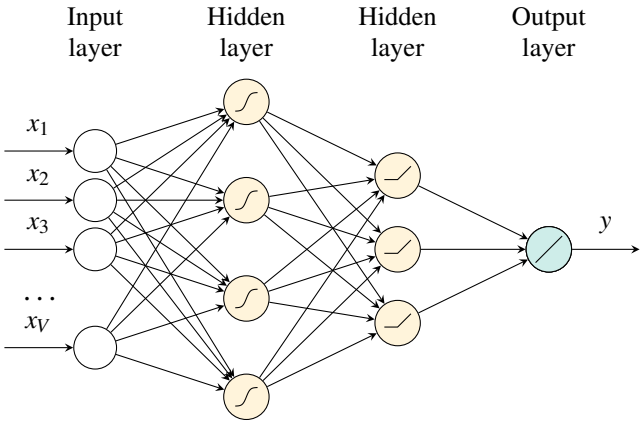


Figure 3.3: An illustration of a *deep neural network* (DNN) with two hidden layers – one with sigmoid and the second with ReLU activation functions; note the different icons inside the neurons.

neuron, is the *rectified linear unit*: $\text{ReLU}(x) = \max(0,x)$, see Fig. 3.2. Then the output from a trained network will have a following structure:

$$y = \sum_{j=1}^{3} u_j \,\text{ReLU}\left(\sum_{i=1}^{V} w_{i,j} x_i\right), \tag{3.3}$$

where $w_{i,j}$ is the weight on a connection between the i-th input and the j-th hidden neuron and u_j is the weight on a connection between the j-th hidden neuron and the output. By adding another hidden layer, we obtain a (relatively simple) *deep neutral network* (DNN), where the non-linearities are stacked on top of non-linearities. An example of such a network is presented in Fig. 3.3.

Despite the more complex architecture, from a forecaster's perspective, the DNN in Fig. 3.3 is as simple to use as the (shallow) NN with one hidden layer in Fig. 3.2, and even as the NN regression in Fig. 3.1. The only differences would be in the network definition and the computational complexity. The models would behave differently and

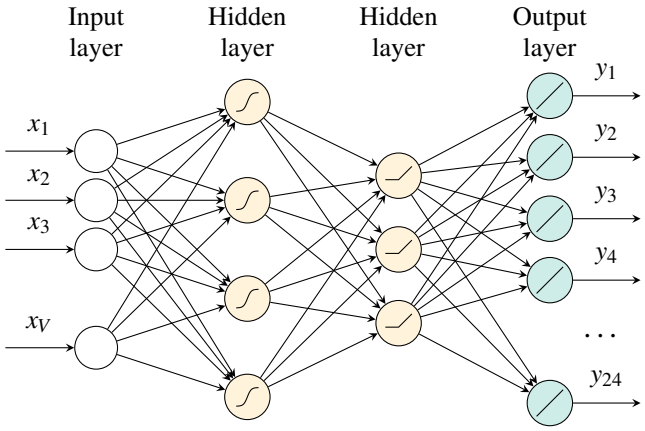


Figure 3.4: An illustration of a *deep neural network* (DNN) with two hidden layers and 24 outputs.

possibly require different hyperparameters, but the training would follow the same rules for all variants.

3.3 Models for day-ahead electricity markets

The majority of day-ahead electricity market designs involve hourly auctions that are resolved simultaneously for all 24 hours of the next day (Mayer and Trück, 2018). This opens up multiple modeling possibilities (Ziel and Weron, 2018):

- treating each hour of the day as a separate time series and estimating them independently, hence not taking into account the joint distribution of the innovations for different hours → this approach is often called *multivariate*;
- treating all hours of the day as one time series and building a model with one output that is applied to all 24 hours of the day → *univariate* approach;
- creating a single model with 24 outputs, hence explicitly or implicitly modeling the joint distribution of the innovations for different hours → *multi-output* or *fully multivariate* approach.

The choice of the best modeling framework is not trivial. For instance, Cuaresma et al. (2004) and Ziel and Weron (2018) argue that the multivariate approach outperforms its univariate counterpart and vector autoregressive (VAR) multi-output models. However, they do not consider multi-output NN-based models. The latter have three advantages (Nargale and Patil, 2016; Lago et al., 2018a; Bento et al., 2018):

- the joint distribution of the innovations for different hours is (implicitly) modeled;
- training one larger model is faster than training 24 smaller ones;

- the modeled dependencies (coefficients) in one part of the network generalize, i.e., try to fit in the best possible way all outputs, and in the another part try to fit data in a way that makes each hourly output unique.

To better understand the third point, the reader is referred to Fig. 3.4. Each weight affects all 24 outputs, and the only factor that differentiates the outputs is the last layer of connections. This means, that the first two layers of connections are, as part of the training process, trying to fit the weights in a way that is best for the general behavior of the model (minimize the average error across all hourly outputs) and the last layer of connections tries to specialize the output neurons for the respective hours.

3.4 Rolling calibration windows

As the conditions in electricity markets change rapidly, it is important to recalibrate models with updated data; for day-ahead predictions this is typically done once per day. At the same time, the oldest data in the training sample get less and less relevant. Therefore, 'rolling' the window is preferred over 'expanding' it. Formally, the rolling calibration window scheme consists of the following steps (note that prices for day d were determined on $d-1$):

- estimate the model using S samples of data for days $\{d-S, d-S+1, ..., d-1\}$,
- use the estimated model to forecast prices for day d.

The computational complexity is significant, i.e., for a one year long test period we have to train 365 daily or 8760 hourly models. However, compared to a model trained only once and reused for every future date, the rolling calibration window scheme should allow the model to reflect the current market situation much better. Moreover, from the perspective of the computational time, even the more complex approaches proposed in the literature take at most minutes to recalibrate on a typical desktop, leaving enough time for the decision maker to act.

3.5 Overfitting and methods that prevent it

When fitting a high-order polynomial to a small number of points, the variability of the fitted curve is too high for any reasonable interpolation or extrapolation. The same holds for more complex models. This phenomenon is called *overfitting* and is particularly dangerous for non-linear models, like neural networks. When a NN fits the training data too well, it no longer is able to generalize to the unseen data, see Fig. 3.5. The mean absolute error (MAE) for the training data (used to adjust the model parameters) decreases over time. However, after an initial decrease, the MAE increases over time

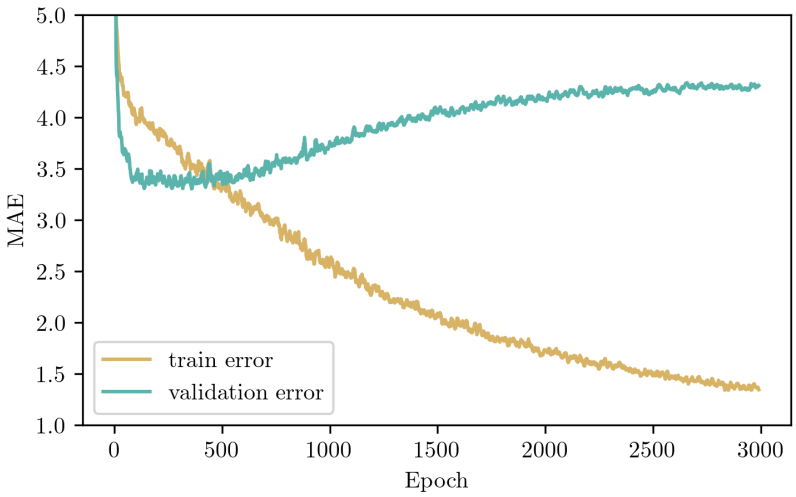


Figure 3.5: The trajectories of the loss metric (mean absolute error, MAE) on sample training and validation power market data.

for the validation (unseen) data. One can clearly identify the point at which the model starts to overfit and loses the ability to generalize (ca. 500 epochs).

In forecasting models, overfitting is addressed by either selecting (as a hyperparameter; see Section 3.6) the length of the training period or by using an early stopping criterion (Yao et al., 2007). Early stopping is the preferred choice as it is more flexible. The optimal length is chosen automatically at each step and can adapt to the changing data patterns. However, early stopping has a downside – it requires setting aside validation data that are not used for the training process.

3.6 Hyperparameter optimization

Hyperparameters are parameters that are not optimized during network training – they have to be set beforehand. The optimal choice of hyperparameters is crucial in achieving a good predictive performance. However, the problem of choosing the right combination gets more complex, the more complex the model is. Starting from the neural network architecture and activation functions, through the choice of the optimization algorithm and its tunables (e.g., for the ADAM algorithm we can set the learning rate and the rates of decay respectively to the first and the second moment estimates), regularization, normalization, dropout, batch size, to the selection of relevant inputs. Typically, some of the hyperparameters are fixed (e.g., only models with two hidden layers are considered), and for the rest of them, an optimization study is performed.

Hyperparameter optimization can be performed in several ways. For instance, hy-

perparameters can be tuned manually (Catalão et al., 2007). In practice, however, such an approach would be very difficult to use as hyperparameters may need to be reestimaed periodically. Alternatively, an automated optimization can be performed (Pravin et al., 2022):

- using a grid search, i.e., a discrete set of possible values is determined for every hyperparameter, and some – or all – combinations are tested on the validation set (Amjady and Keynia, 2009; Panapakidis and Dagoumas, 2016; Marcjasz, 2020);
- using a Bayesian method – such as the tree-structured Parzen estimator – the possible values are determined for each hyperparameter (discrete or continuous) and the algorithm tries new combinations of hyperparameters based on the performance of the combinations tested so far and uncertainty of the estimated performance (Bergstra et al., 2011; Lago et al., 2018b);
- using a genetic algorithm, e.g., particle swarm optimization, ant colony optimization or bat optimization, that describes a – typically inspired by nature – way to select new hyperparameter sets based on the results of previous runs, allowing for both discrete and continuous values (Shafie-khah et al., 2011; Yousaf et al., 2021; Lima de Campos et al., 2020).

For large hyperparameter spaces Bayesian methods seem to be the best choice – a grid search becomes computationally infeasible, while the heuristics used in the genetic algorithms might be slower to converge to reasonable values. The process of preparing a hyperparameter calibration study is, however, mostly effortless thanks to the abundance of libraries that implement these methods, for example, the open-source Optuna and Hyperopt packages for Python (Akiba et al., 2019; Bergstra et al., 2013).

Given the scope of the hyperparameter optimization task and its relevance for the accuracy of the forecasting models, some form of optimization is a necessity. This induces three issues:

- computational complexity,
- requirement for more data (optimization should be performed *ex-ante*) and
- the need for a reasonable design of the optimization study.

While there are no clear rules, a common choice for the hyperparameter optimization is to set aside a portion (e.g., 20%) of the data from the initial calibration window to validate the models. The validation set can be chosen randomly, however, it is important to use exactly the same data in each iteration to allow for the comparison of results (see **Paper 2**). Note, that the validation set for hyperparameter optimization does not coincide with the validation set in the rolling calibration study.

Chapter 4

Summary of results

4.1 Best practices for using machine learning in EPF

Recently there has been a growing interest in using deep learning to forecast electricity prices (Maciejowska et al., 2023). However, the lack of standardized testing and reporting practices has hindered accurate comparisons and – effectively – a wider adoption of models (Aggarwal et al., 2009). Moreover, some of the studies published are inconclusive, come to contrary conclusions or do not validate the claims correctly (see **Paper 2**). Even if the method is described in-depth in the paper, the replication is often impossible, as typically some implementation details are left out. Taking into consideration also the computational time it takes to replicate such a set of results, the researchers typically only compare their newly proposed methods to the simplest approaches. This calls for an open-source, unified benchmark, that was proposed in other fields, such as computer vision (Dollar et al., 2012) or cloud computing (Gan et al., 2019).

Lastly, the hyperparameter optimization – which is a key component of the neural network training – is done differently by the authors of different papers and there is no clearly defined standard approach. Many articles propose new (e.g., genetic) algorithms, but often the validation of their performance is insufficient. Therefore, there is a need for well-performing, automatic methods that can be easily applied in different scenarios.

4.1.1 A robust hyper-parameter optimization scheme for neural network models (Paper 1)

The hyper-parameter optimization is not only time consuming, but also requires a long history of the data (that cannot be later reused as the out-of-sample test data). On the other hand, the electricity price time series are prone to structural breaks, hence using a year or two of the data preceding the out-of-sample test period might result in the hyper-parameters to be optimized using a completely different data (in terms of magnitude of

price spikes, relation between the electricity load and price, average price level over the weekends etc.) than are observed later, in the out-of-sample test period.

Paper 1 successfully attempts to find optimal hyper-parameter sets using the data from past observations regarding one electricity market and applying them to a different dataset. It resulted in a lower error score than when hyper-parameters optimized on the tested electricity market were used, most probably thanks to the fact that the dataset used for the hyper-parameter optimization contains a well-balanced sample of price spikes, low prices and seasonal effects, while structural breaks were observed in both of the test datasets.

The second key outcome of the study is a robust hyper-parameter selection and aggregation scheme. The final proposed model averaged 15 individual neural networks – for each of 1, 2 and 3-year long calibration windows, 5 best hyper-parameter sets were chosen. The hyperparameters were optimized using the data from a different electricity market. The model outperformed similar combinations of 15 ensembled NNs that used data from the same market for hyper-parameter optimization and the out-of-sample testing on both of the tested markets (Nord Pool and PJM). The conclusion holds also for when the data is preprocessed using a variance-stabilizing transformation prior to the modeling; a fast and well-performing asinh transformation was used (Schneider, 2011; Uniejewski et al., 2018).

The study also concluded that in every test case, the NN models outperformed a LASSO benchmark that used the same input information, while taking similar time to train the models (the worst-case scenario for NN models was 2 minutes per one day of forecast for the final ensemble, whereas LASSO took 30-40s).

Publication details

- Published as: G. Marcjasz (2021), *Forecasting Electricity Prices Using Deep Neural Networks: A Robust Hyper-Parameter Selection Scheme*, Energies, 13(18), 4605.
- JCR classification: *Energy & Fuels*. Scopus classification: *Mathematics: Control and Optimization*; *Engineering*; *Energy*. IF = 3.2. MEiN: 140 pts, assigned to the Management and Quality Studies (NZJ) discipline.

4.1.2 Literature review, a set of best practices and an open-access benchmark (Paper 2)

Paper 2 serves as a comprehensive review of point EPF methods, with a main focus on machine learning and hybrid methods. The study outlines key aspects of a well-designed and replicable research in EPF, discusses a set of best practices new studies should follow, and proposes two well-described EPF models (LEAR, DNN). Both models are available publicly on GitHub platform as Python code and a collection of result files

ready for an evaluation.

The literature review is a holistic view on papers published after the review of Weron (2014) and considers recent advances in the field of statistical, deep learning and hybrid models. The last group is defined as models that contain multiple "modules": algorithms for data decomposition, feature selection, data clustering, multiple combined models and a heuristic to determine the hyper-parameter values. The review concludes that the possibility of choosing of a well-performing model from published literature is hindered due to lack of common test sets between different studies, often insufficient model and data descriptions and short test periods.

The aforementioned obstacles in comparing EPF models encouraged us to formulate a set of best practices – guidelines for replicable EPF studies that can be compared with different methods on a well-described testing ground comprising five distinct datasets. Some of the best practices we propose new EPF studies should follow are:

- using a sufficiently long test period – we recommend at least one year of data to be used as the out-of-sample test period,
- comparing against other well-established literature methods so the proposed method can be easily evaluated,
- making the researched models open access to eliminate the possibility of misinterpretation of the description and bugs in the implementation,
- considering (and communicating) the computational time that the new method takes.

The methods used in **Paper 2** are evaluated on a two year long out-of-sample period on five datasets. The codes are available on a public GitHub repository, which from the time the paper has been published attracted the researchers and encouraged them to send detailed questions regarding the implementation elements. At the time of writing this thesis, the repository was forked over 50 times and received over 130 stars on GitHub.

Publication details

- Published as: J. Lago, G. Marcjasz, B. De Schutter, R. Weron (2021) *Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark*, Applied Energy, 293, 116983
- JCR classification: *Energy & Fuels*; *Engineering*. Scopus classification: *Engineering*; *Environmental Science: Management, Monitoring, Policy and Law*; *Energy*. IF = 11.2, MEiN: 200 pts, assigned to the Economics & Finance discipline.
- My contribution amounted to ca. 30% and concerned designing the study, developing the methods, concluding analyses and drafting the paper.

4.2 NBEATSx – interpretable neural networks for EPF (Paper 3)

The neural network models – especially the deep ones – do not offer the same insights as for example, a linear counterpart. Due to a sheer number of the weights adjusted and heavily non-linear transformations applied to the data, checking the sensitivity of the neural network prediction on the changing or explaining which of the inputs made the prediction higher/lower is a cumbersome and in many cases, impossible task. The interpretability that is lacking in NN-based methods is, on the other hand, often an important factor when incorporating a model as a tool to aid the decision-making in daily operation. The lack of easily understandable outcomes is slowing down the model adoption in real-world applications (Rudin, 2019). For example, due to highly non-linear structure of the DNN model from **Paper 2** an expert can, at best, make an educated guess about what drives the prices and what is the sensitivity to various scenarios. On the other hand, having an interpretable model, decision-maker can investigate the outputs and understand the model better without the need to trust the outcome of a black box solution – which is especially important in a situation where the possibility to audit the decisions is a strict requirement (Carvalho et al., 2019).

Paper 3 proposes a novel extension to neural basis expansion analysis (NBEATS; a sequence-to-sequence model based on a system of deep neural networks) that allows for the inclusion of exogenous factors in the model. The wider acknowledgment of sequence modeling in the literature was initially sparked by the success of the ES-RNN hybrid method (comprising of Holt-Winters exponential smoothing and multiple LSTM stacks) in the M4 competition (Makridakis et al., 2018). The paper introducing the NBEATS method (Oreshkin et al., 2020) followed soon, proposing a purely machine learning method that outperformed the ES-RNN model on the M4 dataset by 3%. Our extension to the method addresses a crucial (from the EPF standpoint) shortcoming of the model – lack of exogenous input series support and uses the benchmarks introduced in **Paper 2** to compare against state-of-the-art literature methods. The resulting forecasts outperform the DNN model in most cases. Additionally one of the NBEATSx variants provides forecast that can be decomposed into trend, seasonal and exogenous effects. At the same time, the model augmented with exogenous series outperform its NBEATS (with no exogenous factors) counterparts significantly. The method was mentioned in a recent Forbes article as a technology of the future.

The method described in the paper, in general, decomposes the signal in a cascade of nonlinear projections onto different basis functions, organized in stacks of blocks, as depicted in Figure 4.1, the outputs of the stacks is summed to generate the model's output. The choice of basis function allows us to make the model output interpretable – in the paper we chose a configuration with a polynomial trend, harmonic functions as the seasonal effects and the exogenous base. An alternative, generic model (that re-

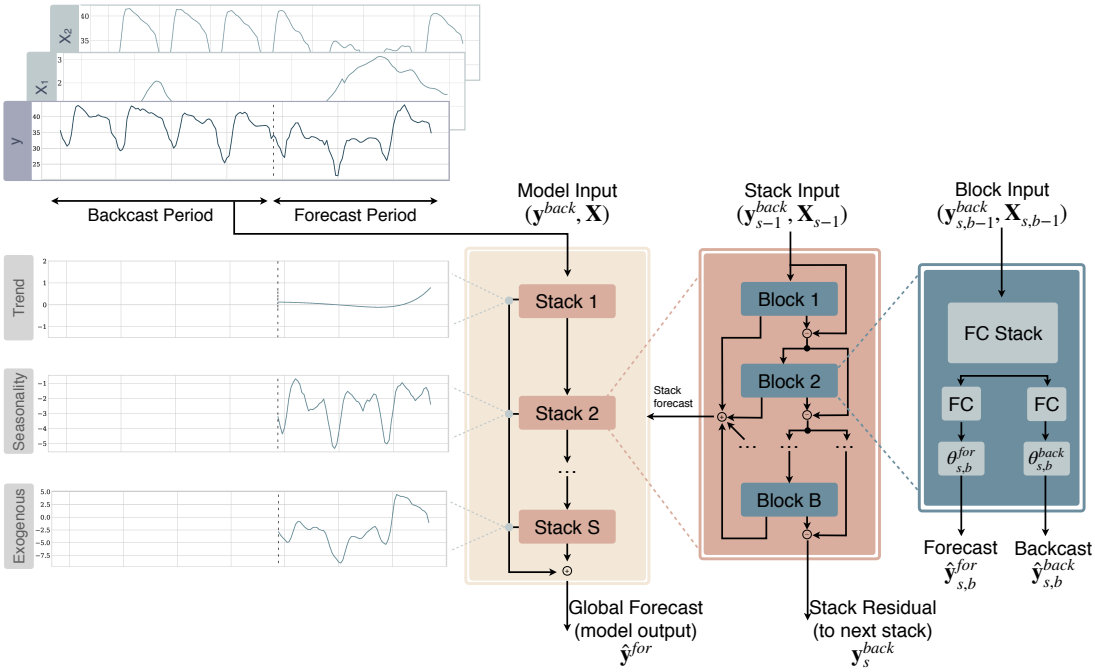


Figure 4.1: The NBEATSx structure showing additively aggregated stacks, which correspond to different bases (that in turn correspond to different components of the price: trend, seasonal component and exogenous variable impact). Source: Olivares et al. (2023).

sembles a DNN network more) was also evaluated – the performance of the generic and interpretable was similar, with a significant difference on only one of the five considered datasets.

Publication details

- Published as: K. Olivares, C. Challu, G. Marcjasz, R. Weron, A. Dubrawski (2022), *Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx*, International Journal of Forecasting, 39(2), 884-900.
- JCR classification: *Economics*; *Management*. Scopus classification: *Business, Management and Accounting*. IF = 7.9, MEiN: 140 pts, assigned to the Management and Quality Studies (NZJ) discipline.
- My contribution amounted to ca. 25% and concerned designing the study, taking part in method develompment, validating the results and drafting the paper.

4.3 Distributional neural networks for EPF (Paper 4)

De Gooijer and Hyndman (2006) state that the prediction intervals and probabilistic forecasts became much more common in practical applications thanks to the practitioners realizing the limitations of point forecasts. However, as Weron (2014) pointed out almost a decade later, the probabilistic methods did not gain much attention in the EPF literature, and despite the steady increase in the number of articles in this field, the research focused on probabilistic EPF is still scarce. Therefore, there is a need for well-performing probabilistic forecasting methods.

When compared to point forecasts, probabilistic ones carry additional information – the uncertainty of the prediction. The latter can be used to manage the risk more efficiently. For instance, the point forecast (i.e., the expected value) of the day-ahead price of electricity can be slightly higher for hour A than for hour B, but at the same time a probabilistic forecast can reveal that for hour B there is a much higher probability of having a higher price than for hour A. Moreover, probabilistic forecasts can indicate that there is a certain probability of an extreme event, or that the model is very uncertain of the forecast. These additional insights – when used correctly – can be valuable assets to a decision maker.

Typically, probabilistic forecasts are derived from point predictions, using historical simulation, bootstrapping or *quantile regression averaging* (QRA; see Nowotarski and Weron, 2018). However, there are also approaches that directly model

- the parameters of a parametric distribution, see Fig. 4.2 for an illustration of a *distributional deep neural network* (DDNN),
- the percentiles (as in the GEFCom2014 competition, see Hong et al., 2016) or a set of adequately selected quantiles approximating the predictive distribution (like in *quantile neural networks*, QNN; see Moon et al., 2021).

More formally, in the probabilistic forecasting task we are looking for an assessment of a probability that the observation will have a given value. Very commonly used are interval forecasts – they describe prediction intervals (PIs) with confidence level $1-\alpha$ (e.g., $\alpha = 10\%$ resulting in a 90% PI) in which a true value will lie with $1-\alpha$ probability (for a two-sided interval, which is commonly applied to EPF, the interval is defined to be between quantiles $\frac{\alpha}{2}$ and $1-\frac{\alpha}{2}$, e.g., between 5th and 95th percentile for a 90% PI). Increasing the number of the forecasted quantiles, we can arrive at a set of all percentiles, which is a good approximation of the whole distribution – such output was required for submitting the results in a Global Energy Forecasting Competition 2014. Alternatively, some methods allow for an output in form of an exact distribution (e.g., normal random variable with given mean and variance). The variety of probabilistic forecasts makes comparing different approaches more complex. Additionally, the error measures are also less straightforward than in the point forecasting. Commonly used, and at the

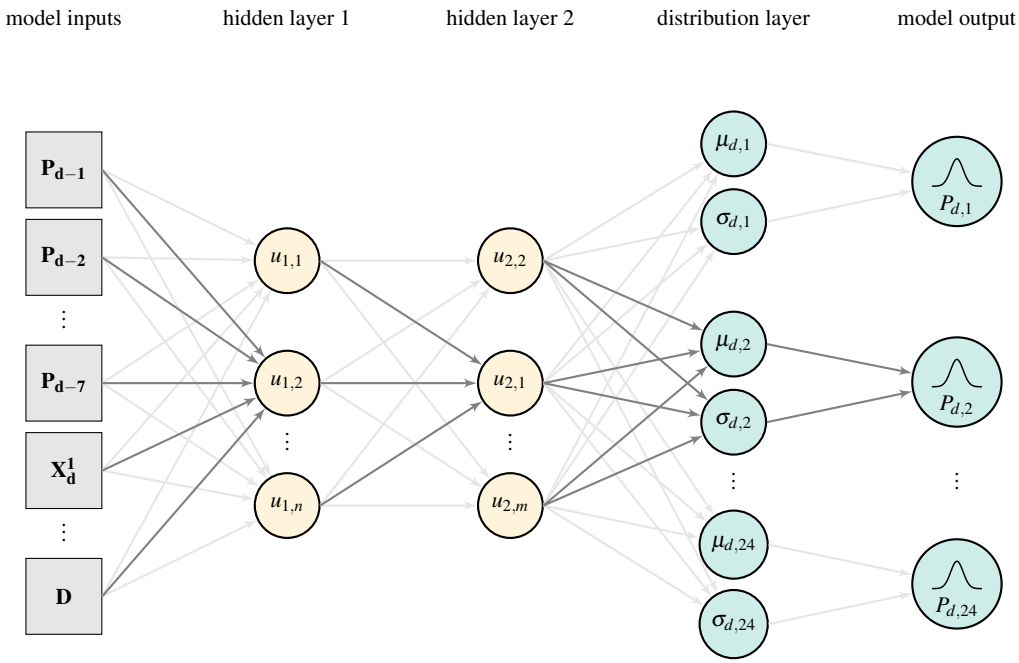


Figure 4.2: An illustration of a *distributional deep neural network* (DDNN) with 24 $\mathrm{N}(\mu_{d,h}, \sigma_{d,h})$-distributed outputs, corresponding to 24 hourly day-ahead prices $P_{d,1}, ..., P_{d,24}$. The gray boxes in the input layer represent 24-dimensional vectors of prices $\mathbf{P_{d-1}}, ..., \mathbf{P_{d-7}}$ or exogenous variables $\mathbf{X_d}, ...$, and a 7-dimensional vector of daily dummies $\mathbf{D}$. Based on Fig. 7 in Jędrzejewski et al. (2022)

same time very versatile is Pinball Loss, which is a measure of accuracy of a quantile forecast. Having a fine-grained distribution approximation (e.g., all percentiles), we can compact the accuracy score of the model into a single value by averaging the scores for all quantiles.

Paper 4 describes a novel forecasting framework that – based on neural networks – predicts a distribution of the price forecast for all 24 hourly subperiods. The main advantage of the method is its simplicity: the amount of changes compared to the point forecasting model that uses neural networks is minimal. The Distributional Deep Neural Network (DDNN) model proposed in the paper is an automated approach that outperforms the literature alternatives that use quantile regression averaging (QRA) as an intermediate step between the point and the probabilistic forecasts.

Overall, there are two differences between the DNN and DDNN models. Firstly, the DDNN model has an additional layer (called parameter layer), that is located between the last hidden layer and the output layer. It is a densly-connected layer, that outputs parameter values (e.g., locations and scales of normal distributions for each of 24 hours).

To allow for a more flexible model, the DDNN model uses a separate sub-layers in the parameter layer for each of chosen distribution's parameters, as illustrated in Fig. 4.2 for the normal distribution (2 parameters per output). The output layer is then responsible for transforming the parameters into distributions. The second difference is the loss function used during training. Distributional neural networks maximize the likelihood, as opposed to minimizing the MAE or RMSE in their point counterparts.

The model is tested for two output distributions: normal and Johnson SU, both of which outperform the two benchmark models that use QRA – and LEAR or DNN point forecasts, respectively. One of the crucial elements of the DDNN model is forecast averaging. Two probabilistic forecast aggregation schemes ("horizontal" – averaging quantiles and "vertical" – averaging probabilities) are tested and the horizontal one is not only statistically significantly better than the vertical counterpart in the Diebold and Mariano (1995) test (performed on the Pinball loss series), but also produces probabilistic forecasts that pass the Kupiec coverage test (Kupiec, 1995) for more hours of the day (the test is performed separately for each hour).

Aside from the statistical evaluation of the forecasts, a real-world application simulation is performed. Assuming the perspective of a battery owner, we try to schedule a daily cycle of charge and discharge that will result in the highest profit (both overall and per-transaction, to allow for the assesment of the cost of using the battery itself). The exercise results in two main findings: the probabilistic foreacsts are useful and allow for a better scheduling and the proposed DDNN methods outperform other tested forecasting models.

Publication details

- Published as: G. Marcjasz, M. Narajewski, R. Weron, F. Ziel (2023), *Distributional neural networks for electricity price forecasting*, Energy Economics, 125, 106843.
- JCR classification: *Economics*. Scopus classification: *Economics, Econometrics and Finance*; *Energy*. IF = 12.8. MEiN: 200 pts, assigned to the Management and Quality Studies (NZJ) discipline.
- My contribution amounted to ca. 30% and concerned designing the study, developing and testing the models, analyzing the results and drafting the paper.

4.4 Intraday trading strategies based on trajectory forecasts (Paper 5)

Probabilistic forecasts have another drawback besides complexity – the temporal dependency between consecutive periods is (usually) not reflected in the outputs. This means, that for two subsequent future horizons, we obtain two densities (or approximations)

without any information on the correlation between the two values. In other words, the prediction for the second horizon is not conditional on the realization of the first one.

Trajectory (or *ensemble*) forecasts attempt to mitigate this problem. They provide a way of generating multiple independent realizations for all future horizons that consider the temporal dependency in the output. Having such an ensemble of trajectories, one is able to, for example, compute the probability of a certain threshold to be exceeded for a number of consecutive periods. This can be helpful in planning an operation schedule for a power plant that has long ramp-up and ramp-down times.

The forecasts of trajectories of future values are not uncommon – a most notable example are the weather forecasting models, in which there is a strong dependency of the predicted value on the predicted values for preceding time steps (Pinson et al., 2009). When it comes to the energy markets, the importance of correctly forecasting the inter-period dependency structure is much less pronounced in the literature – despite a clear advantage of such an approach in certain scenarios.

Using the auction-based day-ahead market as an example, the possible benefit can be seen for the battery-backed RES (or reservoir-based hydro power plant), where the operator can better assess the risks and opportunities. For example, having multiple scenarios one can determine the probability of the occurrence of a long enough period of cheaper electricity to charge the batteries (or pump water to the reservoir) mid-day for it to be sold in the evening period of high prices. Even with probabilistic forecast for each hour of the day, such an assessment is impossible, as the dependency structure between prices in the consecutive prices is not contained there.

Trajectory forecasts can be even more beneficial for operators that participate in the continuous ID market. Since every product is traded for at least a couple of hours, there is not one price, but rather a trajectory, which is very volatile (Narajewski and Ziel, 2020; Uniejewski et al., 2019a). The moment at which the operator decides to purchase or sell the electricity dictates the economic result.

Paper 5 builds on the framework introduced in Serafin et al. (2022) and introduces a novel modeling approach – a DDNN-based (see **Paper 4**) method to forecast price paths on German continuous intraday electricity market. The paper contributes the DDNN ensemble that uses JSU-distributed networks and applies the Gaussian Copula on such probabilistic forecasts to reconstruct the temporal dependency between the consecutive subperiods – the models from the original paper were outperformed in terms of both the strategy profits and the accuracy of probabilistic forecasts used. The DDNN-based method achieved profit higher than the second-best model by ca. 2000 EUR – for comparison, Serafin et al. list the advantage of the temporal dependency modeling using Gaussian Copula on LASSO-QRA probabilistic models over using a historical errors on the point LASSO forecasts to equal roughly 500 EUR. This also proves the existence of an economic incentive behind more complex, but at the same time more accurate probabilistic forecasting models.

The paper also discusses the effect of ensemble size on the profit – we find that the returns diminish significantly with the ensemble size approaching 5 – which is still computationally feasible in time frame assumed in the trading strategy. To the best of our knowledge, this paper is the first to apply the distributional neural networks to forecasting intraday electricity prices. Based on the limited testing, the method was significantly better than combination of DNN with QRA and Gaussian Copula.

Lastly, and most importantly, **Paper 5** extends the benchmark of Serafin et al. (2022) by providing a more realistic variant, closely resembling the decision that a renewable energy producer might face in the bidding process. The transaction side depends on the day-ahead forecast error (the strategy assumes that the energy producer sells all of the generation forecasted day-ahead on the spot market, and uses the intraday market to balance the surplus or the deficit based on the more precise forecasts available closer to the delivery. The DDNN-based method outperforms other tested approaches on all 4 test datasets.

Publication details

- **Submitted to Energy Economics (under review)** as: G. Marcjasz, T. Serafin, R. Weron, *Trading on short-term path forecasts of intraday electricity markets with distributional neural networks*
- My contribution amounted to ca. 40% and concerned designing the study, preparing and testing the DDNN model, analyzing the results and drafting the paper.

Chapter 5

Auxiliary results

Aside from the papers that constitute the thesis, I have co-authored 10 other papers in the EPF field. They do not directly contribute to the thesis objectives or their contribution is limited, therefore they are not an integral part of the thesis. For the sake of completeness, in this section, I briefly describe the main findings from these studies:

- G. Marcjasz, T. Serafin, R. Weron (2018), *Selection of calibration windows for day-ahead electricity price forecasting*, Energies, 11, 2364
- K. Hubicka, G. Marcjasz, R. Weron (2019), *A note on averaging day-ahead electricity price forecasts across calibration windows*, IEEE Transactions on Sustainable Energy, 10(1), 321-323
- B. Uniejewski, G. Marcjasz, R. Weron (2019), *On the importance of the long-term seasonal component in day-ahead electricity price forecasting Part II — Probabilistic forecasting*, Energy Economoics, 79, 171-182
- G. Marcjasz, B. Uniejewski, R. Weron (2019), *On the importance of the long-term seasonal component in day-ahead electricity price forecasting with NARX neural networks*, International Journal of Forecasting, 35(4), 1520-1532
- B. Uniejewski, G. Marcjasz, R. Weron (2019), *Understanding intraday electricity markets: Variable selection and very short-term price forecasting using LASSO*, International Journal of Forecasting, 35(4), 1533-1547
- G. Marcjasz, B. Uniejewski, R. Weron (2020) *Beating the naïve — Combining LASSO with naïve intraday electricity price forecasts*, Energies, 13(7), 1667
- G. Marcjasz, B. Uniejewski, R. Weron (2020) *Probabilistic electricity price forecasting with NARX networks: Combine point or probabilistic forecasts?*, International Journal of Forecasting, 36(2), 466-479
- A. Jędrzejewski, G. Marcjasz, R. Weron (2021) *Importance of the long-term seasonal component in day-ahead electricity price forecasting revisited: Parameter-rich models estimated via the LASSO*, Energies, 14(11), 3249

- A. Jędrzejewski, J. Lago, G. Marcjasz, R. Weron (2022) *Electricity price forecasting: The dawn of machine learning*, IEEE Power and Energy Magazine, 20, (3), 24-31
- T. Serafin, G. Marcjasz, R. Weron (2022), *Trading on short-term path forecasts of intraday electricity prices*, Energy Economics, 112, 106125

In Hubicka et al. (2019) we introduce a novel concept in EPF: producing a forecast as an average of multiple runs of the same model that use different calibration window lengths. In a short empirical study, we show that simple schemes that average a (small) number of forecasts produced using a set of diverse calibration window lengths (especially a combination of very short and very long windows) are very effective and perform significantly better than the best single calibration window length chosen *ex-post*. The improvement was observed for both linear (estimated using ordinary least squares) and neural net (NN) models, and was equal to ca. 7% and 4%, respectively.

In Marcjasz et al. (2018), an article that eventually got published earlier than Hubicka et al. (2019), we extend the latter by considering an improved, weighted averaging scheme that performed on par on better than its equally-weighted counterpart. The paper also serves as a comprehensive comparison of different (with regards to the calibration windows used) averaging schemes across three datasets, testing not only the forecast averaging schemes, but also the effect of the inclusion of exogenous factors and application of variance stabilizing transformations. The proposed weighted averaging scheme is also fully automatic (it does not require any manual tuning) and the weights are computed based on only the last day, which enables the model to quickly adapt to the current market situation.

Uniejewski et al. (2019b) is one of three articles (that partially share the title) on the importance of long-term seasonal component (LTSC) modeling for EPF. All three articles extend the paper of Nowotarski and Weron (2016), which proposed a framework that decomposes the time series into a seasonal and a stochastic component, and models them separately. We consider probabilistic forecasting models and add an additional step to the seasonal component modeling framework – we also decompose the exogenous series. This results in a significant improvement in forecast accuracy. Moreover, forecast aggregation schemes that improve the Pinball score further were proposed (for the probabilistic forecasts).

The second LTSC-focused article is Marcjasz et al. (2019) in which we look at neural network models in the same setting. We conclude that the NN models outperform the linear counterparts and provide an additional possibility of averaging across multiple independent runs, which is also shown to improve the forecasts significantly. The LTSC framework applied to the NN models yields even greater improvements than in the case of linear models.

In the third article, Marcjasz et al. (2020b), we perform a comprehensive study that concerns forecast averaging. We answer the question of whether to average multiple point forecasts and use the latter to construct a probabilistic forecast, or to make a

probabilistic forecast from each of the point ones and then combine the probabilistic forecasts. We conclude that combining probabilistic forecasts yields more accurate predictive distributions.

In Jędrzejewski et al. (2021), we revisit the LTSC modeling framework, applying it to much larger (in terms of the number of inputs) models. We show that significant accuracy gains can be achieved for such more complex linear models, and additionally we discuss the order of applying seasonal decomposition and variance stabilizing transformations, and introduce forecast averaging schemes that outperform single forecasting models.

In Uniejewski et al. (2019a) we develop models for the German intraday electricity market. The dataset we use contains transactions from the continuous trading for hourly products, however, our analysis is limited to forecasting only the ID3 price index – a volume-weighted average price of the transactions in a 3-hour window preceding the delivery. The model is based on a LASSO-estimated parameter-rich structure, which allows us to identify the factors that drive the prices in the intraday market. We find that the most important predictor for the ID3 index is – alongside the day-ahead price for the forecasted hour – the last available ID3 index (in the setting of the paper, the ID3 price of a product with delivery 4 hours earlier than the forecasted one).

In Marcjasz et al. (2020a) we extend Uniejewski et al. (2019a). By using more granular transaction data and additional exogenous inputs, we are able to propose an averaging scheme that is able to outperform the naïve forecast of the ID3 price index, which Narajewski and Ziel (2020) report as the best-performing model. The naïve forecast of the ID3 index is in this case a volume-weighted transaction price of the last 15 minutes of continuous trading directly preceding the moment of forecast generation. Similarly to the previous paper, a LASSO-estimated parameter-rich model was used.

In Serafin et al. (2022) we introduce a novel economic benchmark based on a trading strategy in the German intraday market. We use trajectory forecasts to derive prediction bands which are then used as a time-dependent price level that – when exceeded – is a signal to enter the market. Together with the selection of different methods generating prediction bands, the benchmark allows to assess the economic differences between point, probabilistic and trajectory forecasts, and is also used in **Paper 5**. We show that trajectory forecasts are the most profitable of all considered approaches, hence providing evidence that more complex modeling techniques can lead to higher profits. Lastly, Jędrzejewski et al. (2022) is a review that discusses the developments in EPF in the last 25 years. The paper focuses on feature selection/model shrinkage and increasing model complexity.

Chapter 6

Conclusions

The aim of this thesis was to make a significant contribution to the field of electricity price forecasting (EPF) by developing robust, reliable and – when possible – interpretable DNN-based approaches for short-term point, probabilistic and ensemble forecasting of electricity prices. To address this aim, four objectives have been set and achieved:

- **Identify the most common problems encountered in EPF machine learning research, present a set of best practices and publish open access codes for well-performing benchmark models:**

  Based on a comprehensive literature review, **Paper 2** points to the most prominent shortcomings of the published machine learning studies, and proposes a set of best practices. The key contribution, however, lies in the open-access benchmark that consists of two state-of-the-art models (LEAR and DNN) and their results on five datasets. Moreover, the paper highlights the importance of hyperparameter optimization and forecast averaging. Both **Paper 1** and **Paper 2** propose selection schemes that use multiple hyperparameter sets obtained independently and average the forecasts for improved stability and accuracy of the outcome.

- **Develop an interpretable DNN model for point EPF that outperforms state-of-the-art benchmarks:**

  Paper 3 proposes a novel extension to the NBEATS framework of Oreshkin et al. (2020), that significantly improves prediction accuracy in EPF tasks. By introducing a stack that performs the projection onto exogenous variables, the NBEATSx model allows for (partial) interpretability and performance that surpasses that of the DNN benchmark of **Paper 2**.

- **Construct distributional DNNs that directly yield predictive distributions and are superior to state-of-the-art probabilistic models in terms of both statistical and economic measures:**

  Paper 4 introduces a new probabilistic forecasting method for EPF – the distributional deep neural network (DDNN) – that directly outputs parameters of a parametric distribution (Gaussian or JSU). It is tested in a real-world decision problem involving day-ahead trading and battery storage, and yields higher per transaction profits than state-of-the-art probabilistic benchmarks.

- **Develop a decision support method that uses distributional DNNs to generate trajectories of ID prices, then use it to construct profitable trading strategies:**

  Paper 5 proposes a new method to obtain trajectory forecasts based on DDNN probabilistic predictions and temporal dependence captured by a Gaussian copula. It extends the strategy of Serafin et al. (2022) by considering a more realistic evaluation approach with trading decisions depending on the errors of wind generation forecasts. The results indicate that utilizing DDNN predictions significanlty outperforms LEAR-based ones, both in terms of statistical error measured and trading profits.

In summary, this thesis explores the applicability of deep learning for various electricity price forecasting tasks, from the standpoint of decision-makers. Across all four objectives, deep neural network (DNN) models – when carefully calibrated – consistently demonstrated superior performance in point (**Paper 1** – **Paper 3**), probabilistic (**Paper 4**) and ensemble (**Paper 5**) forecasting compared to state-of-the-art-benchmarks. Furthermore, the introduced methods exhibit flexibility, suggesting potential applicability to other domains within the realm of energy modeling, including but not limited to renewable generation forecasting. Finally, the thesis also highlights the importance of following the best practices outlined in **Paper 2**, as robust comparisons and replicability are key to research excellence.

Bibliography

M. Afrasiabi, M. Mohammadi, M. Rastegar, L. Stankovic, S. Afrasiabi, M. Khazaei. Deep-based conditional probability density function forecasting of residential loads. *IEEE Transactions on Smart Grid*, 11(4):3646–3657, 2020.

S. K. Aggarwal, L. M. Saini, A. Kumar. Electricity price forecasting in deregulated markets: A review and evaluation. *International Journal of Electrical Power & Energy Systems*, 31(1):13–22, 2009. ISSN 01420615.

T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2623–2631, 2019.

N. Amjady, F. Keynia. Day-ahead price forecasting of electricity markets by mutual information technique and cascaded neuro-evolutionary algorithm. *IEEE Transactions on Power Systems*, 24(1):306–318, 2009.

P. Bento, J. Pombo, M. Calado, S. Mariano. A bat optimized neural network and wavelet transform approach for short-term price forecasting. *Applied Energy*, 210:88–97, 2018.

J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24, pages 2546–2554. Curran Associates, Inc., 2011.

J. Bergstra, D. Yamins, D. Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In S. Dasgupta, D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 115–123, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

D. Bertsimas, N. Kallus. From predictive to prescriptive analytics. *Management Science*, 66(3):1025–1044, 2020.

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

J. Catalão, S. Mariano, V. Mendes, L. Ferreira. Short-term electricity prices forecast-

ing in a competitive market: A neural network approach. *Electric Power Systems Research*, 77(10):1297–1304, 2007.

J. C. Cuaresma, J. Hlouskova, S. Kossmeier, M. Obersteiner. Forecasting electricity spot-prices using linear univariate time-series models. *Applied Energy*, 77(1):87–106, 2004.

T. Davenport, J. Harris. *Competing on Analytics: The New Science of Winning*. Harvard Business School Press, 2007.

J. G. De Gooijer, R. J. Hyndman. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3):443–473, 2006.

K. De Vos. Negative wholesale electricity prices in the German, French and Belgian day-ahead, intra-day and real-time markets. *The Electricity Journal*, 28(4):36–50, 2015.

D. Delen, S. Ram. Research challenges and opportunities in business analytics. *Journal of Business Analytics*, 1(1):2–12, 2018.

F. Diebold, R. Mariano. Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13:253–265, 02 1995.

P. Dollar, C. Wojek, B. Schiele, P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (4):743–761, 2012.

G. Dudek. Multilayer perceptron for GEFCom2014 probabilistic electricity price forecasting. *International Journal of Forecasting*, 32:1057–1060, 2016.

G. Dudek. STD: A seasonal-trend-dispersion decomposition of time series. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–13, 2023.

Energy-Charts. Energy-charts. `https://energy-charts.info`, 2023. Accessed: 2023-08-13.

ENTSO-E. ENTSO-E Transparency Platform, 2022. URL `https://transparency.entsoe.eu`. Accessed: 2022-11-08.

EPEX. EPEX SPOT. `https://www.epexspot.com/en`, 2023. Accessed: 2023-08-12.

Forbes. Problem-solving with deep tech: five fields to watch out for in the next 10 years. `https://www.forbes.com`, 2023. Accessed: 2023-09-10.

Y. Gan, Y. Zhang, D. Cheng, A. Shetty, P. Rathi, *et al.* An open-source benchmark suite for microservices and their hardware-software implications for cloud & edge systems. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '19, page 3–18, New York, NY, USA, 2019.

A. Gianfreda, F. Ravazzolo, L. Rossini. Comparing the forecasting performances of linear models for electricity prices with high RES penetration. *International Journal of Forecasting*, 36:974–986, 2020.

I. Goodfellow, Y. Bengio, A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org/`.

R. Griffin. *Fundamentals of Management*. Cengage Learning, 2021. ISBN 9780357517550.

S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, USA, 1st edition, 1994. ISBN 0023527617.

J. Heizer, B. Render, H. J. Weiss. *Principles of Operations Management*. Pearson College Div, 2008. ISBN 978-0132343282.

T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, R. J. Hyndman. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. *International Journal of Forecasting*, 32(3):896–913, 2016. ISSN 0169-2070.

T. Hong, P. Pinson, Y. Wang, R. Weron, D. Yang, H. Zareipour. Energy forecasting: A review and outlook. *IEEE Open Access Journal of Power and Energy*, 7:376–388, 2020.

K. Hubicka, G. Marcjasz, R. Weron. A note on averaging day-ahead electricity price forecasts across calibration windows. *IEEE Transactions on Sustainable Energy*, 10 (1):321–323, 2019.

T. Janke, F. Steinke. Forecasting the price distribution of continuous intraday electricity trading. *Energies*, 12(22):4262, 2019.

T. Januschowski, J. Gasthaus, Y. Wang, D. Salinas, V. Flunkert, M. Bohlke-Schneider, L. Callot. Criteria for classifying forecasting methods. *International Journal of Forecasting*, 36:167–177, 2020.

A. Jędrzejewski, G. Marcjasz, R. Weron. Importance of the long-term seasonal component in day-ahead electricity price forecasting revisited: Parameter-rich models estimated via the lasso. *Energies*, 14(11):3249, 2021.

A. Jędrzejewski, J. Lago, G. Marcjasz, R. Weron. Electricity price forecasting: The dawn of machine learning. *IEEE Power and Energy Magazine*, 20(3):24–31, 2022.

O. A. Karabiber, G. Xydis. Electricity price forecasting in the danish day-ahead market using the tbats, ann and arima methods. *Energies*, 12(5):928, 2019.

D. Keles, J. Scelle, F. Paraschiv, W. Fichtner. Extended forecast methods for day-ahead electricity spot prices applying artificial neural networks. *Applied Energy*, 162:218–230, 2016. ISSN 0306-2619.

D. P. Kingma, J. Ba. Adam: A method for stochastic optimization, 2014.

E. Kraft, D. Keles, W. Fichtner. Modeling of frequency containment reserve prices with econometrics and artificial intelligence. *Journal of Forecasting*, 39(8):1179–1197, 2020.

E. Kraft, M. Russo, D. Keles, V. Bertsch. Stochastic optimization of trading strategies in sequential electricity markets. *European Journal of Operational Research*, 308(1): 400–421, 2023.

M. Kraus, S. Feuerriegel, A. Oztekin. Deep learning in business analytics and operations research: Models, applications and managerial implications. *European Journal of Operational Research*, 281(3):628–641, 2020.

P. H. Kupiec. Techniques for verifying the accuracy of risk measurement models. *The Journal of Derivatives*, 3(2), 1995.

J. Lago, F. De Ridder, B. De Schutter. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. *Applied Energy*, 221: 386–405, 2018a.

J. Lago, F. De Ridder, P. Vrancx, B. De Schutter. Forecasting day-ahead electricity prices in Europe: The importance of considering market integration. *Applied Energy*, 211:890–903, 2018b.

J. Lago, G. Marcjasz, B. De Schutter, R. Weron. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293:116983, 2021.

K. Lepenioti, A. Bousdekis, D. Apostolou, G. Mentzas. Prescriptive analytics: Literature review and research challenges. *International Journal of Information Management*, 50:57–70, 2020.

L. M. Lima de Campos, J. H. Almeida Pereira, D. S. Duarte, R. C. L. de Oliveira. Bio-inspired system for electricity price forecast in the brazilian market. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.

K. Maciejowska. Assessing the impact of renewable energy sources on the electricity price level and variability – a quantile regression approach. *Energy Economics*, 85: 104532, 2020.

K. Maciejowska, B. Uniejewski, R. Weron. Forecasting electricity prices. In *Oxford Research Encyclopedia of Economics and Finance*. Oxford University Press, 2023. DOI: 10.1093/acrefore/9780190625979.013.667.

S. Makridakis, E. Spiliotis, V. Assimakopoulos. The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802–808, 2018.

G. Marcjasz. Forecasting electricity prices using deep neural networks: A robust hyperparameter selection scheme. *Energies*, 13(18):4605, 2020.

G. Marcjasz, T. Serafin, R. Weron. Selection of calibration windows for day-ahead electricity price forecasting. *Energies*, 11:2364, 2018.

G. Marcjasz, B. Uniejewski, R. Weron. On the importance of the long-term seasonal component in day-ahead electricity price forecasting with NARX neural networks. *International Journal of Forecasting*, 35:1520–1532, 2019.

G. Marcjasz, B. Uniejewski, R. Weron. Beating the naïve – combining LASSO with naïve intraday electricity price forecasts. *Energies*, 13(7):1667, 2020a.

G. Marcjasz, B. Uniejewski, R. Weron. Probabilistic electricity price forecasting with NARX networks: Combine point or probabilistic forecasts? *International Journal of Forecasting*, 36:466–479, 2020b.

A. Mashlakov, T. Kuronen, L. Lensu, A. Kaarna, S. Honkapuro. Assessing the per-

formance of deep learning models for multivariate probabilistic energy forecasting. *Applied Energy*, 285:116405, 2021.

K. Mayer, S. Trück. Electricity markets around the world. *Journal of Commodity Markets*, 9:77–100, 2018.

S. J. Moon, J.-J. Jeon, J. S. H. Lee, Y. Kim. Learning multiple quantiles with neural networks. *Journal of Computational and Graphical Statistics*, 30(4):1238–1248, 2021.

M. Narajewski, F. Ziel. Econometric modelling and forecasting of intraday electricity prices. *Journal of Commodity Markets*, 19:100107, 2020.

K. K. Nargale, S. B. Patil. Day ahead price forecasting in deregulated electricity market using artificial neural network. In *2016 International Conference on Energy Efficient Technologies for Sustainability (ICEETS)*, pages 527–532, 2016.

J. Nowotarski, R. Weron. On the importance of the long-term seasonal component in day-ahead electricity price forecasting. *Energy Economics*, 57:228–235, 2016.

J. Nowotarski, R. Weron. Recent advances in electricity price forecasting: A review of probabilistic forecasting. *Renewable and Sustainable Energy Reviews*, 81:1548–1568, 2018.

K. G. Olivares, C. Challu, G. Marcjasz, R. Weron, A. Dubrawski. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting*, 39:884–900, 2023.

OMIE. Operador del Mercado Ibérico de Energía – Polo Español. `https://www.omie.es`, 2022. Accessed: 2022-12-10.

B. N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, pages 1–21, 2020. URL `https://openreview.net/forum?id=r1ecqn4YwB`.

B. N. Oreshkin, G. Dudek, Paweka, E. Turkina. N-beats neural network for mid-term electricity load forecasting. *Applied Energy*, 293:116918, 2021. ISSN 0306-2619.

I. P. Panapakidis, A. S. Dagoumas. Day-ahead electricity price forecasting via the application of artificial neural network based models. *Applied Energy*, 172:132–151, 2016.

F. Petropoulos, D. Apiletti, V. Assimakopoulos, M. Z. Babai, D. K. Barrow, *et al.* Forecasting: theory and practice. *International Journal of Forecasting*, 38(3):705–871, 2022. ISSN 0169-2070.

F. Petropoulos, G. Laporte, E. Aktas, S. A. Alumur, C. Archetti, *et al.* Operational research: Methods and applications. *Journal of the Operational Research Society*, 2024. DOI: 10.1080/01605682.2023.2253852.

P. Pinson, H. Madsen, H. A. Nielsen, G. Papaefthymiou, B. Klöckl. From probabilistic forecasts to statistical scenarios of short-term wind power production. *Wind Energy*, 12(1):51–62, 2009.

PJM. Pennsylvania-new jersey-maryland interconnection website. https://pjm.com, 2022. Accessed: 2022-12-11.

P. S. Pravin, J. Z. M. Tan, Z. Wu. Performance evaluation of various hyperparameter tuning strategies for forecasting uncertain parameters used in solving stochastic optimization problems. In *2022 IEEE International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, pages 301–306, 2022.

M. Qi, H.-Y. Mak, Z.-J. M. Shen. Data-driven research in retail operations—a review. *Naval Research Logistics (NRL)*, 67(8):595–616, 2020.

M. Rubaszek, Z. Karolak, M. Kwas, G. S. Uddin. The role of the threshold effect for the dynamics of futures and spot prices of energy commodities. *Studies in Nonlinear Dynamics & Econometrics*, 24(5):20190068, 2020.

C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 2019.

S. Schneider. Power spot price models with negative prices. *Journal of Energy Markets*, 4(4):77–102, 2011.

T. Serafin, G. Marcjasz, R. Weron. Trading on short-term path forecasts of intraday electricity prices. *Energy Economics*, 112:106125, 2022.

M. Shafie-khah, M. P. Moghaddam, M. Sheikh-El-Eslami. Price forecasting of day-ahead electricity markets using a hybrid forecast method. *Energy Conversion and Management*, 52(5):2165–2169, 2011.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58:267–288, 1996.

B. Uniejewski, J. Nowotarski, R. Weron. Automated variable selection and shrinkage for day-ahead electricity price forecasting. *Energies*, 9(8), 2016. ISSN 1996-1073.

B. Uniejewski, R. Weron, F. Ziel. Variance stabilizing transformations for electricity spot price forecasting. *IEEE Transactions on Power Systems*, 33(2):2219–2229, 2018.

B. Uniejewski, G. Marcjasz, R. Weron. Understanding intraday electricity markets: Variable selection and very short-term price forecasting using LASSO. *International Journal of Forecasting*, 35:1533–1547, 2019a.

B. Uniejewski, G. Marcjasz, R. Weron. On the importance of the long-term seasonal component in day-ahead electricity price forecasting: Part II – Probabilistic forecasting. *Energy Economics*, 79:171–182, 2019b.

R. Weron. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, 30(4):1030–1081, 2014.

P. M. Williams. Using neural networks to model conditional multivariate densities. *Neural Computation*, 8(4):843–854, 1996.

Y. Yao, L. Rosasco, A. Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26:289–315, 2007.

A. Yousaf, R. M. Asif, M. Shakir, A. U. Rehman, F. Alassery, H. Hamam,

O. Cheikhrouhou. A novel machine learning-based price forecasting for energy management systems. *Sustainability*, 13(22):12693, 2021.

Y. Zhou, A. Scheller-Wolf, N. Secomandi, S. Smith. Electricity trading and negative prices: Storage vs. disposal. *Management Science*, 62(3):880–898, 2016.

F. Ziel. Forecasting electricity spot prices using lasso: On capturing the autoregressive intraday structure. *IEEE Transactions on Power Systems*, 31(6):4977–4987, 2016.

F. Ziel, R. Weron. Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks. *Energy Economics*, 70:396–420, 2018.

K. Özen, D. Yıldırım. Application of bagging in day-ahead electricity price forecasting and factor augmentation. *Energy Economics*, 103:105573, 2021.

Paper 1

A robust hyper-parameter optimization scheme for neural network models

Grzegorz Marcjasz

energies

MDPI

Article

Forecasting Electricity Prices Using Deep Neural Networks: A Robust Hyper-Parameter Selection Scheme

Grzegorz Marcjasz

Department of Operations Research and Business Intelligence, Wrocław University of Science and Technology, 50-370 Wrocław, Poland; grzegorz.marcjasz@pwr.edu.pl

Received: 22 July 2020; Accepted: 2 September 2020; Published: 4 September 2020

Abstract: Deep neural networks are rapidly gaining popularity. However, their application requires setting multiple hyper-parameters, and the performance relies strongly on this choice. We address this issue and propose a robust ex-ante hyper-parameter selection procedure for the day-ahead electricity price forecasting that, when used jointly with a tested forecast averaging scheme, yields high performance throughout three-year long out-of-sample test periods in two distinct markets. Being based on a grid search with models evaluated on long samples, the methodology mitigates the noise induced by local optimization. Forecast averaging across calibration window lengths and hyper-parameter sets allows the proposed methodology to outperform a parameter-rich least absolute shrinkage and selection operator (LASSO)-estimated model and a deep neural network (DNN) with non-optimized hyper-parameters in terms of the mean absolute forecast error.

Keywords: electricity price forecasting; artificial neural network; deep learning; machine learning; hyper-parameter optimization

1. Introduction

The majority of electricity trading in Europe takes place during the day-ahead auctions, which are held once a day (typically before or around noon) and determine the prices for the physical delivery of electricity during each load period of the next day. This results in a vector-like time series and uniquely defines the time when price forecasts for all 24 h of the next day have to be available [1].

The "batch" determination of prices for the whole day at once and intraday load patterns introduce very strong daily and weekly seasonalities. This is typically addressed by using a multivariate modeling framework. The price distribution is heavy-tailed, with both positive and negative spikes. The sole presence of spikes suggests that the predictive model should operate on data transformed in a way that stabilizes (or reduces) the variance.

Overall, electricity price forecasting (EPF) is a very demanding task, as even the best-performing solutions for one market might not be well suited for data of different origin. The recent comeback of the artificial intelligence methods, especially artificial neural networks (ANN), as an effect of advancements in both the software and the hardware available, enabled researchers to efficiently utilize them for forecasting. A rising trend can be observed in the EPF literature, with multiple papers reporting good performance of such models [2,3]. The simplest use case is to calibrate a parsimonious model (e.g., ARX1) using the neural network topology. The research interest is, however, not limited to such sumple approaches and deep learning models (with deep neural networks (DNNs) as an example) also note increased popularity.

The question whether it is feasible to use different (but describing the same phenomenon) data to find the best hyper-parameter sets is one of the main purposes of the study. Additionally, the paper

aims to provide insights about the influence of several factors on the final results, including the origin of the data, variance stabilizing transformations (VST) and calibration window lengths used for training models.

The rest of the paper is structured as follows. Section 2 provides a brief EPF literature overview, Section 3 describes several auxiliary techniques, then Section 4 presents the datasets used in the study. Section 5 introduces the benchmarks and models used. The results in terms of Mean Absolute Errors (MAE) are presented in Section 6. Finally, Section 7 wraps up the results and concludes the study.

2. Overview of Machine Learning Techniques in EPF Literature

In the recent years, multiple papers using machine learning techniques to forecast the electricity prices were published [3–8]. However, not all of the above test the proposed methods against statistical-based algorithms using long out-of-sample test periods, and only two papers consider multiple datasets [3,5]. Some authors use the neural networks as building blocks for more sophisticated structures [9–12]. Such a complex approaches are usually evaluated on short test periods (most probably due to the computational time constraints) or their performance is only compared to very simple methods.

Among the recently published papers, one research direction emerges most prominently, namely proposing multi-step frameworks that typically consist of a method for data decomposition, feature selection procedure and optimization algorithm that governs the hyper-parameter selection process [8,13,14]. However, the methods proposed in such studies are seldom compared with the results of the state-of-the-art literature methods, and hence their predictive performance is hard to assess. Part of the EPF literature focuses more on the economic benefits that can be attained with better forecasts [15] or on the price formation process and the impact of the respective fundamental variables on the price [16,17].

Regarding the model estimation itself, the papers that utilize machine learning techniques use mainly artificial neural networks and extreme learning machines (ELM). ELMs are typically used in multi-step frameworks, where decomposition and optimization techniques are used to preprocess the data and select the model hyper-parameters [13,14]. Most of these studies concentrate more on the preprocessing steps than the modeling. On the other hand, the studies that use the ANNs to estimate the model often focus on the design aspect of neural networks (such as activation functions and network topologies) or the methods that facilitaite the process of hyper-parameter selection [4,5,18,19]. However, the possible hyper-parameter space is very broad, and studies cover only its small part, by allowing only some of the potential variables to change, while keeping others fixed. Moreover, the results are strongly dependent on the whole process: From the dataset used, through the preprocessing steps to the input features. Overall, the machine learning methods used in the literature tend to significantly outperform considered benchmarks.

3. Preliminaries

3.1. Available Data Overview

The data used in the study comes from three markets. The Global Energy Forecasting Competition 2014 (GEFCom2014) [20] time series spans an almost three year long period and is used solely for hyper-parameter precalibration. The Nord Pool and PJM series consist of six years of data each, with the first three years used for the hyper-parameter precalibration and the remainder used as an out-of-sample test period for the main phase of the study. This results in three three year long datasets for the hyper-parameter precalibration procedure and two three year long datasets for out-of-sample testing.

The timeline of data used in the study is presented in Figure 1, where the dashed vertical lines mark the ends of the longest initial calibration windows: Gray for the precalibration phase and black for the main testing. Horizontal grid lines indicate the price levels: The solid one, i.e., the lowest, corresponds to 0 and all lines (dashed, solid) are separated by 100 units (USD/MWh for PJM and GEFCom2014, and EUR/MWh for Nord Pool). The three series exhibit very distinct characteristics.

The PJM data noted a long period of pronounced price spikes in the beginning of 2014 (reaching over 500–800 USD/MWh) and nearly no prominent spikes in the test years (2016–2018). The Nord Pool, on the other hand, shows an increased variability over time, along with price spikes to both high and low prices (albeit not reaching the negative values). The reason for such a changing behavior lies, most probably, in the changes in the generation mix and usage patterns. The six year-long history shows that the markets are not stagnant, and the modeling should be performed on the recent data. The third, shorter series (GEFCom2014) is only used to infer the hyper-parameters on, and does not demonstrate a change in the characteristics similar to the other two series.

For each of the data series, an exogenous series of the day-ahead load forecasts is also available. To simplify the notation, the letter p is used to refer to the prices, letter z to refer to the exogenous data and letter x—to present a transformation that is applied identically to both prices and load forecasts, i.e., $x = \{p, z\}$. The notation also makes use of capital letters, subscripts and superscripts to indicate the step of the data preprocessing.

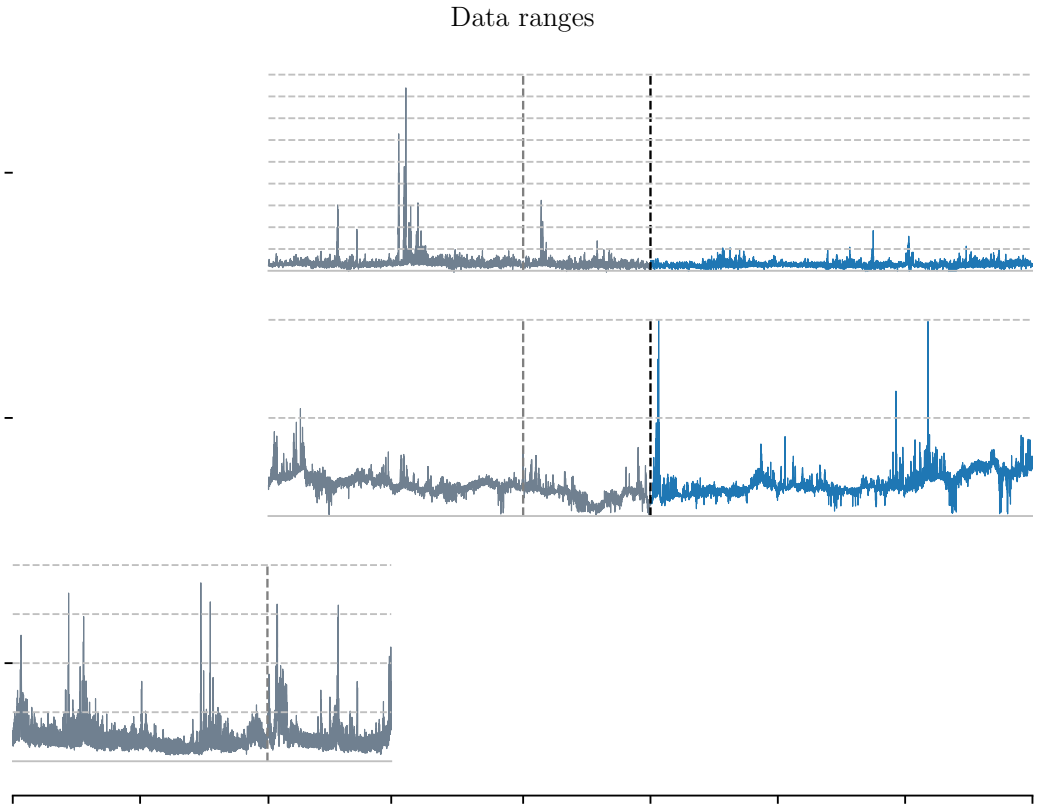


Figure 1. Timeline of the datasets considered in the study. The periods used for hyper-parameter calibration are drawn in gray, whereas the out-of-sample test periods are in blue.

3.2. Hyper-Parameter Choice

The procedure of training a neural network (i.e., the process of fitting the weights to the connections present in the structure) and the topology of the network itself is defined by a collection of hyper-parameters. Firstly, the user can choose from different network-related options, such as the structure (neurons, layers, connections between them, activation functions). Secondly, the choice expands to various parameters that describe the training (e.g., the optimization algorithm, the loss function, training time, sampling, etc.). Note, that there are options that depend on the optimization algorithm, such as the learning rate. However, their impact on the forecasting performance is not covered in this study.

Naturally, the parameters come with predefined default values, which are typically an effect of limited empirical tests by the authors, or are theoretically derived based on the characteristics of the underlying processes. The defaults, albeit being "reasonable", might turn out to be sub-optimal for a specific task. To address this issue, one can resort to some form of hyper-parameter optimization. This can be addressed in multiple ways, each of them having its own strengths and weaknesses. One of the methods—the one used in this study—requires the user to perform an additional numerical experiment to test values from a predefined hyper-parameter grid, using different data than the main out-of-sample test period. The advantage of such an approach are small requirements for information about the impact of each parameter on the forecasting performance, a mere specification of possible values suffices. Moreover, after performing the additional experiment, the method does not impose additional computational burden; the approach presented here implicitly assumes that the best parameter sets remain unchanged over time. On the other hand, the hyper-parameter precalibration procedure is very resource-consuming and any unnecessary value of a parameter inserted into the grid results in significantly increased computational burden. The method is, however, well suited for explanatory analysis, as it can help to understand the impact of each parameter on the final outcome and thus to create better forecasting models in the future. Additionally, a grid search mitigates one major shortcoming of step-by-step optimization techniques: It lessens the impact of the noise on the model chosen. Step-by-step optimization procedures applied on such volatile data are prone to end up in a suboptimal point of the parameter space due to the randomness of the error evolution over consecutive trial runs.

3.3. Calibration Window Length

The length of the calibration window used for fitting models—both statistical and neural network based—is a very important, but overlooked issue. In a recent series of papers [21–23], researchers aim to find combinations of calibration windows that yield the best forecasts and a scheme to aggregate the information contained within. A thorough examination of the impact of the calibration window length on the forecasting accuracy is out of scope of this study. However, motivated by the results, we will use three calibration window lengths: 364, 728 and 1092 days. Note, that the shortest window length we use (364 days) is much longer than the 28- or 56-day windows used in the previous studies. This is a consequence of using richer, multi-parameter input structures.

3.4. Forecast Averaging

Forecast averaging across different calibration window lengths is only one of the possibilities, but can be applied to all of the models used in the study. Another option can be introduced by generating multiple forecasts using different parameters, such as the λ parameter in least absolute shrinkage and selection operator (LASSO) or the hyper-parameter sets for neural networks. The former, however, was not used in this study. Based on a limited numerical experiment, the forecasting performance depends on the λ parameter in a close-to-convex way, which would require either taking poor forecasts into the ensemble, or taking forecasts very similar to each other. For neural network models, especially with a full parameter grid search, one should be able to simply take the top N performing parameter sets and create an ensemble out of them. The top entries obtained this way would present a variety of different parameter combinations, and additionally limiting the negative impact of local optimization and random initialization on the forecast robustness. This leads to the next ensembling possibility, namely averaging consecutive runs of an identically parametrized model [3]. This technique, however, is not utilized in this study, as (i) it requires n times more computational effort for n runs and (ii) further averaging the forecasts averaged over different parameter sets would result in diminishing returns. Naturally, the potential of averaging forecasts does not end here. It is possible to combine forecasts obtained using different models, different estimation techniques, etc. However, the two methods used in this study, namely averaging forecasts of one model, obtained using different calibration window lengths and considering top-performing hyper-parameter sets, are

straight-forward to implement. Moreover, the former has already been shown to be effective in EPF, whereas the latter is a natural choice for the hyper-parameter grid search performed in this study.

3.5. Variance Stabilizing Transformations

Electricity price time series are characterized by pronounced spikes. Having such spiky data, using one of the so-called variance stabilizing sransformations (VSTs) is a well-known technique to obtain more accurate predictions [24–26].

In this study the data series were normalized prior to applying the VST, and the normalization process was as follows. First, we compute the median value a_X of the series X (i.e., a_P for the prices and a_Z for the load forecast) in whole calibration window, then we derive the median absolute deviation (MAD) around the median and adjust it by a factor for asymptotically normal consistency to the standard deviation, i.e., $b_X = \text{MAD}(\mathbf{X})/\Phi^{-1}(0.75)$, where $\Phi^{-1}(0.75) \simeq 0.6745$ is the 75th percentile of the standard normal distribution. The normalized price has the form

$$P^{norm}_{d,h} = \frac{1}{b_P}(P_{d,h} - a_P),$$

and the normalized load forecast $Z^{norm}_{d,h}$ is obtained analogously. Having the normalized series, one can apply the chosen transformation: $x_{d,h} = f(X^{norm}_{d,h})$, where $X^{norm}_{d,h} = P^{norm}_{d,h}$ or $Z^{norm}_{d,h}$.

The VST used in the study is the arc hyperbolic sine (asinh), which is straightforward to implement, symmetric around zero and its inverse—the hyperbolic sine—is also computationally efficient, thus making it feasible for use in electricity price forecasting [26–28]. The asinh transformation is defined as follows:

$$x_{d,h} = \text{asinh}(X^{norm}_{d,h}) = \log\left(X^{norm}_{d,h} + \sqrt{\left(X^{norm}_{d,h}\right)^2 + 1}\right).$$

After computing the forecasts $\widehat{p}_{d,h}$ for the transformed prices, an inverse transformation is applied to obtain the price predictions in terms of normalized prices: $\widehat{P}^{norm}_{d,h} = \sinh(\widehat{p}_{d,h})$. Next, to compute the final predictions $\widehat{P}_{d,h}$ in correct units, one has to invert the normalization procedure: $\widehat{P}_{d,h} = b_P \cdot \widehat{P}^{norm}_{d,h} + a_P$. Note, that only the price series is forecasted in the process. The variance stabilizing transformation, however, is applied independently for both series, i.e., the medians and MADs are computed separately for the prices and loads.

3.6. Early Stopping Condition for Neural Networks

The neural network training process is typically interrupted before the algorithm converges. This is done to avoid over-fitting the network's weights to the presented data and to ensure that the network can accurately infer based on the unseen data [29]. However, the use of early stopping leads to two issues. Firstly, the data has to be split into training and validation samples, leaving less data to train the model on. Secondly, the procedure induces additional hyper-parameters to be determined: A ratio of training to validation data, the method of splitting the series (block, random or a mixture of the two) and the so-called "patience", i.e., a parameter that fixes the number of iterations without validation score improvement after which the training is stopped.

Instead, the study uses a fixed number of iterations (called `epochs`), and the value is chosen empirically as the best-performing in the precalibration phase. Not only such an approach simplifies the study setup, but also serves as a medium to find an optimal training length. Additionally, the model is trained using all of the available data.

3.7. Software

LASSO models were estimated using the scikit-learn [30] library for Python. The neural network models were trained using Python programming language and Keras library (version 2.2.4) [31] with either

Theano or TensorFlow backend (both yielded results with negligible differences in the limited comparison performed). All neural networks were invoked with default floating point precision (`float32`).

4. Datasets

4.1. Modeling Settings

The study can be divided into two parts: Precalibration phase that involves a hyper-parameter grid search, and testing phase in which the tested hyper-parameter sets are used. However, there are several design decisions that were not a part of the hyper-parameter grid:

1. VST used for both phases,
2. calibration window length within the precalibration phase,
3. transferring the hyper-parameter optimization results from one dataset to another.

To assess the need for VSTs when using neural networks, we have performed two repetitions of each experiment, one with the VST (asinh, see Section 3.5) applied, and one without. However, we did not allow to mix the approaches between the phases, meaning that for each precalibration dataset, we have obtained two separate sets of top models. In the testing part of the study, for VST-transformed data we have used only the parameter sets appointed using the VST, and similarly for the non-transformed data.

Lastly, we have included a dataset that does not contain an out-of-sample test set. This was done to check whether it is important to precalibrate on the data from the same market. To evaluate the potential differences between the hyper-parameter "origin", for both of the test datasets, we had three best hyper-parameter sets: The one chosen for the GEFCom2014 data precalibration period, the one chosen for NP data, and the one chosen for PJM data.

Both phases used a rolling calibration window scheme (i.e., a calibration window of fixed length, directly preceding the forecasted day). For example, when forecasting the first day of the out-of-sample test period (i.e., 29 December 2015) with a 1092 day long calibration window, the model is trained on data from 1 January 2013 to 28 December 2015. After that, the calibration window is rolled by one day (2 January 2013–29 December 2015) to model the next day, 30 December 2015, etc.

4.2. Precalibration and Out-Of-Sample Testing Data

The precalibration procedure, i.e., the process of empirically finding the best hyper-parameter set for given test conditions (data origin, VST, see Sections 3.2, 3.3 and 3.5) was performed on three data series. Two of them originate from electricity markets operating in the United States: The data used in GEFCom2014 competition [20] and the Pennsylvania-New Jersey-Maryland (PJM) Interconnection data, the third one is from the Scandinavian electricity exchange (Nord Pool, NP). Each of the data series used for precalibration comprises roughly three years (1092 day long subset for NP and PJM, 1082 days for GEFCom) of hourly data: Marginal prices as well as day-ahead load forecasts. It is worth noting, that these datasets present a distinct overview of liquid and well-established electricity markets. The diversity of generation sources and usage patterns is important for this study, as in the chapters to follow the patterns and dependencies discovered for one dataset will be to some extent used for other markets via the precalibrated hyper-parameters.

The second phase of the procedure uses the best precalibrated parameter sets to make predictions for either a completely different dataset (e.g., GEFCom $\rightarrow$ NP) or from the same market, but a different period (e.g., PJM 2013–2015 $\rightarrow$ PJM 2016–2018). The methodology implicitly assumes that the price response to the input variables (i.e., previous prices and exogenous series) does not vary throughout the years nor between the markets. This limitation is partially weakened by the rolling calibration window scheme used in the study.

The testing is performed on two datasets. In both cases the out-of-sample test period consists of 1092 days directly after the end of the data frame used for precalibration. Thus, the hyper-parameter optimization is done ex-ante from the forecasting point of view. Moreover, the performance of models

based on different datasets is evaluated as well. On the other hand, models do not mix the VST used in the precalibration and the main evaluation periods (meaning that hyper-parameter sets obtained using asinh transformation will only be used in the asinh-transformed test setting). For each of the four test (dataset, VST) settings: (PJM, ID), (PJM, asinh), (NP, ID), (NP, asinh), three possible hyper-parameter collections can be derived: From PJM, NP and GEFCom data.

The longest initial calibration window (1092 days from 01.01.2013 until 28.12.2015) coincides with the data used for hyper-parameter precalibration. The data from 29.12.2015 onwards (next 1092 days, up until 24.12.2018) is used to evaluate the forecasts. Exogenous series for both datasets are visually very similar across the whole range, whereas the behavior of price series varies slightly with time. For the PJM data, the volatility of price series decreases in the test period. On the other hand, an adverse effect can be observed for the Nord Pool market.

5. Methodology

5.1. Benchmark Models

The predictive performance of the proposed approach is compared to a selection of different benchmarks: From the parsimonious ARX model, through parameter-rich LASSO structure to the DNN identical to the one used in the proposed approach, but using the default parameter set. This ensures a comprehensive assessment of not only the potential gains from using the neural networks themselves, but also the gains imposed by the precalibration approach. The latter is especially important given the computational cost of the precalibration procedure.

5.1.1. The Arx Expert Model

The first benchmark is a parsimonious autoregressive structure with exogenous variables (ARX), widely used in EPF. The model was originally proposed by [32] and later adopted by many researchers [28,33,34]. It is referred to as an expert model [35,36] due to the fact that it uses regressors that are derived either by empirical testing or knowledge of experts. The model uses a multivariate setting, which means that for each day, 24 separate models are trained, one corresponding to each hour of the day. It is important to note that such a setting limits the information contained in the model, as no information about hours different than the forecasted is included (apart from the minimum price). Within the model, the price for hour h of day d is modeled via the following formula:

$$p_{d,h} = \underbrace{\beta_{h,1}\, p_{d-1,h} + \beta_{h,2}\, p_{d-2.h} + \beta_{h,3}\, p_{d-7,h}}_{\text{autoregressive effects}} + \underbrace{\beta_{h,4}\, p_{d-1,\min}}_{\text{non-linear effect}} + \underbrace{\beta_{h,5}\, z_{d,h}}_{\text{exogenous variable}} + \\ + \underbrace{\textstyle\sum_{i\in\{1,2,3\}} \beta_{h,i+5}\, D_i}_{\text{weekday dummies}} + \varepsilon_{d,h}, \tag{1}$$

where $p_{d-1,\min} = \min\left\{p_{d-1,1}, p_{d-1,2}, \ldots, p_{d-1,24}\right\}$ is the minimum of the previous day's 24 hourly prices. It serves as both the link with all yesterday's prices and a correction factor for the base price level. The variable $z_{d,h}$ is the load forecast for hour h (see Section 4). Lastly, the three weekday dummies D_1, D_2, D_3 correspond to Saturday, Sunday and Monday, respectively. They allow to better model the intraweek seasonality by describing the different price levels for Saturdays, Sundays and Mondays, when the dependence of prices on the prices from the day before is different than for the rest of the weekdays.

Given the ARX structure (1), the linear responses of price to the regressors are then estimated using ordinary least squares (OLS) on a fixed-length sample of past observations (i.e., the calibration window). The optimal length of the calibration window is strongly dependent on several factors, including the model used, the dataset, the forecasted period and the data transformation. To mitigate the inability of choosing the optimal calibration window ex-ante this study resorts to using three calibration window lengths, corresponding to roughly a year, two years and three years. This approach

allows for an improved performance and stability of the forecasts, while also limiting the computational burden, particularly the computations required to find the best performing calibration window lengths.

5.1.2. Lasso-Derived Parameter-Rich Model

The second benchmark can be seen as a natural enrichment of the ARX structure, as it does not add any regressor of different origin. It rather presents the information regarding all 24 h of the day among the regressors, instead of only selecting data corresponding to the forecasted hour. In theory, it should improve the predictions by adding very similar neighboring hours, as well as including better peak and off-peak base levels. Because of a greatly increased number of regressors (100 instead of 8), a different model estimation method is used, namely the least absolute shrinkage and selection operator (LASSO, originally proposed by [37]), as it limits the number of explanatory variables included in the model by selecting only the most relevant ones. Moreover, the process of regularization is automatic and requires the user to choose only the regularization parameter λ, which can be empirically fitted for the specific data through Cross Validation [38]. Such an approach was used in this study, with 100 values of the parameter tested via 10-fold CV trials. LASSO-estimated parameter-rich models are widely used in EPF literature [28,35,36,39].

The baseline model estimated via the LASSO is simply an augmented version of Equation (1). It models the price for hour h of day d via the formula:

$$p_{d,h} = \underbrace{\sum_{i=1}^{24} \beta_{h,i}\, p_{d-1,i} + \sum_{i=1}^{24} \beta_{h,24+i}\, p_{d-2.i} + \sum_{i=1}^{24} \beta_{h,48+i}\, p_{d-7,i}}_{\text{autoregressive effects}} + \\ + \underbrace{\beta_{h,73}\, p_{d-1,\min}}_{\text{non-linear effect}} + \underbrace{\sum_{i=1}^{24} \beta_{h,73+i}\, z_{d,i}}_{\text{exogenous variables}} + \underbrace{\sum_{i\in\{1,2,3\}} \beta_{h,97+i}\, D_i}_{\text{weekday dummies}} + \varepsilon_{d,h}, \quad (2)$$

where $p_{d-1,\min} = \min\left\{p_{d-1,1}, p_{d-1,2}, \ldots, p_{d-1,24}\right\}$ indicates the lowest price of yesterday, similarly as in Equation (1). Note, that in this case, the minimum is always included in the model twice (as opposed to the ARX model, where double inclusion takes place only once per 24 trained models). However, it does not create linearly dependent columns (the lowest price is observed in different hours of the day), and has an independent parameter estimated. By analogy, the D_i variables correspond to weekday dummies for Saturday, Sunday and Monday, whereas $z_{d,i}$ is the forecasted load for i-th hour of day d.

It is important to note, that albeit having exactly the same structure and inputs, the model is fitted separately for each hour of the day (i.e., we use a multivariate framework, see Section 5.1.1), and by extension, the regularization and selection take place for each hour independently. This allows to define the model in an universal way, which can be seen as the largest available parameter space, yet still allows to properly derive 24 potentially different models—each specifically tailored for a single hour.

5.1.3. The Default Dnn Benchmark

The last benchmark was constructed with a two-way comparison in mind. Firstly, it is used to compare the non-linear structure with the linear data description of the LASSO. The second comparison, however, is even more interesting, as it allows to measure the gains from using the computationally expensive two-step approach.

The benchmark assumes the use of all default hyper-parameter values (see Section 3.2). Unfortunately, there are two exceptions, without which the model would produce completely unusable forecasts. First of them is the `epochs` parameter that governs the maximum number of full iterations of the algorithm over the training set. The default value of a single epoch is not suitable in our case, therefore to provide a benchmark that is able to achieve reasonable results, the `epochs` value was set to 500. The value was chosen as the most popular entry among the best-performing models in the

precalibration phase, which also means that most of the precalibrated DNN models are trained for 500 epochs. There was no early stopping condition, meaning that each calibration of a neural network consisted of exactly 500 iterations over the full training dataset.

All neural network models use exactly the same information as in Equation (2), however, in an multi-output (vectorized, similarly to vector autoreggresion) framework with all 24 h of the day modeled at once. This is done due to the significant computational burden of model estimation. Modeling the whole 24-hour vector at once, but resorting to only one model trained per day of forecast, requires only about 5% of the CPU time when compared to the framework with 24 independent models, each predicting a single hour of the day (i.e., a typical framework used by ARX and LASSO benchmarks). Moreover, the nonlinear, dense structure of a network should in theory allow to minimize the potentially negative impact of such a setting, which has been shown to occur for some datasets [28]. Therefore, the DNN model can be visualized as shown in Figure 2.

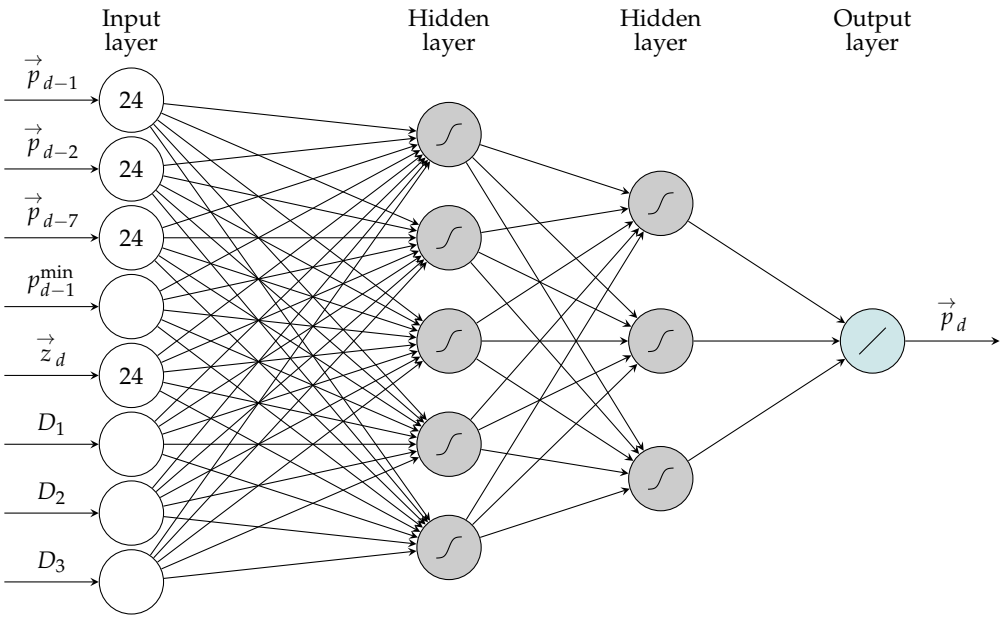


Figure 2. Visualization of the Deep Neural Network (DNN) structure, used in both the DNN benchmark and the main DNN models (hidden layer sizes are not reflected here; the net used in the study comprised 100 neurons in each hidden layer). The arrows over symbols correspond to 24 separate input/output neurons, with a dense connection structure.

As mentioned, the benchmark uses default hyper-parameter values in the Keras library. More precisely, the network structure of choice was a densely connected network with 100 input neurons, two hidden layers with 100 neurons each and 24 output neurons, the training length was fixed at 500 `epochs` of `ADAM` stochastic gradient descent-based optimizer with MAE loss function and a `batch_size` of 32. The `validation_split` parameter was set to 0 (i.e., there was only training set) and activation functions were set to `sigmoid` and `linear`, respectively for both hidden layers and for the output layer. See Section 5.3 for descriptions of the hyper-parameters.

The second exception from the default values was the activation function. When not specified explicitly, Keras always uses the linear activation (more specifically, the identity function). This results in a fully linear model, and does not perform well for EPF. To address that, all of the implemented activation functions were tested in this setting. The `sigmoid` activation scored well (despite being slightly outperformed by very similar `hard-sigmoid` in some cases), and was chosen as the ex-post best performer for the DNN default benchmark.

5.2. Fine-Tuned Neural Network Model

This DNN model uses exactly the same network structure and inputs as the DNN benchmark (see Section 5.1.3), but differs in terms of the hyper-parameters (see Sections 3.2 and 5.3). The main idea behind the two-step fine-tuning proposed in this study is to automatically evaluate hyper-parameter sets and use the best 5 of them to compute the final forecasts (the number 5 was chosen via an additional numerical experiment, using more yields diminishing returns, while increasing the computational time). The resulting forecasts are later averaged (creating an ensemble) across both the hyper-parameter sets and calibration window lengths. The aggregation process is described in Section 5.4. As the ensemble smoothens the random noise of the respective forecasts and generally performs better than the input forecasts, it is treated as the final output model of the approach.

5.3. Training Parameters Grid Search

The first stage of the approach involves forecasting the validation dataset using a wide range of hyper-parameter sets. To avoid making any assumptions, the parameter space is formed as a full grid of 1008 possible combinations of four parameters. There are two categorical options, namely the optimization algorithm and the activation function of all hidden neurons (which was identical for both hidden layers of deep networks). Secondly, two numerical (integer) parameters were included.

The first one—the batch size—governs the number of training samples presented at once to the optimizer. This translates to the number of samples after which the weights in the network are updated. The second numerical parameter controls the maximum epoch count. An epoch during training corresponds to one full iteration over all training samples. The maximum value of this parameter was chosen to be very high for this problem, while still maintaining the computational time on a rational level. Further increasing of this value would most likely result in overfitting. In conjunction with the batch size, it controls the frequency and the total count of weight updates in the network. It is important to note, that only fixed epoch counts are used in the study, i.e., the training in every case consisted of the specified iterations over the dataset and no early stopping condition was implemented. This is a possible point of future improvement, however, this approach—at a cost of a larger hyper-parameter space—provides an empirically validated combination of parameters, see Section 3.6 for discussion.

The specific values tested during the precalibration procedure were as follows:

- Seven optimization algorithms based on the stochastic gradient descent: `Adam`, `Adamax`, `Adagrad`, `Adadelta`, `Nadam`, `SGD`, `RMSprop`.
- Four activations: `eLU`, `ReLU`, `tanh`, `sigmoid`, same activation function was used for every hidden neurons, and the output layer was always linear.
- Four values of max epochs: 50, 100, 200, 500.
- Eight values of the batch size: 16, 32, 64, 96, 128, 192, 256, 384, 768.

As a result, for each of the (Data, VST) tuples, the list of hyper-parameters with corresponding errors is obtained. The errors are computed based on the forecast using a rolling calibration window of model and a specific parameter set. Note, that the considered hyper-parameter space is not exhaustive and did not cover e.g., the network depth or width, which were fixed.

5.4. Parameter Choice and Forecast Aggregation

After performing the parameter space grid search described in Section 5.3, the parameter sets are ranked according to the mean absolute error. Then, for each of the (Data, VST) tuples, the five sets yielding the lowest MAE are selected. The final forecasts use an ensemble of forecasts obtained using all five derived parameter sets, see Section 3.4. Note, that this kind of averaging (i.e., averaging outcomes of forecasts with different hyper-parameters) is applicable only to neural networks in the two-step approach.

However—as discussed in Section 3.3—there is a potential for improvements in forecast accuracy by using multiple models with an identical structure, but obtained using different calibration window lengths. As argued by [21], there are some window length combinations that yield robust gains, but the testing was not conducted for parameter-rich neural networks. Nevertheless, to optimize the outcomes and benefit from the technique to a limited extent, all models and benchmarks are evaluated using three lengths of calibration windows. The results are presented for each of them separately and as an ensemble of forecasts. The lengths: 52, 104 and 156 weeks, which corresponds to roughly 1, 2 and 3 years were chosen ad-hoc. This also applies to the forecasts obtained using the two-step approach; for each of the top five parameter sets, three forecasts are computed and additionally an ensemble over calibration windows is derived. Moreover, the ensemble across different parameter sets is created for each calibration window length as well as an ensemble consisting of 15 individual forecast runs.

6. Results

The section presents the obtained results and discusses the differences between the methods. The structure is as follows. Firstly, the results of the hyper-parameter space search are described with the analysis of the most often picked values. Secondly, the predictive accuracy of benchmark models is summarized. Next, the results of the fine-tuned DNNs are presented, and the importance of the precalibration dataset is investigated. Lastly, the computational feasibility is studied, with the assessment of the computational complexity of both the fine-tuned DNNs and the benchmark models.

6.1. Preliminary Hyper-Parameter Space Search

This section contains a brief discussion on the diversity of the top parameter sets across different precalibration conditions, as well as the most common occurrences of some of the parameter values. However, it is important to note that while there are some patterns, they might be solely caused by the random nature of the network training methods and might not be applicable to other (even similar) forecasting tasks. The precalibration procedure itself does not rely on those patterns and is in principle invariant to aforementioned randomness.

The first—and potentially the most important—observation regards the activation function used in the hidden layers. Among the precalibration parameter set results `sigmoid` was present most often (23 of 30 times), with 5 `tanh` and 2 exponential linear unit (`eLU`) entries. Interestingly, rectified linear unit (`ReLU`) activation did not appear even once among the top results, in spite of its popularity in the literature [5,6].

The second observation concerns the network training iterations over the dataset. Most commonly picked parameter sets were trained for 500 epochs. Only seven out of 30 best-performing parameter sets contained shorter (200 epochs) training.

As far as the optimizer is concerned, there was no unanimity. The `ADAM`, `ADAMAX` and `ADAGRAD` optimizers were chosen most commonly, however, every optimizer aside from stochastic gradient descent (`SGD`) occurred at least once in all 30 top picks.

6.2. Benchmark Models

Each of the benchmark models was computed for three calibration window lengths of 364, 728 and 1092 days and tested on all 1092 days of the out-of-sample test window. Additionally, an ensemble of these three forecasts was derived (by taking an arithmetic mean of the forecasts for each hour). The results in terms of MAE are presented in Table 1. It is important to note how well the default DNN benchmark preforms in all of the cases, being on par with or outperforming the second-best LASSO.

6.3. Fine-Tuned and Aggregated Forecasts

Each of the benchmark models was computed for three calibration window lengths of 364, 728 and 1092 days and tested on all 1092 days of the out-of-sample test window. Additionally, an ensemble

of these three forecasts was derived (by taking an arithmetic mean of the forecasts for each hour). The results in terms of MAE are presented in Table 1. It is important to note how well the default DNN benchmark preforms in all of the cases, being on par with or outperforming the second-best LASSO.

Table 1. Mean absolute errors for the benchmarks. The lowest score for each of the VSTs and datasets is marked in bold.

ID-Transformed Data								
Benchmark	**PJM**				**Nord Pool**			
	364	**728**	**1092**	**ensemble**	**364**	**728**	**1092**	**ensemble**
ARX1	3.476	3.581	4.128	3.614	2.690	2.608	2.563	2.587
LASSO	3.239	3.280	3.632	3.178	2.125	2.044	2.020	2.001
DNN default	3.183	3.241	3.970	**3.103**	2.018	2.060	2.043	**1.903**
Asinh-Transformed Data								
Benchmark	**PJM**				**Nord Pool**			
	364	**728**	**1092**	**ensemble**	**364**	**728**	**1092**	**ensemble**
ARX1	3.303	3.305	3.350	3.290	2.536	2.574	2.537	2.519
LASSO	3.076	3.047	3.054	3.000	2.048	2.039	2.009	1.984
DNN default	3.158	3.156	3.230	**2.989**	2.033	2.115	2.087	**1.945**

This section presents the results obtained using deep neural networks with hyper-parameters chosen via the precalibration procedure. The non-transformed data (i.e., with ID VST) and asinh-transformed data are treated separately. It is important to note, that due to the small differences between the hyper-parameter sets obtained using different datasets, presented here are the average error metrics of three ensembles, each based on different precalibration dataset.

Aggregate relative metrics comparing the fine-tuned ensembles with both DNN default and LASSO ensembles are presented in Table 2.

Table 2. Relative improvements in MAE when using the fine-tuned DNN ensemble instead of the benchmark models.

Fine-Tuned DNN vs.	**Dataset**	**ID**	**Asinh**
LASSO benchmark	Nord Pool	7.25%	5.28%
	PJM	7.44%	2.90%
default DNN benchmark	Nord Pool	2.23%	1.37%
	PJM	5.05%	1.77%

6.3.1. Models Calibrated to Raw Data

When considering models calibrated to raw data, for PJM dataset some of the individual (non-aggregated) networks were able to outperform the best benchmark (the DNN default benchmark, which is an ensemble itself) that scored 3.103, whereas final ensemble of tuned DNNs—2.929. The situation changes slightly for Nord Pool data, where ensembling is needed to outperform the best benchmark (also DNN default, with score of 1.903)—no single model does so. However, the final forecast of the fine-tuned DNN model—namely the ensemble across both the top parameter sets and different calibration window lengths—scores significantly lower (1.853). The difference is even larger when we consider the second best benchmark: The ensemble of 3 LASSO forecasts. Such a benchmark yielded MAEs of 3.178 and 2.001, respectively for PJM and Nord Pool, which makes the obtained DNN results even more significant and shows the shortcomings of using linear models on data as volatile as observed in electricity markets.

One can also conclude that averaging across different calibration window lengths yields very robust estimates, especially for PJM data. It is important to note that for PJM data, individual models trained using the 3 year long calibration window are underperforming when compared to shorter windows. This is mainly due to the period of extremely high volatility at the beginning of 2014. Using the 3 year long calibration window includes this information in the model for over a third of the out-of-sample test period. The impact of these outliers can be seen clearly for all three benchmarks. On the other hand, the forecasts chosen via the precalibration procedure are not such strongly penalized, which shows the potential of this methodology.

6.3.2. Models Calibrated to Asinh-Transformed Data

The overall outcome is very similar for the asinh-transformed data; after applying the VST, the fine-tuned DNN forecasts are also able to outperform the best benchmark. What is interesting to note, is that the 3 year long calibration window is no longer underperforming for the PJM dataset. Even more so, it performs on par or better than the shorter ones. The same observation can be made for the benchmarks (besides the default DNN benchmark, but that is most likely due to randomness), including the DNN default. The final ensemble scores were equal 2.909, 2.989 and 3.000, respectively for fine-tuned DNNs, DNN default and LASSO ensembles.

The Nord Pool data exhibit a situation where none of the single forecasts are able to outperform the best benchmark. The ensembles across calibration window lengths or hyper-parameter sets are mostly able to match the best benchmark score with little improvement. In this case, however, the gain from using all 15 forecasts instead of 3 (ensemble across calibration window lengths) or 5 (across top parameter sets) is very substantial, and the final forecast scores MAE of 1.875, which significantly outperforms both the DNN default (1.945) and LASSO ensembles (1.984).

Overall, the improvement imposed by the VST was negligible for ensemble forecasts: For Nord Pool data there was even a (slight) performance degradation, for both DNN default benchmark and the optimized networks. When the effects of applying the VST on linear structures (ARX, LASSO) are considered, this implies that the non-linear representation of a neural network is able to effectively model the strongly non-linear relations found in the data without resorting to any external techniques. This makes the neural networks a very universal modeling tool, as it removes the need of choosing the right VST for the data (the asinh function used in this study is only one of many possible functions, see [26] for a comparison performed using linear models).

6.4. Importance of the Precalibration Dataset for the Fine-Tuned Forecast

As can be seen in Figures 3 and 4, the fine-tuned neural network ensembles outperform significantly all of the benchmarks. Interestingly, the performance differences between different datasets of origin of the hyper-parameters are not strongly pronounced, with GEFCom data (dotted bars) being slightly better than the two other datasets. This result may seem counter-intuitive (it should be better to use the same origin of data for both the hyper-parameter selection and the testing), however the GEFCom data window for used for the precalibration serves well as an "exemplary" EPF dataset, with uniformly distributed spikes and without visible structural changes throughout.

These examples show that the hyper-parameters sets can successfully be found using data of different origin (but describing the same phenomenon), and that the GEFCom dataset is slightly better than the remaining two datasets that we have considered. However, regardless of the choice, the results show that an ex-ante hyper-parameter selection resulting in accurate forecasts is possible.

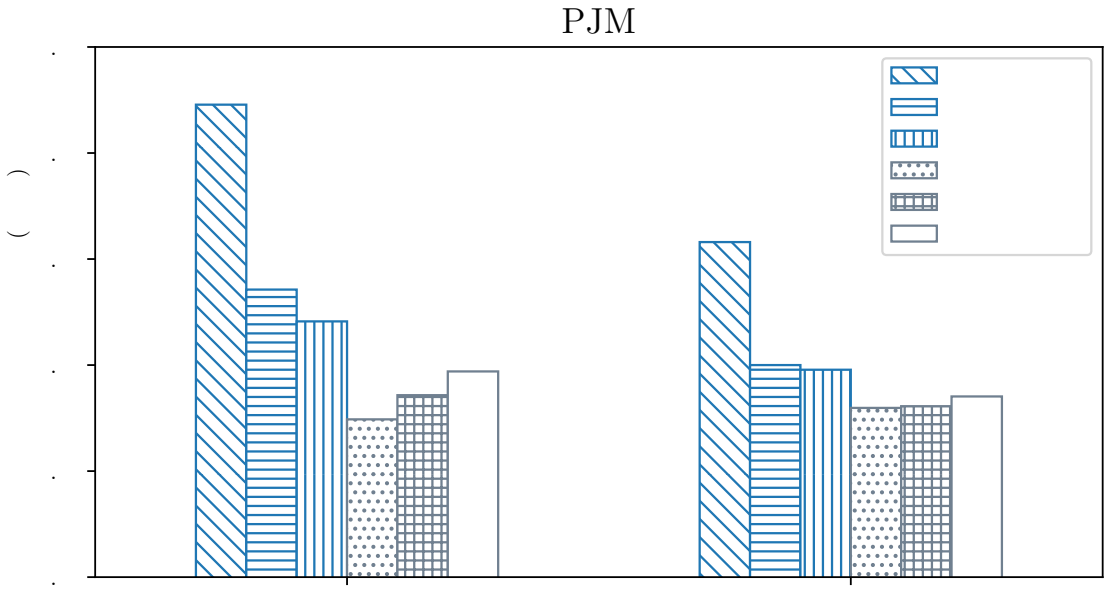


Figure 3. Visualization of the MAE errors for PJM data across the benchmark (darker shade) and fine-tuned (lighter shade) ensembles, separately for identity (ID) (left part) and asinh-transformed data (right part). Each bar represents the ensemble of all forecasts made using a given model (represented by a filling pattern). The `@DATA` notation refers to the precalibration dataset used for hyper-parameter selection.

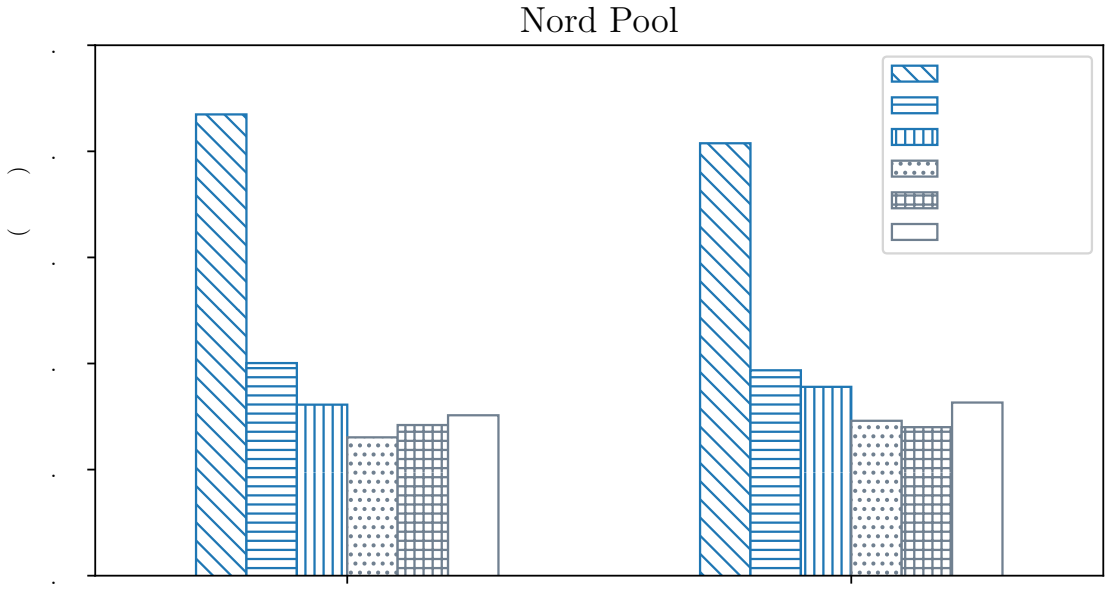


Figure 4. Visualization of the MAE errors for Nord Pool data across the benchmark (darker shade) and fine-tuned (lighter shade) ensembles, separately for ID (left part) and asinh-transformed data (right part). Each bar represents the ensemble of all forecasts made using a given model (represented by a filling pattern). The `@DATA` notation refers to the precalibration dataset used for hyper-parameter selection.

6.5. Computational Feasibility of Proposed Solutions

To assess the computational feasibility of the neural network methods, timed runs were performed on a machine equipped with an Intel Core i5-3570 CPU. Additionally, benchmark generation times are included for reference (ARX1 and LASSO). All listed times refer to the generation of price forecasts for one day (extracted from a longer sample to reduce the initialization overhead). The times listed for DNN models reflect the worst-case scenario over all of the hyper-parameter sets. Generation times for

final forecasts (i.e., an ensemble over three or 15 runs) are reported, and hyper-parameter optimization step is not considered here—it is assumed that an initial computational effort can be computed over a long period of time or on a multi-node computer cluster, and there is no need to rerun the optimization in the day-to-day operation. The results overview is presented in Table 3.

Table 3. Approximate times needed to generate one day of the forecasts using different methods. Note the differences in the units. The time listed for the DNN is a worst-case scenario, see the description in the text.

	ARX	LASSO	Fine-Tuned DNN
time	ca. 15 ms	30–40 s	below 120 s

The parsimonious structure of ARX model and an exact solution obtained via matrix operations allow for generation of an ensemble forecast for the next day in approximately 15 milliseconds, with negligible dependence on the VST. Both the LASSO benchmark and DNN model are orders of magnitude slower to obtain, due to the iterative estimation. LASSO, despite being a simple linear model, utilizes cross validation, which greatly increases the times needed to compute forecasts; it takes 41.2 and 32.8 s respectively for the ID-transformed and asinh-transformed data, with the difference coming most probably from the faster convergence.

Assessing the times for the DNN model is not straightforwards, hence, a worst-case scenario is presented. Specifically, we assume that for every calibration window length we take the longest time across all hyper-parameter sets as the time needed to generate a single forecast, which is then multiplied by 5 (ensemble across hyper-parameter sets). This results in the longest runtimes, 116 and 121 s respectively for ID and asinh transformations. The time does not vary much between them due to the fixed number of iterations over the dataset.

To sum up, the DNN model, even in the worst-case scenario is computationally feasible, i.e., the result can be obtained in near real-time on a contemporary hardware. The LASSO model can be accelerated significantly by either limiting the cross validation space for the regularization parameter or by using a fixed parameter (or a parameter based on information criteria) instead.

7. Conclusions

Obtaining the accurate day-ahead electricity price forecasts is crucial for any entity that heavily bases the profitability on the electricity prices, e.g., a power generating company. The accurate forecasts can greatly benefit the decision-making process in such a utility and can lead to better (in terms of the financial outcome) decisions being made. In the long run, availability of more accurate forecasts for the market participants can lead to lowering the price volatility, which in turn could result in e.g., lower risk of the long-term investments that are strongly dependent on the electricity prices. Another use of the more accurate price forecasts lies in the growing field of demand response and its role in the transformation of the electricity markets [40,41].

The deep neural network used in the study allows, along with the hyper-parameter optimization and ensembling scheme, for a significant outperformance of statistical-based approaches. This kind of modeling provides the user with a set of very well-performing and feature-rich tools, which—as shown by the default DNN benchmark—works relatively well with little tuning. However, the tuning itself is no longer an option, but rather a necessity, as using pure defaults would yield a 1-epoch trained network with the linear activation function in the hidden layers. The largest performance penalty would obviously come from training limited to 1 epoch, however, the activation function itself also plays a major role in achieving a well-rounded model.

The two-step approach—as proposed here—proves to be a very robust technique, with excellent performance across almost all of the test scenarios and can be applied in practice. It does, however, come at a price. The generation of the precalibration forecasts alone is a serious computational difficulty. On the other hand, the approach uses many independent forecasts (in this case, 1008 for each dataset,

VST pair), which makes it feasible to run in parallel on high-performance computer clusters or using cloud computing. With that in mind, the precalibration procedure does not need to be reapplied (often). This study assumed that the precalibration procedure would not be repeated for the whole three year long test period. Moreover, if the precalibration was to be updated periodically (e.g., every three months) it would be much faster because only incremental updates of the forecasts are needed, which would not be straightforward to implement using step-by-step hyper-parameter optimization.

The main takeaway from this study is that the hyper-parameters can be automatically tailored to fit a specific task, and the selection can be performed ex-ante. Such forecasts consistently outperform even the DNN benchmark used in the study. Looking at the results from a different perspective, all of the neural network-based models were much better-performing than the LASSO model estimated using the same information. The performance gain is best visible on the non-transformed PJM data, which exhibits a change in the volatility of the price series. Non-linear layers enable the model to effectively cope with the data that proves difficult to forecast using linear models.

The research, however, does not conclude the idea of using a neural network precalibrated using a hyper-parameter grid search fully. While a well-performing application is presented, there are still interesting aspects left for future research, such as the network topology (deep or shallow, dense or sparse?) or the periodic hyper-parameter recalibration.

Funding: This work was partially supported by the Ministry of Science and Higher Education (MNiSW, Poland) through grant No. 0219/DIA/2019/48. Calculations have been carried out using resources provided by the Wrocław Center for Networking and Supercomputing (WCSS; http://wcss.pl) under grant no. 466.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Huisman, R.; Huurman, C.; Mahieu, R. Hourly electricity prices in day-ahead markets. *Energy Econ.* **2007**, *29*, 240–248. [CrossRef]
2. Cruz, A.; Muñoz, A.; Zamora, J.L.; Espínola, R. The effect of wind generation and weekday on Spanish electricity spot price forecasting. *Electr. Power Syst. Res.* **2011**, *81*, 1924 – 1935. [CrossRef]
3. Marcjasz, G.; Uniejewski, B.; Weron, R. On the importance of the long-term seasonal component in day-ahead electricity price forecasting with NARX neural networks. *Int. J. Forecast.* **2019**, *35*, 1520 –1532. [CrossRef]
4. 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*, 890–903. [CrossRef]
5. Lago, J.; Ridder, F.D.; Schutter, B.D. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. *Appl. Energy* **2018**, *221*, 386–405. [CrossRef]
6. Chinnathambi, R.A.; Plathottam, S.J.; Hossen, T.; Nair, A.S.; Ranganathan, P. Deep Neural Networks (DNN) for Day-Ahead Electricity Price Markets. In Proceedings of the 2018 IEEE Electrical Power and Energy Conference (EPEC), Toronto, ON, Canada, 10–11 October 2018; pp. 1–6.
7. Schnürch, S.; Wagner, A. Machine Learning on EPEX Order Books: Insights and Forecasts. *arXiv* **2019**, arXiv:1906.06248.
8. Zhang, J.; Tan, Z.; Wei, Y. An adaptive hybrid model for short term electricity price forecasting. *Appl. Energy* **2020**, *258*, 114087. [CrossRef]
9. Gareta, R.; Romeo, L.M.; Gil, A. Forecasting of electricity prices with neural networks. *Energy Convers. Manag.* **2006**, *47*, 1770 – 1778. [CrossRef]
10. Kuo, P.H.; Huang, C.J. An Electricity Price Forecasting Model by Hybrid Structured Deep Neural Networks. *Sustainability* **2018**, *10*, 1280. [CrossRef]
11. Xie, X.; Xu, W.; Tan, H. The Day-Ahead Electricity Price Forecasting Based on Stacked CNN and LSTM. In *Intelligence Science and Big Data Engineering*; Peng, Y., Yu, K., Lu, J., Jiang, X., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 216–230.
12. Zahid, M.; Ahmed, F.; Javaid, N.; Abbasi, R.A.; Zainab Kazmi, H.S.; Javaid, A.; Bilal, M.; Akbar, M.; Ilahi, M. Electricity Price and Load Forecasting using Enhanced Convolutional Neural Network and Enhanced Support Vector Regression in Smart Grids. *Electronics* **2019**, *8*, 122. [CrossRef]

13. Yang, W.; Wang, J.; Niu, T.; Du, P. A novel system for multi-step electricity price forecasting for electricity market management. *Appl. Soft Comput.* **2020**, *88*, 106029. [CrossRef]
14. 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. [CrossRef]
15. Kath, C.; Ziel, F. The value of forecasts: Quantifying the economic gains of accurate quarter-hourly electricity price forecasts. *Energy Econ.* **2018**, *76*, 411–423. [CrossRef]
16. Paraschiv, F.; Erni, D.; Pietsch, R. The impact of renewable energies on EEX day-ahead electricity prices. *Energy Policy* **2014**, *73*, 196–210. [CrossRef]
17. Díaz, G.; Coto, J.; Gómez-Aleixandre, J. Prediction and explanation of the formation of the Spanish day-ahead electricity price through machine learning regression. *Appl. Energy* **2019**, *239*, 610–625. [CrossRef]
18. 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*, 218–230. [CrossRef]
19. Halužan, M.; Verbič, M.; Zorić, J. Performance of alternative electricity price forecasting methods: Findings from the Greek and Hungarian power exchanges. *Appl. Energy* **2020**, *277*, 115599. [CrossRef]
20. Hong, T.; Pinson, P.; Fan, S.; Zareipour, H.; Troccoli, A.; Hyndman, R.J. Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond. *Int. J. Forecast.* **2016**, *32*, 896–913. [CrossRef]
21. Hubicka, K.; Marcjasz, G.; Weron, R. A Note on Averaging Day-Ahead Electricity Price Forecasts Across Calibration Windows. *IEEE Trans. Sustain. Energy* **2019**, *10*, 321–323. [CrossRef]
22. Marcjasz, G.; Serafin, T.; Weron, R. Selection of Calibration Windows for Day-Ahead Electricity Price Forecasting. *Energies* **2018**, *11*, 2364. [CrossRef]
23. Serafin, T.; Uniejewski, B.; Weron, R. Averaging Predictive Distributions Across Calibration Windows for Day-Ahead Electricity Price Forecasting. *Energies* **2019**, *12*, 2561. [CrossRef]
24. Janczura, J.; Trück, S.; Weron, R.; Wolff, R. Identifying spikes and seasonal components in electricity spot price data: A guide to robust modeling. *Energy Econ.* **2013**, *38*, 96–110. [CrossRef]
25. Diaz, G.; Planas, E. A Note on the Normalization of Spanish Electricity Spot Prices. *IEEE Trans. Power Syst.* **2016**, *31*, 2499–2500.
26. Uniejewski, B.; Weron, R.; Ziel, F. Variance Stabilizing Transformations for Electricity Spot Price Forecasting. *IEEE Trans. Power Syst.* **2018**, *33*, 2219–2229. [CrossRef]
27. Schneider, S. Power spot price models with negative prices. *J. Energy Mark.* **2011**, *4*, 77–102. [CrossRef]
28. Ziel, F.; Weron, R. Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks. *Energy Econ.* **2018**, *70*, 396–420. [CrossRef]
29. Geman, S.; Bienenstock, E.; Doursat, R. Neural Networks and the Bias/Variance Dilemma. *Neural Comput.* **1992**, *4*, 1–58. [CrossRef]
30. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
31. Chollet, F. Keras. 2015. Available online: https://keras.io (accessed on 4 September 2020).
32. Misiorek, A.; Trück, S.; Weron, R. Point and interval forecasting of spot electricity prices: Linear vs. non-linear time series models. *Stud. Nonlinear Dyn. Econom.* **2006**, *10*, 2. [CrossRef]
33. Gaillard, P.; Goude, Y.; Nedellec, R. Additive models and robust aggregation for GEFCom2014 probabilistic electric load and electricity price forecasting. *Int. J. Forecast.* **2016**, *32*, 1038–1050. [CrossRef]
34. Weron, R.; Misiorek, A. Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models. *Int. J. Forecast.* **2008**, *24*, 744–763. [CrossRef]
35. Uniejewski, B.; Nowotarski, J.; Weron, R. Automated Variable Selection and Shrinkage for Day-Ahead Electricity Price Forecasting. *Energies* **2016**, *9*, 621. [CrossRef]
36. Ziel, F. Forecasting Electricity Spot Prices Using LASSO: On Capturing the Autoregressive Intraday Structure. *IEEE Trans. Power Syst.* **2016**, *31*, 4977–4987.
37. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B* **1996**, *58*, 267–288. [CrossRef]
38. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer Series in Statistics; Springer: New York, NY, USA, 2009; Chapter 7.
39. Marcjasz, G.; Uniejewski, B.; Weron, R. Beating the Naïve—Combining LASSO with Naïve Intraday Electricity Price Forecasts. *Energies* **2020**, *13*, 1667. [CrossRef]

40. O'Connell, N.; Pinson, P.; Madsen, H.; O'Malley, M. Benefits and challenges of electrical demand response: A critical review. *Renew. Sustain. Energy Rev.* **2014**, *39*, 686–699. [CrossRef]
41. Larsen, E.M.; Pinson, P.; Leimgruber, F.; Judex, F. Demand response evaluation and forecasting—Methods and results from the EcoGrid EU experiment. *Sustain. Energy Grids Netw.* **2017**, *10*, 75–83. [CrossRef]

Paper 2

Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark

Jesus Lago, Grzegorz Marcjasz, Bart De Schutter, Rafał Weron

Applied Energy 293 (2021) 116983

Contents lists available at ScienceDirect

Applied Energy

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

Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark

Jesus Lago [a,*], Grzegorz Marcjasz [b], Bart De Schutter [a], Rafał Weron [b]

[a] Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands
[b] Department of Operations Research and Business Intelligence, Wrocław University of Science and Technology, Wrocław, Poland

ARTICLE INFO

Keywords:
Electricity price forecasting
Regression model
Deep learning
Open-access benchmark
Forecast evaluation
Best practices

ABSTRACT

While the field of electricity price forecasting has benefited from plenty of contributions in the last two decades, it arguably lacks a rigorous approach to evaluating new predictive algorithms. The latter are often compared using unique, not publicly available datasets and across too short and limited to one market test samples. The proposed new methods are rarely benchmarked against well established and well performing simpler models, the accuracy metrics are sometimes inadequate and testing the significance of differences in predictive performance is seldom conducted. Consequently, it is not clear which methods perform well nor what are the best practices when forecasting electricity prices. In this paper, we tackle these issues by comparing state-of-the-art statistical and deep learning methods across multiple years and markets, and by putting forward a set of best practices. In addition, we make available the considered datasets, forecasts of the state-of-the-art models, and a specifically designed `python` toolbox, so that new algorithms can be rigorously evaluated in future studies.

1. Introduction

The increasing penetration of renewable energy sources in today's power systems makes electricity generation more volatile and the resulting electricity prices harder to predict than ever before [1–4]. On the other hand, advances in *electricity price forecasting* (EPF) constantly provide new tools with the ultimate objective of narrowing the gap between predictions and actual prices. The progress in this field, however, is not steady and easy to follow. In particular, as concluded by all major review publications, comparisons between EPF methods are very difficult since studies use different datasets, different software implementations, and different error measures; the lack of statistical rigor complicates these analyses even further [5–8]. In particular:

- There are several studies comparing *machine learning* (ML) and statistical methods but the conclusions of these studies are contradictory. Typically, studies considering advanced statistical techniques only compare them with simple ML methods [9–11] and show that statistical methods are obviously better. Conversely, studies proposing new ML methods only compare them with simple statistical methods [12–16] and show that ML models are more accurate.
- In many of the existing studies [17–23] the testing periods are too short to yield conclusive results. In some cases, the test datasets are limited to one-week periods [22,24–30]; this ignores the problem of special days, e.g. holidays, and is not representative for the performance of the proposed algorithms across a whole year. As argued in [5], to have meaningful conclusions, the test dataset should span at least a year.
- Some of the existing papers do not provide enough details to reproduce the research. The three most common issues are: (i) not specifying the exact split between the training and test dataset [31–37], (ii) not indicating the inputs used for the prediction model [35,36,38–40], and (iii) not specifying the dataset employed [21,33,41,42]. This obviously prevents other researchers from validating the research results.

These three problems have aggravated over the last years with the increase in popularity of *deep learning* (DL). While new published papers on DL for EPF appear almost every month, and most claim to develop models that obtain state-of-the-art accuracy, the comparisons performed in those papers are very limited. Particularly, the new DL methods are usually compared with simpler ML methods [28,30,43–47]. This is obviously problematic as such comparisons are not fair. Moreover, as the proposed methods are not compared with other DL algorithms, new DL methods are continuously being proposed but it is unclear how the different models perform relative to each other.

* Corresponding author.
E-mail address: j.lagogarcia@tudelft.nl (J. Lago).

https://doi.org/10.1016/j.apenergy.2021.116983
Received 29 December 2020; Received in revised form 15 April 2021; Accepted 18 April 2021

Similar problems arise in the context of *hybrid methods*. In recent years, very complex hybrid methods have been proposed. Typically, these hybrid models are based on combining a decomposition technique, a feature selection method, an ML regression model, and sometimes a meta-heuristic algorithm for optimization purposes. As with DL algorithms, these studies usually avoid comparisons with well-established methods [21,25,34,42,48–50] or resort to comparisons using outdated methodologies [22,24,26,37,41,51,52]. In addition, while a specific genetic algorithm or decomposition technique is considered, most of the studies do not analyze the effect of selecting a variant of these techniques [21,24,50–52]. Thus, the relative importance of each of the different components of the hybrid methods it is not even clear.

1.1. Motivation and contributions

The above mentioned problems call for three actions. Firstly, implementing in a popular programming environment (e.g. `python`) and making available *a set of simple but powerful open-source forecasting methods*, which can potentially obtain state-of-the-art performance, and that researchers can easily use to evaluate any new forecasting model.

Secondly, collecting and making freely available to the EPF community *a set of representative benchmark datasets* that researchers can use to evaluate and compare their methods using long testing periods. Although, some datasets are available for download without restrictions, e.g. as supplements to published articles [53] or sample transaction data [54], they are typically limited in scope (one market, a 2–3 year timespan or price series only). Hence, conclusions from such datasets are limited, results can hardly be extrapolated to other markets, and the relevance of the studies using such data are not entirely clear.

Thirdly, *putting forward a set of best practices* so that the conclusions of EPF studies become more meaningful and fair comparisons can be made.

In this paper, we try to tackle the above via three distinct contributions:

1. We analyze the existing literature and select what could arguably be considered as state-of-the-art among statistical and machine learning methods: the *Lasso Estimated AutoRegressive* (LEAR) model[1] [55] and the *Deep Neural Network* (DNN) [57], a relatively simple and automated DL method that optimizes hyperparameters and features using Bayesian optimization. Then, we make our models available to other researchers as part of an open-source `python` library (https://github.com/jeslago/epftoolbox) specially designed to provide a common research framework for EPF research [58]. Besides the models, we also provide extensive documentation [59] for the library.
2. We propose a set of five open-access benchmark datasets spanning six years each, that represent a range of well-established day-ahead, auction type power markets from around the globe. The datasets contain day-ahead electricity prices at an hourly resolution and two relevant exogenous variables each. They can be accessed from the mentioned `python` library [58]. Together with the datasets, the library also includes the forecasts of the open-access methods across the five benchmark datasets so that researchers can quickly make further comparisons without having to re-train or re-estimate the models.
3. We provide a set of best practice guidelines to conduct research in EPF so that new studies are more sound, reproducible, and the obtained conclusions are stronger. In addition, we include some of the guidelines, e.g. adequate evaluation metrics or statistical tests, in the mentioned `python` library [58] to provide a common research framework for EPF research.

[1] Originally introduced in [55] under the name *LassoX* and based on the *fARX* model, a parameter-rich autoregressive specification with exogenous variables. The name refers to the *least absolute shrinkage and selection operator* (LASSO) [56] used to jointly select features and estimate their parameters.

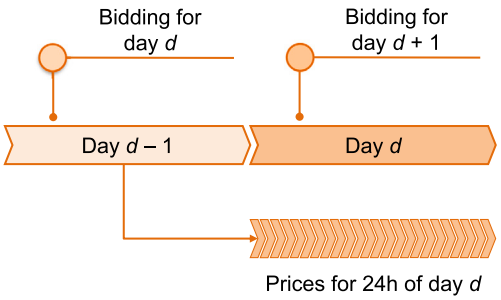


Fig. 1. Illustration of the *day-ahead* auction market, where wholesale sellers and buyers submit their bids before gate closure on day $d-1$ for the delivery of electricity during day d; the 24 hourly prices for day d are set simultaneously, typically around midday.

1.2. Paper structure

The remainder of the paper is organized a follows. Section 2 performs a literature review of the current state of EPF. Sections 3 and 4 respectively present the open-access benchmark datasets and the open-source benchmark models. Section 5 describes the set of guidelines and best practices when performing research in EPF. Section 6 discusses the forecasting results for all five datasets. Finally, Section 7 provides a summary and a checklist of the requirements for meaningful EPF research.

2. Literature review

The field of EPF aims at predicting the spot and forward prices in wholesale markets, either in a point or probabilistic setting. However, given the diversity of trading regulations available across the globe, EPF always has to be tailored to the specific market. For instance, the workhorse of European short-term power trading is the *day-ahead* market with its once-per-day uniform-price auction, see Fig. 1. On the other hand, the Australian National Electricity Market operates as a real-time power pool, where a dispatch price is determined every five minutes and six dispatch prices are averaged every half hour as pool prices [60], while electricity forward markets share many aspects with those of other energy commodities (oil, gas, coal), and quite often are only financially settled [61].

As the field of EPF is very diverse, a complete literature review is out of the scope of this paper. Instead, this section is intended to provide an overview of the three families of methods, i.e. statistical, ML, and hybrid methods, proposed for point forecasting in day-ahead markets since 2014, i.e. since the last comprehensive literature review of Weron [5]. The more recent reviews either focused on short-term [6] and medium-/long-term [7] probabilistic EPF, were not that comprehensive in scope [62,63], or concerned electricity derivatives [61]. Furthermore, our survey puts a special emphasis on DL and hybrid methods as this is the area of EPF characterized by the most rapid development and, at the same time, troubled by non-rigorous empirical studies which motivated us to write this paper in the first place.

2.1. Statistical methods

Most models in this class rely on linear regression and represent the dependent (or output) variable, i.e. the price $p_{d,h}$ for day d and hour h, by a linear combination of independent (or predictor, explanatory) variables, also called regressors, inputs, or features:

$$p_{d,h} = \boldsymbol{\theta}_h \mathbf{X}_{d,h} + \varepsilon_{d,h}, \tag{1}$$

where $\boldsymbol{\theta}_h = [\theta_{h,0}, \theta_{h,1}, \dots, \theta_{h,n}]$ is a row vector of coefficients specific to hour h, $\mathbf{X}_{d,h} = [1, X^1_{d,h}, \dots, X^n_{d,h}]^\top$ is a column vector of inputs and $\varepsilon_{d,h}$ is an error term; the intercept $\theta_{h,0}$ can be set to zero if the data is

demeaned beforehand. Note that here we are using a notation common in day-ahead forecasting, which emphasizes the vector structure of these price series, see Fig. 1. Alternatively we could use single indexing: p_t with $t = 24d + h$. Although the multivariate modeling framework has been shown to be marginally more accurate than the univariate counterpart, both approaches have their pros and cons [64,65].

In the last few years, there have been several key contributions in the field of statistical methods for EPF. Arguably, the most relevant of them has been the appearance of linear regression models with a large number of input features that utilize regularization techniques [56,66]. Classically, the regression model in (1) is estimated using *ordinary least squares* (OLS) by minimizing the *residual sum of squares* (RSS), i.e. squared differences between the predicted and actual values. However, if the number of regressors is large, using the *least absolute shrinkage and selection operator* (LASSO) [56] or its generalization the *elastic net* [66] as implicit feature selection methods have been shown to improve the forecasting results [55,57,64,67–69], also in intraday [70,71] and probabilistic [72,73] EPF. In particular, by jointly minimizing the RSS and a penalty factor of the model parameters (see Section 4.2 for details), these two implicit regularization techniques set some of the parameters to zero and thus effectively eliminate redundant regressors. As shown in the cited studies, these parameter-rich[2] regularized regression models exhibit superior performance. It is important to note that such an approach, called here *Lasso Estimated AutoRegressive* (LEAR), is in fact hybrid since LASSO (and electric nets) are considered ML techniques by some authors. However, we classify it as statistical because the underlying model is *autoregressive* (AR).

Aside from proposing parameter-rich models and advanced estimators, researchers have also improved the field by considering a variety of additional preprocessing techniques. Most notably, models using so-called variance stabilizing transformations [9,64,74,75] and long-term seasonal components [76–79] have been proposed and shown to result in statistically significant improvements. However, the applicability of these two techniques varies greatly: due to very common occurrence of price spikes, variance stabilizing transformations have become a standard and replaced the commonly used logarithmic transformation (no longer applicable due to zeros and negative values[3]) to normalize electricity prices. By contrast, the applicability of long-term seasonal components has been more limited and it is unknown whether their beneficial effect is limited to relatively parsimonious regression models or also holds for parameter-rich models.

A third innovation in the field is an ensemble (i.e. a method that combines individual forecasting models) that combines multiple forecasts of the same model calibrated on different windows. In this context, two different studies [80,81] showed that the best results are obtained with a combination of a few short (spanning 1–4 months) and a few long calibration windows (of approximately two years). Said ensembles were able to significantly outperform predictions obtained for the best ex-post selected calibration window [80–82]. But again, it has not been shown to date whether this effect is limited to relatively parsimonious regression models or also holds for LEAR models.

Interestingly, as [83] argue in an econometric context, in the presence of structural breaks it may be advisable to combine forecasts obtained for calibration windows of different lengths. Longer windows allow for a better fit, while shorter faster adapt to changes. Hence, if a structural break appears, like the COVID-19 pandemic, using models calibrated to shorter windows may better capture changes in the price dynamics. A different, but a potentially also appealing approach has been recently suggested in [84,85]. The authors assume that fundamental and price time series exhibit recurrent regimes with similar dynamics and employ cluster analysis – k-means [84] or k-nearest neighbors [85] – to identify such periods in the past. Then they calibrate models only on data segments which resemble current conditions. As such, they are able to eliminate subperiods that include structural breaks from the calibration sample.

Finally, note that in contrast to financial econometrics, where heteroskedasticity is a basic building block of many state-of-the-art approaches [86], models with *generalized autoregressive conditional heteroskedastic* (GARCH) residuals have been tried for EPF without much success, for a review and discussion see [5]. For instance, [57] compare 27 different models, among them an ARIMA–GARCH model, and find that it performs comparable to a much simpler AR model and ca. 1.5 times worse than the DNN model defined in Section 4.3. As [87] argue, GARCH effects diminish when fundamental and behavioral drivers of the electricity price volatility are taken into account and allowing for the time-varying responses of prices to fundamentals can yield more precise volatility estimates than an explicit GARCH specification.

2.2. Deep learning

In the last five years, a total of 28 deep learning papers in the context of EPF have been published.[4] Moreover, this number has been steadily increasing: while in 2017 there was only one paper, in 2018 there were 11, and in 2019 there were 16. Despite this trend, most of the published studies are very limited: the comparisons are too simplistic, e.g. avoid state-of-the-art statistical methods, and their results cannot be generalized.

The first published DL paper [12] proposes a deep learning network using stacked denoising autoencoders. The paper, despite being the first, provides a better evaluation than most studies: the new method is compared not only against machine learning techniques but also against two statistical methods. Yet, the evaluation is limited as it only considers three months of test data and simple benchmark models. In the second published DL article [57], a DNN for modeling market integration is proposed. While the method is evaluated over a year of data, the proposed model is not compared against other machine learning or statistical methods.

In the third published paper [57], four DL models (a DNNs, two *recurrent neural networks (RNNs)*, and a *convolutional network (CNN)*) are proposed. This study is, to the best of our knowledge, the most complete study up to date. In particular, the proposed DL models are compared using a whole year of data against a benchmark of 23 different models, including 7 machine learning models, 15 statistical methods, and a commercial software. Moreover, among the statistical methods, the comparison includes the fARX-Lasso and fARX-EN, i.e. the state-of-the-art statistical methods. While the study shows the superiority of the DL algorithms, very strong conclusions are not possible as the study only considers a single market.

The studies that followed in 2018 focused on one of three topics: (1) evaluating the performance of different deep recurrent networks [13,23,37,88]; (2) proposing new hybrid methods based on CNNs and LSTMs [14,44,89,90]; or (3) employing regular DNN models [23]. Independently of the focus, they were all more limited than the first and the third studies [12,57] as they failed to compare the new DL

[2] We define a parameter-rich linear model as a model with multiple regressors (dozens, hundreds).

[3] The logarithm of 0 or of a negative value is undefined.

[4] This data is primarily based on a Scopus search in the title, abstract, and keywords: `TITLE-ABS-KEY((((''forecasting electricity'') OR (''predicting electricity'')) AND ((''electricity spot'') OR (''electricity day-ahead'') OR (''electricity price''))) OR ((((''price forecasting'') OR (''price prediction'') OR (''forecasting price'') OR (''predicting price'') OR (''forecasting spikes'') OR (''forecasting VAR'')) AND ((''electricity spot price'') OR (''electricity price'') OR (''electricity market'') OR (''day-ahead market'') OR (''power market''))) AND (''deep'') AND (''learning''))`. We have also run a second, more general query replacing ("deep") AND ("learning") by ("neural") AND ("network"), however, only a few additional papers have been identified.

models with state-of-the-art statistical methods and/or to employ long enough datasets to derive strong conclusions.

In detail, [13] studies the use of RNNs for forecasting electricity prices but the comparison is done in a single market and against simple statistical methods: a seasonal *auto regressive integrated moving average* (ARIMA) model, a Markov regime-switching model, and a self exciting threshold model. Moreover, while the comparison includes other DL methods, it avoids comparison with simpler ML techniques. Ref. [44] proposes a hybrid DL method composed of a CNN and a *long short-term memory* (LSTM) neural network (a type of recurrent network) for forecasting balancing prices. However, the new model is only compared against simple ML benchmarks and the evaluation is done using different periods comprising three months for training and 1 month for testing. Similarly, [14] proposes another hybrid model combining a CNN and an LSTM, but the model is only compared against two simple statistical methods: an *auto regressive moving average* (ARMA) and a GARCH model.

In [23] a regular DNN model is proposed but the model is only evaluated on a test dataset comprising a single day and compared against a simple *multilayer perceptron* (MLP). In [29], the use of an LSTM model for EPF is evaluated, but the method is only compared with three neural networks and a simple statistical method, and the evaluation is done using only 4 weeks of data. Likewise, [88] proposes a model based on an LSTM but a comparison against other methods is not performed and the test dataset only comprises 2 weeks of data. In [37], another LSTM model is proposed but, as in other studies, the test dataset comprises a few months of data and the method is only compared against a simple decision tree and a support vector regressor; moreover, the exact split between the training and test dataset is not specified and it is unclear what is exactly the performance of the model. An exception to these studies is [91] which proposes a series of DL models and compares them for a year of data against several advanced statistical methods such as LASSO and a simpler ML method. The main drawbacks of the study are that it is based on a single market and that it only considers a simple ML method as a benchmark. In addition, the study focuses on intraday electricity prices, while most of the literature (including the current paper) considers forecasting day-ahead electricity prices.

In 2019, the main focus of the papers was the same as in 2018: (1) evaluating the performance of different deep recurrent networks (mostly LSTMs) [16,30,45,47,92–94], (2) proposing new hybrid deep learning methods usually based on LSTMs and CNNs [17,28,36,92,95–97], or (3) employing regular DNN models [15,46,98]. Similarly, as with most studies in 2018, the new studies were more limited than [12,57] as no comparisons with state-of-the-art statistical methods were made and long test datasets were seldom used. In this context, even though some studies [16,98] tried to compare the proposed methods with existing DL models [57], they either failed to re-estimate the benchmark models for the new case study [16] or they overfitted the DL benchmark models [98].

In detail, [30] proposes different LSTM models but the new models are only compared against 5 other ML techniques and using a test period of 4 weeks. In [28], a CNN model is proposed but the new model is just compared against three simple ML methods and using a test dataset that comprises a week. In [45], a model based on an LSTM is proposed but it is only compared against three simple ML methods and for a period of 12 weeks. In [46], the performance of a DNN is compared to that of an SVR model and, as the comparison only includes these two models, it is obviously very limited. In [15], a DNN is used as part of a two-step forecasting method; as in many other studies, the comparison is performed for one month of data and limited to two simple ML models (a SVR and an MLP) and a standard linear model. In [47], two DL models are proposed but the models are only compared to very simple ML methods (extreme learning machines and standard MLPs) and using a test dataset spanning eight months. In [16], a bidirectional LSTM to forecast prices in the French market is proposed; however, the study only considers historical prices as input features and the proposed method is only compared against DL models and a simple autoregressive model. In addition, the benchmark DL models are copied from [57] (a completely different case study that considers exogenous inputs and a different market) without re-tuning the hyperparameters to the new case study.

In [98], a neural network that uses data from order books is proposed and compared against DL methods from the literature, e.g. the ones proposed in [57]. While the new model outperforms existing DL methods, the DL methods from the literature are trained to overfit the training dataset.[5] Therefore, the comparison is not meaningful (the DL benchmark models will necessarily perform poorly in the test dataset) and it cannot be assessed how the new model performs. In [95], a hybrid DL forecasting method is proposed based on stacked denoising autoencoders for pre-training, regular autoencodes for feature selection, and a rough DNN as a forecasting method. As in other studies, the method is only compared against simple ML models. Moreover, the importance of each of the four modules of the hybrid method is not studied and the authors do not re-calibrate the models with new data: the models are trained once and evaluated over a whole year. Similarly, [96] proposes a CNN hybrid model that uses mutual information, random forests, gray correlation analysis, and recursive feature elimination for feature selection. Unlike most models, the algorithm is trained to classify prices instead of predicting their scalar values; however, details of how this process is done are not provided. In addition, the method is only compared against simple ML methods and evaluated for less than a year of data (the study uses one year for testing and training but the split is not specified). Likewise, [36] proposes a hybrid model based on CNNs and RNNs in the context of microgrids; as in other studies, the method is evaluated in a small dataset, it is not compared against state-of-the-art statistical methods, and the exact split between training and test datasets is not specified.

2.3. Hybrid methods

Within the field of EPF, the research area that has received the most attention in the last 5 years has been hybrid forecasting methods. In this time frame, more than 100 articles proposing new hybrid methods have been published,[6] i.e. approximately 5 times more than articles based on DL. Hybrid models are very complex forecasting frameworks that are composed of several algorithms. Usually, they comprise at least two of the following five modules:

- An algorithm for decomposing data.
- An algorithm for feature selection.
- An algorithm to cluster data.
- One or more forecasting models whose predictions are combined.
- Some type of heuristic optimization algorithm to either estimate the models or their hyperparameters.

[5] In the training dataset, the proposed model and some naive ML benchmark models yield a *root mean square error* (RMSE) of ca. 6. For the test dataset, for the same models, the RMSE is between 9 and 12. By contrast, the training error of the benchmark DL model is 2, and the test error is 20. Having a training error that is 1/3 of the error of other models but a test error that is 10 times larger than the training error is a clear sign for overfitting (especially when for the rest of the models the test error is just 1.5 larger than the training error).

[6] This data is based on two searches in Scopus looking for keywords in the title, abstract, and keywords. The first search is based on the following query `TITLE-ABS-KEY(((forecast*) OR (predict*)) AND (electricity) AND (price*) AND (hybrid))`. The second search is very similar but replacing the keyword `hybrid` by `neural AND network`. Note that, while this search is not as complete as the one for DL, it provides enough material for building an overview of the state of the field.

In terms of decomposition methods, the most widely used technique is the wavelet transform [17,19,22,24,34,41,49,51,52,99]. Alternative methods include empirical mode decomposition (EMD) [32,100], the Hilbert–Huang transform which uses EMD to decompose a signal and then applies Hilbert spectral analysis [101], variational mode decomposition [27,48], and singular spectrum analysis [102,103].

For feature selection, the most commonly utilized algorithms are correlation analysis [32,41,42,104,105] and the mutual information technique [18,42,52,106–108]. Other algorithms include classification and regression trees with recursive feature elimination [50] or ReliefF [50].

For clustering data, the algorithms are usually based on one of the following four: k-means [26,109], self-organizing maps [19,26,110], enhanced game theoretic clustering [26], or fuzzy clustering [52,111]

The selection of forecasting models is much more diverse. The most widely used method is the standard MLP [19,20,32,41,42,51,102,103,105,107,108], followed by the *adaptive network-based fuzzy inference system* (ANFIS) [19,100,106], radial basis function network [20,24,111], and autoregressive models like ARMA or ARIMA [20,22,24,100]. Other models include LSTM [17], linear regression [50], extreme learning machine [22,50], CNN [50], Bayesian neural network [26,110], exponential GARCH [100], echo state neural network [27], Elman neural networks [18], and support vector regressors [20]. It is important to note that in many of the approaches, the hybrid method does not consider a single forecasting model but combines several of them [19,20,24,50,100,108].

Just as for the forecasting model, the diversity of the heuristic optimization algorithms is also large. While the most often utilized algorithm is particle swarm optimization [22,48,51,106,107,111], many other approaches are also used: differential evolution [27], genetic algorithm [106], backtracking search [106], deterministic annealing [111], bat algorithm [41], vaporization precipitation-based water cycle algorithm [104], cuckoo search [103,105], or honey bee mating optimization [24].

In spite of the large number of published works, the research in hybrid methods suffers from the same problems as discussed earlier. First, most of the studies either avoid comparison with well-established methods [18–21,25,27,34,42,48–50,100,104,106,111] or resort to comparisons using outdated methodologies [22,24,26,41,51,52,102,103]. Hence, the accuracy of the new proposed methods cannot be accurately established.

Second, the considered studies usually employ very small datasets consisting either of a few days [17–22] or a few weeks [18,19,22,24–27,41,42,49,51,102–104,106,111]. Thus, drawing conclusions is nearly impossible and it is unclear whether the accuracy results are just the outcome of selecting a convenient test period.

Besides these two problems, for many hybrid methods the effect of selecting variants of the different hybrid components is not analyzed [20,21,24,25,27,41,42,50–52,102,103]. Thus, it is not clear how relevant or useful the individual components are.

2.4. State-of-the-art models

Because of the described problems when comparing EPF models, it is very hard to establish what are the state-of-the-art methods. Nevertheless, considering the studies performed in the last years, it can be argued that the LEAR is a very accurate (if not the most accurate) linear model. Moreover, it can also be argued that the accuracy of this model can be further improved by transforming the prices using variance stabilizing transformations, combining forecasts obtained for different calibration windows, and/or using long-term seasonal decomposition.

For the case of ML models, the selection is harder as the existing comparisons are of worse quality. Considering the most complete benchmark study in terms of forecasting models [57], it seems that a simple DNN with two layers is one of the best ML models. In particular, while more complex models, e.g. LSTMs, could potentially be more accurate, there is at the moment no sound evidence to validate this claim.

In the case of hybrid models, establishing what is the best model is an impossible task. Firstly, while many hybrid methods have been proposed, they have not been compared with each other nor with the LEAR or DNN models. Secondly, as most studies do not evaluate the individual influence of each hybrid component, it is also impossible to establish the best algorithms for each hybrid component, e.g. it is unclear what are the best clustering, feature selection method, or data decomposition methods.

With that in mind, we will consider the LEAR and the DNN for the proposed open-access benchmark. In particular, not only are these two methods highly accurate, but they are also relatively simple. As such, we think that they are the best benchmarks to compare new complex EPF forecasting methods with.

3. Open-access benchmark dataset

The first contribution of the paper is to provide a large open-access benchmark dataset on which new methods can be tested, together with the day-ahead forecasts of the proposed open-access methods. In this section, we introduce this dataset, which can be accessed[7] using the `python` library built for this study.

3.1. General characteristics

For a benchmark dataset in EPF to be fair it should satisfy three conditions:

1. comprise several electricity markets so that the capabilities of new models can be tested under different conditions,
2. be long enough so that algorithms can be analyzed using out-of-sample datasets that span 1–2 years, and
3. be recent enough to include the effects of integrating renewable energy sources on wholesale prices.

Based on these conditions, we propose five datasets representing five different day-ahead electricity markets, each of them comprising 6 years of data. The prices of each market have very distinct dynamics, i.e. they all have differences in terms of the frequency and existence of negative prices, zeros, and price spikes. In addition, as electricity prices depend on exogenous variables, each dataset comprises two additional time series: day-ahead forecasts of two influential exogenous factors that differ for each market. The length of each dataset equals 2184 days, which translates to six 364-day "years" or 312 weeks.[8] All available time series are reported using the local time, and the daylight savings are treated by either arithmetically averaging two values for the extra hour or interpolating the neighboring values for the missing observation.

3.2. Nord Pool

The first dataset represents the Nord Pool (NP), i.e. the European power market of the Nordic countries, and spans from 01.01.2013 to 24.12.2018. The dataset contains hourly observations of day-ahead prices, the day-ahead load forecast, and the day-ahead wind generation forecast. The dataset was constructed using data freely available on the

[7] Note that we do not own the data in the dataset. However, it can be freely accessed from different websites, e.g. the ENTSO-E transparency platform [112]. In this context, the proposed `python` library [58,59] provides an interface to easily access the data.

[8] Electricity prices exhibit weekly seasonality. Thus, by approximating a year by 52 weeks because we ensure that the metrics are not impacted by a certain day, e.g. Monday, being harder to predict than the others.

webpage of the Nordic power exchange Nord Pool [54]. Fig. 2(b) (top) displays the electricity price time series of the dataset; as can be seen, the prices are always positives, zero prices are rare, and prices spikes seldom occur.

3.3. PJM

The second dataset is obtained from the *Pennsylvania–New Jersey–Maryland* (PJM) market in the United States. It covers the same time period as Nord Pool, i.e. from 01.01.2013 to 24.12.2018. The three time series are: the zonal prices in the *Commonwealth Edison* (COMED) (a zone located in the state of Illinois) and two day-ahead load forecast series, one describing the system load and the second one the COMED zonal load. The data is freely available on the PJM's website [113]. Fig. 2(b) (bottom) depicts the electricity price time series of the dataset; as with the NP market, the prices are always positive and zero prices are rare; however, unlike the NP market, spikes appear frequently.

3.4. EPEX-BE

The third dataset represents the EPEX-BE market, the day-ahead electricity market in Belgium, which is operated by EPEX SPOT. The dataset spans from 09.01.2011 to 31.12.2016. The two exogenous data series represent the day-ahead load forecast and the day-ahead generation forecast in France. While this selection might be surprising, it has been shown [57] that these two are the best predictors of Belgian prices. The price data is freely available in the ENTSO-E transparency platform [112] and the ELIA website [114], and the load and generation day-ahead forecasts are freely available in [115]. It is important to note that this dataset is particularly interesting because it is harder to predict. Fig. 3 (top) shows the electricity price time series of the dataset; unlike the prices in the PJM and NP markets, negative prices and zero prices appear more frequently, and price spikes are very common.

3.5. EPEX-FR

The fourth dataset represents the EPEX-FR market, the day-ahead electricity market in France, which is also operated by EPEX SPOT. The dataset spans the same period as the EPEX-BE dataset, i.e. from 09.01.2011 to 31.12.2016. Besides the electricity prices, the dataset comprises the day-ahead load forecast and the day-ahead generation forecast. As before, the price data is freely obtained from the ENTSO-E transparency platform [112], and the load and generation day-ahead forecasts are freely available on the webpage of RTE [115], i.e. the *transmission system operator* (TSO) in France. Fig. 3 (middle) displays the electricity price time series of the dataset; as in the EPEX-BE market, negative prices, zero prices, and spikes are very common.

3.6. EPEX-DE

The last dataset describes the EPEX-DE market, the German electricity market, which is also operated by EPEX SPOT. The dataset spans from 09.01.2012 to 31.12.2017. Besides the prices, the dataset comprises the day-ahead zonal load forecast in the TSO Amprion zone and the aggregated day-ahead wind and solar generation forecasts in the zones of the 3 largest[9] TSOs (Amprion, TenneT, and 50Hertz). The price data is freely obtained from the ENTSO-E transparency platform [112], the zonal load day-ahead forecasts is freely available in the website of Amprion [116], and the wind and solar forecasts in the websites of Amprion [116], 50Hertz [117], and TenneT [118]. Fig. 3 (bottom) displays the electricity price time series of the dataset; as can be seen, while negative and zero prices occur more often than in the other four markets, price spikes are more rare.

[9] There are 4 TSOs in Germany.

Table 1
Start and end dates of the testing (out-of-sample) datasets for each electricity market.

Market	Test period
Nord pool	27.12.2016–24.12.2018
PJM	27.12.2016–24.12.2018
EPEX-FR	04.01.2015–31.12.2016
EPEX-BE	04.01.2015–31.12.2016
EPEX-DE	04.01.2016–31.12.2017

3.7. Training and testing periods

For each dataset, the testing period is defined as the last 104 weeks, i.e. the last two years, of the dataset. The exact dates of the testing datasets are defined in Table 1. It is important to note that, as we will argue in Section 5, selecting two years as the testing period is paramount to ensure good research practices in EPF.

Unlike the testing dataset, the training dataset cannot be defined as it will vary between different models. In general, the training dataset will comprise any data that is known prior to the target day. However, the exact data will change depending on two concepts, i.e. calibration window and recalibration:

- While there are four years of data available for estimating the model, it might be desirable to employ only recent data, e.g. to avoid estimating effects that no longer play a role. The amount of past data employed for estimation defines the calibration window.
- The model can be estimated once and then evaluated for the full test dataset, or it can be continuously recalibrated on a daily basis to incorporate the input of recent data.

For example, let us consider predicting the NP prices on 15.02.2017. A model using a calibration window of 52 weeks and no recalibration would employ a training dataset comprising the data between 29.12.2016 and 26.12.2016, i.e. one year prior to the start of the test period. By contrast, a model using a calibration window of 104 weeks and daily recalibration would employ the data between 18.02.2015 and 14.02.2017.

4. Open-access benchmark models

The second contribution of the paper is to provide a set of state-of-the-art forecasting methods as an open-source `python` toolbox. As explained in Section 2.4, the LEAR [55] and the DNN [57] models are not only highly accurate but also relatively simple. Therefore, we implement these two methods and provide their code freely available as part of the proposed toolbox [58,59]. It is important to note that the use of the proposed open-access methods is fully documented and automated so researchers can test and use them without expert knowledge.

For the sake of simplicity, the description provided here is limited to the bare minimum. For further details on the two models we refer to the original papers [55,57].

4.1. Input features

Before describing each model, let us define the input features that are considered. Independently of the model, the available input features to forecast the 24 day-ahead prices of day d, i.e. $\mathbf{p}_d = [p_{d,1}, \dots, p_{d,24}]^\top$, are the same:

- Historical day-ahead prices of the previous three days and one week ago, i.e. $\mathbf{p}_{d-1}$, $\mathbf{p}_{d-2}$, $\mathbf{p}_{d-3}$, $\mathbf{p}_{d-7}$.
- The day-ahead forecasts of the two variables of interest (see Section 3 for details) for day d available on day $d-1$, i.e. $\mathbf{x}_d^1 = [x_{d,1}^1, \dots, x_{d,24}^1]^\top$ and $\mathbf{x}_d^2 = [x_{d,1}^2, \dots, x_{d,24}^2]^\top$; note that the variables of interest are different for each market.

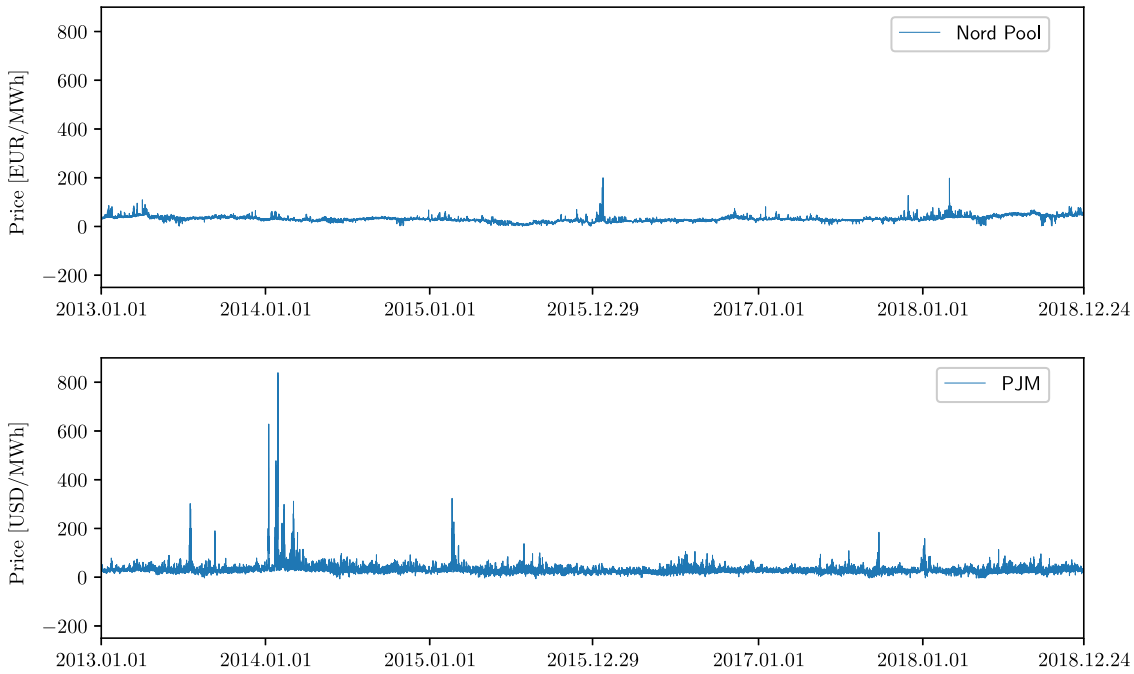


Fig. 2. Electricity price time series for two of the five datasets, i.e. Nord Pool and PJM, considered in the open-access benchmark dataset. Note that each dataset also includes two exogenous time series that are not plotted here.

- Historical day-ahead forecasts of the variables of interest the previous day and one week ago, i.e. $\mathbf{x}^1_{d-1}$, $\mathbf{x}^1_{d-7}$, $\mathbf{x}^2_{d-1}$, $\mathbf{x}^2_{d-7}$.
- A dummy variable $\mathbf{z}_d$ that represents the day of the week. In the case of the linear model, following the standard practice in the literature [55,69,81], this is a binary vector $\mathbf{z} = [z_{d,1}, \dots, z_{d,7}]^\top$ that encodes every day of the week by setting all elements to zero except the element that identifies the day of the week, e.g. $[1,0,0,0,0,0,0]$ represents Monday and $[0,1,0,0,0,0,0]$ Tuesday. In the case of the neural network, for the sake of simplicity, the day of the week is modeled with a multi-value input $z_d \in \{1, \dots, 7\}$.

Overall, we consider a total of 247 available input features for each LEAR model and 241 input features for each DNN model. It is important to note that, while the available input features are the same, the LEAR and DNN models utilize a different feature selection procedure. Namely, each of the LEAR models finds the optimal set of features using LASSO as an embedded feature selection, i.e. each model uses L1-regularization to select among the 247 features. On the other hand, in the DNN model, as in the original study [57], the input features are optimized together with the hyperparameters using the tree Parzen estimator [119] (see Section 4.3 for details). Finally, it should be emphasized that for both types of models the feature selection is fully automated and does not require expert intervention.

4.2. The LEAR model

The *Lasso Estimated AutoRegressive* (LEAR) model is a parameter-rich ARX structure estimated using L1-regularization, i.e. the LASSO [56]. It was originally introduced in [55] under the name *LassoX*. The LEAR is based on the so-called *full ARX* or *fARX* model, a parameter-rich autoregressive specification with exogenous variables, which in turn is inspired by the general autoregressive model defined by Equation (2) in [68], with some important differences. While fARX includes fundamentals and a much richer seasonal structure, it does not look too far into the past and concentrates only on the last week of data. Note, that very similar models to the LEAR were used in [64] under the name $24lasso_{DoW,nl}$ and in [69] under the name $24Lasso_1$.

To enhance the model, as empirically tested and recommended in [9,64,69], the data is preprocessed with the *area* (or *inverse*) *hyperbolic sine* variance stabilizing transformation:

$$\operatorname{asinh}(x) = \log\left(x + \sqrt{x^2+1}\right), \tag{2}$$

where x is the price standardized by subtracting the in-sample median and dividing by the median absolute deviation adjusted by a factor for asymptotically normal consistency to the standard deviation, for details see [9]. Long-term seasonal decomposition is not considered for the sake of simplicity; particularly, while it has been shown to further improve the performance of the LEAR, we leave it out for future research.

As in [81], to further enhance the model, we recalibrate it daily over different calibration window lengths: 8 weeks, 12 weeks, 3 years, and 4 years. We consider short windows (8–12 weeks) in combination with long windows (3–4 years) because it has been empirically shown to lead to better results [81]. In this context, we consider a minimum of 8 weeks as lower windows might not have enough information to correctly estimate parameter-rich models [81].

The LEAR model to predict price $p_{d,h}$ on day d and hour h is defined by:

$$\begin{aligned} p_{d,h} =& f(\mathbf{p}_{d-1}, \mathbf{p}_{d-2}, \mathbf{p}_{d-3}, \mathbf{p}_{d-7}, \mathbf{x}^i_d, \mathbf{x}^i_{d-1}, \mathbf{x}^i_{d-7}, \boldsymbol{\theta}_h) + \varepsilon_{d,h} \\ =& \sum_{i=1}^{24} \theta_{h,i} \cdot p_{d-1,i} + \sum_{i=1}^{24} \theta_{h,24+i} \cdot p_{d-2,i} \\ &+ \sum_{i=1}^{24} \theta_{h,48+i} \cdot p_{d-3,i} + \sum_{i=1}^{24} \theta_{h,72+i} \cdot p_{d-7,i} \\ &+ \sum_{i=1}^{24} \theta_{h,96+i} \cdot x^1_{d,i} + \sum_{i=1}^{24} \theta_{h,120+i} \cdot x^2_{d,i} \\ &+ \sum_{i=1}^{24} \theta_{h,144+i} \cdot x^1_{d-1,i} + \sum_{i=1}^{24} \theta_{h,168+i} \cdot x^2_{d-1,i} \\ &+ \sum_{i=1}^{24} \theta_{h,192+i} \cdot x^1_{d-7,i} + \sum_{i=1}^{24} \theta_{h,216+i} \cdot x^2_{d-7,i} \\ &+ \sum_{i=1}^{7} \theta_{h,240+i} \cdot z_{d,i} + \varepsilon_{d,h} \end{aligned} \tag{3}$$

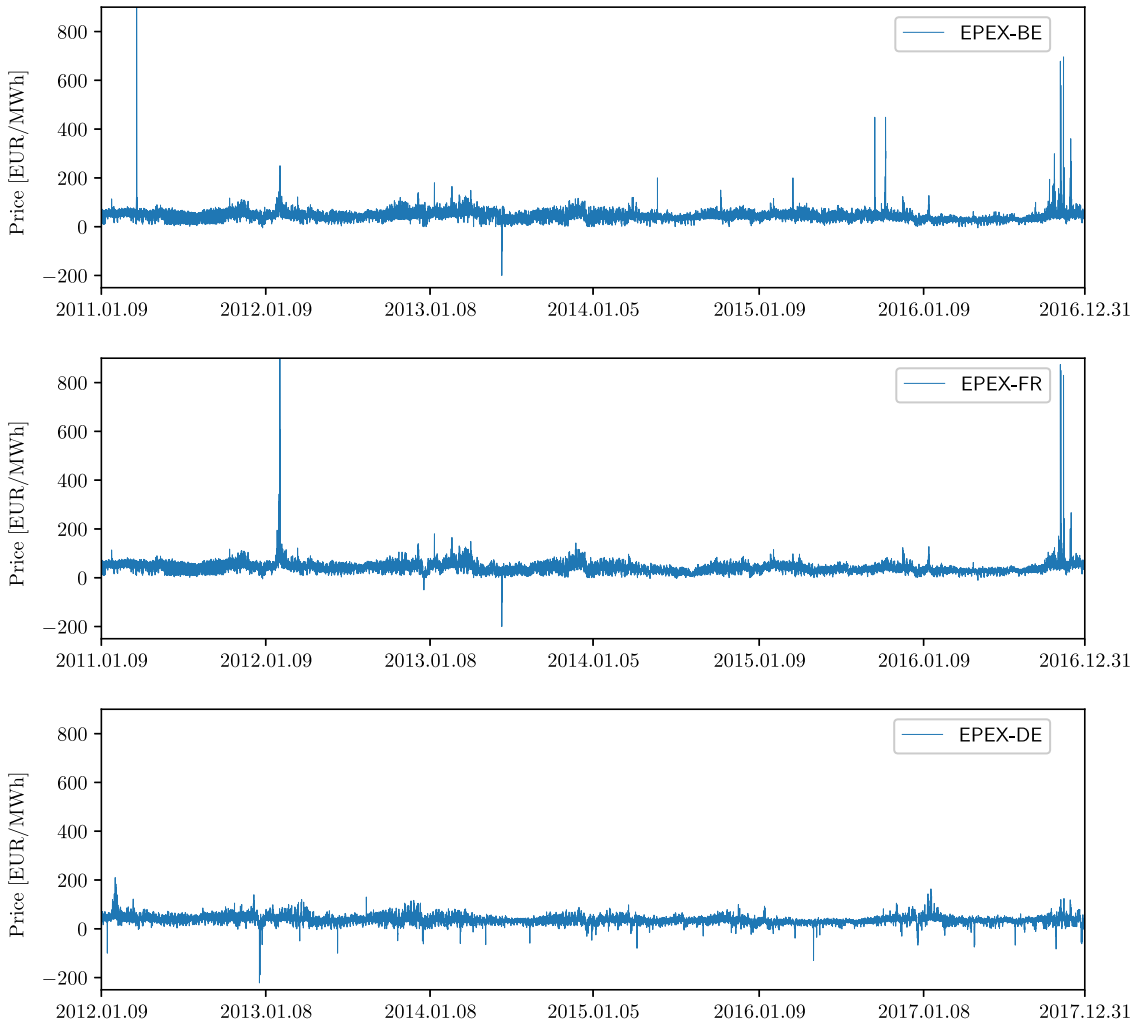


Fig. 3. Electricity price time series for three of the five datasets, i.e. EPEX-BE, EPEX-FR, and EPEX-DE, considered in the open-access benchmark dataset. Note that each dataset also includes two exogenous time series that are not plotted here. The EPEX-BE and EPEX-FR time series are similar because the EPEX-FR and EPEX-BE are highly coupled markets [57]. To keep the plots readable, the upper limit of the y-axis is below the maximum price; this only affects one spike in EPEX-FR and another one in EPEX-BE.

where $\boldsymbol{\theta}_h = [\theta_{h,1}, \dots, \theta_{h,247}]^\top$ are the 247 parameters of the LEAR model for hour h. Many of these parameters become zero when (3) is estimated using LASSO:

$$\hat{\boldsymbol{\theta}}_h = \underset{\boldsymbol{\theta}_h}{\operatorname{argmin}} \ \mathrm{RSS} + \lambda \left\|\boldsymbol{\theta}_h\right\|_1 = \underset{\boldsymbol{\theta}_h}{\operatorname{argmin}} \ \mathrm{RSS} + \lambda \sum_{i=1}^{247} \left|\theta_{h,i}\right|, \tag{4}$$

where $\mathrm{RSS} = \sum_{d=8}^{N_d} (p_{d,h} - \hat{p}_{d,h})^2$ is the sum of squared residuals, $\hat{p}_{d,h}$ the price forecast, N_d is the number of days in the training dataset, and $\lambda \geq 0$ is the *tuning* (or *regularization*) hyperparameter of LASSO. Due to the computational speed of estimating with LASSO, during every daily recalibration, the hyperparameter λ that regulates the L_1 penalty is optimized. This can be done using an *ex-ante* cross-validation procedure [120]. In this study, to further reduce the computational cost, we propose an efficient hybrid approach to perform the optimal selection of λ. See Section 4.2.2 for details.

4.2.1. Regularization hyperparameter

The hyperparameter λ of LASSO can be optimized in multiple ways, each with different advantages and disadvantages. A first approach is to optimize λ once and then keep it fixed for the whole test period. Although it requires very low computation costs, the limitation of this approach is that it assumes that the hyperparameter λ does not change over time. This assumption might hinder the performance of the estimator as the regularization parameter does not change even when the market might do.

A second approach is to recalibrate the hyperparameter on a periodic basis using a validation dataset. Although this method yields good results, tuning the recalibration frequency and calibration window is complicated, the computational cost is large, and the results may vary between datasets [69].

A third option is to recalibrate the hyperparameter periodically, but using *cross-validation* (CV): splitting the data into disjoint partitions, using each possible partition once as a test dataset with the remaining data as the training dataset, and selecting the hyperparameter that performs the best across all partitions [120]. Although this approach is highly accurate, its computation costs are very large.

A fourth option is to periodically update the hyperparameter but using information criteria, e.g. the *Akaike information criterion* (AIC) or the Bayesian information criterion [64,68,121]. As before, this involves training multiple LASSO models to compute the information criteria for each possible hyperparameter value, which in turn leads to a high computational cost.

Lastly, one can use the *least angle regression* (LARS) LASSO [122] for estimating the model instead of the coordinate descent implementation. This estimation procedure has the advantage of computing the whole LASSO solution path, which in turn allows to compute the information criteria or perform CV much faster.

4.2.2. Selecting the regularization hyperparameter

To select λ we propose a hybrid approach. On a daily basis, we estimate the hyperparameter using the LARS method with the in-sample

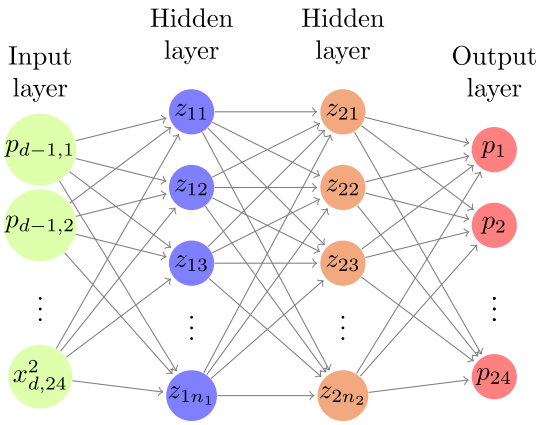


Fig. 4. Visualization of a sample DNN model.

AIC. Then, using the optimal λ obtained from the LARS method, we recalibrate the LEAR using the traditional coordinate descent implementation.

The reason for proposing this hybrid approach is that it provides a good trade-off between computational complexity and accuracy. In particular, it leverages the computational efficiency of LARS for ex-ante λ selection with the predictive performance on short calibration windows of the coordinate descent LASSO.

It is important to note that we have studied multiple approaches to select λ: (i) daily recalibration, CV, with coordinate descent; (ii) daily recalibration, CV, with LARS; (iii) daily recalibration with LARS and AIC. However, the computational cost of the first method was too high (in the same order of magnitude as the cost of the DNN model), and the accuracy of the other two was not good. By contrast, the proposed approach had a performance on par with coordinate descent LASSO using CV, but with a computational cost that was an order of magnitude lower.

4.3. The DNN model

The second model is the DNN [57], one of the simplest DL models whose input features and hyperparameters can be optimized and tailored for each case study without the need for expert knowledge. The DNN is a straightforward extension of the traditional multilayer perceptron (MLP) with two hidden layers.

4.3.1. Structure

The DNN is a deep feedforward neural network that contains 4 layers, employs the multivariate framework (single model with 24 outputs), is estimated using Adam [123], and its hyperparameters and input features are optimized using the tree Parzen estimator [119], i.e. a Bayesian optimization algorithm. Its structure is visualized in Fig. 4.

4.3.2. Training dataset

For estimating the hyperparameters, the training dataset is fixed and comprises the four years prior to the testing period. For evaluating the testing dataset, the DNN is recalibrated on a daily basis using a calibration window of four years.

In all cases, the training dataset is split into a training and a validation dataset, with the latter being used for two purposes: performing early stopping [124] to avoid overfitting and optimizing hyperparameters/features. While the validation dataset always comprises 42 weeks, the split between the training and validation datasets depends on whether the validation dataset is used for hyperparameter/feature selection or for the recalibration step:

- For estimating the hyperparameters, as the validation dataset is used to guide the optimization process, the validation dataset is selected as the last 42 weeks of the training dataset. This is done to keep the training and validation datasets completely independent and to avoid overfitting.[10]
- For the testing phase, as the validation dataset is only used for early stopping, it is defined by randomly selecting 42 weeks out of the total 208 weeks employed for training. This is done to ensure that the dataset used for optimizing the DNN parameters includes up-to-date data.[11]

As example, let us consider the training and evaluation of a DNN in the Nord Pool market. Before evaluating the DNN, the hyperparameter and features of the DNN are optimized. For that, the employed dataset comprises the data between 01.01.2013 and 26.12.2016, of which the training dataset represents the first 166 weeks, i.e. 01.01.2013 to 07.03.2016, and the validation dataset the last 42, i.e. 08.03.2016 to 26.12.2016. During the evaluation of the model, i.e. after the hyperparameter and feature selection, the training and validation datasets comprise the last four years of data but are randomly shuffled. For example, to evaluate the DNN during 15.02.2017, the training and validation datasets would represent the data between 20.02.2013 and 14.02.2017, of which 166 randomly selected weeks would define the training dataset and the remaining 42 the validation dataset.

4.3.3. Hyperparameter and feature selection

As in the original DNN paper [57], the hyperparameters and input features are optimized together using the tree-structured Parzen estimator [119], a Bayesian optimization algorithm based on sequential model-based optimization. To do so, the features are modeled as hyperparameters, with each hyperparameter representing a binary variable that selects whether or not a specific feature is included in the model (as explained in [57]). In more detail, to select which of the 241 available input features are relevant, the method employs 11 decision variables, i.e. 11 hyperparameters:

- Four binary hyperparameters (1–4) that indicate whether or not to include the historical day ahead prices $\mathbf{p}_{d-1}$, $\mathbf{p}_{d-2}$, $\mathbf{p}_{d-3}$, $\mathbf{p}_{d-7}$. The selection is done per day,[12] e.g. the algorithm either selects all 24 hourly prices $\mathbf{p}_{d-j}$ of j days ago or does not select any price from day $d-j$, hence the four hyperparameters.
- Two binary hyperparameters (5–6) that indicate whether or not to include each of the day-ahead forecasts $\mathbf{x}^1_d$ and $\mathbf{x}^2_d$. As with the past prices, this is done for the whole day, i.e. a hyperparameter either selects all the elements in $\mathbf{x}^j_d$ or none.
- Four binary hyperparameters (7–10) that indicate whether or not to include the historical day-ahead forecasts $\mathbf{x}^1_{d-1}$, $\mathbf{x}^2_{d-1}$, $\mathbf{x}^1_{d-7}$, and $\mathbf{x}^2_{d-7}$. This selection is also done per day.
- One binary hyperparameter (11) that indicates whether or not to include the variable z_d representing the day of the week.

In short, 10 binary hyperparameters indicating whether or not to include 24 inputs each and another binary hyperparameter indicating whether or not to include a dummy variable.

[10] Similar as it is done when splitting the dataset between the training and the test dataset.

[11] For hyperparameter optimization, as the validation dataset represents the most recent weeks of data, the neural network is trained with data that is almost one year old. While this is not a big problem when deciding on the DNN structure, it should be avoided during testing to ensure that the DNN captures new market effects.

[12] This is done for the sake of simplicity to speed up the optimization procedure of the feature selection. In particular, an alternative could be to use a binary hyperparameter for each individual historical prices; however, in most markets, that would mean using 24 as many hyperparameters as there are 24 different prices per day.

Besides selecting the features, the algorithm also optimizes eight additional hyperparameters: (1) the number of neurons per layer, (2) the activation function, (3) the dropout rate, (4) the learning rate, (5) whether or not to use batch normalization, (6) the type of data preprocessing technique, (7) the initialization of the DNN weights, and (8) the coefficient for L1 regularization that is applied to each layer's kernel.

Unlike the weights of the DNN that are recalibrated on a daily basis, the hyperparameter and features are optimized only once using the four years of data prior to the testing period. It is important to note that the algorithm runs for a number T of iterations, where at every iteration the algorithm infers a potential optimal subset of hyperparameters/features and evaluates this subset in the validation dataset. For the proposed open-access benchmark models, T is selected as 1500 iterations to obtain a trade-off between accuracy and computational requirements.[13]

4.4. Ensembles

For the open-access benchmark, in order to have benchmark predictions when evaluating ensemble techniques, we also propose ensembles of LEAR and DNNs as open-access benchmarks of ensembles methods. For the LEAR, the ensemble is built as the arithmetic average of forecasts across four calibration window lengths: 8 weeks, 12 weeks, 3 years, and 4 years. For the DNN, the ensemble is built as the arithmetic average of four different DNNs that are estimated by running the hyperparameter/feature selection procedure four times. In particular, the hyperparameter optimization is asymptotically deterministic, i.e. the global optimum is found for an infinite number of iterations. However, for a finite number of iterations and using a different initial random seed, the algorithm is non-deterministic and every run provides a different set of hyperparameters and features. Although each of these hyperparameter/feature subsets represent a local minimum, it is impossible to establish which of the subsets is better as their relative performance on the validation dataset is nearly identical. This effect can be explained by the fact that the DNN is a very flexible model and thus different network architectures are able to obtain equally good results.

4.5. Software implementation

The proposed open-access models are developed in `python`: the LEAR is implemented using the `scikit-learn` library [125] and the DNN model using the `Keras` library [126]. The reason for selecting `python` is that it is one of the most widely used programming languages, especially in the context of ML and statistical inference.

5. Guidelines and best practices in EPF

As motivated in the introduction, the field of EPF suffers from several problems that prevent having reproducible research and establishing strong conclusions on what methods work best. In this section, we outline some of these issues and provide some guidelines on how to address them.

[13] It can be empirically observed that the performance of the models barely improves after 1000 iterations. Moreover, performing 1500 iterations takes approximately just one day on a regular quadcore laptop like the i7-6920HQ, a computation cost very acceptable when the algorithm has to run only once.

5.1. Length of the test period

A common practice in EPF is to evaluate new methods on very short test periods. The typical approach is to evaluate the method on 4 weeks of data [18,19,22,24–26,29,30,41,42,49,51,97,102–107,110], with each week representing one of the four seasons in the year. This is problematic for three reasons:

- Selecting four weeks can lead to cherry-picking the weeks where a given method excels, e.g. a method that performs bad with spikes could be evaluated in a week with fewer spikes, leading in turn to biased estimations of the forecasting accuracy. While this is an ethical issue that most researchers would avoid, establishing four week testing periods as the standard does facilitate malpractice and should be avoided.
- Assuming that the four weeks are randomly selected and no bias is introduced in the selection, it is still not possible to guarantee that these four weeks are representative of the price behavior over a whole year. Particularly, even within a given season, the price dynamics can change dramatically, e.g. during winter there are weeks with a lot of sun and wind but there are also weeks without them. Therefore, picking only a week per season rarely represents the average performance of a forecaster in a given dataset.
- There are situations in the electrical grid that do not occur very often but that can have a very large effect on electricity prices, e.g. when several power plants are under maintenance at the same time. Forecasting methods need to be evaluated under those conditions to ensure that they are also accurate under extreme events. By selecting four weeks most of these effects are neglected.

To avoid this problem, we recommend using a minimum of one year as a testing period. This ensures that forecasting methods are evaluated considering the complete set of effects that take place during the year. To guarantee that all researchers have access to this type of data, the open-access benchmark dataset that we propose contains data from several markets and employs a testing period of two years. In addition, the open-access benchmark can be directly accessed using the proposed `epftoolbox` library [58,59].

5.2. Benchmark models

A second issue with many EPF publications is that new methods are not compared with well-established methods [14,16,18–21,23,25,27,34,36,42,46,48–50,88,100,104,106,111] or resort to comparisons using either outdated methodologies or simplified methods [13,15,22,24,26,28–30,37,41,44,45,47,51,52,95,96,102,103].

This poses a problem since it becomes very hard to establish which algorithms work best and which ones do not. To address this issue, we recommend using well-established state-of-the-art open-source methods and a common benchmark dataset. With that in mind, we have provided and made freely available an open-access benchmark dataset comprising 5 markets (as described in Section 3), and we have implemented, thoroughly tested, and made freely available two state-of-the-art forecasting methods (as described in Section 4) and their day-ahead predictions for all 5 datasets over a period of two years (as described in Section 6). Additionally, we have implemented all these resources in an easy-to-use toolbox [58] and adequately documentated it [59].

5.3. Open-access

A third issue in the field of EPF is that datasets are usually not made publicly available and the code of the proposed methods is not shared. This poses four obvious problems:

- Research cannot be reproduced as data is not available. This goes against one of the main principles of science as all research should be reproducible.
- The progress of EPF research is hindered since it is hard to establish which methodologies work well. Consequently, researchers spend unnecessary time re-evaluating methodologies that have been evaluated already.
- Comparing new methods with published ones becomes very challenging because researchers have to re-implement methods from the literature. As a result, comparisons with state-of-the-art methods are often avoided, and new methods are usually compared with simple and easy-to-implement methods.
- When new methods are proposed, they cannot be compared with published methods under the same circumstances. This leads to comparisons under different conditions and opens up the door to wrong implementations of the original methods, which in turn leads to results that are not correct.

As these problems are critical, we directly try to address them by providing an open-access benchmark/toolbox comprising five datasets, two state-of-the-art methods, and a set of day-ahead forecasts of the latter two methods. In addition, we encourage researchers in EPF to share the developed codes and to either share their datasets or use an open-access benchmark dataset.

5.4. Evaluation metrics for point forecasts

In the field of EPF, the most widely used metrics to measure the accuracy of point forecasts are the *mean absolute error* (MAE), the *root mean square error* (RMSE), and the *mean absolute percentage error* (MAPE):

$$\text{MAE} = \frac{1}{24\,N_\text{d}} \sum_{d=1}^{N_\text{d}} \sum_{h=1}^{24} |p_{d,h} - \hat{p}_{d,h}|, \tag{5}$$

$$\text{RMSE} = \sqrt{\frac{1}{24\,N_\text{d}} \sum_{d=1}^{N_\text{d}} \sum_{h=1}^{24} (p_{d,h} - \hat{p}_{d,h})^2}, \tag{6}$$

$$\text{MAPE} = \frac{1}{24\,N_\text{d}} \sum_{d=1}^{N_\text{d}} \sum_{h=1}^{24} \frac{|p_{d,h} - \hat{p}_{d,h}|}{|p_{d,h}|}, \tag{7}$$

where $p_{d,h}$ and $\hat{p}_{d,h}$ respectively represent the real and forecasted price on day d and hour h, and N_d is the number of days in the out-of-sample test period, i.e. in the test dataset.

Since absolute errors are hard to compare between different datasets, the MAE and RMSE are not always very informative. Moreover, since electricity costs and profits are often linearly dependent on the electricity prices, metrics based on quadratic errors, e.g. RMSE, are hard to interpret and do not accurately represent the underlying problem of most forecasting users. In particular, in most electricity trade applications, the underlying risk, profits, and costs depend linearly on the price and on the forecasting errors. Hence, linear metrics represent better than quadratic metrics the underlying risks of forecasting errors.

Similarly, since MAPE values become very large with prices close to zero (regardless of the actual absolute errors), the MAPE is usually dominated by the periods of low prices and is also not very informative. While the *symmetric mean absolute percentage error* (sMAPE) defined[14] as:

$$\text{sMAPE} = \frac{1}{24\,N_\text{d}} \sum_{d=1}^{N_\text{d}} \sum_{h=1}^{24} 2\frac{|p_{d,h} - \hat{p}_{d,h}|}{|p_{d,h}| + |\hat{p}_{d,h}|} \tag{8}$$

solves some of these issues, it has (as any metric based on percentage errors) a statistical distribution with undefined mean and infinite variance [128].

[14] Note, that there are multiple versions of sMAPE, here we consider the most sensible one according to [127].

5.4.1. Scaled errors

In this context, several studies advocate for the use of scaled errors [5,128,129], where a scaled error is simply the MAE scaled by the in-sample MAE of a naive forecast. A scaled error has the nice interpretation of being lower/larger than one if it is better/worse than the average naive forecast evaluated in-sample.

A metric based on this concept is the *mean absolute scaled error* (MASE), and in the context of one-step ahead forecasting is defined as [128]:

$$\text{MASE} = \frac{1}{N} \sum_{k=1}^{N} \frac{|p_k - \hat{p}_k|}{\frac{1}{n-1}\sum_{i=2}^{n} |p_i^\text{in} - p_{i-1}^\text{in}|}, \tag{9}$$

where p_i^in is the i^th price in the in-sample, i.e. training, dataset (note that in EPF $i = 24d + h$), p_{i-1}^in is the one-step ahead naive forecast of p_i^in, i.e. $\hat{p}_i^\text{in}$, N is the number of out-of-sample (test) datapoints, and n the number of in-sample (training) datapoints. For seasonal time series, the MASE may be defined using the MAE of a seasonal naive model in the denominator [5,129].

5.4.2. Relative measures

While scaled errors do indeed solve the issues of more traditional metrics, they have other associated problems that make them unsuitable in the context of EPF:

1. As MASE depends on the in-sample dataset, forecasting methods with different calibration windows will naturally have to consider different in-sample datasets. As a result, the MASE of each model will be based on a different scaling factor and comparisons between models cannot be drawn.
2. The same argument applies to models with and without rolling windows. The latter will use a different in-sample dataset at every time point while the former will keep the in-sample dataset constant.
3. In ensembles of models with different calibration windows, the MASE cannot be defined as the calibration window of the ensemble is undefined.
4. Drawing comparisons across different time series is problematic as electricity prices are not stationary. For example, an in-sample dataset with spikes and an out-of-sample dataset without spikes will lead to a smaller MASE than if we consider the same market but with the in-sample/out-sample datasets reversed.

To solve these issues, we argue that a better metric is the *relative* MAE (rMAE) [128,130]. Similar to MASE, it normalizes the error by the MAE of a naive forecast. However, instead of considering the in-sample dataset, the naive forecast is built based on the out-of-sample dataset. For day-ahead electricity prices of hourly frequency, rMAE is defined as:

$$\text{rMAE} = \frac{\frac{1}{24N_\text{d}} \sum_{d=1}^{N_\text{d}} \sum_{h=1}^{24} |p_{d,h} - \hat{p}_{d,h}|}{\frac{1}{24N_\text{d}} \sum_{d=1}^{N_\text{d}} \sum_{h=1}^{24} |p_{d,h} - \hat{p}_{d,h}^\text{naive}|}, \tag{10}$$

where the $\frac{1}{24N_d}$ factor cancels out in the numerator and the denominator. There are three natural choices for the naive forecasts:

- $\hat{p}_{d,h}^{\text{naive},1} = p_{d-1,h}$,
- $\hat{p}_{d,h}^{\text{naive},2} = p_{d-7,h}$,
- $\hat{p}_{d,h}^{\text{naive},3} = \begin{cases} p_{d-1,h}, & \text{if } d \text{ is Tue, Wed, Thu, or Fri,} \\ p_{d-7,h}, & \text{if } d \text{ is Sat, Sun, or Mon.} \end{cases}$

In the context of EPF, rMAE using $\hat{p}_{d,h}^{\text{naive},2} = p_{d-7,h}$ is arguably the best choice for two reasons: (i) it is easier to compute than the one based on $\hat{p}_{d,h}^{\text{naive},3}$ and, unlike the rMAE based on $\hat{p}_{d,h}^{\text{naive},1}$, it captures weekly effects; (ii) given a set of forecasting models, the relative

ranking of the accuracy of the models is independent from the naive benchmark used (see the last paragraph of this subsection for an explanation). Hence, in the remainder of the article we will use rMAE to explicitly refer to the rMAE based on $\hat{p}^{\text{naive},2}_{d,h}$. It is important to note that, similar to rMAE, one could also define the *relative* RMSE (rRMSE) by dividing the RMSE of each forecast by the RMSE of a naive forecast.

Since the dependence on the in-sample dataset is removed, using a rolling window is no longer a problem as the out-of-sample dataset stays the same. Similarly, models with different calibration windows can be compared and the rMAE of ensembles is properly defined. Moreover, as the metric is normalized by the MAE of a naive forecast for the same sample, the problem with drawing conclusions in non-stationary time series is mitigated.

Due to its better properties, rMAE should always be used to evaluate new methods in EPF. In particular, while it can be used in conjunction with other metrics, it is important to include and employ rMAE to obtain more fair evaluations and comparisons.

With that in mind, the accuracy of the open-access models in the open-access benchmark dataset is computed considering rMAE, sMAPE, MAPE, MAE, and RMSE. Then, an analysis of the different metrics is provided (see Section 6.4.2). Finally, the forecasts themselves are provided as csv files so that the accuracy results can be updated in case more adequate metrics are developed in the future.

As a final remark, let us note that, given a set of forecasting models, the relative ranking of the accuracy of the models is independent from the naive benchmark used for the rMAE or MASE. Changing it simply changes the denominator but preserves the numerator, and since the change in the denominator is the same across all methods, the relative ranking is preserved. Furthermore, as the numerator is the MAE, it follows that the ranking based on the rMAE or MASE will be the same as that based on the MAE.

5.5. Statistical testing

While using adequate metrics to compare the accuracy of the forecasts is important, it is also necessary to analyze whether any difference in accuracy is statistically significant. This is paramount to conclude whether the difference in accuracy does really exist and is not simply due to random differences between the forecasts. Despite its importance, the use of statistical testing has been downplayed in the EPF literature [5]. In particular, most publications only compare the accuracy in terms of an error metric and do not analyze the statistical significance of the accuracy differences. This trend needs to be corrected in order to compare forecasting approaches with statistical rigor. Particularly, new studies need to ensure that:

- Any new method is compared against well-established methods using a statistical test.
- The forecasts of the proposed methods are provided as open-access datasets. This ensures that, when new models are proposed, the difference in accuracy with the published methods can be analyzed in terms of statistical testing.

To facilitate statistical testing, we include in the proposed open-source `epftoolbox` library [58,59] the two most widely used statistical tests in EPF, i.e. the Diebold–Mariano and the Giacomini–White tests.

5.5.1. The Diebold–Mariano test

The Diebold–Mariano (DM) test [131] is probably the most commonly used tool to evaluate the significance of differences in forecasting accuracy. It is an asymptotic z-test of the hypothesis that the mean of the *loss differential* series:

$$\Delta^{\text{A,B}}_{d,h} = L(\varepsilon^{\text{A}}_{d,h}) - L(\varepsilon^{\text{B}}_{d,h}) \tag{11}$$

is zero, where $\varepsilon^{\text{Z}}_{d,h} = p_{d,h} - \hat{p}_{d,h}$ is the prediction error of model Z for day d and hour h, and $L(\cdot)$ is the loss function. For point forecasts, we usually take $L(\varepsilon^{\text{Z}}_{d,h}) = |\varepsilon^{\text{Z}}_{d,h}|^p$ with $p = 1$ or 2, which corresponds to the absolute and squared losses, respectively; for probabilistic forecasts, $L(\cdot)$ may be any strictly proper scoring rule, in particular the pinball loss, the *continuous ranked probability score* (CRPS), or the energy score [6,63,65]. Given the loss differential series, we compute the statistic:

$$\text{DM} = \sqrt{N}\frac{\hat{\mu}}{\hat{\sigma}}, \tag{12}$$

where $\hat{\mu}$ and $\hat{\sigma}$ are the sample mean and standard deviation of $\Delta^{\text{A,B}}_{d,h}$, respectively, and N is the length of the out-of-sample test period. Under the assumption of covariance stationarity of $\Delta^{\text{A,B}}_{d,h}$, the DM statistic is asymptotically standard normal, and one- or two-sided asymptotic tail probabilities can be easily computed.

It is important to note three things. Firstly, the DM test is model-free, i.e. it compares forecasts (of models), not models themselves. Secondly, although in the standard formulation [131] the DM test compares forecasts via the null hypothesis of the expected loss differential being zero, it is more informative to compute the p-values of two one-sided tests:

1. with the null hypothesis $H_0 : E(\Delta^{\text{A,B}}_{d,h}) \leq 0$,
2. with the alternative hypothesis null $H_1 : E(\Delta^{\text{A,B}}_{d,h}) \geq 0$.

The lower the p-value,[15] i.e. the closer it is to zero, the more the observed data is inconsistent with the null hypothesis. If the p-value is less than the commonly accepted level of 5%, the null hypothesis is typically rejected. In the DM test, this means that the forecasts of model B are significantly more accurate than those of model A.

Thirdly, the DM test requires (only) that the loss differential be covariance stationary.[16] This may not be satisfied by forecasts in day-ahead markets, since the predictions for all 24 h of the next day are computed at the same time, using the same information set. Hence, following [63], we recommend two variants of the DM test in the context of day-ahead EPF:

- a *univariate* variant with 24 independent tests performed,[17] one for each hour of the day, and comparisons based on the number of hours for which the predictions of one model are significantly better than those of another, i.e. the number of hours for which the null hypothesis is rejected,
- a *multivariate* variant with the test performed jointly for all hours using the 'daily' or multivariate loss differential series:

$$\Delta^{\text{A,B}}_{d} = \|\varepsilon^{\text{A}}_{d}\|_p - \|\varepsilon^{\text{B}}_{d}\|_p, \tag{13}$$

where $\varepsilon^{\text{Z}}_{d}$ is the 24-dimensional vector of prediction errors of model Z for day d, $\|\varepsilon^{\text{Z}}_{d}\|_p = (\sum_{h=1}^{24} |\varepsilon^{\text{Z}}_{d,h}|^p)^{1/p}$ is the pth norm of that vector with $p = 1$ or 2.

The univariate version of the test has the advantage of providing a deeper analysis as it indicates which forecast is significantly better for which hour of the day [6,55,57,65,133,134]. The multivariate version, introduced in [64], enables a better representation of the results as it summarizes the comparison in a single p-value, which can be conveniently visualized using heat maps arranged as chessboards [9,10,69,80], see Fig. 5.

[15] Recall, that the p-value is the probability of obtaining results (in our case — loss differentials) at least as large as the ones actually observed, assuming that the null hypothesis is correct.

[16] Actually covariance stationarity is sufficient but may not be strictly necessary [132].

[17] We assume that a day-ahead market has 24 prices. For markets with prices every half hour, the univariate variant comprises 48 independent tests.

5.5.2. The Giacomini–White test

In some of the more recent EPF studies [81,135,136], the DM test has been replaced by the Giacomini–White (GW) test [137] for *conditional predictive ability*. The latter is preferred because it can be regarded as a generalization of the DM test for *unconditional predictive ability*: while both tests can be used for nested and non-nested models,[18] only the GW test accounts for parameter estimation uncertainty through 'conditioning' [63].

Like the DM test, also the GW test has two variants in day-ahead EPF — the univariate and the multivariate. Without loss of generality, let us focus on the latter. It starts by building a multivariate loss differential series, see (13), for a pair of forecasts (of models A and B). Next, the test considers the following regression:

$$\Delta_d^{\mathrm{A,B}} = \boldsymbol{\phi}' X_{d-1} + \epsilon_d, \tag{14}$$

where X_{d-1} contains elements from the information set on day $d-1$, i.e. a constant and lags of $\Delta_d^{\mathrm{A,B}}$. Note that $\epsilon_d \neq \epsilon_d^{Z}$, i.e. ϵ_d is not the 24-dimensional vector of prediction errors for day d and model Z but simply an error term in the regression. Also note that using this notation the DM test can be written as [138]:

$$\Delta_d^{\mathrm{A,B}} = \mu + \epsilon_d, \tag{15}$$

i.e. with X_{d-1} containing just a constant. Finally, like for the DM test, to check the significance of differences in forecasting accuracy, the p-values of two one-sided tests can be computed. The interpretation and possible visualization (see Fig. 5) are analogous to that of the DM test.

5.6. Recalibration

An issue with many EPF studies is that forecasting models are not recalibrated. Instead, they are often estimated once using the training dataset and directly evaluated over the whole test dataset. This is problematic as it does not represent real-life conditions where forecasting models are retrained (often on a daily basis) to account for the latest market information.

To have models that are evaluated in realistic conditions, they need to be retrained considering the new incoming flow of market information. As an example, for the day-ahead market, a forecasting model should be retrained on a daily basis as new information is available. Considering a testing period of a year, this means that a realistic evaluation requires estimating the forecasting model 365 times.

5.7. Ex-ante hyperparameter optimization

A common issue in the current EPF literature is that the hyperparameter selection is often either done ex-post [49,51,139–142] or its details are not sufficiently explained [13,21,37,48,89,92,102–104,107,110]. As an example, when models based on neural networks are proposed, the details on how the number of neurons are selected are usually not provided. In other cases, while the approach is provided, it is often based on analyzing different configurations of neurons using the test dataset and selecting the one that works best, i.e. ex-post hyperparameter selection.

Not providing enough details on how hyperparameters are selected is an obvious problem as it prevents reproducing research. Similarly, performing hyperparameter optimization ex-post leads to overfitting the test dataset, i.e. the model is partially optimized using the same dataset used for evaluating the model, and it grants the model an unfair and non-existent advantage over other models.

To prevent this, the selection of hyperparameters should be explicitly explained and always performed ex-ante using a validation dataset. With that motivation, for the open-access methods proposed, not only do we explain how the hyperparameters are obtained, but we also provide within the toolbox [58,59] a module for hyperparameter selection and the files containing the results of the hyperparameter optimization of the current study.

5.8. Computation time

An even more common problem is the fact that new models are very rarely compared in terms of their computational requirements [19,20,22,24,32,37,41,42,51,100,102,103,105–108,111]. Although a model might be marginally better than another, it might not be worthwhile to deploy it in a practical application if its computational requirements are much larger. Particularly, higher computational requirements might pose two problems:

1. As mentioned before, forecasting models should ideally be recalibrated on a daily basis. Hence, a forecasting method is only suitable if its computational time allows this recalibration to take place. In this context, the maximum available time for estimating a model will depend on each electricity market but, as a rule of thumb, it can be argued that any model that requires more than 30 min or 1 h will unlikely be suitable for forecasting prices in the spot markets.
2. Besides recalibration, the second issue with computation time is its cost. If the computational capabilities are too large, the benefits of using a marginally better forecast might be lower than the cost of running the forecasting model on a much more expensive computer.

Hence, when new forecasting models are proposed, we argue that it is very important to provide their computation times. Moreover, we also argue that for a model to be better than the existing methods, it does not necessarily have to be the most accurate one. Instead:

1. If its computational time is large, i.e. in the order of minutes, the model should indeed be more accurate than all state-of-the-art models, e.g. DNNs.
2. If its computational time is small, i.e. in the order of seconds, the model should be more accurate than the state-of-the-art models with low computational requirements, e.g. LEAR.

In this article, we provide an analysis of the computational requirements of the proposed open-access models so other researchers can easily make such comparisons.

5.9. Reproducibility

Another related issue is that some studies lack enough details to replicate the research. Missing details vary from study to study but the four most common are:

1. the dataset used for testing and evaluation is not defined [31–37];
2. the dataset used for training is not defined [21,33,35,41,42];
3. the inputs of the model are unclear [35,36,38–40];
4. the selection of hyperparameters is unclear [13,21,37,48,89,92,102–104,107,110].

To correct this, future EPF papers should provide enough details to allow replication and reviewers should verify that all necessary details of the employed datasets are always provided.

[18] This holds as long as the calibration window does not grow with the sample size [138]. This is satisfied for rolling windows, but not for extended calibration windows.

5.10. Data contamination

Another recurrent issue in the EPF literature is data contamination, which appears when a part of the training dataset is used for testing. Particularly, when working with time series data the test dataset should always comprise the last part of the dataset to avoid data contamination. If this is not done, the models can overfit the testing dataset and their accuracy can be overestimated.

Despite the importance of correctly separating the training/validation dataset from the testing dataset, some studies in EPF:

1. Do not specify the split between the training, validation, and test datasets [21,31–37,41,42]. If the datasets are not specified, it is not possible to know whether data contamination occurs.
2. Randomly sample the test dataset from the full dataset [143–146], e.g. in a dataset comprising a year of data randomly selecting 4 weeks for testing and the remaining data for training.
3. Have a partial or total overlap between the training/validation dataset and the testing dataset [51,139,140,147], e.g. by performing hyperparameter optimization ex-post.

To correct this issue, it is important that any future research in EPF ensures that: (1) the split between the datasets is correctly described; (2) the test dataset does never overlap with the training or validation datasets; (3) the test dataset is always selected as the last segment of the full dataset.

5.11. Software toolboxes

A less pressing yet relevant issue is the use of state-of-the-art software toolboxes. When comparing new methods with methods from the literature, the latter should be modeled using adequate toolboxes. Particularly, it is important to use toolboxes that are continuously updated as implementing methods using outdated libraries leads to unfair evaluations.

For example, in the context of neural networks, there are several open-source state-of-the-art toolboxes [126] that are continuously updated and that grant access to the latest development in the field of DL. Yet, in the context of EPF, new methods are often compared with neural networks that are modeled using the `MATLAB` toolbox [32,38,41,42,49,102,103,105,144], a toolbox that for many years was outdated and did not include many of the neural network developments that are critical in EPF, e.g. state-of-the-art activation functions or stochastic gradient descent algorithms [57]. As a result, many of the existing comparisons in EPF are based on evaluations where the accuracy of neural networks might be underestimated.

Besides using state-of-the-art software toolboxes, e.g. the `python` library `keras` for deep learning, it is also important to employ (whenever possible) free-to-access libraries so that research can be replicated by anyone.

5.12. Combining forecasts

As a final guideline, it is important to indicate the importance of ensembles in the context of EPF. In general, although exceptions exist [148], combining different models leads to a higher accuracy [81,134] and it is thus a good idea to build forecasts based on multiple models. However, as even the arithmetic average improves the accuracy of individual models, new ensemble techniques should be studied in comparison with other ensemble techniques, i.e. as done in [134], and not simply w.r.t. the individual models.

To maximize the forecasting accuracy, it is important to employ diverse forecasts [148], e.g. forecasts generated using different data or different models. For EPF, the former can be achieved by considering models trained using different calibration window lengths [80,135] and the latter using different modeling techniques or different sets of hyperparameters. To further maximize the performance, the number of models used in the ensemble should be limited [148], e.g. 4–10, especially in the case of heavy-tailed data for which large ensembles tend to contain outliers more often, resulting in less accurate forecasts.

With that in mind, as part of the open-access benchmark and toolbox [58,59], we also propose a series of simple ensemble techniques. Particularly, as explained in Section 4, we provide an ensemble of four LEAR models that are estimated over different calibration windows and combined using a simple arithmetic average and another ensemble using four DNNs that are estimated for different hyperparameters and combined using the arithmetic average.

6. Evaluation of state-of-the-art methods

In this section, we present the results of the open-source benchmark methods for all five datasets. For the sake of clarity, we divide the section into two parts respectively comprising the results for the error metrics and the results for statistical testing.

6.1. Accuracy metrics

We first start by presenting the results of the open-access benchmark models in terms of accuracy metrics.

6.1.1. Individual models

Table 2 compares the performance of the two individual models and their 4 variations in terms of rMAE, MAE, MAPE, SMAPE, and RMSE. The LEAR model is displayed for 4 different calibration windows representing 56, 84, 1092, and 1456 days, i.e. 8 weeks, 12 weeks, 3 years, and 4 years. The four DNNs are obtained by performing the hyperparameter/feature optimization process four times and using the best hyperparameter/feature selection of every run (see Sections 4.4 and 4.3.3 for further details).[19] Several observations can be made:

- The MAPE seems an unreliable metric as it completely disagrees with the other three linear metrics and the quadratic metric. In particular, while the rMAE, MAE, and sMAPE agree on what the best model is in all the cases, the MAPE almost never does so. This unreliability can be further seen in the German market: while the MAPE and sMAPE metrics usually have similar orders of magnitude, in the case of the German market the MAPE is approximately 10 times larger. This effect is due to prices in Germany being negative and very close to 0, leading in turn to very large absolute percentage errors that bias the MAPE.
- The DNN models seem to be more accurate than the LEAR models. Particularly, in terms of linear metrics, the best model across the five marketplaces is a DNN. Moreover, the majority of the DNN models perform better than the four LEAR models.
- Although the RMSE displays slightly different results, this is expected as the metric is based on quadratic errors and not linear ones. Nonetheless, it still shows the superiority of the DNN model: even though the DNN is estimated by minimizing absolute errors (unlike LEAR), the DNN is better in 3 of the 5 datasets. Moreover, even though the DNN seems to be worse in two markets, the RMSE metric does not correctly represent the underlying problem (see Sections 5.4 and 6.4.1) and it can be argued that it is not the best metric to assess the performance of EPF models.

[19] Note that, for the sake of simplicity, the features and hyperparameter selection for each model are not provided. However, they can be obtained from the website [58] accompanying this study.

Table 2
Comparison between the two individual state-of-the-art open-source methods in terms of rMAE, MAE, MAPE, sMAPE, and RMSE. Each of the two methods is listed for four different configurations. The gray cells represent the best model for a given metric.

		DNN_1	DNN_2	DNN_3	DNN_4	$LEAR_{56}$	$LEAR_{84}$	$LEAR_{1092}$	$LEAR_{1456}$
NP	rMAE	0.435	0.512	0.414	0.455	0.475	0.472	0.482	0.481
	MAE	1.797	2.118	1.712	1.883	1.964	1.952	1.993	1.990
	MAPE [%]	5.738	6.527	5.584	5.814	6.336	6.357	6.099	6.144
	sMAPE [%]	5.167	5.982	4.970	5.367	5.656	5.619	5.641	5.658
	RMSE	3.474	3.859	3.360	3.489	3.671	3.664	3.605	3.604
PJM	rMAE	0.511	0.499	0.491	0.486	0.550	0.548	0.490	0.489
	MAE	3.234	3.157	3.105	3.075	3.477	3.467	3.098	3.095
	MAPE [%]	30.622	29.345	27.554	27.975	32.520	32.341	30.279	30.239
	sMAPE [%]	12.518	12.212	12.271	12.004	13.677	13.576	12.331	12.538
	RMSE	6.231	6.773	5.200	5.498	5.718	5.709	5.264	5.142
EPEX BE	rMAE	0.620	0.621	0.620	0.597	0.682	0.669	0.649	0.653
	MAE	6.299	6.308	6.297	6.068	6.924	6.798	6.594	6.634
	MAPE [%]	24.650	27.710	27.578	25.466	32.878	32.343	26.256	22.645
	sMAPE [%]	14.543	14.723	14.980	14.106	16.197	15.954	16.867	17.293
	RMSE	16.360	16.666	16.115	15.950	16.371	16.291	16.458	16.420
EPEX FR	rMAE	0.575	0.573	0.554	0.591	0.638	0.624	0.580	0.597
	MAE	4.218	4.198	4.063	4.334	4.681	4.575	4.250	4.378
	MAPE [%]	14.284	13.757	15.160	15.513	19.031	18.087	14.955	14.896
	sMAPE [%]	12.124	11.698	11.488	12.176	13.427	13.281	13.250	14.054
	RMSE	11.772	12.345	11.880	12.354	11.732	10.759	11.337	11.462
EPEX DE	rMAE	0.407	0.422	0.406	0.394	0.506	0.499	0.450	0.451
	MAE	3.716	3.850	3.706	3.592	4.619	4.555	4.108	4.118
	MAPE [%]	77.145	137.449	100.214	90.578	129.763	133.580	128.295	124.191
	sMAPE [%]	14.970	15.356	15.508	14.680	17.600	17.491	16.984	17.054
	RMSE	6.796	7.304	6.271	6.080	8.122	7.923	6.996	6.987

Table 3
Comparison between the ensembles of the state-of-the-art open-source methods in terms of rMAE, MAE, MAPE, sMAPE, and RMSE. The comparison also includes, for each market, the best individual performing DNN and LEAR model in terms of rMAE and MAE, i.e. the two most reliable metrics. The gray cells represent the best model for a given metric.

		DNN Ensemble	LEAR Ensemble	Best[a] DNN	Best LEAR
NP	rMAE	0.407	0.420	0.414	0.472
	MAE	1.683	1.738	1.717	1.952
	MAPE [%]	5.384	5.533	5.584	6.357
	sMAPE [%]	4.880	5.009	4.970	5.619
	RMSE	3.319	3.362	3.360	3.664
PJM	rMAE	0.452	0.476	0.486	0.489
	MAE	2.862	3.013	3.075	3.095
	MAPE [%]	27.478	30.134	27.975	30.239
	sMAPE [%]	11.331	11.980	12.004	12.538
	RMSE	5.040	5.127	5.498	5.142
EPEX BE	rMAE	0.578	0.604	0.597	0.649
	MAE	5.870	6.140	6.068	6.594
	MAPE [%]	24.892	20.720	25.466	26.256
	sMAPE [%]	13.446	14.546	14.106	16.867
	RMSE	15.966	15.974	15.950	16.458
EPEX FR	rMAE	0.527	0.543	0.554	0.580
	MAE	3.866	3.980	4.063	4.250
	MAPE [%]	13.601	14.680	15.160	14.955
	sMAPE [%]	10.812	11.566	11.488	13.250
	RMSE	11.867	10.676	11.880	11.337
EPEX DE	rMAE	0.374	0.433	0.394	0.450
	MAE	3.413	3.955	3.592	4.108
	MAPE [%]	94.434	122.412	90.578	128.295
	sMAPE [%]	14.078	15.747	14.680	16.984
	RMSE	5.927	7.079	6.080	6.996

[a]Best in terms of rMAE/MAE.

6.1.2. Ensembles

The results for the ensemble methods are listed in Table 3, which compares the performance of the two ensemble models and the best DNN and LEAR models in terms of the rMAE metric, i.e. arguably the most reliable metric. From the table, several observations can be made:

- As already argued in Section 5.12, combining models usually improves the accuracy. Particularly, the ensemble of DNNs is better than the best individual DNN model for all five markets and for all reliable metrics. Similarly, the ensemble of LEAR models is better than the best individual LEAR model for all markets and reliable metrics. The exception to this observation are the MAPE and RMSE metrics but, as already noted, MAPE is an unreliable metric and RMSE does not correctly represent the underlying problem of EPF.
- As before, in terms of rMAE, the ensemble of DNNs is the most accurate model across all markets, which again seems to suggest that the DNN models are more accurate than the LEAR models.

6.2. Statistical testing

In this section, we present the results of the open-access benchmark models in terms of the statistical tests. For the sake of simplicity, we

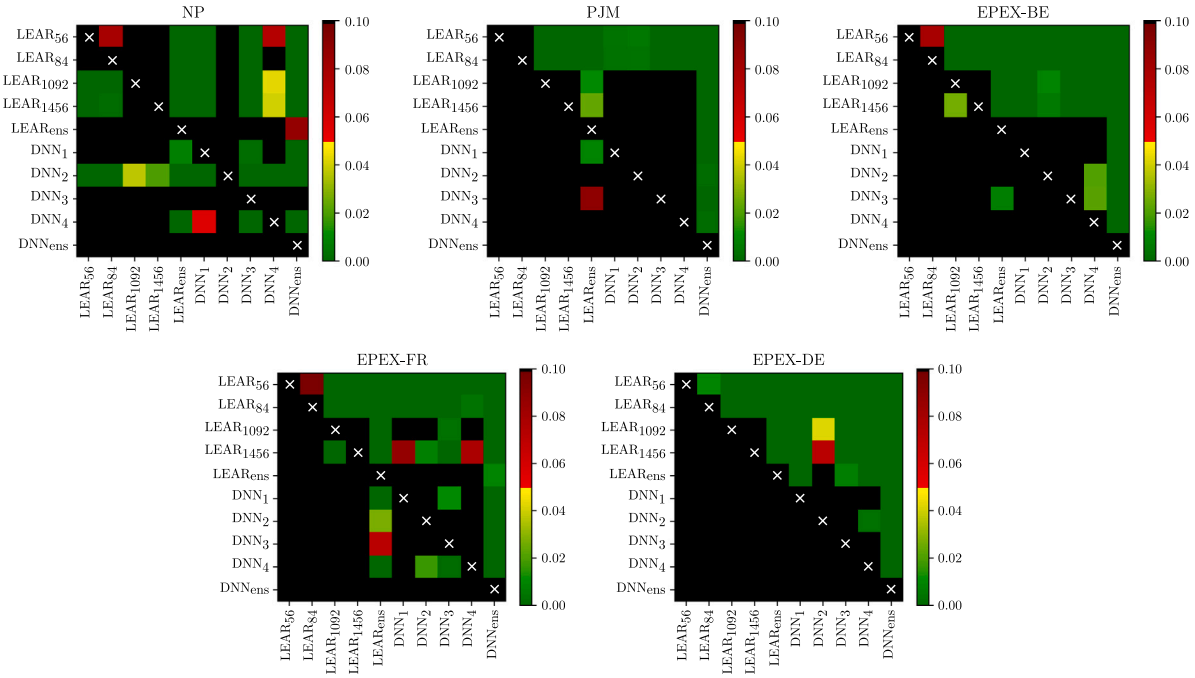


Fig. 5. Results of the GW test with the multivariate loss differential series (16) for the eight individual models and the two ensembles. A heat map is used to indicate the range of the obtained p-values for each of the five markets. The closer the p-values are to zero (dark green), the more significant the difference is between the forecasts of a model on the X-axis (better) and the forecasts of a model on the Y-axis (worse). Black color indicates p-values above the color map limit, i.e. p-values larger or equal than 0.10.

present together the results for individual methods and ensembles. The results are based on the multivariate GW test using the $L1$ norm in (13), i.e. with the following loss differential series:

$$\Delta_d^{\mathrm{A,B}} = \sum_{h=1}^{24} |\varepsilon_{d,h}^{\mathrm{A}}| - \sum_{h=1}^{24} |\varepsilon_{d,h}^{\mathrm{B}}|. \tag{16}$$

While squared losses could also be used, we do not consider them here because absolute errors better represent the underlying problem in EPF, see Section 6.4.1 for a discussion.

In Fig. 5 we display the results for the five markets. More precisely, we use heat maps arranged as chessboards to indicate the range of the obtained p-values. The closer they are to zero (dark green) the more significant is the difference between the forecasts of a model on the X-axis (better) and the forecasts of a model on the Y-axis (worse). For instance, for the EPEX-DE market the first row is green indicating that the forecasts of LEAR_{56} are significantly outperformed by those of all other models. We can observe that:

- For all markets the last column is green indicating that the forecasts of the ensemble of DNNs are statistically significantly better than the predictions of all the other models for all 5 datasets. The only exception is the LEAR ensemble and the NP market, a scenario in which the difference in forecasting accuracy is not statistically significant.
- The forecasts of $\mathrm{LEAR}_{\mathrm{ens}}$ are statistically significantly better than those of all individual LEAR models. Together with the previous observation, i.e. the superiority of the DNN ensemble, it shows that the predictions of ensemble models usually improve upon the forecasting accuracy of individual methods.
- In two datasets (EPEX-BE and EPEX-DE), the forecasts of all the individual DNN methods are statistically significantly better than those of the individual LEAR models. In the EPEX-FR dataset, the forecasts of all the individual DNN methods are statistically significantly better than 3 out of the 4 individual LEAR models. For PJM, there are 2 DNN models whose forecasts are statistically significantly better than those of all LEAR models.
- Aside from the poor-performing DNN_2 model for the NP dataset, the forecasts of the individual LEAR models are never significantly better than those of the individual DNN models. Overall, it seems that forecasts based on DNNs are more likely to obtain significantly better results; this is particularly true for the DNN ensemble.

6.3. Computation time

As described in Section 5.8, besides comparing the predictive accuracy, it is also necessary to analyze the computation time of the forecasting methods. Table 4 lists a comparison of the computation time required for estimating the models considered, i.e. the time required to recalibrate each model on a daily basis. As the computation time is non-deterministic, its value is given as a range. These data were obtained using a regular laptop quadcore CPU, i.e. the i7-6920HQ.

As can be observed, although the LEAR model performs slightly worse than the DNN model, its computation time is 30 to 100 times lower; particularly, when considering the maximum computation time of both methods, the LEAR model is 50 times faster.

6.4. Discussion and remarks

In this section, we provide some final remarks behind the motivation of the metrics employed, we briefly analyze the influence of the different metrics considered, and provide a discussion on comparing new models.

6.4.1. Absolute vs. squared errors

Throughout the text, we have mostly considered accuracy metrics based on absolute/linear errors, i.e. metrics that evaluate the accuracy of predicting the median of the distribution. Since the LEAR model is estimated by minimizing squared errors, thus leading to forecasts of the mean [129], one could argue that a metric/test based on squared errors should be preferred. While the argument has some merits, we focused on absolute metrics for three reasons:

Table 4
Computation time that each benchmark model requires to perform a daily recalibration.

	Time
LEAR	1–10 s
LEAR ensemble	20–25 s
DNN	2–5 min
DNN ensemble	8–20 min

1. The metric used to evaluate the accuracy should be the one that better represents the underlying problem. In the case of EPF, since the cost of purchasing electricity is linear, linear metrics are arguably the best to quantify the risk associated with forecasting errors.
2. While we provided the RMSE results, they are qualitatively the same as for MAE/rMAE. Hence, as absolute errors better represent the underlying problem of EPF and the results are similar, the RMSE results are not analyzed here in detail due to space limitations.
3. While the LEAR model is indeed estimated using squared errors, this is partly done because the techniques to efficiently estimate the LASSO, e.g. coordinate descent, are based on square errors. This gives the LEAR model a computational advantage over the DNN. An alternative would be to use regularized quantile regression [149] leading, however, to an increased computational burden with little benefits on the accuracy in terms of MAE/rMAE.

6.4.2. Metrics

The obtained results validate the general guidelines proposed in Section 5.4 regarding accuracy metrics: research in EPF should avoid MAPE and only use metrics like sMAPE or RMSE in conjunction with any version of rMAE. Particularly, the results validate the following four claims:

1. MAE is as reliable as rMAE. However, as the errors are not relative, comparison between datasets is not possible and rMAE is preferred.
2. sMAPE is more reliable than MAPE and it agrees with MAE/rMAE. Yet, it has the problem of an undefined mean and an infinite variance. Thus, it is less reliable than rMAE.
3. MAPE is not a reliable metric as it gives more importance to datapoints close to zero. As such, using MAPE can lead to misleading results and wrong conclusions.
4. RMSE is more reliable than MAPE but it does not represent correctly the underlying risks of EPF. Hence, it should not be used alone to evaluate forecasting models.

6.4.3. Performance of open-access models

Based on the extensive comparison of Sections 6.1–6.3, it can be concluded that the models based on DL are more likely to outperform those based on statistical methods. This is especially true in the context of DL ensemble models as the ensemble of DNNs obtains results that are statistically significantly better than any other model.

However, while DNNs outperformed the LEAR models, the latter are still the state-of-the-art in terms of low complexity and computational cost. In particular, their performance is very close to that of DNNs, but with the advantage of having computational costs that are up to 100 times lower. As such, they are the best available option when decision making has to be done within seconds.

In short, new models for EPF should either be compared against LEAR models or DNNs depending on the decision time that is available. For a method to be considered more accurate than state-of-the-art methods, it should either be more accurate than the DNN model, or more accurate than LEAR but with similar or lower computational requirements.

7. Checklist to ensure adequate EPF research

As a final contribution, and with the goal of facilitating the work of reviewers of future EPF publications, we provide a short checklist to evaluate whether any new research in EPF satisfies the requirements to be reproducible and lead to meaningful conclusions:

1. The test dataset comprises at least a year of data.
2. Any new model is tested against state-of-the-art open-access models, e.g. the ones provided here.
3. The computational cost of new methods is evaluated and compared against the computational cost of existing methods.
4. The employed datasets are open-access.
5. The study is based on multiple markets.
6. rMAE is employed as one of the accuracy metrics to evaluate forecasting accuracy.
7. Statistical testing is used to assess whether differences in performance are significant.
8. Forecasting models are recalibrated on a daily basis and not simply estimated once and evaluated in the full out-of-sample dataset.
9. Hyperparameters are estimated using a validation dataset that is different from the test dataset.
10. The split and dates of the dataset are explicitly stated.
11. All the inputs of the model are explicitly defined.
12. The test dataset is selected as the last section of the full dataset and does not contain any overlapping data with the training or validation datasets.
13. State-of-the-art and free toolboxes are used for modeling the benchmark models.

While this is just a very short summary of the guidelines described in Section 5, we think it is very useful to have them summarized together for quick evaluations of new research.

8. Conclusion

In this paper, we have derived a set of best practices for performing research in *electricity price forecasting* (EPF). Particularly, as the field of EPF lacks a rigorous approach to compare and to evaluate new forecasting models, we have analyzed different factors affecting the quality of the research, e.g. dataset size or accuracy metrics, and we have proposed solutions to ensure that new research is adequate, reproducible, and useful.

In addition, as comparisons in EPF are often done using unique datasets that no other researchers have access to, we have proposed an extensive open-access benchmark dataset comprising 6 years of recent data in 5 different markets. The aim of the benchmark dataset is to provide a common framework for future research so that new methods can be validated under the same conditions and meaningful comparisons can be obtained. To facilitate future research, we have developed an open-source `python` library named `epftoolbox` [58,59] that provides easy access to these datasets.

Similarly, as new methods in EPF are often not compared with well-established methods, we have proposed several state-of-the-art open-source models based on statistical methods and deep learning. The methods are tuned automatically and require no expert knowledge in order to be used. These methods are provided as open-source within the proposed `epftoolbox` library [58,59] so that other researchers can employ them as benchmarks in their own studies. Although the proposed methods are currently developed in `python`, we would like to extend the support to other languages; in that spirit, we encourage other researchers to help us do so.

Finally, to have a complete open-access benchmark, we have evaluated the two proposed open-access methods in the open-access dataset and we have provided the results in terms of accuracy metrics and

statistical testing. Using these results, we have shown that deep neural networks are more likely to outperform LEAR methods but that the latter are the best model for applications with short decision timeframes. Moreover, we have also shown that ensemble methods often obtain significantly better results than their individual counterparts. Based on the same results, we have also showed the importance of the guidelines as to what constitutes good practices for the rigorous use of models, metrics, and statistical tests in EPF research. The most notable guidelines were that MAPE is an unreliable metric that should be avoided, that statistical testing is mandatory to obtain meaningful conclusions, and that the length of the test dataset should be at least one year.

CRediT authorship contribution statement

Jesus Lago: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization, Project administration. **Grzegorz Marcjasz:** Methodology, Resources, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - review & editing, Visualization. **Bart De Schutter:** Supervision, Writing - review & editing, Funding acquisition. **Rafał Weron:** Conceptualization, Investigation, Resources, Supervision, Writing - review & editing, Funding acquisition.

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 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), the Ministry of Science and Higher Education (MNiSW, Poland) through grant No. 0219/DIA/2019/48 and the National Science Center (NCN, Poland) through grant No. 2018/30/A/HS4/00444.

References

[1] Brancucci Martinez-Anido C, 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.

[2] Gianfreda A, Parisio L, Pelagatti M. The impact of RES in the Italian day-ahead and balancing markets. Energy J 2016;37:161–84. http://dx.doi.org/10.5547/01956574.37.si2.agia.

[3] Grossi L, Nan F. Robust forecasting of electricity prices: Simulations, models and the impact of renewable sources. Technol Forecast Soc Change 2019;141:305–18. http://dx.doi.org/10.1016/j.techfore.2019.01.006.

[4] Maciejowska K. Assessing the impact of renewable energy sources on the electricity price level and variability – A quantile regression approach. Energy Econ 2020;85:104532. http://dx.doi.org/10.1016/j.eneco.2019.104532.

[5] 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.

[6] 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.

[7] Ziel F, Steinert R. Probabilistic mid- and long-term electricity price forecasting. Renew Sustain Energy Rev 2018;94:251–66. http://dx.doi.org/10.1016/j.rser.2018.05.038.

[8] Hong T, Pinson P, Wang Y, Weron R, Yang D, Zareipour H. Energy forecasting: A review and outlook. IEEE Open Access J Power Energy 2020;7:376–88.

[9] Uniejewski B, Weron R, Ziel F. Variance stabilizing transformations for electricity spot price forecasting. IEEE Trans Power Syst 2018;33(2):2219–29. http://dx.doi.org/10.1109/tpwrs.2017.2734563.

[10] Marcjasz G, Uniejewski B, Weron R. On the importance of the long-term seasonal component in day-ahead electricity price forecasting with NARX neural networks. Int J Forecast 2019;35(4):1520–32. http://dx.doi.org/10.1016/j.ijforecast.2017.11.009.

[11] 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.

[12] Wang L, Zhang Z, Chen J. Short-term electricity price forecasting with stacked denoising autoencoders. IEEE Trans Power Syst 2017;32(4):2673–81. http://dx.doi.org/10.1109/TPWRS.2016.2628873.

[13] Ugurlu U, Oksuz I, Tas O. Electricity price forecasting using recurrent neural networks. Energies 2018;11(5):1255. http://dx.doi.org/10.3390/en11051255.

[14] Zhang W, Cheema F, Srinivasan D. Forecasting of electricity prices using deep learning networks. In: Proceedings of the 2018 IEEE PES Asia-Pacific Power and Energy Engineering Conference. 2018, p. 451–6. http://dx.doi.org/10.1109/APPEEC.2018.8566313.

[15] Luo S, Weng Y. A two-stage supervised learning approach for electricity price forecasting by leveraging different data sources. Appl Energy 2019;242:1497–512. http://dx.doi.org/10.1016/j.apenergy.2019.03.129.

[16] Chen Y, Wang Y, Ma J, Jin Q. BRIM: An accurate electricity spot price prediction scheme-based bidirectional recurrent neural network and integrated market. Energies 2019;12(12):2241. http://dx.doi.org/10.3390/en12122241.

[17] Chang Z, Zhang Y, Chen W. Electricity price prediction based on hybrid model of Adam optimized LSTM neural network and wavelet transform. Energy 2019;187:115804. http://dx.doi.org/10.1016/j.energy.2019.07.134.

[18] Gao W, Darvishan A, Toghani M, Mohammadi M, Abedinia O, Ghadimi N. Different states of multi-block based forecast engine for price and load prediction. Int J Electr Power Energy Syst 2019;104:423–35. http://dx.doi.org/10.1016/j.ijepes.2018.07.014.

[19] Nazar MS, Fard AE, Heidari A, Shafie-khah M, Catalão JP. Hybrid model using three-stage algorithm for simultaneous load and price forecasting. Electr Power Syst Res 2018;165:214–28. http://dx.doi.org/10.1016/j.epsr.2018.09.004.

[20] Zhou L, Wang B, Wang Z, Wang F, Yang M. Seasonal classification and RBF adaptive weight based parallel combined method for day-ahead electricity price forecasting. In: Proceedings of the 2018 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference. 2018, p. 1–5. http://dx.doi.org/10.1109/isgt.2018.8403372.

[21] Singh N, Hussain S, Tiwari S. A PSO-based ANN model for short-term electricity price forecasting. In: Advances in Intelligent Systems and Computing. 2018, p. 553–63. http://dx.doi.org/10.1007/978-981-10-7386-1_47.

[22] 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.

[23] Chinnathambi RA, Plathottam SJ, Hossen T, Nair AS, Ranganathan P. Deep neural networks (DNN) for day-ahead electricity price markets. In: Proceedings of the 2018 IEEE Electrical Power and Energy Conference. 2018, p. 1–6. http://dx.doi.org/10.1109/epec.2018.8598327.

[24] Olamaee J, Mohammadi M, Noruzi A, Hosseini SMH. Day-ahead price forecasting based on hybrid prediction model. Complexity 2016;21(S2):156–64. http://dx.doi.org/10.1002/cplx.21792.

[25] Darudi A, Javidi MH, Bashari M. Electricity price forecasting using a new data fusion algorithm. IET Gener Transm Distrib 2015;9(12):1382–90. http://dx.doi.org/10.1049/iet-gtd.2014.0653.

[26] Ghayekhloo M, Azimi R, Ghofrani M, Menhaj M, Shekari E. A combination approach based on a novel data clustering method and Bayesian recurrent neural network for day-ahead price forecasting of electricity markets. Electr Power Syst Res 2019;168:184–99. http://dx.doi.org/10.1016/j.epsr.2018.11.021.

[27] Victoire AA, Gobu B, Jaikumar S, Arulmozhi N, Kanimozhi P, Victoire A. Two-stage machine learning framework for simultaneous forecasting of price-load in the smart grid. In: Proceedings of the 2018 IEEE International Conference on Machine Learning and Applications. 2018, p. 1081–6. http://dx.doi.org/10.1109/icmla.2018.00176.

[28] Zahid M, Ahmed F, Javaid N, Abbasi R, Zainab Kazmi H, Javaid A, et al. Electricity price and load forecasting using enhanced convolutional neural network and enhanced support vector regression in smart grids. Electronics 2019;8(2):122. http://dx.doi.org/10.3390/electronics8020122.

[29] Jiang L, Hu G. Day-ahead price forecasting for electricity market using long-short term memory recurrent neural network. In: Proceedings of the 2018 International Conference on Control, Automation, Robotics and Vision. 2018, p. 949–54. http://dx.doi.org/10.1109/icarcv.2018.8581235.

[30] Zhou S, Zhou L, Mao M, Tai H, Wan Y. An optimized heterogeneous structure LSTM network for electricity price forecasting. IEEE Access 2019;7:108161–73. http://dx.doi.org/10.1109/ACCESS.2019.2932999.

[31] Aggarwal A, Tripathi MM. A novel hybrid approach using wavelet transform, time series time delay neural network, and error predicting algorithm for day-ahead electricity price forecasting. In: Proceedings of the International Conference on Computer Applications in Electrical Engineering-Recent Advances. 2017, p. 199–204. http://dx.doi.org/10.1109/cera.2017.8343326.

[32] Hong Y-Y, Liu C-Y, Chen S-J, Huang W-C, Yu T-H. Short-term LMP forecasting using an artificial neural network incorporating empirical mode decomposition. Int Trans Electr Energy Syst 2014;25(9):1952–64. http://dx.doi.org/10.1002/etep.1949.

[33] Talari S, Shafie-khah M, Osório G, Wang F, Heidari A, Catalão J. Price forecasting of electricity markets in the presence of a high penetration of wind power generators. Sustainability 2017;9(11):2065. http://dx.doi.org/10.3390/su9112065.

[34] Singh N, Mohanty SR, Shukla RD. Short term electricity price forecast based on environmentally adapted generalized neuron. Energy 2017;125:127–39. http://dx.doi.org/10.1016/j.energy.2017.02.094.

[35] Khan GM, Arshad R, Khan NM. Efficient prediction of dynamic tariff in smart grid using CGP evolved artificial neural networks. In: Proceedings of the 2017 IEEE International Conference on Machine Learning and Applications. 2017, p. 493–8. http://dx.doi.org/10.1109/icmla.2017.0-113.

[36] Afrasiabi M, Mohammadi M, Rastegar M, Kargarian A. Multi-agent microgrid energy management based on deep learning forecaster. Energy 2019;186:115873. http://dx.doi.org/10.1016/j.energy.2019.115873.

[37] Zhu Y, Dai R, Liu G, Wang Z, Lu S. Power market price forecasting via deep learning. In: Proceedings of the 44th Annual Conference of the IEEE Industrial Electronics Society. 2018, http://dx.doi.org/10.1109/iecon.2018.8591581.

[38] 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.

[39] Shrivastava NA, Panigrahi BK, Lim M-H. Electricity price classification using extreme learning machines. Neural Comput Appl 2014;27(1):9–18. http://dx.doi.org/10.1007/s00521-013-1537-1.

[40] Jiang P, Ma X, Liu F. A new hybrid model based on data preprocessing and an intelligent optimization algorithm for electrical power system forecasting. Math Probl Eng 2015;2015:1–17. http://dx.doi.org/10.1155/2015/815253.

[41] Bento P, Pombo J, Calado M, Mariano S. A bat optimized neural network and wavelet transform approach for short-term price forecasting. Appl Energy 2018;210:88–97. http://dx.doi.org/10.1016/j.apenergy.2017.10.058.

[42] Khajeh MG, Maleki A, Rosen MA, Ahmadi MH. Electricity price forecasting using neural networks with an improved iterative training algorithm. Int J Ambient Energy 2017;39(2):147–58. http://dx.doi.org/10.1080/01430750.2016.1269674.

[43] 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;211:890–903. http://dx.doi.org/10.1016/j.apenergy.2017.11.098.

[44] Kuo P-H, Huang C-J. An electricity price forecasting model by hybrid structured deep neural networks. Sustainability 2018;10(4):1280. http://dx.doi.org/10.3390/su10041280.

[45] Mujeeb S, Javaid N, Ilahi M, Wadud Z, Ishmanov F, Afzal M. Deep long short-term memory: A new price and load forecasting scheme for big data in smart cities. Sustainability 2019;11(4):987. http://dx.doi.org/10.3390/su11040987.

[46] Atef S, Eltawil AB. A comparative study using deep learning and support vector regression for electricity price forecasting in smart grids. In: Proceedings of the 2019 IEEE International Conference on Industrial Engineering and Applications. 2019, p. 603–7. http://dx.doi.org/10.1109/IEA.2019.8715213.

[47] Mujeeb S, Javaid N. ESAENARX and DE-RELM: Novel schemes for big data predictive analytics of electricity load and price. Sustainable Cities Soc 2019;51:101642. http://dx.doi.org/10.1016/j.scs.2019.101642.

[48] 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.

[49] Peter SE, Raglend IJ. Sequential wavelet-ANN with embedded ANN-PSO hybrid electricity price forecasting model for Indian energy exchange. Neural Comput Appl 2016;28(8):2277–92. http://dx.doi.org/10.1007/s00521-015-2141-3.

[50] Naz A, Javed M, Javaid N, Saba T, Alhussein M, Aurangzeb K. Short-term electric load and price forecasting using enhanced extreme learning machine optimization in smart grids. Energies 2019;12(5):866. http://dx.doi.org/10.3390/en12050866.

[51] Anamika, Peesapati R, Kumar N. Electricity price forecasting and classification through wavelet–dynamic weighted PSO–FFNN approach. IEEE Syst J 2018;12(4):3075–84. http://dx.doi.org/10.1109/jsyst.2017.2717446.

[52] Gao W, Sarlak V, Parsaei MR, Ferdosi M. Combination of fuzzy based on a meta-heuristic algorithm to predict electricity price in an electricity markets. Chem Eng Res Des 2018;131:333–45. http://dx.doi.org/10.1016/j.cherd.2017.09.021.

[53] Hong T, Pinson P, Fan S, Zareipour H, Troccoli A, Hyndman RJ. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. Int J Forecast 2016;32(3):896–913. http://dx.doi.org/10.1016/j.ijforecast.2016.02.001.

[54] Nord Pool website. URL www.nordpoolspot.com.

[55] 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.

[56] Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc Ser B Stat Methodol 1996;267–88. http://dx.doi.org/10.1111/j.2517-6161.1996.tb02080.x.

[57] Lago J, De Ridder F, De Schutter B. Forecasting spot electricity prices: deep learning approaches and empirical comparison of traditional algorithms. Appl Energy 2018;221:386–405. http://dx.doi.org/10.1016/j.apenergy.2018.02.069.

[58] Epftoolbox library. URL https://github.com/jeslago/epftoolbox.

[59] Epftoolbox documentation. URL https://epftoolbox.readthedocs.io.

[60] Mayer K, Trück S. Electricity markets around the world. J Commodity Mark 2018;9:77–100. http://dx.doi.org/10.1016/j.jcomm.2018.02.001.

[61] Aïd R. Electricity derivatives. Springer; 2015, http://dx.doi.org/10.1007/978-3-319-08395-7.

[62] Maciejowska K, Weron R. Electricity price forecasting. In: Wiley StatsRef: Statistics reference online. Wiley; 2019, p. 1–9. http://dx.doi.org/10.1002/9781118445112.stat08215.

[63] Weron R, Ziel F. Electricity price forecasting. In: Soytas U, Sari R, editors. Routledge handbook of energy economics. Routledge; 2018, p. 506–21. http://dx.doi.org/10.4324/9781315459653-36.

[64] Ziel F, Weron R. Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks. Energy Econ 2018;70:396–420. http://dx.doi.org/10.1016/j.eneco.2017.12.016.

[65] Gianfreda A, Ravazzolo F, Rossini L. Comparing the forecasting performances of linear models for electricity prices with high RES penetration. Int J Forecast 2020;36:974–86. http://dx.doi.org/10.1016/j.ijforecast.2019.11.002.

[66] Zou H, Hastie T. Regularization and variable selection via the elastic net. J R Stat Soc Ser B Stat Methodol 2005;67(2):301–20. http://dx.doi.org/10.1111/j.1467-9868.2005.00503.x.

[67] 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.

[68] Ziel F. Forecasting electricity spot prices using lasso: On capturing the autoregressive intraday structure. IEEE Trans Power Syst 2016;31(6):4977–87. http://dx.doi.org/10.1109/tpwrs.2016.2521545.

[69] Uniejewski B, Weron R. Efficient forecasting of electricity spot prices with expert and LASSO models. Energies 2018;11(8):2039. http://dx.doi.org/10.3390/en11082039.

[70] Uniejewski B, Marcjasz G, Weron R. Understanding intraday electricity markets: Variable selection and very short-term price forecasting using LASSO. Int J Forecast 2019;35(4):1533–47.

[71] Narajewski M, Ziel F. Ensemble forecasting for intraday electricity prices: Simulating trajectories. Appl Energy 2020;279:115801.

[72] Muniain P, Ziel F. Probabilistic forecasting in day-ahead electricity markets: Simulating peak and off-peak prices. Int J Forecast 2020;36(4):1193–210.

[73] Uniejewski B, Weron R. Regularized quantile regression averaging for probabilistic electricity price forecasting. Energy Econ 2021;95:105121.

[74] Schneider S. Power spot price models with negative prices. J Energy Mark 2011;4(4):77–102. http://dx.doi.org/10.21314/jem.2011.079.

[75] Diaz G, Planas E. A note on the normalization of Spanish electricity spot prices. IEEE Trans Power Syst 2016;31(3):2499–500. http://dx.doi.org/10.1109/tpwrs.2015.2449757.

[76] Nowotarski J, Tomczyk J, Weron R. Robust estimation and forecasting of the long-term seasonal component of electricity spot prices. Energy Econ 2013;39:13–27. http://dx.doi.org/10.1016/j.eneco.2013.04.004.

[77] Nowotarski J, Weron R. On the importance of the long-term seasonal component in day-ahead electricity price forecasting. Energy Econ 2016;57:228–35. http://dx.doi.org/10.1016/j.eneco.2016.05.009.

[78] Lisi F, Pelagatti M. Component estimation for electricity market data: Deterministic or stochastic? Energy Econ 2018;74:13–37. http://dx.doi.org/10.1016/j.eneco.2018.05.027.

[79] Marcjasz G, Uniejewski B, Weron R. On the importance of the long-term seasonal component in day-ahead electricity price forecasting with NARX neural networks. Int J Forecast 2019;35(4):1520–32. http://dx.doi.org/10.1016/j.ijforecast.2017.11.009.

[80] Hubicka K, Marcjasz G, Weron R. A note on averaging day-ahead electricity price forecasts across calibration windows. IEEE Trans Sustain Energy 2019;10(1):321–3. http://dx.doi.org/10.1109/tste.2018.2869557.

[81] Marcjasz G, Serafin T, Weron R. Selection of calibration windows for day-ahead electricity price forecasting. Energies 2018;11(9):2364. http://dx.doi.org/10.3390/en11092364.

[82] Maciejowska K, Uniejewski B, Serafin T. PCA forecast averaging – Predicting day-ahead and intraday electricity prices. Energies 2020;13(14):3530.

[83] Pesaran M, Timmermann A. Selection of estimation window in the presence of breaks. J Econometrics 2007;137(1):134–61.

[84] De Marcos R, Bunn D, Bello A, Reneses J. Short-term electricity price forecasting with recurrent regimes and structural breaks. Energies 2020;13(20):5452.

[85] Nitka W, Serafin T, Sotiros D. Forecasting electricity prices: Autoregressive hybrid nearest neighbors (ARHNN) method. In: ICCS 2021. In: Lecture Notes in Computer Science. 2021 [forthcoming].

[86] Westerlund J, Narayan P. Testing for predictability in conditionally heteroskedastic stock returns. J Financ Econ 2015;13(2):342–75.

[87] Karakatsani N, Bunn D. Fundamental and behavioural drivers of electricity price volatility. Stud Nonlinear Dyn Econom 2010;14(4):4.

[88] Mujeeb S, Javaid N, Akbar M, Khalid R, Nazeer O, Khan M. Big data analytics for price and load forecasting in smart grids. Lecture Notes on Data Engineering and Communications Technologies, Springer; 2018, p. 77–87. http://dx.doi.org/10.1007/978-3-030-02613-4_7.

[89] Xie X, Xu W, Tan H. The day-ahead electricity price forecasting based on stacked CNN and LSTM. Lecture Notes in Computer Science, Springer International Publishing; 2018, p. 216–30. http://dx.doi.org/10.1007/978-3-030-02698-1_19.
[90] 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):2093. http://dx.doi.org/10.3390/en11082093.
[91] Kolberg JK, Waage K. Artificial Intelligence and Nord Pool's intraday electricity market Elbas: a demonstration and pragmatic evaluation of employing deep learning for price prediction: using extensive market data and spatio-temporal weather forecasts [Master's thesis], Norwegian School of Economics; 2018.
[92] Xu J, Baldick R. Day-ahead price forecasting in ERCOT market using neural network approaches. In: Proceedings of the tenth ACM International Conference on Future Energy Systems. 2019, p. 486–91. http://dx.doi.org/10.1145/3307772.3331024.
[93] Meier J-H, Schneider S, Schmidt I, Schüller P, Schönfeldt T, Wanke B. ANN-based electricity price forecasting under special consideration of time series properties. In: Information and Communication Technologies in Education, Research, and Industrial Applications. Springer International Publishing; 2019, p. 262–75. http://dx.doi.org/10.1007/978-3-030-13929-2_13.
[94] Chang Z, Zhang Y, Chen W. Effective Adam-optimized LSTM neural network for electricity price forecasting. In: Proceedings of the 2018 IEEE International Conference on Software Engineering and Service Science. 2018, p. 245–8. http://dx.doi.org/10.1109/icsess.2018.8663710.
[95] Jahangir H, Tayarani H, Baghali S, Ahmadian A, Elkamel A, Aliakbar Golkar M, et al. A novel electricity price forecasting approach based on dimension reduction strategy and rough artificial neural networks. IEEE Trans Ind Inf 2019;16(4):2369–81. http://dx.doi.org/10.1109/TII.2019.2933009.
[96] Ahmad W, Javaid N, Chand A, Shah SYR, Yasin U, Khan M, et al. Electricity price forecasting in smart grid: A novel E-CNN model. In: Web, Artificial Intelligence and Network Applications. Springer International Publishing; 2019, p. 1132–44. http://dx.doi.org/10.1007/978-3-030-15035-8_109.
[97] Aineto D, Iranzo-Sánchez J, Lemus-Zúñiga LG, Onaindia E, Urchueguía JF. On the influence of renewable energy sources in electricity price forecasting in the Iberian market. Energies 2019;12(11):2082. http://dx.doi.org/10.3390/en12112082.
[98] Schnürch S, Wagner A. Machine learning on EPEX order books: Insights and forecasts. 2019, arXiv preprint arXiv:1906.06248.
[99] Zhang J, Tan Z, Li C. A novel hybrid forecasting method using GRNN combined with wavelet transform and a GARCH model. Energy Sources B: Econ Plann Policy 2015;10(4):418–26. http://dx.doi.org/10.1080/15567249.2011.557685.
[100] Zhang J-L, Zhang Y-J, Li D-Z, Tan Z-F, Ji J-F. Forecasting day-ahead electricity prices using a new integrated model. Int J Electr Power Energy Syst 2019;105:541–8. http://dx.doi.org/10.1016/j.ijepes.2018.08.025.
[101] Kurbatsky V, Sidorov D, Spiryaev V, Tomin N. Forecasting nonstationary time series based on Hilbert–Huang transform and machine learning. Autom Remote Control 2014;75(5):922–34.
[102] Varshney H, Sharma A, Kumar R. A hybrid approach to price forecasting incorporating exogenous variables for a day ahead electricity market. In: Proceedings of the 2016 IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems. 2016, p. 1–6. http://dx.doi.org/10.1109/icpeices.2016.7853355.
[103] 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.
[104] Bisoi R, Dash PK, Das PP. Short-term electricity price forecasting and classification in smart grids using optimized multikernel extreme learning machine. Neural Comput Appl 2018;32:1457–80. http://dx.doi.org/10.1007/s00521-018-3652-5.
[105] Kim MK. Short-term price forecasting of Nordic power market by combination Levenberg–Marquardt and Cuckoo search algorithms. IET Gener Transm Distrib 2015;9(13):1553–63. http://dx.doi.org/10.1049/iet-gtd.2014.0957.
[106] Pourdaryaei A, Mokhlis H, Illias HA, Kaboli SHA, Ahmad S. Short-term electricity price forecasting via hybrid backtracking search algorithm and ANFIS approach. IEEE Access 2019;7:77674–91. http://dx.doi.org/10.1109/access.2019.2922420.
[107] Ebrahimian H, Barmayoon S, Mohammadi M, Ghadimi N. The price prediction for the energy market based on a new method. Econ Res-Ekonomska Istraživanja 2018;31(1):313–37. http://dx.doi.org/10.1080/1331677x.2018.1429291.
[108] Abedinia O, Amjady N, Shafie-khah M, Catalão J. Electricity price forecast using Combinatorial Neural Network trained by a new stochastic search method. Energy Convers Manage 2015;105:642–54. http://dx.doi.org/10.1016/j.enconman.2015.08.025.
[109] Itaba S, Mori H. An electricity price forecasting model with fuzzy clustering preconditioned ANN. Electr Eng Japan 2018;204(3):10–20. http://dx.doi.org/10.1002/eej.23094.
[110] Ghofrani M, Azimi R, Najafabadi FM, Myers N. A new day-ahead hourly electricity price forecasting framework. In: Proceedings of the 2017 North American Power Symposium. 2017, p. 1–6. http://dx.doi.org/10.1109/naps.2017.8107269.
[111] Itaba S, Mori H. A fuzzy-preconditioned GRBFN model for electricity price forecasting. Procedia Comput Sci 2017;114:441–8. http://dx.doi.org/10.1016/j.procs.2017.09.010.
[112] ENTSO-E transparency platform. URL https://transparency.entsoe.eu/.
[113] PJM website. URL www.pjm.com.
[114] Elia. Grid data. URL http://www.elia.be/en/grid-data/dashboard.
[115] RTE. Grid data. URL https://data.rte-france.com/.
[116] Amprion website. URL https://www.amprion.net/.
[117] 50 Hertz website. URL https://www.50hertz.com/.
[118] TenneT website. URL https://www.tennet.eu/.
[119] Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for hyper-parameter optimization. In: Advances in Neural Information Processing Systems. 2011, p. 2546–54.
[120] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning. Springer Series in Statistics, New York, NY, USA: Springer New York Inc.; 2001, http://dx.doi.org/10.1007/978-0-387-21606-5.
[121] Ziel F, Steinert R, Husmann S. Efficient modeling and forecasting of electricity spot prices. Energy Econ 2015;47:98–111. http://dx.doi.org/10.1016/j.eneco.2014.10.012.
[122] Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. Ann Statist 2004;32(2):407–99. http://dx.doi.org/10.1214/009053604000000067.
[123] Kingma DP, Ba J. Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR. 2015, http://arxiv.org/abs/1412.6980.
[124] 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.
[125] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Mach Learn Res 2011;12:2825–30.
[126] Chollet F. Keras. In: GitHub repository. GitHub; 2015, URL https://github.com/fchollet/keras.
[127] Hyndman RJ. Errors on percentage errors. 2014, URL https://robjhyndman.com/hyndsight/smape/.
[128] 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.
[129] Hyndman RJ, Athanasopoulos G. Forecasting: Principles and Practice. OTexts; 2018.
[130] Narayan PK, Bannigidadmath D. Are Indian stock returns predictable? J Bank Financ 2015;58:506–31.
[131] 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.
[132] Diebold FX. Comparing predictive accuracy, twenty years later: A personal perspective on the use and abuse of Diebold–Mariano tests. J Bus Econom Statist 2015;33(1):1–9. http://dx.doi.org/10.1080/07350015.2014.983236.
[133] Bordignon S, Bunn DW, Lisi F, Nan F. Combining day-ahead forecasts for British electricity prices. Energy Econ 2013;35:88–103. http://dx.doi.org/10.1016/j.eneco.2011.12.001.
[134] 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.
[135] Serafin T, Uniejewski B, Weron R. Averaging predictive distributions across calibration windows for day-ahead electricity price forecasting. Energies 2019;12(13):2561. http://dx.doi.org/10.3390/en12132561.
[136] Marcjasz G, Lago J, Weron R, Schutter BD. 2020. Neural networks in day-ahead electricity price forecasting: single vs. multiple outputs,
[137] Giacomini R, White H. Tests of conditional predictive ability. Econometrica 2006;74(6):1545–78. http://dx.doi.org/10.1111/j.1468-0262.2006.00718.x.
[138] Giacomini R, Rossi B. Forecasting in macroeconomics. In: Handbook of Research Methods and Applications in Empirical Macroeconomics. Edward Elgar Publishing; 2013, p. 381–408. http://dx.doi.org/10.4337/9780857931023.00024.
[139] Ibrahim NNAN, Razak IAWA, Sidin SSM, Bohari ZH. Electricity price forecasting using neural network with parameter selection. In: Intelligent and Interactive Computing. Springer Singapore; 2019, p. 141–8. http://dx.doi.org/10.1007/978-981-13-6031-2_33.
[140] 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.
[141] Singh NK, Singh AK, Tripathy M. Short-term load/price forecasting in deregulated electric environment using ELMAN neural network. In: Proceedings of the 2015 International Conference on Energy Economics and Environment. 2015, p. 1–6. http://dx.doi.org/10.1109/energyeconomics.2015.7235086.
[142] Reddy SS, Jung C-M, Seog KJ. Day-ahead electricity price forecasting using back propagation neural networks and weighted least square technique. Front Energy 2016;10(1):105–13. http://dx.doi.org/10.1007/s11708-016-0393-y.
[143] Nascimento J, Pinto T, Vale Z. Day-ahead electricity market price forecasting using artificial neural network with spearman data correlation. In: Proceedings of the 2019 IEEE PowerTech Conference. 2019, p. 1–6. http://dx.doi.org/10.1109/ptc.2019.8810618.

[144] Kotur D, Zarkovic M. Neural network models for electricity prices and loads short and long-term prediction. In: Proceedings of the 2016 International Symposium on Environmental Friendly Energies and Applications. 2016, p. 1–5. http://dx.doi.org/10.1109/efea.2016.7748787.

[145] Monteiro C, Ramirez-Rosado I, Fernandez-Jimenez L, Conde P. Short-term price forecasting models based on artificial neural networks for intraday sessions in the Iberian electricity market. Energies 2016;9(9):721. http://dx.doi.org/10.3390/en9090721.

[146] Monteiro C, Fernandez-Jimenez L, Ramirez-Rosado I. Explanatory information analysis for day-ahead price forecasting in the Iberian electricity market. Energies 2015;8(9):10464–86. http://dx.doi.org/10.3390/en80910464.

[147] Anamika, Kumar N. Market-clearing price forecasting for Indian electricity markets. In: Proceeding of International Conference on Intelligent Communication, Control and Devices. Springer Singapore; 2016, p. 633–42. http://dx.doi.org/10.1007/978-981-10-1708-7_72.

[148] Atiya AF. Why does forecast combination work so well? Int J Forecast 2020;36(1):197–200. http://dx.doi.org/10.1016/j.ijforecast.2019.03.010.

[149] Li Y, Zhu J. L1-norm quantile regression. J Comput Graph Statist 2008;17:163–85. http://dx.doi.org/10.1198/106186008x289155.

Paper 3

Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx

Kin G. Olivares, Christian Challu, Grzegorz Marcjasz, Rafał Weron,
Artur Dubrawski

International Journal of Forecasting 39 (2023) 884–900

Contents lists available at ScienceDirect

International Journal of Forecasting

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

Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx

Kin G. Olivares [a,*], Cristian Challu [a], Grzegorz Marcjasz [b], Rafał Weron [b], Artur Dubrawski [a]

[a] Auton Lab, School of Computer Science, Carnegie Mellon University, United States
[b] Department of Operations Research and Business Intelligence, Wroclaw University of Science and Technology, Poland

ARTICLE INFO

Keywords:
Deep learning
NBEATS and NBEATSx models
Interpretable neural network
Time series decomposition
Fourier series
Electricity price forecasting

ABSTRACT

We extend neural basis expansion analysis (NBEATS) to incorporate exogenous factors. The resulting method, called NBEATSx, improves on a well-performing deep learning model, extending its capabilities by including exogenous variables and allowing it to integrate multiple sources of useful information. To showcase the utility of the NBEATSx model, we conduct a comprehensive study of its application to electricity price forecasting tasks across a broad range of years and markets. We observe state-of-the-art performance, significantly improving the forecast accuracy by nearly 20% over the original NBEATS model, and by up to 5% over other well-established statistical and machine learning methods specialized for these tasks. Additionally, the proposed neural network has an interpretable configuration that can structurally decompose time series, visualizing the relative impact of trend and seasonal components and revealing the modeled processes' interactions with exogenous factors. To assist related work, we made the code available in a dedicated repository.

1. Introduction

In the last decade, significant progress has been made in the application of deep learning to forecasting tasks, with models such as the exponential smoothing recurrent neural network (ESRNN; Smyl 2019) and neural basis expansion analysis (NBEATS; Oreshkin, Carpov, Chapados, and Bengio 2020) outperforming classical statistical approaches in the recent M4 competition (Makridakis, Spiliotis, & Assimakopoulos, 2020). Despite this success we still identify two possible improvements, namely the integration of time-dependent exogenous variables as their inputs and the interpretability of the neural network outputs.

Neural networks have proven powerful and flexible, yet there are several situations where our understanding of the model's predictions can be as crucial as their accuracy, which constitutes a barrier for their wider adoption. The interpretability of the algorithm's outputs is critical because it encourages trust in its predictions, improves our knowledge of the modeled processes, and provides insights that can improve the method itself.

Additionally, the absence of time-dependent covariates makes these powerful models unsuitable for many applications. For instance, electricity price forecasting (EPF) is a task where covariate features are fundamental to obtain accurate predictions. For this reason, we chose this challenging application as a test ground for our proposed forecasting methods.

In this work, we address the two mentioned limitations by first extending NBEATS, allowing it to incorporate temporal and static exogenous variables, and second, by further exploring the interpretable configuration of NBEATS and showing its use as a time-series signal decomposition tool. We refer to the new method as NBEATSx. The main contributions of this paper include:

* Corresponding author.
E-mail address: kdgutier@cs.cmu.edu (Kin G. Olivares).

https://doi.org/10.1016/j.ijforecast.2022.03.001

(i) **Incorporation of Exogenous Variables:** We propose improvements to the NBEATS model to incorporate time-dependent as well as static exogenous variables. For this purpose, we designed a special substructure built with convolutions, to clean and encode useful information from these covariates, while respecting time dependencies present in the data. These enhancements greatly improve the accuracy of the NBEATS method, and extend its interpretability capabilities, which are so rare in neural forecasting.

(ii) **Interpretable Time Series Signal Decomposition:** Our method combines the power of nonlinear transformations provided by neural networks with the flexibility to model multiple seasonalities and simultaneously account for interaction events such as holidays and other covariates, all while remaining interpretable. The extended NBEATSx architecture can decompose its predictions into the classic set of level, trend, and seasonality, and identify the effects of exogenous covariates.

(iii) **Time Series Forecasting Comparison:** We showcase the use of the NBEATSx model on five EPF tasks, achieving state-of-the-art performance on all of the considered datasets. We obtain accuracy improvements of almost 20% in comparison to the original NBEATS and ESRNN architectures, and of up to 5% over other well-established machine learning, EPF-tailored methods (Lago, Marcjasz, De Schutter, & Weron, 2021a).

The remainder of the paper is structured as follows. Section 2 reviews relevant literature on the developments and applications of deep learning to sequence modeling and current approaches to EPF. Section 3 introduces mathematical notation and describes the NBEATSx model. Section 4 explores our model's application to time series decomposition and forecasting over a broad range of electricity markets and time periods. Finally, Section 5 discusses possible directions for future research, wraps up the results, and concludes the paper.

2. Literature review

2.1. Deep learning and sequence modeling

The deep learning methodology (DL) has demonstrated significant utility in solving sequence modeling problems, with applications to natural language processing, audio signal processing, and computer vision. This subsection summarizes the critical DL developments in sequence modeling that are building blocks of the NBEATS and ESRNN architectures.

For a long time, sequence modeling with neural networks and recurrent neural networks (RNNs; Elman 1990) was treated as synonymous. The hidden internal activations of the RNNs propagated through time provided these models with the ability to encode the observed past of the sequence. This explains their great popularity in building different variants of sequence-to-sequence models (Seq2Seq) applied to natural language processing (Graves, 2013) and machine translation (Sutskever, Vinyals, & Le, 2014). Most progress on RNNs was made possible by architectural innovations and novel training techniques that made their optimization easier, and involved popular designs such as long short-term memory (LSTM; Gers, Cummins, and Schmidhuber 2000) and gated recurrent units (GRUs; Chung, Gülçehre, Cho, and Bengio 2014).

The adoption of convolutions and skip-connections within the recurrent structures were important precursors for new advancements in sequence modeling, as using deeper representations endowed longer effective memory for the models. Examples of such precursors could be found in WaveNet for audio generation and machine translation (van den Oord et al., 2016), as well as the dilated RNN (DRNN; Chang et al. 2017) and the temporal convolutional network (TCN; Bai, Kolter, and Koltun 2018).

Nowadays, Seq2Seq models and their derivatives can learn complex nonlinear temporal dependencies efficiently; their use in the time series analysis domain has been a great success. Seq2Seq models have recently showed better forecasting performance than classical statistical methods, while greatly simplifying the forecasting systems into single-box models, such as the multi-quantile convolutional neural network (MQCNN; Wen, Torkkola, Narayanaswamy, and Madeka 2017), the exponential smoothing recurrent neural network (ESRNN; Smyl 2019), and neural basis expansion analysis (NBEATS; Oreshkin et al. 2020). For quite a while, academia resisted broadly adopting these new methods (Makridakis, Spiliotis, & Assimakopoulos, 2018), although their evident success in challenges such as the M4 competition has motivated their wider adoption by the forecasting research community (Benidis et al., 2020).

2.2. Electricity price forecasting

The electricity price forecasting (EPF) task aims at predicting the spot (balancing, intraday, day-ahead) and forward prices in wholesale markets. Since the workhorse of short-term power trading is the day-ahead market with its once-per-day uniform-price auction (Mayer & Trück, 2018), the vast majority of research has focused on predicting electricity prices for the 24 h of the next day, either in a point (Lago et al., 2021a; Weron, 2014) or a probabilistic setting (Nowotarski & Weron, 2018). There also are studies on EPF for very short-term (Narajewski & Ziel, 2020) as well as mid- and long-term horizons (Ziel & Steinert, 2018). The recent expansion of renewable energy generation and large-scale battery storage has induced complex dynamics to the already volatile electricity spot prices, turning the field into a prolific subject on which to test novel forecasting ideas and trading strategies (Chitsaz, Zamani-Dehkordi, Zareipour, & Parikh, 2018; Gianfreda, Ravazzolo, & Rossini, 2020; Uniejewski & Weron, 2021).

Out of the numerous approaches to EPF developed over the last two decades, two classes of models are of particular importance when predicting day-ahead prices: statistical (also called econometric or technical analysis), in most

cases based on linear regression, and computational intelligence (also referred to as artificial intelligence, nonlinear learning, or machine learning), with neural networks being the fundamental building block). Among the latter, many of the recently proposed methods utilize deep learning (Lago, De Ridder, and De Schutter 2018, Marcjasz 2020, Wang, Zhang, and Chen 2017), or are hybrid solutions that typically comprise data decomposition, feature selection, clustering, forecast averaging, and/or heuristic optimization to estimate the model (hyper-) parameters (Li & Becker, 2021; Nazar, Fard, Heidari, Shafie-khah, & ao P.S. Catalão, 2018).

Unfortunately, as argued by Lago et al. (2021a), the majority of the neural network EPF-related research is limited to single-market test periods and suffers from a lack of well-performing and established benchmark methods and incomplete descriptions of the pipeline and training methodology, resulting in poor reproducibility of the results. To address these shortcomings, our models are compared across two-year out-of-sample periods from five power markets and using two highly competitive benchmarks recommended in previous studies: the lasso-estimated autoregressive (LEAR) model and a (relatively) parsimonious deep neural network (DNN).

3. NBEATSx model

As a general overview, the NBEATSx framework decomposes the objective signal by performing separate local nonlinear projections of the target data onto basis functions across its different blocks. Fig. 1 depicts the general architecture of the model. Each block consists of a fully connected neural network (FCNN; Rosenblatt 1961), which learns expansion coefficients for the backcast and forecast elements. The backcast model is used to clean the inputs of subsequent blocks, while the forecasts are summed to compose the final prediction. The blocks are grouped in stacks. Each of the potentially multiple stacks specializes in a different variant of basis functions.

To continue the description of NBEATSx, we introduce the following notation: the objective signal is represented by the vector $\mathbf{y}$; the inputs for the model are the backcast window vector $\mathbf{y}^{back}$ of length L and the forecast window vector $\mathbf{y}^{for}$ of length H, where L denotes the length of the lags available as classic autoregressive features and H is the forecast horizon treated as the objective. While the original NBEATS only admits as regressor the backcast period of the target variable $\mathbf{y}^{back}$, NBEATSx incorporates covariates in its analysis, denoted with the matrix $\mathbf{X}$. Fig. 1 shows an example where the target variable is the hourly electricity price, the backcast vector has a length L of 96 h, and the forecast horizon H is 72 h. In the example, the covariate matrix $\mathbf{X}$ is composed of wind power production and electricity load. For the EPF comparative analysis of Section 4.3.6, the horizon considered is $H = 24$, which corresponds to day-ahead predictions, while backcast inputs $L = 168$ correspond to a week of lagged values.

For its predictions, the NBEATS model only receives a local vector of inputs corresponding to the backcast period, making the computations exceptionally fast. The model can still represent longer time dependencies through its local inputs from the exogenous variables; for example, it can learn long seasonal effects from calendar variables.

Overall, as shown in Fig. 1, NBEATSx is composed of S stacks of B blocks each. The input $\mathbf{y}^{back}$ of the first block consists of L lags of the target time series $\mathbf{y}$ and the exogenous matrix $\mathbf{X}$, while the inputs of each of the subsequent blocks include residual connections with the backcast output of the previous block. We will describe in detail in the next subsections the blocks, stacks, and model predictions.

3.1. Blocks

For a given sth stack and bth block within it, the NBEATSx model performs two transformations, depicted in the blue rectangle of Fig. 1. The first transformation, defined in Eq. (1), takes the input data $(\mathbf{y}^{back}_{s,b-1}, \mathbf{X}_{s,b-1})$ and applies a fully connected neural network (FCNN; Rosenblatt 1961) to learn hidden units $\mathbf{h}_{s,b} \in \mathbb{R}^{N_h}$ that are linearly adapted into the forecast $\boldsymbol{\theta}^{for}_{s,b} \in \mathbb{R}^{N_s}$ and backcast $\boldsymbol{\theta}^{back}_{s,b} \in \mathbb{R}^{N_s}$ expansion coefficients, where N_s denotes the dimension of the stack basis.

$$\begin{aligned} \mathbf{h}_{s,b} &= \mathbf{FCNN}_{s,b}\left(\mathbf{y}^{back}_{s,b-1}, \mathbf{X}_{b-1}\right) \\ \boldsymbol{\theta}^{back}_{s,b} &= \mathbf{LINEAR}^{back}\left(\mathbf{h}_{s,b}\right) \qquad \boldsymbol{\theta}^{for}_{s,b} = \mathbf{LINEAR}^{for}\left(\mathbf{h}_{s,b}\right) \end{aligned} \tag{1}$$

The second transformation, defined in Eq. (2), consists of a basis expansion operation between the learnt coefficients and the block's basis vectors $\mathbf{V}^{back}_{s,b} \in \mathbb{R}^{L\times N_s}$ and $\mathbf{V}^{for}_{s,b} \in \mathbb{R}^{H\times N_s}$. This transformation results in the backcast $\hat{\mathbf{y}}^{back}_{s,b}$ and forecast $\hat{\mathbf{y}}^{for}_{s,b}$ components.

$$\hat{\mathbf{y}}^{back}_{s,b} = \mathbf{V}^{back}_{s,b}\,\boldsymbol{\theta}^{back}_{s,b} \quad \text{and} \quad \hat{\mathbf{y}}^{for}_{s,b} = \mathbf{V}^{for}_{s,b}\,\boldsymbol{\theta}^{for}_{s,b} \tag{2}$$

3.2. Stacks and residual connections

The blocks are organized into stacks using the doubly residual stacking principle, which is described in Eq. (3) and depicted in the brown rectangle of Fig. 1. The residual backcast $\mathbf{y}^{back}_{s,b+1}$ allows the model to subtract the component associated to the basis of the s,b-th stack and block $\mathbf{V}^{back}_{s,b}$ from $\mathbf{y}^{back}$, which can be also thought of as a sequential decomposition of the modeled signal. In turn, this methodology helps with the optimization procedure, as it prepares the inputs of the subsequent layer, making the downstream forecast easier. The stack forecast $\mathbf{y}^{for}_{s}$ aggregates the partial forecasts from each block.

$$\mathbf{y}^{back}_{s,b+1} = \mathbf{y}^{back}_{s,b} - \hat{\mathbf{y}}^{back}_{s,b} \qquad \text{and} \qquad \hat{\mathbf{y}}^{for}_{s} = \sum_{b=1}^{B} \hat{\mathbf{y}}^{for}_{s,b} \tag{3}$$

3.3. Model predictions

The final predictions $\hat{\mathbf{y}}^{for}$ of the model, shown in the yellow rectangle of Fig. 1, are obtained by the summation of all the stack predictions.

$$\hat{\mathbf{y}}^{for} = \sum_{s=1}^{S} \hat{\mathbf{y}}^{for}_{s} \tag{4}$$

The additive generation of the forecast implies a very intuitive decomposition of the prediction components when the bases within the blocks are interpretable.

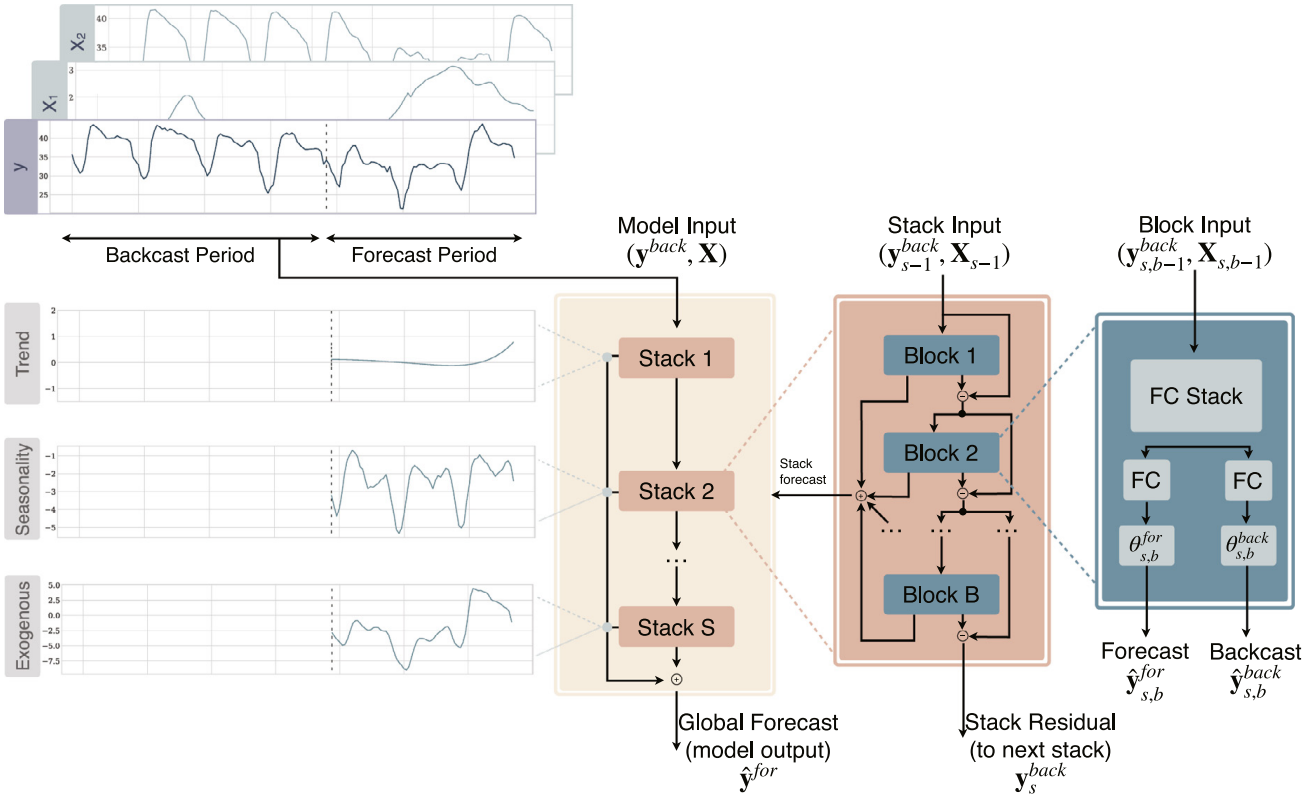


Fig. 1. The building blocks of NBEATSx are structured as a system of multilayer fully connected networks with ReLU-based nonlinearities. Blocks overlap using the doubly residual stacking principle for the backcast $\hat{\mathbf{y}}^{back}_{s,b}$ and forecast $\hat{\mathbf{y}}^{for}_{s,b}$ outputs of the bth block within the sth stack. The final predictions $\hat{\mathbf{y}}^{for}$ are composed by aggregating the outputs of the stacks.

3.4. NBEATSx configurations

The original neural basis expansion analysis method proposed two configurations based on the assumptions encoded in the learning algorithm by selecting the basis vectors $\mathbf{V}^{back}_{s,b}$ and $\mathbf{V}^{for}_{s,b}$ used in the blocks from Eq. (2). A mindful selection of restrictions to the basis allows the model to output an interpretable decomposition of the forecasts, while allowing the basis to be freely determined can produce more flexible forecasts by effectively removing any constraints on the form of the basis functions.

In this subsection, we present both interpretable and generic configurations, explaining in particular how we propose to include the covariates in each case. We limit ourselves to the analysis of the forecast basis, as the backcast basis analysis is almost identical, only differing by its extension over time. We show an example in Appendix A.1.

3.4.1. Interpretable configuration

The choice of basis vectors relies on time series decomposition techniques that are often used to understand the structure of a given time series and patterns of its variation. Work in this area ranges from classical smoothing methods and their extensions such as X-11-ARIMA, X-12-ARIMA, and X-13-ARIMA-SEATS, to modern approaches such as TBATS (Livera, Hyndman, & Snyder, 2011). To encourage interpretability, the blocks within each stack may use harmonic functions, polynomial trends, and exogenous variables directly to perform their projections. The partial forecasts of the interpretable configuration are described through Eqs. (5)–(7).

$$\hat{\mathbf{y}}^{trend}_{s,b} = \sum_{i=0}^{N_{pol}} \mathbf{t}^{i}\, \theta^{trend}_{s,b,i} \equiv \mathbf{T}\, \boldsymbol{\theta}^{trend}_{s,b} \tag{5}$$

$$\hat{\mathbf{y}}^{seas}_{s,b} = \sum_{i=0}^{\lfloor H/2-1 \rfloor} \cos\left(2\pi i \frac{\mathbf{t}}{N_{hr}}\right) \theta^{seas}_{s,b,i} + \sin\left(2\pi i \frac{\mathbf{t}}{N_{hr}}\right) \theta^{seas}_{s,b,i+\lfloor H/2 \rfloor} \equiv \mathbf{S}\, \boldsymbol{\theta}^{seas}_{s,b} \tag{6}$$

$$\hat{\mathbf{y}}^{exog}_{s,b} = \sum_{i=0}^{N_x} \mathbf{X}_i\, \theta^{exog}_{s,b,i} \equiv \mathbf{X}\, \boldsymbol{\theta}^{exog}_{s,b} \tag{7}$$

where the time vector $\mathbf{t}^{\intercal} = [0, 1, 2, \ldots, H-2, H-1]/H$ is defined discretely. When the basis $\mathbf{V}^{for}_{s,b}$ is $\mathbf{T} = [\mathbf{1}, \mathbf{t}, \ldots, \mathbf{t}^{N_{pol}}] \in \mathbb{R}^{H \times (N_{pol}+1)}$, where N_{pol} is the maximum polynomial degree, the coefficients are those of a polynomial model for the trend. When the bases $\mathbf{V}^{for}_{s,b}$ are harmonic $\mathbf{S} = [\mathbf{1}, \cos(2\pi \frac{\mathbf{t}}{N_{hr}}), \ldots, \cos(2\pi \lfloor H/2 - 1 \rfloor \frac{\mathbf{t}}{N_{hr}}), \ldots, \sin(2\pi \frac{\mathbf{t}}{N_{hr}}), \ldots, \sin(2\pi \lfloor H/2 - 1 \rfloor \frac{\mathbf{t}}{N_{hr}})] \in \mathbb{R}^{H \times (H-1)}$, the coefficient vector $\boldsymbol{\theta}^{for}_{s,b}$ can be interpreted as Fourier transform coefficients, the hyper-parameter N_{hr} controls the harmonic oscillations. The exogenous basis expansion can be thought as a time-varying local regression when the basis is the matrix $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_{N_x}] \in \mathbb{R}^{H \times N_x}$, where N_x is the number of

Table 1
Datasets used in our empirical study. For the five day-ahead electricity markets considered, we report the test period dates and two influential covariate variables.

Market	Exogenous variable 1	Exogenous variable 2	Test period
NP	day-ahead load	day-ahead wind generation	27-12-2016 to 24-12-2018
PJM	two-day-ahead system load	two-day-ahead COMED load	27-12-2016 to 24-12-2018
EPEX-FR	day-ahead load	day-ahead total France generation	04-01-2015 to 31-12-2016
EPEX-BE	day-ahead load	day-ahead total France generation	04-01-2015 to 31-12-2016
EPEX-DE	day-ahead zonal load	day-ahead wind and solar generation	04-01-2016 to 31-12-2017

exogenous variables. The resulting models can flexibly reflect common structural assumptions, in particular using the interpretable bases, as well as their combinations.

In this paper, we propose including one more type of stack to specifically represent the exogenous variable basis, as described in Eq. (7) and depicted in Fig. 1. In the original NBEATS framework (Oreshkin et al., 2020), the interpretable configuration usually consists of a trend stack followed by a seasonality stack, each containing three blocks. Our NBEATSx extension of this configuration consists of three stacks, one for each type of factor (trend, seasonal, and exogenous). We refer to this interpretable and its enhanced interpretable configuration as the NBEATS-I and NBEATSx-I models, respectively.

3.4.2. Generic configuration

For the generic configuration, the basis of the nonlinear projection in Eq. (2) corresponds to canonical vectors, that is $\mathbf{V}^{for}_{s,b} = I_{H\times H}$, an identity matrix of dimensionality equal to the forecast horizon H that matches the coefficient's cardinality $|\boldsymbol{\theta}^{for}_{s,b}| = H$.

$$\hat{\mathbf{y}}^{gen}_{s,b} = \mathbf{V}^{for}_{s,b}\,\boldsymbol{\theta}^{for}_{s,b} = \boldsymbol{\theta}^{for}_{s,b} \tag{8}$$

This basis enables NBEATSx to effectively behave like a classic fully connected neural network (FCNN). The output layer of the FCNN inside each block has H neurons that correspond to the forecast horizon, each producing the forecast for one particular time point of the forecast period. This can be understood as the basis vectors being learned during optimization, allowing the waveform of the basis of each stack to be freely determined in a data-driven fashion. Compared to the interpretable counterpart described in Section 3.4.1, the constraints on the form of the basis functions are removed. This affords the generic variant more flexibility and power at representing complex data, but it can also lead to less interpretable outcomes and potentially escalated risk of overfitting.

For the NBEATSx model with the generic configuration, we propose a new type of exogenous block that learns a context vector $\mathbf{C}_{s,b}$ from the time-dependent covariates with an *encoder* convolutional sub-structure:

$$\hat{\mathbf{y}}^{exog}_{s,b} = \sum_{i=1}^{N_c} C_{s,b,i}\,\theta^{for}_{s,b,i} \equiv \mathbf{C}_{s,b}\boldsymbol{\theta}^{for}_{s,b} \qquad \text{with} \qquad \mathbf{C}_{s,b} = \mathrm{TCN}(\mathbf{X}) \tag{9}$$

In the previous equation, a temporal convolutional network (TCN; Bai et al. 2018) is employed as an *encoder*, but any neural network with a sequential structure will be compatible with the backcast and forecast branches of the model, and could be used as an *encoder*. For example, WaveNet (van den Oord et al., 2016) can be an effective alternative to RNNs, as it is also able to capture long-term dependencies and the interactions of covariates by stacking multiple layers, while dilations help it keep the models computationally tractable. In addition, convolutions have a very convenient interpretation as a weighted moving average of signal filters. The final linear projection and the additive composition of the predictions can be interpreted as a *decoder*.

The original NBEATS configuration includes only one generic stack with dozens of blocks, while our proposed model includes both the generic and exogenous stacks, with the order determined via data-driven hyperparameter tuning. We refer to this configuration as the NBEATSx-G model.

3.4.3. Exogenous variables

We distinguish the exogenous variables by whether they reflect static or time-dependent aspects of the modeled data. The *static* exogenous variables carry time-invariant information. When the model is built with common parameters to forecast multiple time series, these variables allow information to be shared within groups of time series with similar static variable levels. Examples of static variables include designators such as identifiers of regions and groups of products, among others.

As for the *time-dependent* exogenous covariates, we discern two subtypes. First, we consider seasonal covariates from the natural frequencies in the data. These variables are useful for NBEATSx to identify seasonal patterns and special events inside and outside the window lookback periods. Examples of these are the trends and harmonic functions from Eq. (5) and Eq. (6). Second, we identify domain-specific temporal covariates unique to each problem. The EPF setting typically includes day-ahead forecasts of electricity load and production levels from renewable energy sources.

4. Empirical evaluation

4.1. Electricity price forecasting datasets

To evaluate our method's forecasting capabilities, we consider short-term electricity price forecasting tasks, where the objective is to predict day-ahead prices. Five major power markets[1] are used in the empirical evaluation, all comprising hourly observations of the prices and two influential temporal exogenous variables that

[1] For the sake of reproducibility we only consider datasets that are openly accessible in the EPFtoolbox library https://github.com/jeslago/epftoolbox (Lago et al., 2021a).

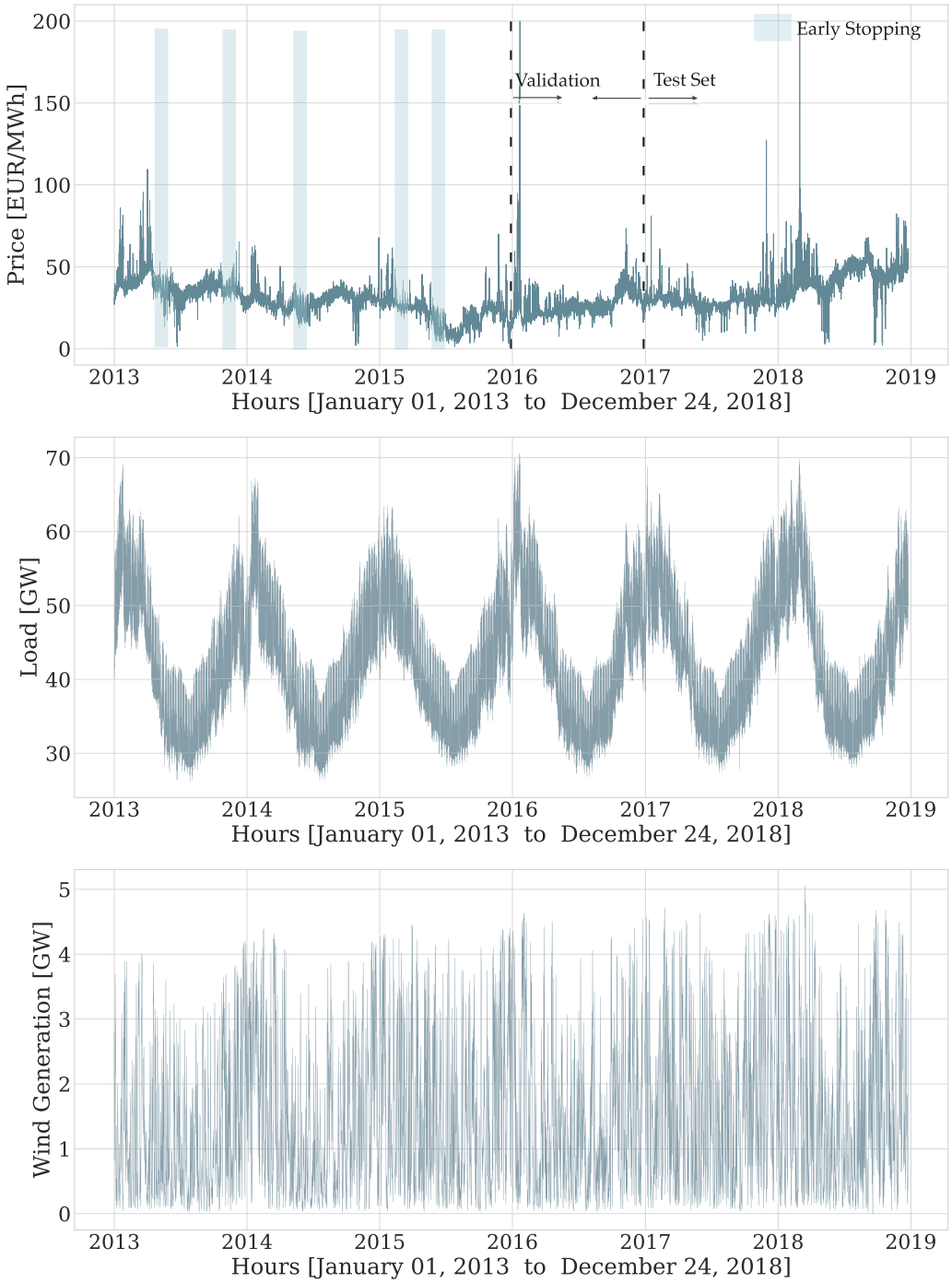


Fig. 2. The top panel shows the day-ahead electricity price time series for the Nord Pool (NP) market. The second and third panels show the day-ahead forecast for the system load and wind generation. The training data are composed of the first four years of each dataset. The validation set is the year that follows the training data (between the first and second dotted lines). For the held-out test set, the last two years of each dataset are used (marked by the second dotted line). During the evaluation, we recalibrate the model, updating the training set to incorporate all available data before each daily prediction. The recalibration uses an early stopping set of 42 weeks randomly chosen from the updated training set (a sample selection is marked with blue rectangles in the top panel).

extend for 2184 days (312 weeks, six years). From the six years of available data for each market, we hold two years out to test the forecasting performance of the algorithms. The length and diversity of the test sets allow us to obtain accurate and highly comprehensive measurements of the robustness and the generalization capabilities of the models.

Table 1 summarizes the key characteristics of each market. The Nord Pool electricity market (NP), corresponding to the exchange among Nordic countries, contains the hourly prices and day-ahead forecasts of load and wind generation. The second dataset is the Pennsylvania–New Jersey—Maryland market in the United States (PJM), which contains hourly zonal prices in the

Commonwealth Edison (COMED) and two-day-ahead forecasts of load at the system and COMED zonal levels. The remaining three markets are obtained from the integrated European Power Exchange (EPEX). The Belgian (EPEX-BE) and French (EPEX-FR) markets share the day-ahead forecast generation in France as covariates, since it is known to be one of the best predictors for Belgian prices (Lago, De Ridder, Vrancx, & De Schutter, 2018). Finally, the German market (EPEX-DE) contains the hourly prices, day-ahead load forecasts, and the country-level wind and solar generation day-ahead forecast.

Fig. 2 displays the NP electricity price time series and its corresponding covariate variables to illustrate the datasets. The NP market is the least volatile among the considered markets, since most of its power comes from hydroelectric generation, renewable source volatility is negligible, and zero spikes are rare. The PJM market is transitioning from coal generation to natural gas and some renewable sources. Zero spikes are rare, but the system exhibits higher volatility than NP. In the EPEX-BE and EPEX-FR markets, negative prices and spikes are more frequent, and as time passes, these markets begin to show increasing signs of integration. Finally, the EPEX-DE market shows few price spikes, but the most frequent negative and zero price events, due in great part to the impact of renewable sources.

The exogenous covariates are normalized, following best practices drawn from the EPF literature (Uniejewski, Weron, & Ziel, 2018). Preprocessing the inputs of neural networks is essential to accelerate and stabilize the optimization (LeCun, Bottou, Orr, & Müller, 1998).

4.2. Interpretable time series signal decomposition

In this subsection, we demonstrate the versatility of the proposed method and show how a careful selection of the inductive bias, constituted by the assumptions used to learn the modeled signal, endows NBEATSx with an outstanding ability to model complex dynamics while enabling human understanding of its outputs, turning it into a unique and exciting tool for time series analysis. Our method combines the power of nonlinear transformations provided by neural networks with the flexibility to model multiple seasons that can be fractional, while simultaneously accounting for interaction events such as holidays and other covariates. As described above, the interpretable configuration of the NBEATSx architecture computes time-varying coefficients for slowly changing polynomial functions to model the trend, harmonic functions to model the cyclical behavior of the signal, and exogenous covariates. Here, we show how this configuration can decompose a time series into the classic set of level, trend, and seasonality components, while identifying the covariate effects.

In this time series signal decomposition example, we show how the NBEATSx-I model benefits over NBEATS-I by explicitly accounting for information carried by exogenous covariates. Fig. 3 shows the NP electricity market's hourly price (EUR/MWh) for December 18, 2017, which was a day with high prices due to high load. Other days showed a less pronounced difference between the results obtained with the original NBEATS-I and the NBEATSx-I. We selected a day with a higher-than-normal load for exposition purposes, to demonstrate qualitative differences in the forecasts. We can see a substantial difference in the forecast residual magnitudes in the bottom row of Fig. 3. The original model shows a strong negative bias. On the other hand, NBEATSx-I is able to capture the evidently substantial explanatory value of the exogenous features, resulting in a much more accurate forecast.

4.3. Comparative analysis

4.3.1. Evaluation metrics

To ensure the comparability of our results with the existing literature, we opted to follow the widely accepted practice of evaluating the accuracy of point forecasts with the following metrics: mean absolute error (MAE), relative mean absolute error (rMAE),[2] symmetric mean absolute percentage error (sMAPE), and root mean squared error (RMSE), defined as:

$$MAE = \frac{1}{24N_d}\sum_{d=1}^{N_d}\sum_{h=1}^{24}|y_{d,h} - \hat{y}_{d,h}|$$

$$rMAE = \frac{\sum_{d=1}^{N_d}\sum_{h=1}^{24}|y_{d,h} - \hat{y}_{d,h}|}{\sum_{d=1}^{N_d}\sum_{h=1}^{24}|y_{d,h} - \hat{y}^{naive}_{d,h}|}$$

$$sMAPE = \frac{200}{24N_d}\sum_{d=1}^{N_d}\sum_{h=1}^{24}\frac{|y_{d,h} - \hat{y}_{d,h}|}{|y_{d,h}| + |\hat{y}_{d,h}|}$$

$$RMSE = \sqrt{\frac{1}{24N_d}\sum_{d=1}^{N_d}\sum_{h=1}^{24}\left(y_{d,h} - \hat{y}_{d,h}\right)^2}$$

where $y_{d,h}$ and $\hat{y}_{d,h}$ are the actual value and the forecast of the time series at day d and hour h for our experiments given the two years of each test set $N_d = 728$.

While regression-based models are estimated by minimizing squared errors, to train neural networks we minimize absolute errors (see Section 4.3.3 below). Hence, both the MAE and RMSE are highly relevant in our context. Since they are not easily comparable across datasets – and given the popularity of such errors in forecasting practice (Makridakis et al., 2020)– we have additionally computed a percentage and a relative measure. The sMAPE is used as an alternative to MAPE, which in the presence of values close to zero may degenerate (Hyndman & Koehler, 2006). The rMAE is calculated instead of a scaled measure used in the M4 competition for reasons explained in Sec. 5.4.2. of Lago et al. (2021a).

4.3.2. Statistical tests

To assess which forecasting model provides better predictions, we rely on the Giacomini–White test (GW; Giacomini and White 2006) of the multi-step conditional

[2] The naïve forecast method in EPF corresponds to a similar day rule, where the forecast for a Monday, Saturday, and Sunday equals the value of the series observed on the same weekday of the previous week, while the forecast for Tuesday, Wednesday, Thursday, and Friday is the value observed on the previous day.

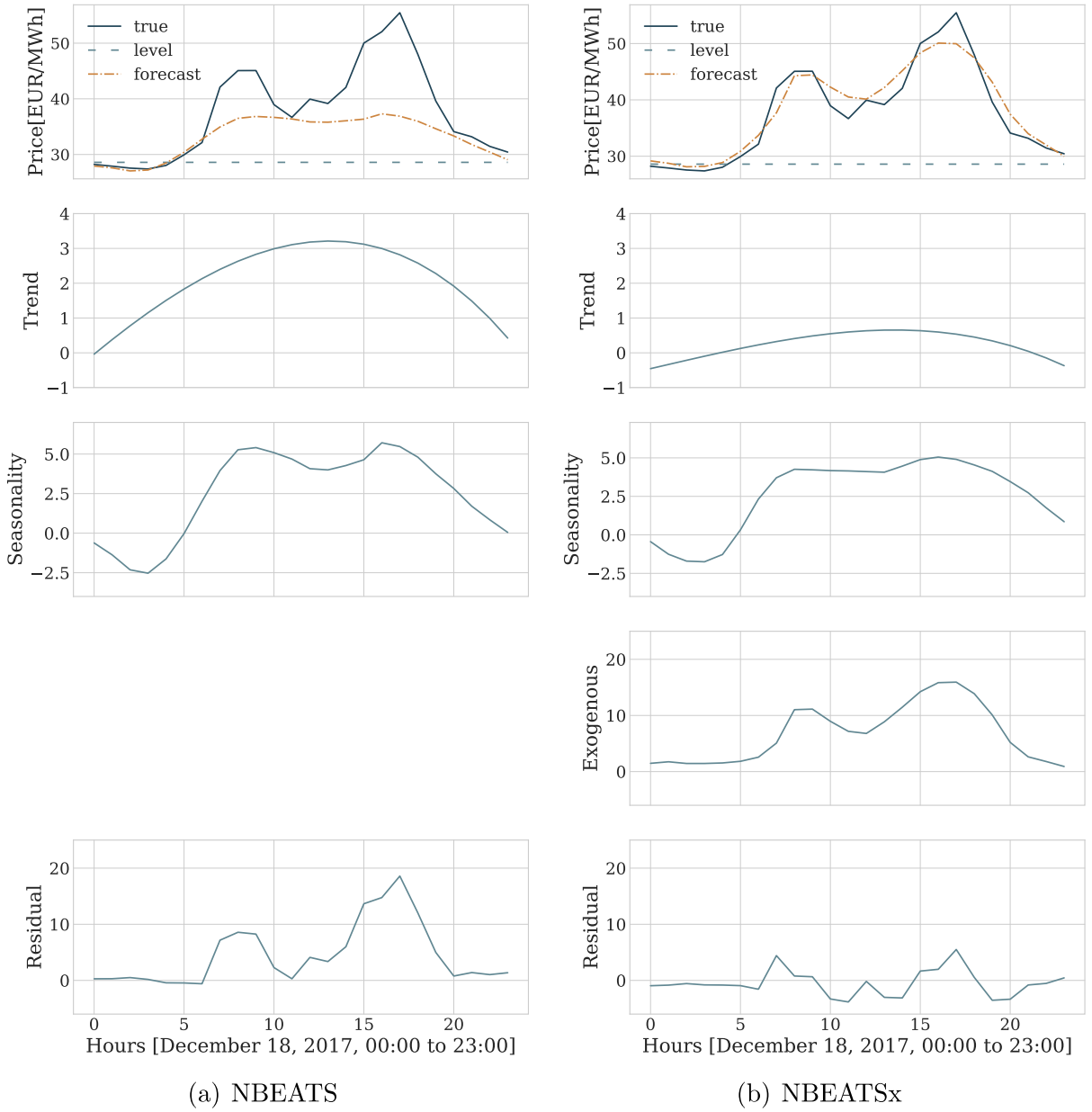


Fig. 3. Time series signal decomposition for NP electricity price day-ahead forecasts using interpretable variants of NBEATS and NBEATSx. The graphs in the top row show the original signal and the level; the latter is defined as the last available observation before the forecast. The second row shows the polynomial trend components, the third and fourth rows display the complex seasonality modeled by nonlinear Fourier projections and the exogenous effects of the electricity load on the price, respectively. The graphs in the bottom row show the unexplained variation of the signal. The use of electricity load and production forecasts turns out to be fundamental for accurate price forecasting.

predictive ability, which can be interpreted as a generalization of the Diebold–Mariano test (DM; Diebold and Mariano 1995), widely used in the forecasting literature. Compared with the DM or other unconditional tests, the GW test is valid under general assumptions, such as the heterogeneity rather than the stationarity of data. The GW test examines the null hypothesis of equal accuracy specified in Eq. (10), measured by the $L1$ norm of the daily errors of a pair of models A and B, conditioned on the information available at that moment[3] in time $\mathcal{F}_{d-1}$.

$$H_0 : \mathbb{E}\left[\|\mathbf{y}_d - \hat{\mathbf{y}}_d^A\|_1 - \|\mathbf{y}_d - \hat{\mathbf{y}}_d^B\|_1 \mid \mathcal{F}_{d-1}\right] \equiv \mathbb{E}\left[\Delta_d^{A,B} \mid \mathcal{F}_{d-1}\right] = 0 \tag{10}$$

4.3.3. Training methodology

The cornerstone of the training methodology for NBEATSx and the benchmark models included in this work is the definition and use of the training, validation, early stopping, and test datasets depicted in Fig. 2. The training set for each of the five markets comprises the

[3] In practice, the available information $\mathcal{F}_{d-1}$ is replaced with a constant and lags of the error difference $\Delta_d^{A,B}$, and the test is performed using a linear regression with a Wald-like test. When the conditional information considered is only the constant variable, one recovers the original DB test.

first three years of data, and the test set includes the last two years of data. The validation set is defined as the year between the training and test set coverages. The early stopping set, used for regularization, is either randomly sampled or corresponds to 42 weeks following the time span of the training set. These sets are used in the hyperparameter optimization phase and recalibration phase that we describe below.

During the hyperparameter optimization phase, model performance measured on the validation set is used to guide the exploration of the hyperparameter space defined in Table 2. During the recalibration phase, the optimally selected model, as defined by its hyperparameters, is re-trained for each day to include newly available information before the test inference. In this phase, an early stopping set provides a regularization signal for the retraining optimization.

To train the neural network, we minimize the mean absolute error (MAE) using stochastic gradient descent with adaptive moments (ADAM; Kingma and Ba 2014). Fig. A.2 in the Appendix compares the training and validation trajectories for NBEATS and NBEATSx, as diagnostics to assess the differences of the methods. The early stopping strategy halts the training procedure if a specified number of consecutive iterations occur without improvements in the loss measured on the early stopping set (Yao, Rosasco, & Andrea, 2007).

The NBEATSx model is implemented and trained in PyTorch (https://pytorch.org/) and can be run with both CPU and GPU resources. The code is available publicly in a dedicated repository to promote the reproducibility of the presented results and to support related research.

4.3.4. Hyperparameter optimization

We follow the practice of Lago, De Ridder, and De Schutter (2018) to select the hyperparameters that define the model, input features, and optimization settings. During this phase, the validation dataset is used to guide the search for well-performing configurations. To compare the benchmarks and NBEATSx, we rely on the same automated selection process: a Bayesian optimization technique that efficiently explores the hyperparameter space using tree-structured Parzen estimators (HYPEROPT; Bergstra, Bardenet, Bengio, and Kégl 2011). The architecture, optimization, and regularization hyperparameters are summarized in Table 2. To have comparable results, during the hyperparameter optimization stage we used the same number of configurations as in Lago, De Ridder, and De Schutter (2018). Note, that some of the methods do not require any hyperparameter optimization – e.g., the AR1 benchmark – and some might only have one hyper-parameter to be determined, such as the regularization parameter in the LEARx method, which is typically computed using the information criteria or cross-validation.

4.3.5. Ensembling

In many recent forecasting competitions, and particularly in the M4 competition, most of the top-performing models were ensembles (Atiya, 2020). It has been shown that in practice, combining a diverse group of models can be a powerful form of regularization to reduce the variance of predictions (Breiman, 1996; Hubicka, Marcjasz, & Weron, 2018; Nowotarski, Raviv, Trück, & Weron, 2014).

The techniques used by the forecasting community to induce diversity in the models are plentiful. The original NBEATS model obtained its diversity from three sources, training with different loss functions, varying the size of the input windows, and bagging models with different random initializations (Oreshkin et al., 2020). They used the median as the aggregation function for 180 different models. Interestingly, the original model did not rely on regularization, such as L2 or dropout, as (Oreshkin et al., 2020) found it to be good for the individual models but detrimental to the ensemble.

In our case, we ensemble the NBEATSx model using two sources of diversity. The first comes from a data augmentation technique controlled by the sampling frequency of the windows used during training, as defined in the data parameters from Table 2. The second source of diversity comes from whether we randomly select the early stopping set or instead use the last 42 weeks preceding the test set. Combining the data augmentation and early stopping options, we obtain four models that we ensemble using the arithmetic mean as the aggregation function. This technique is also used by the DNN benchmark (Lago, De Ridder, & De Schutter, 2018; Lago et al., 2021a).

4.3.6. Forecasting results

We conducted an empirical study involving two types of autoregressive models (AR1 and ARx1; Weron 2014), the lasso-estimated autoregressive model (LEARx; Uniejewski, Nowotarski, and Weron 2016), a parsimonious deep neural network (DNN; Lago, De Ridder, and De Schutter 2018, Lago et al. 2021a), the original neural basis expansion analysis without exogenous covariates (NBEATS; Oreshkin et al. 2020), and the exponential smoothing recurrent neural network (ESRNN; Smyl 2019). This experiment examined the effects of including the covariate inputs and comparing NBEATSx with state-of-the-art methods for the electricity price day-ahead forecasting task.

Table 3 summarizes the performance of the ensembled models, where the NBEATSx ensemble shows prevailing performance. It improves 18.77% on average for all metrics and markets when compared with the original NBEATS, and 20.6% when compared to ESRNN without time-dependent covariates. For the ensembled models, the NBEATSx RMSE improved on average 4.68%, MAE improved 2.53%, rMAE improved 1.97%, and sMAPE improved 1.25%. When comparing the NBEATSx ensemble against the DNN ensemble on individual markets, NBEATSx improved by 5.38% on the Nord Pool market, by 2.48% on the French market, and 2.81% on the German market. There was a non-significant difference in NBEATSx performance on the PJM and BE markets of 0.24% and 1.1%, respectively.

Fig. 4 provides a graphical representation of the statistical significance from the Giacomini–White test (GW) for the six ensembled models across the five markets for the MAE evaluation metric. A similar significance analysis

Table 2
Hyperparameters of NBEATSx networks. They are common to all presented datasets. We list the typical values we considered in our experiments. The configuration that performed best on the validation set was selected automatically.

Hyperparameter	Considered values
Architecture parameters	
Input size, size of autorregresive feature window.	$L \in 168$
Output size is the forecast horizon for day-ahead forecasting.	$H \in \{24\}$
List for architecture's type/number of stacks.	{[identity, TCN], [TCN, Identity] [Identity, WaveNet], [Wavenet, Identity], }
Type of activation used across the network.	{SoftPlus,SeLU,PreLU,Sigmoid,ReLU, TanH, LReLU}
Blocks separated by residual links per stack (shared across stacks).	{[1,1,1], [1, 1]}.
FCNN layers within each block.	{2}
FCNN hidden neurons on each layer of a block.	$N_h \in \{50, \ldots, 500\}$
Exogenous Temp. convolution filter size (Equation 9)	$\{2, \ldots, 10\}$
Only interpretable, degree of trend polynomials.	$N_{pol} \in \{2, 3, 4\}$
Only interpretable, number of Fourier basis (seasonality smoothness).	$N_{hr} \in 1, 2$
Whether NBEATSx coefficients take input $\mathbf{X}$ (Equation (1)).	{True, False}
Optimization and regularization parameters	
Initialization strategy for network weights.	{orthogonal, he_norm, glorot_norm}
Initial learning rate for regression problem.	Range(5e−4,1e−2)
The number of samples for each gradient step.	{256, 512}
The decay constant allows a large initial lr to escape local minima.	{0.5}
Number of times the learning rate is halved during train.	{3}
Maximum number of gradient descent iterations.	{30000}
Iterations without validation loss improvement before stop.	{10}
Frequency of validation loss measurements.	{100}
Whether batch normalization is applied after each activation.	{True, False}
The probability for dropout of neurons for all in the projection layers.	Range(0,1)
The probability for dropout of neurons for the exogenous encoder.	Range(0,1)
Constant to control the Lasso penalty used on the coefficients.	Range(0, 0.1)
Constant that controls the influence of L2 regularization of weights.	Range(1e−5,1e−0)
The objective loss function with which NBEATSx trained.	{MAE}
Random weeks from full dataset used to validate.	{42}
Number of iterations of hyperparameter search.	{1500}
Random seed that controls initialization of weights.	DiscreteRange(1,1000)
Data parameters	
Rolling window sample frequency, for data augmentation.	{1, 24}
Number of time windows included in the full dataset.	4 years
Number of validation weeks used for early stopping strategy.	{40, 52}
Normalization strategy of model inputs.	{none, median, invariant, std }

was conducted for the single models. The models included in the significance tests are the same as in Table 3: LEAR, DNN, ESRNN, NBEATS, and our proposed methods, NBEATSx-G and NBEATSx-I. The p-value of each comparison shows whether the performance improvement of the model's predictions corresponding to the column index of a cell in the grids shown in Fig. 4 over the model's predictions corresponding to the row of this cell of the grid is statistically significant. The NBEATSx-G model outperformed the DNN model in NP and DE, while NBEATSx-I outperformed it in NP, FR, and DE. Moreover, no benchmark model significantly outperformed NBEATSx-I and NBEATSx-G in any market.

In Appendix, we observe similar results for the single best models chosen from the four possible configurations of the ensemble components described in Section 4.3.5.

Table A.2 summarizes the accuracy of the predictions measured with the MAE, and Fig. A.3 displays the significance of the GW test. Ensembling improves the accuracy of NBEATSx by 3% on average across all markets, when compared to the single best models.

Finally, regarding the computational time complexity NBEATSx maintains good perfor- mance. As shown in Table A.1 in the Appendix, the time necessary to compute day-ahead predictions is in the order of miliseconds and comparable to that of the LEAR and DNN benchmarks. Additionally, the average time needed to perform a recalibration only takes circa 50 percent more than the relatively parsimonious DNN.

5. Conclusions

We presented NBEATSx: a new method for univariate time series forecasting with exogenous variables. It extends the well-performing neural basis expansion analysis. The resulting neural-based method has several valuable properties that make it suitable for a wide range of forecasting tasks. The network is fast to optimize, as it is mainly composed of fully connected layers. It can produce interpretable results, and achieves state-of-the-art performance on forecasting tasks where consideration of exogenous variables is fundamental.

We demonstrated the utility of the proposed method using a set of benchmark datasets from the electricity price forecasting domain, but it can be straightforwardly applied to forecasting problems in other domains. A qualitative evaluation showed that the interpretable configuration of NBEATSx can provide valuable insights to the

Table 3
Forecast accuracy measures for day-ahead electricity price predictions of ensembled models. The ESRNN and NBEATS models do not include time-dependent covariates. The reported metrics are the mean absolute error (MAE), relative mean absolute error (rMAE), symmetric mean absolute percentage error (sMAPE), and root mean squared error (RMSE). The smallest errors in each row are highlighted in bold.

		AR1	ESRNN	NBEATS	ARx1	LEARx*	DNN	NBEATSx-G	NBEATSx-I
NP	MAE	2.26	2.09	2.08	2.01	1.74	1.68	**1.58**	1.62
	rMAE	0.71	0.66	0.66	0.63	0.55	0.53	**0.50**	0.51
	sMAPE	6.47	6.04	5.96	5.84	5.01	4.88	**4.63**	4.70
	RMSE	4.08	3.89	3.94	3.71	3.36	3.32	**3.16**	3.27
PJM	MAE	3.83	3.59	3.49	3.53	3.01	**2.86**	2.91	2.90
	rMAE	0.79	0.74	0.72	0.73	0.62	**0.59**	0.60	0.60
	sMAPE	14.5	14.12	13.57	13.64	11.98	**11.33**	11.54	11.61
	RMSE	6.24	5.83	5.64	5.74	5.13	5.04	5.02	**4.84**
EPEX-BE	MAE	7.2	6.96	6.84	7.19	6.14	**5.87**	5.95	6.11
	rMAE	0.88	0.85	0.83	0.88	0.75	**0.72**	0.73	0.75
	sMAPE	16.26	15.84	15.80	16.11	14.55	**13.45**	13.86	14.02
	RMSE	18.62	16.84	17.13	18.07	15.97	15.97	**15.76**	15.80
EPEX-FR	MAE	4.65	4.65	4.74	4.56	3.98	3.87	3.81	**3.79**
	rMAE	0.78	0.78	0.80	0.76	0.67	0.65	**0.64**	**0.64**
	sMAPE	13.03	13.22	13.30	12.7	11.57	10.81	**10.59**	10.69
	RMSE	13.89	11.83	12.01	12.94	**10.68**	11.87	11.50	11.25
EPEX-DE	MAE	5.74	5.60	5.31	4.36	3.61	3.41	3.31	**3.29**
	rMAE	0.71	0.70	0.66	0.54	0.45	0.42	**0.41**	**0.41**
	sMAPE	21.37	20.97	19.61	17.73	14.74	14.08	**13.99**	**13.99**
	RMSE	9.63	9.09	8.99	7.38	6.51	5.93	5.72	**5.65**

*The LEARx results for EPEX-DE differ from (Lago et al., 2021a)—the values presented there are revised (Lago, Marcjasz, De Schutter, & Weron, 2021b).

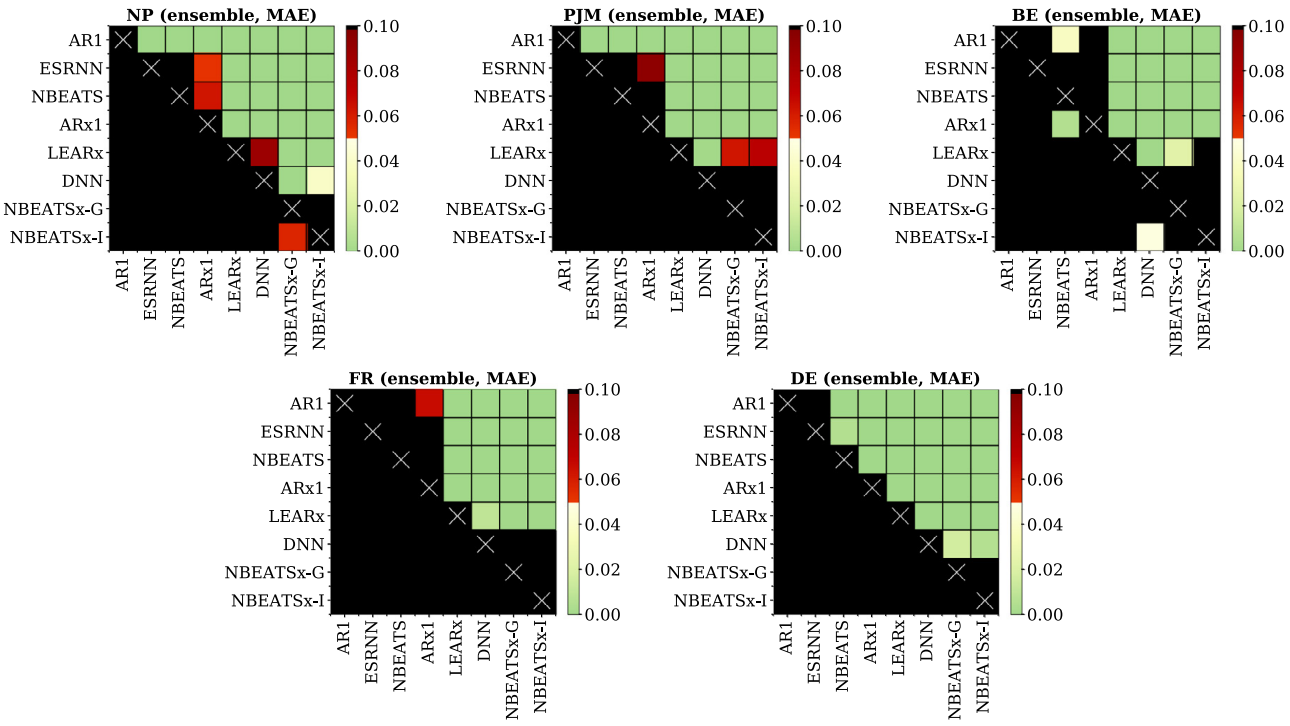


Fig. 4. Results of the Giacomini–White test for the day-ahead predictions with the mean absolute error (MAE) applied to pairs of the ensembled models on the five electricity markets datasets. Each grid represents one market. Each colored cell in a grid is plotted black, unless the predictions of the model corresponding to its column of the grid outperform the predictions of the model corresponding to its row of the grid. The color scale reflects the significance of the difference in MAE, with solid green representing the lowest p-values.

analyst, as it explains the variation of the time series by separating it into trend, seasonality, and exogenous components, in a fashion analogous to classic time series decomposition. Regarding the quantitative forecasting performance, we observed no significant differences between ESRNN and NBEATS without exogenous variables. At the same time, NBEATSx improved over NBEATS by nearly 20%, and by up to 5% over LEAR and DNN models specialized for electricity price forecasting tasks. Finally, we found no significant trade-offs between the

accuracy and interpretability of NBEATSx-G and NBEATSx-I predictions.

The neural basis expansion analysis is a very flexible method capable of producing accurate and interpretable forecasts, yet there is still room for improvement. For instance, augmentation of the harmonic functions towards wavelets or replacement of the convolutional encoder that would generate the covariate basis with smoothing alternatives such as splines. Additionally, one can extend the current non-interpretable method by regularizing its outputs with smoothness constraints.

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 was partially supported by the Defense Advanced Research Projects Agency (award FA8750-17-2-0130), the National Science Foundation (grant 2038612), the Space Technology Research Institutes grant from NASA's Space Technology Research Grants Program, the U.S. Department of Homeland Security (award 18DN-ARI-00031), the Ministry of Education and Science (MEiN, Poland; grant 0219/DIA/2019/48 to Grzegorz Marcjasz), the National Science Center (NCN, Poland; grant 2018/30/A/HS4/00444 to Rafał Weron). We would like to thank Stefania La Vattiata for the upbeat exposition of NBEATSx.

Appendix

A.1. Forecast and backcast bases

As discussed in Section 3.4, the interpretable configuration of the NBEATSx method performs basis projections into polynomial functions for the trends, harmonic functions for the seasonalities and exogenous variables. As shown in Fig. A.1, both the forecast and the backcast components of the model rely on similar basis functions, and the only difference depends upon the span of their time indexes. For this work in the EPF application of NBEATS, the backcast horizon corresponds to 168 hours while the forecast horizon corresponds to 24.

A.2. Training and validation curves

To study the effects of exogenous variables on the NBEATS model, we performed model training procedure diagnostics. Fig. A.2 shows the training and validation *mean absolute error (MAE)* for the NBEATS and NBEATSx models as training progresses. The curves correspond to the hyperparameter optimization phase described in Section 4.3.4. The models trained with and without exogenous variables display a considerable difference in their training and validation errors, as observed by the two separate clusters of trajectories. The exogenous variables—in this case, the electricity load and production forecasts—significantly improve the neural basis expansion analysis.

A.3. Computational time

We measured the computational time of the top four best algorithms with two metrics: the recalibration of the ensemble models selected from the hyperparameter optimization, and the computation of the predictions. For these experiments, we used a GeForce RTX 2080 GPU for the neural network models and an Intel(R) Xeon(R) Silver 4210 CPU @ 2.20 GHz for LEAR.

The training time of the *recalibration phase* of NBEATSx remains efficient, as it still trains in 75 and 81 s, increasing by 30 s on the relatively simple DNN. The computational time of the prediction remains within milliseconds. Finally the *hyperparameter optimization* scales linearly with respect to the time of the *recalibration phase* and the evaluation steps of the optimization. In the case of NBEATSx-G, the approximate time of a hyperparameter search of 1000 steps is two days.[4]

A.4. Best single models

Table A.2 shows that the best NBEATSx models yield improvements of 14.8% on average across all the evaluation metrics when compared to its NBEATS counterpart without exogenous covariates, and improvements of 23.9% when compared to ESRNN without time-dependent covariates. A perhaps more remarkable result is the statistically significant improvement of forecast accuracy over the LEAR and DNN benchmarks, ranging from 0.75% to 7.2% across all metrics and markets, with the exception of BE. Compared to the DNN, the RMSE improved on average 4.9%, the MAE improved 3.2%, the rMAE improved 3.0%, and the sMAPE improved 1.7%. When comparing the best NBEATSx models against the best DNN on individual markets, NBEATSx improved by 3.18% on the Nord Pool market (NP), 2.03%–2.65% on the French (FR) market, and 5.24% on the German (DE) power markets. The positive difference in performance for the Belgian (BE) market of 0.53% was not statistically significant.

Fig. A.3 provides a graphical representation of the GW test for the six best models across the five markets for the MAE evaluation metric. The models included in the significance tests are the same as in Tables A.2: LEAR, DNN, ESRNN, NBEATS, and our proposed methods, NBEATSx-G and NBEATSx-I. The p-value of each individual comparison shows whether the improvement in performance (measured by the MAE or RMSE) of the x-axis model over the y-axis model is statistically significant. Both the NBEATSx-G and NBEATSx-I models outperformed the LEAR and DNN models in all markets, with the exception of Belgium. Moreover, no benchmark model outperformed NBEATSx-I or NBEATSx-G on any market.

[4] For comparability, we used 1000 steps (Lago et al., 2021a), but restricting this to 300 steps yielded similar results.

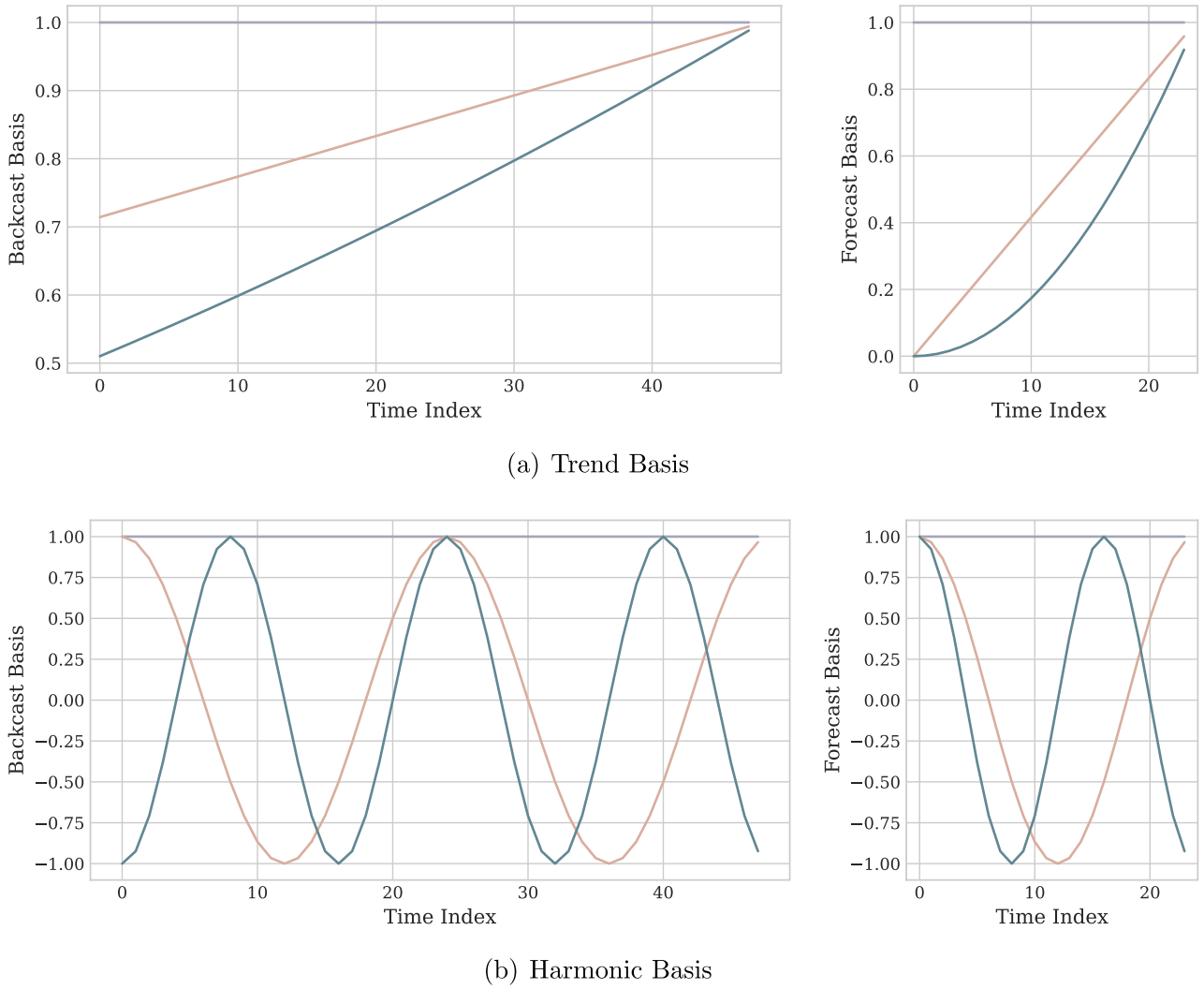


Fig. A.1. Examples of polynomial and harmonic bases included in the interpretable configuration of the neural basis expansion analysis. The slowly varying bases allow NBEATS to model trends and seasonalities.

Table A.1
Computational time performance in seconds for the top four most accurate models for the day-ahead electricity price forecasting task in the NP market, averaged for the four elements of the ensembles. (The time performance for the rest of the markets was almost identical.).

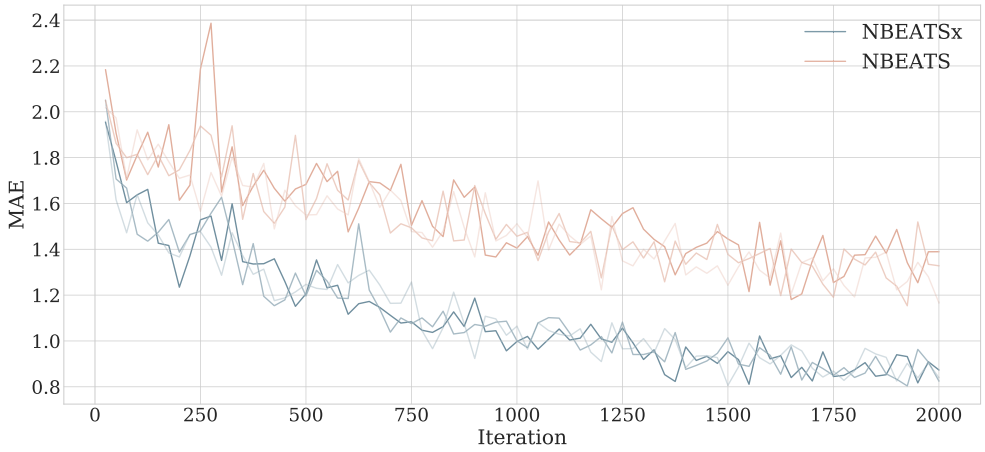
	LEARx	DNN	NBEATSx-G	NBEATSx-I
Recalibration	18.57	50.65	75.02	81.61
Prediction	0.0032	0.0041	0.0048	0.0054

A.5. Comments on hyperparameter optimization

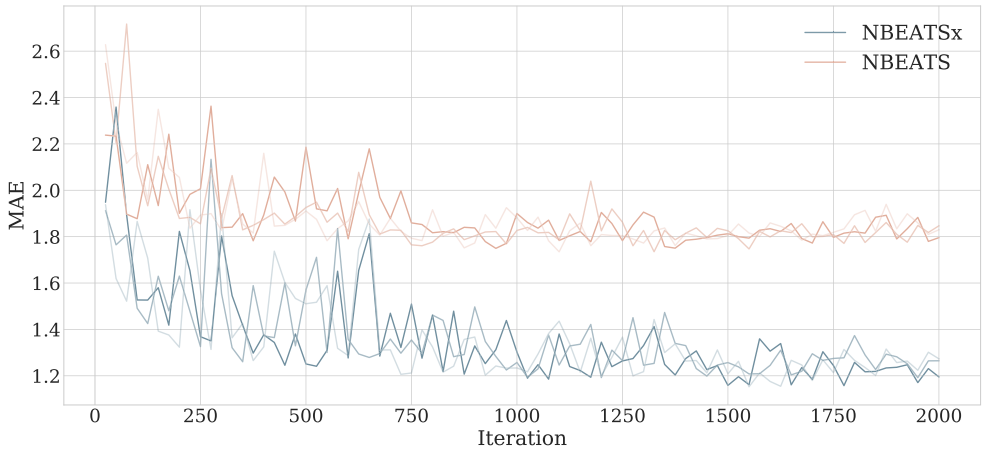
In this Section, we summarize observations and key empirical findings from the extensive hyperparameter optimization on the space defined by Table 2 for the four models composing each dataset ensemble. These observations and regularities of the optimally selected hyperparameters are important to create a more efficient and informed hyperparameter space and possibly guide future experiments with the `NBEATSx` architecture.

Interpretable configuration observations:

1. Among quadratic, cubic and fourth degree polynomials, $N_{pol} \in \{2, 3, 4\}$, the most common basis selected for the day-ahead EPF task was quadratic, $N_{pol} = 2$. As shown in Fig. 3, the combination of quadratic trend and harmonics already describes the electricity price average daily profiles successfully. Linear trends were omitted from exploration as they showed to be fairly restrictive. In experiments on longer forecast horizons ($H > 24$), beyond the scope of this paper, we observed that more trend flexibility tended to be beneficial.
2. We did not observe preferences in the harmonic basis spectrum controlled by $N_{hr} \in \{1, 2\}$, the hyperparameter that controls the number of oscillations of the basis in the forecast horizon. We



(a) Train set



(b) Validation set

Fig. A.2. Training and validation *Mean Absolute Error (MAE)* curves on the NP market. We show the curves for NBEATSx-G with exogenous variables and NBEATS without exogenous variables as a function of the optimization iterations. We define the four curves by a different random seed used for initialization.

believe this is due to the flexibility of the harmonic basis $S \in \mathbb{R}^{H\times(H-1)}$ that already covers a broad spectrum of frequencies. Our intuition dictates that $N_{hr} = 1$ is a good setting unless there is an apparent mismatch between the time- series frequency and the number of recorded observations that one could have in a Nyquist-frequency under-sampling or over-sampling phenomenon (Koopmans, 1995). This, however, is beyond the scope of this paper.

Hyperparameter optimization regularities:

1. Regarding the optimal activation functions, we found that the most selected ones were `SeLU`, `PreLU`, and `Sigmoid`, while activations like `ReLU`, `TanH`, and `LReLU` were consistently outperformed. `Sigmoid` activations tend to make the optimization of the network difficult when the networks grow in depth.
2. Surprisingly, the stochastic gradient batch size consistently preferred 256 and 512 over 128 windows. Our selection of the `ADAM` optimizer over classic `SGD` could explain these observations. The machine learning community believes that more extended `SGD` optimization with mini batches tends to have better generalization properties (Keskar, Mudigere, Nocedal, Smelyanskiy, & Tang, 2017). Additional research on the area would be interesting.
3. The batch normalization technique was often detrimental in combination with the doubly-residual stack strategy of the `NBEATSx` method. The residual signals tend to be close to zero, making the normalization numerically unstable.
4. The robust median normalization of the exogenous variables was consistently preferred over alternatives like standard deviation normalization.
5. Regarding the hidden units of the `FCNN` layers, the optimal parameters did not favor an information bottleneck behavior (Tishby, Pereira, & Bialek, 1999). Almost half of the optimal models had a small number of hidden units followed by a larger number of hidden units.

Table A.2
Forecast accuracy measures for day-ahead electricity prices for the best single model out of the four models described in the Section 4.3.5. ESRNN and NBEATS are the original implementations and do not include time-dependent covariates. The reported metrics are the mean absolute error (MAE), relative mean absolute error (rMAE), symmetric mean absolute percentage error (sMAPE), and root mean squared error (RMSE). The smallest errors in each row are highlighted in bold.

		AR1	ESRNN	NBEATS	ARx1	LEARx*	DNN	NBEATSx-G	NBEATSx-I
NP	MAE	2.28	2.11	2.11	2.11	1.95	1.71	**1.65**	1.68
	rMAE	0.72	0.67	0.67	0.67	0.62	0.54	**0.52**	0.53
	sMAPE	6.51	6.09	6.06	6.1	5.62	4.97	**4.83**	4.89
	RMSE	4.08	3.92	3.98	3.84	3.60	3.36	**3.27**	3.33
PJM	MAE	3.88	3.63	3.48	3.68	3.09	3.07	3.02	**3.01**
	rMAE	0.8	0.75	0.72	0.76	0.64	0.63	**0.62**	**0.62**
	sMAPE	14.66	14.26	13.56	14.09	12.54	12.00	11.97	**11.91**
	RMSE	6.26	5.87	5.59	5.94	5.14	5.20	5.06	**5.00**
EPEX-BE	MAE	7.04	7.01	6.83	7.05	6.59	**6.07**	6.14	6.17
	rMAE	0.86	0.86	0.83	0.86	0.80	**0.74**	0.75	0.75
	sMAPE	16.29	15.95	16.03	16.21	15.95	**14.11**	14.68	14.52
	RMSE	17.25	16.76	16.99	17.07	16.29	15.95	15.46	**15.43**
EPEX-FR	MAE	4.74	4.68	4.79	4.85	4.25	4.06	3.98	**3.97**
	rMAE	0.80	0.78	0.80	0.86	0.71	0.68	**0.67**	**0.67**
	sMAPE	13.49	13.25	13.62	16.21	13.25	11.49	**11.07**	11.29
	RMSE	13.68	11.89	12.09	17.07	**10.75**	11.77	11.61	11.08
EPEX-DE	MAE	5.73	5.64	5.37	4.58	3.93	3.59	3.46	**3.37**
	rMAE	0.71	0.70	0.67	0.57	0.49	0.45	0.43	**0.42**
	sMAPE	21.22	21.09	19.71	18.52	16.80	14.68	14.78	**14.34**
	RMSE	9.39	9.17	9.03	7.69	6.53	6.08	5.84	**5.64**

*The LEARx results for EPEX-DE differ from (Lago et al., 2021a)—the values presented there are revised (Lago et al., 2021b).

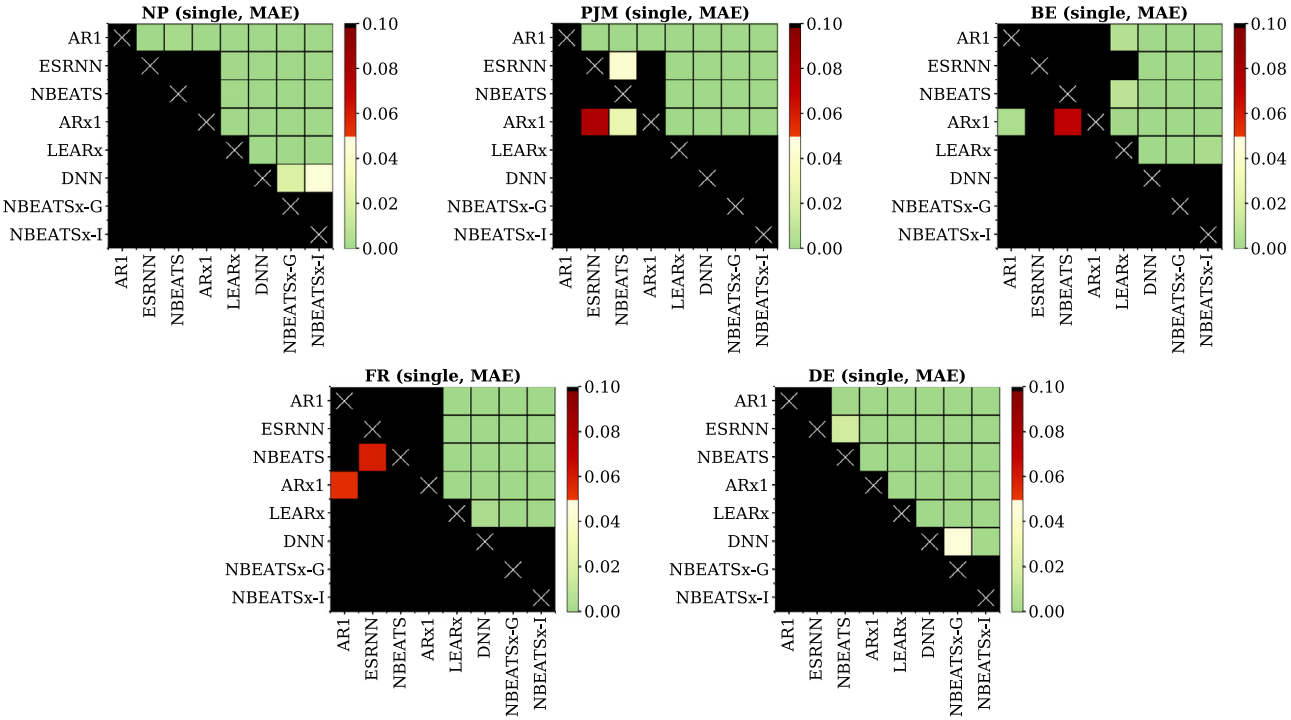


Fig. A.3. Results of the Giacomini–White test for the day-ahead predictions with the mean absolute error (MAE) applied to pairs of the single models on the five electricity markets datasets. Each grid represents one market. Each colored cell in a grid is plotted black, unless the predictions of the model corresponding to its column of the grid outperform the predictions of the model corresponding to its row of the grid. The color scale reflects the significance of the difference in MAE, with solid green representing the lowest p-values.

References

Atiya, A. F. (2020). Why does forecast combination work so well? *International Journal of Forecasting*, *36*(1), 197–200. http://dx.doi.org/10.1016/j.ijforecast.2019.03.010, M4 Competition. URL: https://www.sciencedirect.com/science/article/pii/S0169207019300779.

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *Computing Research Repository*, arXiv:1803.01271.

Benidis, K., Rangapuram, S. S., Flunkert, V., Wang, B., Maddix, D., Turkmen, C., et al. (2020). Neural forecasting: Introduction and literature overview. *Computing Research Repository*, arXiv:2004.10240.

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems, Vol. 24* (pp. 2546–2554). Curran Associates, Inc., URL: https://proceedings.neurips.cc/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. http://dx.doi.org/10.1023/A:1018054314350.

Chang, S., Zhang, Y., Han, W., Yu, M., Guo, X., Tan, W., et al. (2017). Dilated recurrent neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), *Advances in neural information processing systems, Vol. 30*. Curran Associates, Inc., URL: https://proceedings.neurips.cc/paper/2017/file/32bb90e8976aab5298d5da10fe66f21d-Paper.pdf.

Chitsaz, H., Zamani-Dehkordi, P., Zareipour, H., & Parikh, P. (2018). Electricity price forecasting for operational scheduling of behind-the-meter storage systems. *IEEE Transactions on Smart Grid*, *9*(6), 6612–6622. http://dx.doi.org/10.1109/TSG.2017.2717282.

Chung, J., Gülçehre, Ç., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014, workshop on deep learning*. arXiv:1412.3555.

Diebold, F., & Mariano, R. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *13*, 253–265. http://dx.doi.org/10.1080/07350015.1995.10524599, URL: https://www.sas.upenn.edu/~fdiebold/papers/paper68/pa.dm.pdf.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211, URL: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1.

Gers, F. A., Cummins, F., & Schmidhuber, J. (2000). Learning to forget: continual prediction with LSTM. *Neural Computation*, *12*, 2451–2471, URL: https://digital-library.theiet.org/content/conferences/10.1049/cp_19991218.

Giacomini, R., & White, H. (2006). Tests of conditional predictive ability. *Econometrica*, *74*(6), 1545–1578. http://dx.doi.org/10.1111/j.1468-0262.2006.00718.x, URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-0262.2006.00718.x.

Gianfreda, A., Ravazzolo, F., & Rossini, L. (2020). Comparing the forecasting performances of linear models for electricity prices with high RES penetration. *International Journal of Forecasting*, *36*(3), 974–986. http://dx.doi.org/10.1016/j.ijforecast.2019.11.002, URL: https://www.sciencedirect.com/science/article/pii/S0169207019302596.

Graves, A. (2013). Generating sequences with recurrent neural networks. *Computing Research Repository*, arXiv:1308.0850.

Hubicka, K., Marcjasz, G., & Weron, R. (2018). *A note on averaging day-ahead electricity price forecasts across calibration windows*: *HSC research reports HSC/18/03*, Hugo Steinhaus Center, Wroclaw University of Technology, URL: https://ideas.repec.org/p/wuu/wpaper/hsc1803.html.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688. http://dx.doi.org/10.1016/j.ijforecast.2006.03.001, URL: http://www.sciencedirect.com/science/article/pii/S0169207006000239.

Keskar, N., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. (2017). On large-batch training for deep learning: Generalization gap and sharp minima. URL: http://arxiv.org/abs/1609.04836 published as a conference paper at the 5th International Conference for Learning Representations (ICLR), Toulon, France, 2017.

Kingma, D. P., & Ba, J. (2014). ADAM: A method for stochastic optimization. Published as a conference paper at the 3rd International Conference for Learning Representations (ICLR), San Diego, 2015. URL: http://arxiv.org/abs/1412.6980.

Koopmans, L. (1995). *The spectral analysis of time series*. Elsevier.

Lago, J., De Ridder, F., & De Schutter, B. (2018). Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. *Applied Energy*, *221*, 386–405. http://dx.doi.org/10.1016/j.apenergy.2018.02.069, URL: http://www.sciencedirect.com/science/article/pii/S030626191830196X.

Lago, J., De Ridder, F., Vrancx, P., & De Schutter, B. (2018). Forecasting day-ahead electricity prices in Europe: The importance of considering market integration. *Applied Energy*, *211*, 890–903. http://dx.doi.org/10.1016/j.apenergy.2017.11.098, URL: https://www.sciencedirect.com/science/article/pii/S0306261917316999.

Lago, J., Marcjasz, G., De Schutter, B., & Weron, R. (2021a). Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, *293*, Article 116983. http://dx.doi.org/10.1016/j.apenergy.2021.116983, URL: https://www.sciencedirect.com/science/article/pii/S0306261921004529.

Lago, J., Marcjasz, G., De Schutter, B., & Weron, R. (2021b). *Erratum to 'Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark' [Appl. Energy 293 (2021) 116983]*: *WORking papers in Management Science (WORMS) WORMS/21/12*, Department of Operations Research and Business Intelligence, Wroclaw University of Science and Technology, URL: https://ideas.repec.org/p/ahh/wpaper/worms2112.html.

LeCun, Y., Bottou, L., Orr, G. B., & Müller, K. R. (1998). Efficient BackProp. In *Neural networks: Tricks of the trade* (pp. 9–50). Berlin, Heidelberg: Springer Berlin Heidelberg, http://dx.doi.org/10.1007/3-540-49430-8_2.

Li, W., & Becker, D. (2021). Day-ahead electricity price prediction applying hybrid models of LSTM-based deep learning methods and feature selection algorithms under consideration of market coupling. *Energy*, *237*, Article 121543.

Livera, A. M. D., Hyndman, R. J., & Snyder, R. D. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, *106*(496), 1513–1527. http://dx.doi.org/10.1198/jasa.2011.tm09771.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PLoS One*, *13*(3), Article e0194889, URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0194889.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, *36*(1), 54–74. http://dx.doi.org/10.1016/j.ijforecast.2019.04.014, M4 Competition. URL: https://www.sciencedirect.com/science/article/pii/S0169207019301128.

Marcjasz, G. (2020). Forecasting electricity prices using deep neural networks: A robust hyper-parameter selection scheme. *Energies*, *13*(18), Article 13184605.

Mayer, K., & Trück, S. (2018). Electricity markets around the world. *Journal of Commodity Markets*, *9*, 77–100. http://dx.doi.org/10.1016/j.jcomm.2018.02.001.

Narajewski, M., & Ziel, F. (2020). Econometric modelling and forecasting of intraday electricity prices. *Journal of Commodity Markets*, *19*, Article 100107. http://dx.doi.org/10.1016/j.jcomm.2019.100107.

Nazar, M. S., Fard, A. E., Heidari, A., Shafie-khah, M., & ao P.S. Catalão, J. (2018). Hybrid model using three-stage algorithm for simultaneous load and price forecasting. *Electric Power Systems Research*, *165*, 214–228. http://dx.doi.org/10.1016/j.epsr.2018.09.004.

Nowotarski, J., Raviv, E., Trück, S., & Weron, R. (2014). An empirical comparison of alternative schemes for combining electricity spot price forecasts. *Energy Economics*, *46*(C), 395–412. http://dx.doi.org/10.1016/j.eneco.2014.07.0, URL: https://ideas.repec.org/a/eee/eneeco/v46y2014icp395-412.html.

Nowotarski, J., & Weron, R. (2018). Recent advances in electricity price forecasting: A review of probabilistic forecasting. *Renewable and Sustainable Energy Reviews*, *81*, 1548–1568. http://dx.doi.org/10.1016/j.rser.2017.05.234.

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-BEATS: neural basis expansion analysis for interpretable time series forecasting. In *8th international conference on learning representations, ICLR 2020*. URL: https://openreview.net/forum?id=r1ecqn4YwB.

Rosenblatt, F. (1961). *Principles of neurodynamics. Perceptrons and the theory of brain mechanisms*: *Technical report*, Cornell Aeronautical Lab Inc Buffalo NY.

Smyl, S. (2019). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, http://dx.doi.org/10.1016/j.ijforecast.2019.03.017.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems, Vol. 27*. Curran Associates, Inc..

Tishby, N., Pereira, F., & Bialek, W. (1999). The information bottleneck method. In *The 37th annual Allerton Conf. on Communication, Control, and Computing* (pp. 368–377). URL: https://arxiv.org/abs/physics/0004057.

Uniejewski, B., Nowotarski, J., & Weron, R. (2016). Automated variable selection and shrinkage for day-ahead electricity price forecasting. *Energies*, *9*(8), URL: https://www.mdpi.com/1996-1073/9/8/621.

Uniejewski, B., & Weron, R. (2021). Regularized quantile regression averaging for probabilistic electricity price forecasting. *Energy Economics*, *95*, Article 105121. http://dx.doi.org/10.1016/j.eneco.2021.105121, URL: https://www.sciencedirect.com/science/article/pii/S0140988321000268.

Uniejewski, B., Weron, R., & Ziel, F. (2018). Variance stabilizing transformations for electricity spot price forecasting. *IEEE Transactions on Power Systems*, *33*(2), 2219–2229. http://dx.doi.org/10.1109/TPWRS.2017.2734563.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., et al. (2016). WaveNet: A generative model for raw audio. *CoRR*, arXiv:1609.03499.

Wang, L., Zhang, Z., & Chen, J. (2017). Short-term electricity price forecasting with stacked denoising autoencoders. *IEEE Transactions on Power Systems*, *32*(4), 2673–2681. http://dx.doi.org/10.1109/TPWRS.2016.2628873.

Wen, R., Torkkola, K., Narayanaswamy, B., & Madeka, D. (2017). A multi-horizon quantile recurrent forecaster. In *31st conference on neural information processing systems NIPS 2017, time series workshop*. URL: https://arxiv.org/abs/1711.11053.

Weron, R. (2014). Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, *30*(4), 1030–1081. http://dx.doi.org/10.1016/j.ijforecast.2014.08.008, URL: https://www.sciencedirect.com/science/article/pii/S0169207014001083.

Yao, Y., Rosasco, L., & Andrea, C. (2007). On early stopping in gradient descent learning. *Constructive Approximation*, *26*(2), 289–315.

Ziel, F., & Steinert, R. (2018). Probabilistic mid- and long-term electricity price forecasting. *Renewable and Sustainable Energy Reviews*, *94*, 251–266, URL: https://arxiv.org/abs/1703.10806.

Paper 4

Distributional neural networks for electricity price forecasting

Grzegorz Marcjasz, Michał Narajewski, Rafał Weron, Florian Ziel

Energy Economics 125 (2023) 106843

Contents lists available at ScienceDirect

Energy Economics

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

Distributional neural networks for electricity price forecasting

Grzegorz Marcjasz [a,*], Michał Narajewski [b], Rafał Weron [a], Florian Ziel [b]

[a] Department of Operations Research and Business Intelligence, Wrocław University of Science and Technology, 50-370 Wrocław, Poland
[b] House of Energy Markets and Finance, University of Duisburg–Essen, 45141 Essen, Germany

ARTICLE INFO

JEL classification:
C44
C45
C46
C22
C53
Q47

Keywords:
Distributional neural network
Probabilistic forecasting
Quantile regression
LASSO
Electricity prices
Johnson's SU distribution

ABSTRACT

We present a novel approach to probabilistic electricity price forecasting which utilizes distributional neural networks. The model structure is based on a deep neural network containing a so-called probability layer, i.e., the outputs of the network are parameters of the normal or Johnson's SU distribution. To validate our approach, we conduct a comprehensive forecasting study complemented by a realistic trading simulation with day-ahead electricity prices in the German market. The proposed distributional deep neural network outperforms state-of-the-art benchmarks by over 7% in terms of the continuous ranked probability score and by 8% in terms of the per-transaction profits. The obtained results not only emphasize the importance of higher moments when modeling volatile electricity prices, but also – given that probabilistic forecasting is the essence of risk management – provide important implications for managing portfolios in the power sector.

1. Introduction

Trading in competitive markets requires precise probabilistic forecasts. Therefore, the attention of researchers and practitioners is slowly but gradually shifting from point to probabilistic forecasting (Petropoulos et al., 2022). It is not different in electricity markets. The point *electricity price forecasting* (EPF) literature is very broad, and the topic is well-researched (Weron, 2014; Lago et al., 2021). However, proper risk optimization, which is mandatory in the highly volatile and uncertain electricity markets, can only be performed using probabilistic forecasts, which provide a much more detailed view of the phenomenon under study. This has not gone unnoticed by researchers, however, the literature on probabilistic energy forecasting is much scarcer than on point forecasting (Hong et al., 2020; Nowotarski and Weron, 2018; Ziel and Steinert, 2018).

For the last three decades, the primary object of interest in electricity price forecasting has been the day-ahead market (Jędrzejewski et al., 2022), which is the main electricity spot trading place. However, the intraday (Uniejewski et al., 2019; Narajewski and Ziel, 2020a,b; Oksuz and Ugurlu, 2019; Janke and Steinke, 2019; Maciejowska et al., 2021; Maciejowska, 2022) and balancing markets (Kraft et al., 2020; Browell and Gilbert, 2022; Janczura and Wójcik, 2022) have been studied as well.

The two most widely used model classes in point EPF are regressions, more recently estimated via the *least absolute shrinkage and selection operator* or LASSO (Ziel, 2016; Ziel and Weron, 2018; Uniejewski et al., 2019; Narajewski and Ziel, 2020a), and neural networks (Dudek, 2016; Oksuz and Ugurlu, 2019; Zhou et al., 2019; Luo and Weng, 2019; Zahid et al., 2019; Lago et al., 2018; Keles et al., 2016). The latter are often components of complex, hybrid structures (Jahangir et al., 2019; Zhang et al., 2020; Oreshkin et al., 2021; Olivares et al., 2023).

The probabilistic EPF literature utilizes mainly quantile regression on point forecasts (Marcjasz et al., 2020; Maciejowska, 2020; Nowotarski and Weron, 2015; Maciejowska and Nowotarski, 2016), bootstrapping of point forecasts' residuals (Wan et al., 2013; Ziel and Steinert, 2018; Narajewski and Ziel, 2022; Nowotarski and Weron, 2018), and recurrent neural networks (RNN; Mashlakov et al., 2021; Brusaferri et al., 2020).

1.1. Our contribution

In this paper, we propose a novel probabilistic EPF approach based on distributional neural networks. More specifically, we consider a 'vanilla' *deep neural network* (DNN), i.e., a multi-layer perceptron in which the information propagates only forward. We utilize the TensorFlow (Abadi et al., 2015) and Keras (Chollet et al., 2015) frameworks,

* Corresponding author.
E-mail address: grzegorz.marcjasz@pwr.edu.pl (G. Marcjasz).

https://doi.org/10.1016/j.eneco.2023.106843
Received 7 September 2022; Received in revised form 19 June 2023; Accepted 26 June 2023
Available online 17 July 2023

and let the output layer be parameters of the normal or Johnson's SU distribution (Johnson, 1949). Note, that the only difference between a distributional and a standard network that provides point forecasts is in the output layer (Salinas et al., 2020; Barnes and Barnes, 2021; Barnes et al., 2021). Thus, if we have already built a neural network model for point forecasting, it is very easy to convert it to a distributional one. Even though distributional neural networks are not a new concept (Nix and Weigend, 1994; Williams, 1996), they have not attracted much attention. To the best of our knowledge, the only existing distributional neural networks in the energy forecasting literature use mixtures of normal distributions obtained using complex structures comprising convolutional neural networks (CNN) and gated recurrent units (GRU) (Afrasiabi et al., 2020) or RNNs (Mashlakov et al., 2021; Brusaferri et al., 2020). The *distributional deep neural network* (DDNN) proposed in this paper is far less complex than the CNNs, GRUs and RNNs, easier to interpret and computationally less demanding.

We evaluate model performance using a rolling window forecasting and trading study with day-ahead electricity prices in Germany. The DDNN is benchmarked against naive bootstrapping and two well-performing point EPF approaches (Lago et al., 2021): a LASSO-estimated autoregression (LEAR) and a DNN, both combined with quantile regression averaging (QRA) of Nowotarski and Weron (2015) for converting point predictions into probabilistic ones. Although a considerable amount of the EPF literature concerns forecast combinations (Hubicka et al., 2018; Serafin et al., 2019; Karabiber and Xydis, 2019), due to the complexity of aggregating predictive distributions (Berrisch and Ziel, 2022), here we utilize only two simple averaging schemes with equal weights. They allow to stabilize the neural network predictions.

The major contributions of our study are as follows:

1. We are the first to utilize the DDNN architecture and one of the first to consider distributional neural networks in electricity price forecasting.
2. Our approach is fully automated and can be used for datasets with similar characteristics, e.g., from other power markets. The code is open-source and available on GitHub https://github.com/gmarcjasz/distributionalnn. If needed, point forecasts can be easily derived from the predictive distributions.
3. The proposed DDNN outperforms state-of-the-art benchmarks (including LEAR and DNN models combined with QRA) by over 7% in terms of the *continuous ranked probability score* (CRPS) and by 8% in terms of the per-transaction profits.
4. We are the first in EPF and one of the first in probabilistic forecasting to use Johnson's SU distribution. Our results provide evidence for the importance of considering higher moments in EPF.
5. Given that probabilistic forecasting is the essence of risk management, our study provides power market participants with a new, significantly more accurate tool – as measured by the Diebold and Mariano (1995) test – for assessing risks related to trading power portfolios.

The remainder of this manuscript is structured as follows. Section 2 introduces the reader to the concept of distributional neural networks. Section 3 provides an overview of the market and data used in the application study. The models, including the DDNN and the hyperparameter tuning, are described in Section 4. The application study together with the results are presented in Section 5. The paper is concluded with a discussion of the main findings in Section 6.

2. The distributional deep neural network (DDNN) model

We assume that the reader is familiar with and understands the concept of the (feed-forward) deep neural networks (DNN). In this section, we briefly recall the definition and the mathematics behind it to underline the difference between the DNN with point and probabilistic output layers.

2.1. Architecture

Let $\boldsymbol{X} \in \mathbb{R}^{D\times N}$ be the input matrix with N denoting the number of features and D the number of observations. Further, let $\boldsymbol{H}_i \in \mathbb{R}^{D\times h_i}$ be the output matrix of ith hidden layer, $\boldsymbol{W}_i \in \mathbb{R}^{h_{i-1}\times h_i}$ and $\boldsymbol{b}_i \in \mathbb{R}^{D\times h_1}$ be the corresponding hidden-layer weights and bias where $h_i \in \mathbb{N}$ is the number of neurons in ith hidden layer with $h_0 = N$ and thus $\boldsymbol{H}_0 = \boldsymbol{X}$. Additionally, denote $a_i(\cdot)$ the ith activation function. Then, for $i \in \{1, \dots, I\}$ we have

$$\boldsymbol{H}_i = a_i\left(\boldsymbol{H}_{i-1}\boldsymbol{W}_i + \boldsymbol{b}_i\right). \tag{1}$$

Now, we got to the point where the DNN with point and probabilistic output layers differ. That is to say, in the standard DNN we calculate the output $\boldsymbol{O} \in \mathbb{R}^{D\times S}$, where S is the number of modeled features. Formally,

$$\boldsymbol{O} = \boldsymbol{H}_I\boldsymbol{W}_{I+1} + \boldsymbol{b}_{I+1} \tag{2}$$

are the values returned by the network. Such DNN is optimized given the true observation matrix $\boldsymbol{Y} \in \mathbb{R}^{D\times S}$ with respect to point losses, e.g. the mean squared error (MSE) or mean absolute error (MAE). In the case of the DDNN, the parameter layer $\boldsymbol{\Theta} \in \mathbb{R}^{D\times S\cdot P}$ consists of P distribution parameters for each of the S modeled features. It is however computed in the same manner as in Eq. (2). The final output is made by creating a $D\times S$-dimensional matrix of the assumed distributions $\boldsymbol{F}(\boldsymbol{\Theta}; x)$. The network is then optimized given the true observation matrix $\boldsymbol{Y} \in \mathbb{R}^{D\times S}$ with respect to probabilistic losses, e.g. by maximizing the likelihood for a parametric distribution or by minimizing the continuous ranked probability score (CRPS).

Fig. 1 provides an example with $I = 2$ hidden layers and this setting we use in the remainder of the manuscript. The number of neurons in the hidden layers is arbitrary, but the same for both DNNs in order to underline the difference between the point and probabilistic networks. We see clearly that the input and hidden layers are identical for both DNNs and only the output part differs.

As a final remark of the subsection, we discuss the multivariate output which consists of multiple features and the possible probabilistic distributions. Namely, we allow in the definition for S output features, and in our setting they are all $S = 24$ hours of the electricity prices of the following day. This can be done as all the day-ahead electricity prices are published at once, and therefore they can share the input regressor set. In other applications this is rather not the case, however such a multivariate setting may still be preserved if one considers S similar time series to be forecasted that may benefit from common regressors.

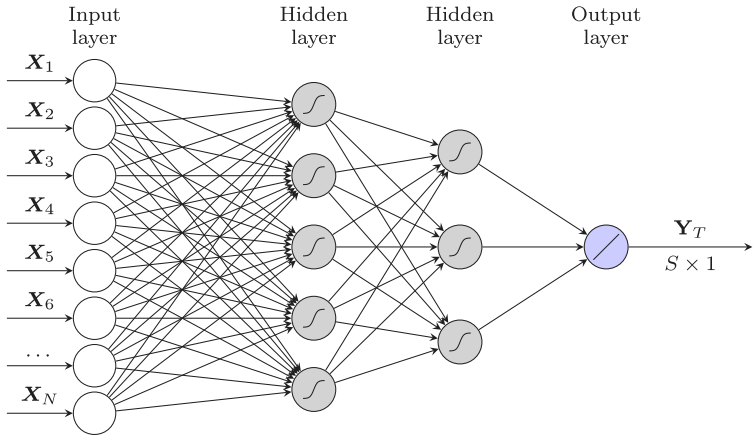
The probabilistic output layer may consist of nearly any implemented probabilistic distribution. Based on application, these can be, e.g., binomial or Poisson if we deal with a discrete problem, gamma or beta if we deal with a continuous problem, but supported only on the positive line, or normal, t or Johnson's SU if supported on the whole real line. As the electricity prices may be both positive and negative, we use in our study the two-parametric normal and four-parametric Johnson's SU distributions.

2.2. Regularization

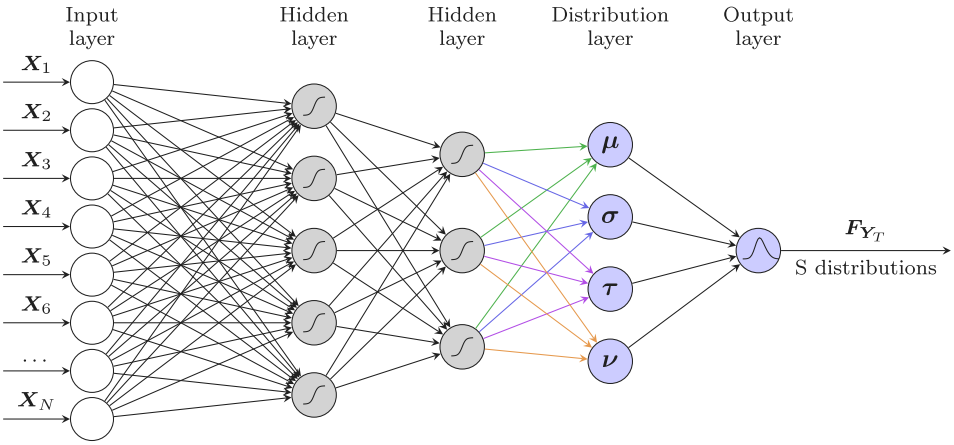
The danger of overfitting the model can be tackled in the DDNN similarly as in the standard one. One could use regularization, a dropout layer or early stopping. We use all of these in our forecasting study, however we approach the regularization of the parameter layer differently.

The DNN design allows for L_p regularization of every hidden layer $\boldsymbol{H}_i$, its weights $\boldsymbol{W}_i$, and bias $\boldsymbol{b}_i$. Applying it to the DNN we get the following loss with regularization

$$\begin{aligned}\mathcal{L}_{\text{reg}}(\boldsymbol{Y}, \boldsymbol{O}) = \mathcal{L}(\boldsymbol{Y}, \boldsymbol{O}) &+ \sum_{i=0}^{I} \lambda_{1,i}\|\boldsymbol{H}_i\|_p \\ &+ \sum_{i=0}^{I} \lambda_{2,i}\|\boldsymbol{W}_{i+1}\|_p + \sum_{i=0}^{I} \lambda_{3,i}\|\boldsymbol{b}_{i+1}\|_p\end{aligned} \tag{3}$$



(a) Deep neural network (DNN) with a multivariate output layer



(b) Distributional deep neural network (DDNN) with a multivariate output layer

Fig. 1. Comparison of the DNN and DDNN. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $\|\cdot\|_p$ represents the L_p norm. One can flexibly choose between the types of regularization, use both or none, and choose to regularize only some part of the network, e.g., only $\boldsymbol{H}_1$ layer and $\boldsymbol{W}_2$ weights. The $\lambda_{j,i}$ parameters are subject to hyperparameter tuning. The regularization of the DDNN may be done in the same way as described in Eq. (3), however, we could also regularize each of the distributional parameters separately as follows

$$\begin{aligned}\mathcal{L}_{\text{reg}}(\boldsymbol{Y}, \boldsymbol{F}(\boldsymbol{\Theta}; x)) &= \mathcal{L}(\boldsymbol{Y}, \boldsymbol{F}(\boldsymbol{\Theta}; x)) + \sum_{i=0}^{I-1} \lambda_{1,i} \|\boldsymbol{H}_i\|_k \\ &+ \sum_{i=0}^{I-1} \lambda_{2,i} \|\boldsymbol{W}_{i+1}\|_k + \sum_{i=0}^{I-1} \lambda_{3,i} \|\boldsymbol{b}_{i+1}\|_k \\ &+ \sum_{p=1}^{P} \left(\lambda_{1,I,p} \|\boldsymbol{H}_I\|_k + \lambda_{2,I,p} \|\boldsymbol{W}_{I+1}\|_k + \lambda_{3,I,p} \|\boldsymbol{b}_{I+1}\|_k\right).\end{aligned} \tag{4}$$

The difference between Eqs. (3) and (4) is the regularization of the last layer. Namely, in Eq. (3) we regularize the whole output layer using the same $\lambda_{j,I}$ values, whereas in Eq. (4) each parameter $p \in \{1, \ldots, P\}$ is regularized using its own $\lambda_{j,I,p}$ values. The color arrows in Fig. 1(b) denote separate kernel $\boldsymbol{W}_{I+1}$ regularization for each of the distributions' parameters. The reason to use such a differentiation is the possibility to use different amount of inputs' information for each distribution parameter, what was already observed in the literature (Narajewski and Ziel, 2020b).

3. The data

The goal in the empirical case study is forecasting day-ahead electricity prices in Germany. This section familiarizes the reader with the utilized data, especially the input features and the forecasting objective. The electricity markets in Europe consist of derivative, spot and balancing parts (Viehmann, 2017). The most important is the spot market, particularly the day-ahead auction. It takes place once a day at noon where all S products of the following day are traded in a uniform price auction (Weron and Ziel, 2019); typically S = 24. As all hours of the following day are traded at once, all of them are based on the same set of information. Therefore, in our study we model all the prices using exactly the same input features, what supports the multivariate output of the DDNN presented in Section 2.

The considered data is publicly available and has been downloaded from the ENTSOE (2022) Transparency platform. It spans six years of hourly observations from 01.01.2015 to 31.12.2020. The study uses a rolling window scheme, which mimics the daily business practice and is a standard procedure in the EPF literature (Weron, 2014; Weron and Ziel, 2019). The initial in-sample period spans from 01.01.2015 to 26.12.2018, i.e., $D = 4{\cdot}364 = 1456$ days (or $24{\cdot}1456 = 34{,}944$ h). For the purpose of hyperparameter tuning, we split it additionally to training and validation sets. The out-of-sample period starts on 27.12.2018, and ends on 31.12.2020, however the first 182 observations are used to obtain the QRA forecasts and thus are excluded from the analysis. Therefore, the final out-of-sample test set for probabilistic predictions uses 554 days of data. The models are retrained every day using the most recent D observations and the hyperparameters obtained in the tuning that is run on initial in-sample dataset.

Fig. 2. Time series plots of the considered data.

Table 1
Descriptive statistics of the data used for the initial calibration window and the out-of-sample test period.

Series	Mean	Std	Min	Q25	Median	Q75	Max
	In-sample data (01.01.2015–26.12.2018)						
Price (EUR/MWh)	34.8	16.4	−130.1	25.6	33.2	43.2	163.5
Load forecast (MWh)	55 117.4	9 543.1	28 823.6	47 233.0	55 046.0	63 403.6	75 912.2
RES forecast (MWh)	14 766.4	9 429.7	574.2	7 003.9	13 004.9	21 074.4	53 703.2
EUA (EUR/tCO_2)	8.7	4.8	3.9	5.3	7.2	8.6	25.2
API2 Coal (EUR/t)	63.7	14.6	37.9	51.1	67.1	76.6	88.6
TTF Gas (EUR/MWh)	18.3	3.8	10.6	15.3	18.2	20.7	29.2
Brent Oil (EUR/bbl.)	49.6	9.4	25.6	43.2	48.7	55.1	75.2
	Out-of-sample data (27.12.2018–31.12.2020)						
Price (EUR/MWh)	34.2	17.0	−90.0	26.3	35.2	44.0	200.0
Load forecast (MWh)	54 326.5	9 378.2	32 425.3	46 641.9	54 213.3	62 132.4	73 743.6
RES forecast (MWh)	19 363.2	11 310.6	793.9	10 218.1	17 763.3	27 161.2	62 490.1
EUA (EUR/tCO_2)	24.8	3.0	15.2	23.3	25.0	26.5	33.3
API2 Coal (EUR/t)	50.6	8.2	37.2	44.4	49.0	54.7	76.1
TTF Gas (EUR/MWh)	12.2	4.4	3.7	9.1	12.3	15.2	23.3
Brent Oil (EUR/bbl.)	47.6	11.8	17.8	37.0	53.1	57.2	66.8

Fig. 2 shows plots of the considered features together with the dates and study stages mentioned above. Table 1 reports the descriptive statistics of the data in the initial calibration window (upper part) and the out-of-sample test period (lower part). The data contains the day-ahead (DA) electricity prices, DA load forecasts, DA renewable energy sources (RES) forecasts, EU emission allowance prices and fuel: coal, oil and natural gas prices. The RES forecast is a sum of wind offshore, wind onshore and solar generation day-ahead forecasts. The DA prices and load forecasts exhibit strong daily, weekly and annual seasonality. Thus, we model each hour of the day separately within a single neural network and also utilize the weekday dummies. We do not construct any regressor explaining the annual behavior as it is well described by the load data. On the other hand, the RES forecasts exhibit only daily and annual seasonality and the EUA and fuel prices are random-walk type processes. Because these conclusions may not be readily apparent from Fig. 2, consult Ziel and Weron (2018), Sgarlato and Ziel (2022) and Billé et al. (2022) for more insights.

Fig. 3 presents histograms of prices for selected hours. Additionally, we fit there normal and Johnson's SU distributions and plot their densities. Both distributions belong to the location-scale family. The normal distribution $\mathcal{N}(\mu, \sigma^2)$ is a well-known two-parametric distribution with μ being the location and σ the scale parameters. The Johnson's SU distribution $\mathcal{J}(\mu, \sigma, \nu, \tau)$ was first investigated by Johnson (1949) as a transformation of the normal distribution. It is a four-parametric distribution with μ being the location, σ the scale, ν the skewness and τ the tail-weight parameters. So far, it has not found application with distributional neural networks. However, it is often used in the

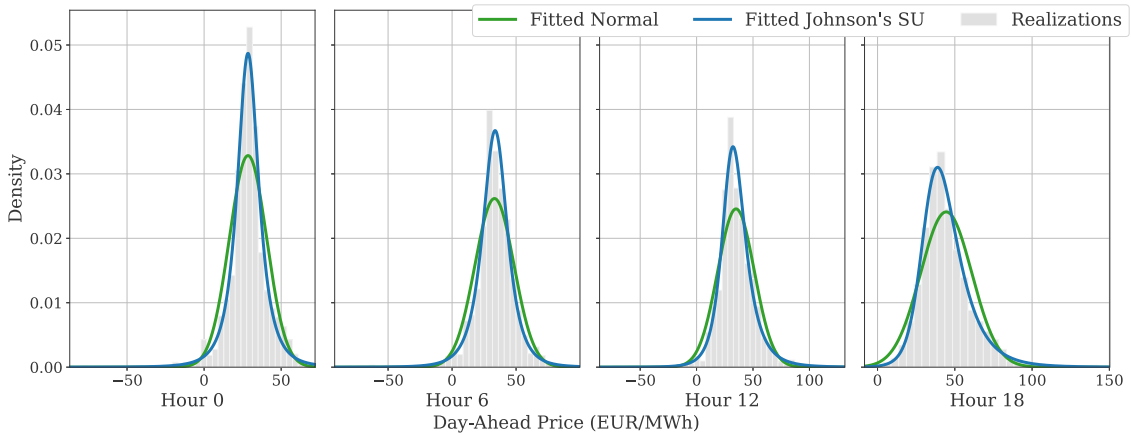


Fig. 3. Histograms of prices for selected hours with fitted densities of normal and Johnson's SU distributions. The plots are based on the in-sample (training and validation) data.

Table 2
CRPS scores and p-values of the Kolmogorov–Smirnov test for the normal and Johnson's SU (JSU) distributions fitted to electricity prices in the initial calibration window (01.01.2015–26.12.2018; see the upper part of Table 1) for four selected hours.

	Hour 0		Hour 6		Hour 12		Hour 18	
	Normal	JSU	Normal	JSU	Normal	JSU	Normal	JSU
CRPS	3.258	3.216	4.029	3.983	4.322	4.277	4.499	4.441
KS-test p-value	0.000	0.391	0.000	0.254	0.000	0.152	0.000	0.624

context of energy commodities (Patra, 2021; Gianfreda and Bunn, 2018; Abramova and Bunn, 2020).

Based on Fig. 3 we suspect that the Johnson's SU distribution is more suitable for modeling the electricity prices than the normal. We observe heavy tails and skewness what cannot be explained by the latter. Moreover, as reported in Table 2, the Johnson's SU yields lower CRPS values than the normal distribution and the p-values of the Kolmogorov–Smirnov test for the equality of distributions (Massey, 1951) clearly indicate that the Johnson's SU better fits electricity prices. Thus, in the forecasting study we use both distributions to emphasize the gain that comes from using the more flexible distribution.

4. Models and estimation

4.1. Input features

Let us recall that we forecast the $S = 24$ day-ahead prices on day T, i.e. $\boldsymbol{Y}_T = (Y_{T,1}, Y_{T,2}, \dots, Y_{T,S})'$. The following 221 (227 for LEAR model) input features are available for all considered models:

- Past day-ahead prices of the previous three days and one week ago, i.e. $\boldsymbol{Y}_{T-1}$, $\boldsymbol{Y}_{T-2}$, $\boldsymbol{Y}_{T-3}$, and $\boldsymbol{Y}_{T-7}$.
- Day-ahead forecasts of the total load for day T, i.e. $\boldsymbol{X}^L_T = (X^L_{T,1}, X^L_{T,2}, \dots, X^L_{T,S})'$, as well as the past values of the previous day and one week ago, i.e. $\boldsymbol{X}^L_{T-1}$, and $\boldsymbol{X}^L_{T-7}$.
- Day-ahead forecasts of renewable energy sources (RES) for day T, i.e. $\boldsymbol{X}^{RES}_T = (X^{RES}_{T,1}, X^{RES}_{T,2}, \dots, X^{RES}_{T,S})'$, as well as the past values of the previous day, i.e. $\boldsymbol{X}^{RES}_{T-1}$.
- EU emission allowance most recent closing price, i.e. X^{EUA}_{T-2}.
- Fuels most recent closing prices, i.e. X^{Coal}_{T-2}, X^{Gas}_{T-2}, and X^{Oil}_{T-2}.
- Weekday dummies, i.e. $\boldsymbol{DoW}_d(T)$ for $d = 1, 2, \dots, 7$ for the LEAR model or $DoW(T) = 1, \dots, 7$ for neural network approaches.

Both neural networks and LEAR model select the relevant input features automatically. For neural network models, the feature selection is performed in the hyperparameter optimization step for groups of input variables. There is a total of 14 groups considered: 9 correspond to $S = 24$ inputs each (marked in bold above), whereas the remaining 5 — to single variables (X^{EUA}_{T-2}, X^{Coal}_{T-2}, X^{Gas}_{T-2}, X^{Oil}_{T-2}, and $DoW(T)$). The LEAR model selects the variables through the L_1 regularization applied to each of the 227 input variables separately.

The forecasting exercise is performed on day $T-1$ before the day-ahead auction. That is to say, we possess only the information available at around 11:30 CET on day $T-1$. The considered input does not violate this assumption, and therefore we use e.g. $T-2$ lag for the EUA and fuels prices.

4.2. Probabilistic neural networks

4.2.1. The DDNN model and its hyperparameters

The probabilistic neural network model uses the DDNN described in Section 2. The model consists of 2 hidden layers, S output distributions, and various number of input features. The output distributions are assumed to be either normal or Johnson's SU and each of them defines a separate model. We regularize the model through input feature selection, dropout layer and L_1 regularization of the hidden layers and weights. All these are subject to hyperparameter tuning. Additionally, we tune the activation functions, the number of neurons, and the learning rate. The detailed list of hyperparameters and the process is described in Section 4.2.2.

The model is built and estimated using the TensorFlow (Abadi et al., 2015) and Keras (Chollet et al., 2015) frameworks. The hyperparameter optimization is performed with the help of Optuna (Akiba et al., 2019) package, 4 times for result stability reasons, each time consisting of 2048 iterations. We report the results for each of the 4 optimized hyperparameter sets, as well as for 3 different ensembles of the four distributions, as described in Section 5. The model consists naturally also of components that are not tuned in the hyperparameter optimization. That is to say, the model uses additionally an input normalization, negative loglikehood as the loss function, Adam optimizing algorithm, and early stopping callback with patience of 50 epochs. The batch size is fixed to 32, and the maximum number of epochs to 1500. For the rolling prediction, the dataframe was shuffled and 20% was left out for validation.

Probabilistic neural networks are denoted in the later parts of the paper using **DDNN-{distribution}-{run}** scheme, where **{distribution}** is either **Normal** (or **N**) or **JSU** and **{run}** is either a number from 1 to 4 (corresponding to the individual hyperparameter sets), or an indicator of the ensemble of the four: **pEns** for the vertical average or **qEns** for the horizontal averaging. See Section 4.2.2 for the description of different schemes. Note, that when the choice of the distribution is obvious, as in Fig. 5, the **{distribution}** term may be missing in the model acronym.

4.2.2. Hyperparameter tuning

The neural network models (both point and distributional) underwent the hyperparameter optimization considering below hyperparameters and their potential values:

- Indicator for inclusion of input features described in Section 4.1 (14 hyperparameters).
- Dropout layer — whether to use the dropout layer after the input layer, and if yes at what rate. The rate parameter is drawn from (0, 1) interval (up to 2 hyperparameters — the rate is not optimized if dropout layer is not present in the model).
- Number of neurons in the hidden layers. The values are chosen from integers from [16, 1024] interval (1 hyperparameter per layer).
- Activation functions used in each of the hidden layers. The possible functions are: elu, relu, sigmoid, softmax, softplus, and tanh (1 hyperparameter per layer).
- L_1 regularization for hidden layers — whether to use the L_1 regularization on the hidden layers and their weights and if yes at what rate. The rate is drawn from $(10^{-5}, 10)$ interval on a log-scale (up to 2 hyperparameters per layer — inclusion of L_1 for the layer and the rate).
- L_1 regularization for distribution layer – separate for each of the P distribution parameters, where $P = 2$ for normal and $P = 4$ for Johnson's SU distributions – whether to use the L_1 regularization and if yes at what rate (a total of $2P$ hyperparameters; rates identical to the hidden layer regularization).
- Learning rate for the Adam algorithm chosen from the $(10^{-5}, 10^{-1})$ interval on a log-scale (1 hyperparameter).

The process consists of 2048 iterations of the optimization algorithm which are performed in a hybrid batch-rolling approach. Having the first four years (1456 days) at disposal, we split them into training data (the first 1092 days) and validation data (the last 364 days). Note, that the first day of the out-of-sample test window is the day after the end of the hyperparameter validation data, as illustrated on Fig. 2 (i.e., there is no data contamination). The hybrid approach is needed to balance two opposing factors. On one hand, a batch estimation (using a single estimation on NN weights) would be less computationally demanding (we would only have 1 neural network trained for every considered hyperparameter set), however the results of such an experiment are very volatile. The best hyperparameter set chosen using the accuracy metric of only a single run would not – in general – guarantee a good predictive performance. On the other hand, a rolling setting identical to the one used later (i.e., with a daily recalibration) would be infeasible to compute (as it would take roughly 364 times longer than for batch approach — we would have 364 neural networks trained for every hyperparameter set). The hybrid approach we have chosen uses 13 recalibrations of neural network models with batches of 28 days estimated using each of the nets. Training data is rolled by 28 days after each step.

As mentioned earlier, to counteract the local behavior of the hyperparameter optimizer, we repeat the process four times for each of the neural networks. We observe that the predictive performance across the separate hyperparameter sets is not consistent, however the simple aggregation schemes described below provide results consistently better than any of the inputs.

The first of the aggregation schemes is a mixture distribution, which corresponds to averaging the distributions vertically. However, having two distributions with disjoint pdfs (e.g., two copies of the same distributions significantly shifted), the resulting mixture will be very wide, and might have a "gap" in the middle. A more robust alternative is considered, which utilizes horizontal (quantile) averaging — i.e., a quantile of an ensemble is computed as an arithmetic mean of the same quantiles from all distributions considered. Such an aggregation in an edge case described earlier would result in an unimodal ensemble distribution, which is much sharper than the vertically averaged one.

4.3. Benchmarks

4.3.1. The naive model

The first and the simplest benchmark model that we consider is the well-known and widely utilized (Weron, 2014; Ziel and Weron, 2018) **naive** model. It requires no parameter tuning. Its formula is as follows

$$\mathbb{E}\left(\boldsymbol{Y}_T\right) = \begin{cases} \boldsymbol{Y}_{T-7}, & \mathbf{DoW}_d(T) = 1 \text{ for } d = 1, 6, 7, \\ \boldsymbol{Y}_{T-1}, & \text{otherwise.} \end{cases} \tag{5}$$

In other words, the **naive** model uses the prices of yesterday to forecast the prices on Tuesday, Wednesday, Thursday and Friday, and the last week's prices on Monday, Saturday and Sunday. The price distributions are obtained using the bootstrap method which was first proposed by Efron (1979). We receive the distributions by adding the in-sample bootstrapped errors to the forecasted expected price

$$\hat{\boldsymbol{Y}}_T^m = \widehat{\mathbb{E}\left(\boldsymbol{Y}_T\right)} + \hat{\varepsilon}_T^m \text{ for } m = 1, \ldots, M \tag{6}$$

where $\hat{\varepsilon}_T^m$ are drawn with replacement in-sample residuals for day T, i.e., we sample from the set of $\hat{\varepsilon}_d = \boldsymbol{Y}_d - \hat{\boldsymbol{Y}}_d$ for $d = 1, \ldots, D$.

4.3.2. The LEAR model combined with QRA

The first of the models that use the structure presented in Section 4.1 is LEAR point forecasting model that uses Quantile Regression Averaging (QRA) to generate probabilistic forecasts. The LEAR model utilizes the LASSO regularization (Tibshirani, 1996). Such an approach eliminates the need for an additional input selection, as the algorithm itself indirectly chooses the most relevant inputs. The regularization parameter (the sole hyperparameter of the LEAR model) is obtained using 7-fold cross validation and a grid of 100 values automatically chosen by a least angle regression (LARS) based estimator. The LEAR approach encompasses a forecast averaging scheme proposed by Lago et al. (2021) – four independent forecasts are generated for each hour (based on 56, 84, 1092 and 1456 day rolling calibration windows) and the final output is their simple average. Such an approach allows for a balance of the ability to adapt to rapidly-changing market conditions (thanks to the shorter calibration windows) with robustness coming from the use of long windows. It was shown to provide forecasts that – on average – are on par or better than all of the comprising forecasts considered separately (Lago et al., 2021).

There are two ways of using a set of four separate forecasts or an ensemble: one that uses the whole information directly (i.e., the separate forecasts), which we will denote **QRA** (Quantile Regression Averaging) and **QRM** (Quantile Regression committee Machine) that uses the ensemble of the point predictions (Marcjasz et al., 2020).

Aside from the input data, the QRA and QRM approaches are identical — both use quantile regression with 182 day rolling calibration window to produce the forecast for each of the 99 percentiles, which approximate the predictive distribution relatively well.

The LEAR models' results are denoted by **LEAR-{CAL}** for the point forecast estimated using **{CAL}** calibration days (e.g., **LEAR-1456** for the longest calibration window), **LEAR-Ens** for an hour-by-hour average of all 4 point forecasts and **LEAR-QRA** and **-QRM** for the probabilistic forecasts.

4.3.3. The DNN model combined with QRA

The second set of benchmarks utilizes a point neural network model. It differs from the probabilistic counterpart only in the output construction in the network and hyperparameters corresponding to the missing distribution layer (see Sections Section 2, 4.2 and Fig. 1). The rest of the model setting remains unchanged: DNN model has the same inputs, the same hyperparameter selection and uses the same calibration window lengths and training and validation splits. The loss function for the network is MAE, whereas DDNN uses log-likelihood.

Similarly to the DDNN, for the (point) DNN we also derive four independently-trained hyperparameter sets. This allows us to *(i)* measure the robustness of the predictions and *(ii)* apply two quantile-regression based methods (QRA and QRM), similarly as for the LEAR point predictions, also using a 182 day rolling calibration window.

The results are marked with **DNN-n** for the point forecasts (where $n = 1, \dots, 4$ or **Ens**) and **DNN-QRA** and **DNN-QRM**, respectively for percentile forecasts obtained using quantile regression on the four separate point forecasts and their ensemble.

5. Empirical results

The EPF literature is clear that forecast averaging is often the key to achieving accurate forecasts (Lago et al., 2021; Marcjasz et al., 2020; Hubicka et al., 2018; Bordignon et al., 2013; Weron, 2014; Uniejewski and Maciejowska, 2022). Here, we also aggregate multiple forecast runs to improve the result accuracy and robustness. However, considering probabilistic instead of point forecasts significantly increases the complexity of the aggregation schemes that can be applied. As the detailed discussion is out of scope of this paper, we opted to include only the simple aggregations, based on the equally-weighted averaging or distribution mixing.

On the probabilistic forecasts side, we have four hyperparameter sets chosen in four separate hyperparameter optimization runs for both the normal and JSU DDNNs. We report the errors of each of them separately, as well as the result of two aggregation schemes: an equally-weighted mixture of the four resulting distributions (vertical aggregation) or a mean of values for a given quantile (horizontal aggregation).

5.1. Statistical evaluation

5.1.1. Point forecasts

While the paper focuses on probabilistic forecasting, we are also interested in the accuracy of the point forecasts. The latter can be easily derived from the predictive distributions. Following the best EPF practices (Weron and Ziel, 2019; Lago et al., 2021), we report two point-oriented metrics: the mean absolute error (MAE) and the root mean squared error (RMSE).

5.1.2. Probabilistic forecasts

When it comes to the probabilistic forecasts, we use the continuous ranked probability score (CRPS), or rather its approximation — the average (or aggregate) pinball score across 99 percentiles (Gneiting, 2011; Hong et al., 2016; Nowotarski and Weron, 2018):

$$\mathrm{CRPS}_{d,h} = \frac{1}{99}\sum_{q=1}^{99} \mathrm{PS}(\hat{Y}^q_{d,h}, Y_{d,h}, q) \tag{7}$$

with the pinball score for quantile q given by:

$$\begin{aligned}\mathrm{PS}(\hat{Y}^q_{d,h}, Y_{d,h}, q) &= \\ &= \begin{cases} (1-q)\left(\hat{Y}^q_{d,h} - Y_{d,h}\right) & \text{for } Y_{d,h} < \hat{Y}^q_{d,h}, \\ q\left(Y_{d,h} - \hat{Y}^q_{d,h}\right) & \text{for } Y_{d,h} \geq \hat{Y}^q_{d,h}, \end{cases}\end{aligned} \tag{8}$$

where $\hat{Y}^q_{d,h}$ is the forecast of the qth quantile of $Y_{d,h}$.

5.1.3. Testing for statistical significance

For each hour of the day, we perform the Kupiec (1995) test for unconditional coverage for the 50% and 90% prediction intervals (PIs). For the CRPS, we aggregate the losses across all predicted hours, whereas for the Kupiec test, we provide the number of hours which passed the test.

Moreover, we use the Diebold and Mariano (1995) test (DM) to assess differences in predictive accuracy with the CRPS as the loss function. Following Lago et al. (2021) and Ziel and Weron (2018), we consider the multivariate loss differential series:

$$\Delta^d_{A,B} = \|L^d_A\|_1 - \|L^d_B\|_1, \tag{9}$$

which defines the difference in the L_1 norm of loss vectors for models A and B. Here $L^d_Z = \{L^{d,1}_Z, \dots, L^{d,24}_Z\}$ denotes the 24-dimensional vector of prediction errors of model Z for day d, and $\|L^d_Z\|_1 = \sum_{h=1}^{24} |L^{d,h}_Z|$. For each pair of models, we compute the p-values of two one-sided DM tests — one with the null hypothesis $\mathcal{H}_0 : \mathbb{E}(\Delta^d_{A,B}) \leq 0$, which corresponds to the outperformance of model B forecasts ($\rightarrow$ worse) by those of model A ($\rightarrow$ better), and the second with the reverse null $\mathcal{H}_1 : \mathbb{E}(\Delta^d_{A,B}) > 0$, or equivalently $\mathbb{E}(\Delta^d_{B,A}) < 0$. As in the standard DM test, we assume that the loss differential series is covariance stationary.

5.2. Economic evaluation

Statistical error measures may not necessarily reflect the economic value of reducing prediction errors (Hong et al., 2020; Maciejowska et al., 2023). Hence, following Uniejewski (2023), we complement them here by a case study, which compares profits from a realistic trading strategy that utilizes battery storage and day-ahead electricity price forecasts. Such a strategy can be deployed by, e.g., a company that manages a virtual power plant with an energy storage system (Shabanzadeh et al., 2017).

5.2.1. Quantile-based trading strategy

Without loss of generality, let us assume that the total usable capacity of the battery is $B = 2$ MWh, and that the efficiency of both charging and discharging processes is 90% (for a total efficiency of 81%). The strategy proposed by Uniejewski (2023) aims to find the most profitable hours of the day to submit a buy order for $\frac{1}{0.9}$ MWh at hour $h1$ (which would result in charging the battery by 1 MWh) and a sell bid for 0.9 MWh at hour $h2 > h1$ (the amount of energy from discharging 1 MWh). Both are limit orders with the limits determined by probabilistic forecasts.

More precisely, as shown in Fig. 4, the selection of $h1$ and $h2$ is based on maximizing the difference in median price forecasts for the next day, i.e., $\Delta\hat{Y}^{0.5}_d = \hat{Y}^{0.5}_{d,h2} - \hat{Y}^{0.5}_{d,h1}$, while the limits are determined by the risk appetite α of the trader. Namely, the trader places two orders — a buy order with price limit $\hat{Y}^{1-q}_{d,h1}$ and a sell order with price limit $\hat{Y}^{q}_{d,h2}$, where $q = \frac{1-\alpha}{2}$. Note, that unlike (Uniejewski, 2023), we assume that the orders are placed only if $0.9\hat{Y}^q_{d,h2} - \frac{1}{0.9}\hat{Y}^{1-q}_{d,h1} > 0$, i.e., if we expect the transaction to be profitable. The risk appetite α, i.e., the prediction interval (PI) width, is set once for the whole evaluation period. However, for comparison purposes we consider $\alpha \in \{90\%, 80\%, 70\%, 60\%, 50\%\}$.

In such a setting it is possible that not all orders are executed. Hence, if at midnight the battery is fully charged ($B = 2$ MWh), a market order to sell 0.9 MWh is placed before hour $h1$, so that the "starting" battery charge state is 1 MWh. Likewise, if at midnight the battery is discharged ($B = 0$ MWh), a market order to buy $\frac{1}{0.9}$ MWh is placed before hour $h2$. Such a behavior allows us to always submit two limit orders – buy and sell – regardless of the number of executed orders on the previous day. The profit maximizing selection of particular hours to submit these market orders is made using a linear optimization solver.

5.2.2. Benchmark strategies

Apart from the quantile-based strategy described in Section 5.2.1, we also consider two benchmark strategies. The first, dubbed 'unlimited bids' and denoted by $\mathrm{unl}_{\mathrm{median}}$, assumes that the trader submits market orders for hours $h1$ and $h2$. In other words, only the moments of market entry are determined, not the price limits. In such a case the profit is always equal to $0.9Y_{d,h2} - \frac{1}{0.9}Y_{d,h1}$, and can be negative. The second benchmark, dubbed 'fixed hours', assumes that market buy orders are

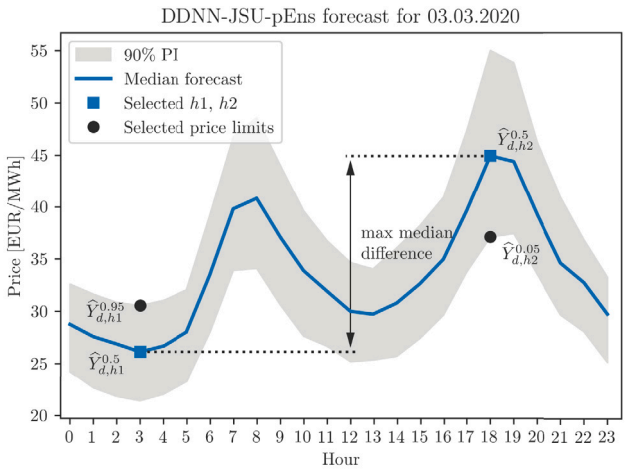


Fig. 4. Illustration of the quantile-based trading strategy for the 90% PIs of the DDNN-JSU-pEns model on 03.03.2020. The blue line represents the median forecast $\hat{Y}^{0.5}_{d,h}$ for $h = 1, 2, \ldots, 24$, whereas the gray area corresponds to the 90% PI. The selected hours $h1$ and $h2$ are marked by blue squares, while the corresponding price limits $\hat{Y}^{0.95}_{d,h1}$ and $\hat{Y}^{0.05}_{d,h2}$ by black dots.

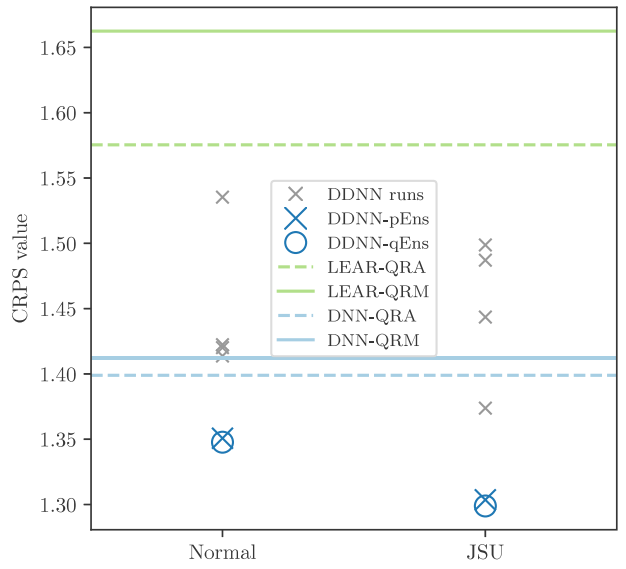


Fig. 5. CRPS values for the DDNN models and the benchmarks. Gray markers correspond to the single hyperparameter set results, whereas color ones to the combination schemes utilizing these four runs. Dashed lines mark the QRA method, while solid ones QRM. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

always submitted for hour 3 and market sell orders for hour 19. These two hours were selected *ex-post* as the ones with the lowest and the highest prices in the whole out-of-sample period, respectively.

5.2.3. Economic evaluation metrics

We report two trading strategy metrics: the total profit across the whole out-of-sample test period of 554 days and the per-transaction profit. Note, that we ignore transaction costs and the impact of the charging cycle on battery life. That said, the per-transaction profit is the more important metric as it can easily be adjusted for actual transaction costs and battery wear.

Table 3
Comparison of point (MAE, RMSE) and probabilistic (CRPS, Kupiec test) forecasting accuracy for the considered models. The best result in each column is emphasized in bold.

	MAE	RMSE	CRPS	Kupiec 50%	Kupiec 90%
LEAR-Ens	4.372	6.375	–	–	–
DNN-Ens	3.610	5.850	–	–	–
naive	9.336	14.358	3.585	**21**	**23**
LEAR-QRA	4.161	6.676	1.575	10	8
LEAR-QRM	4.285	6.788	1.662	6	3
DNN-QRA	3.668	5.845	1.399	6	10
DNN-QRM	3.670	**5.821**	1.412	9	8
DDNN-N-pEns	3.663	5.962	1.351	2	6
DDNN-N-qEns	3.670	5.962	1.348	13	20
DDNN-JSU-pEns	**3.542**	6.146	1.304	1	4
DDNN-JSU-qEns	3.564	6.174	**1.299**	14	13

5.3. Results

5.3.1. MAE, RMSE and CRPS

In terms of the CRPS, as can be seen in Table 3 and Fig. 5, the LEAR-based methods are much worse than the neural network-based approaches. The differences can be also observed in Fig. 6, where QRA-based approaches have very narrow prediction intervals. Note, that the widest depicted interval is the 98% one. However, the performance of the NN models is not robust — run-to-run, the CRPS values differ by as much as 10%. As discussed in Section 5.4, this is not known *ex-ante*, therefore an aggregation scheme is needed. After ensembling, regardless of the aggregation scheme applied (vertical, horizontal with mean, horizontal with median), we see similar performance. The normally-distributed networks yield CRPS ≈ 1.35, whereas JSU ones yield CRPS ≈ 1.30, i.e., ca. 3%–4% better than the normal. DNN-QR-type methods can be placed between the DDNN ensembles and the individual runs. Finally, as can be seen in Table 3, the best performing DDNN yields over 7% lower CRPS than the best non-DDNN model (1.299 vs. 1.399).

As shown in Table 3, the neural network-based models are better than LEAR-based ones also for the point forecasts. Interestingly, the ensemble of DNN forecasts has the third lowest error — both in terms of MAE and RMSE. The best model according to MAE is DDNN-JSU-pEns, followed by its -qEns counterpart — the two models with the best CRPS values. However, these models are worse w.r.t. the RMSE than all other NN-based models. On the other hand, we see the lowest RMSE for DNN-QR based methods, closely followed by the DNN-Ens model. The DDNNs are ca. 2% (normal) and 7% (JSU) worse. There are only minor differences between the vertical and horizontal aggregation schemes.

5.3.2. The Kupiec test

Additionally, we performed the Kupiec test for unconditional coverage with 5% significance level for 50% and 90% prediction intervals. From what can be seen in Table 3, QRA seems to perform better than QRM — for both the LEAR and point DNN models. However, the QR-based approaches pass the Kupiec test for at most 10 h of the day. The probabilistic DDNNs, on the other hand, show mixed performance. The p-Ens forecasts are worse than most other methods, while q-Ens are better than QR-based predictions. As the p-Ens models sport the CRPS similar to the q-Ens ones, the latter are a much better choice, especially when chosen with a more robust median quantile instead of mean. Note, however, that the worst overall method (Naive benchmark) provides the best coverage for both 50 and 90% PIs.

5.3.3. The Diebold-Mariano test

The results of the DM test for the CRPS are visualized in Fig. 7. Each rectangle above the "chessboard" diagonal corresponds to a one-sided test of the null H_0 with model A being on the X-axis and model B on the Y-axis, see Section 5.1.3, while each rectangle below the diagonal to a one-sided test of the reverse null H_1. For instance, the first row is dark green indicating that the forecasts of the naive model are significantly

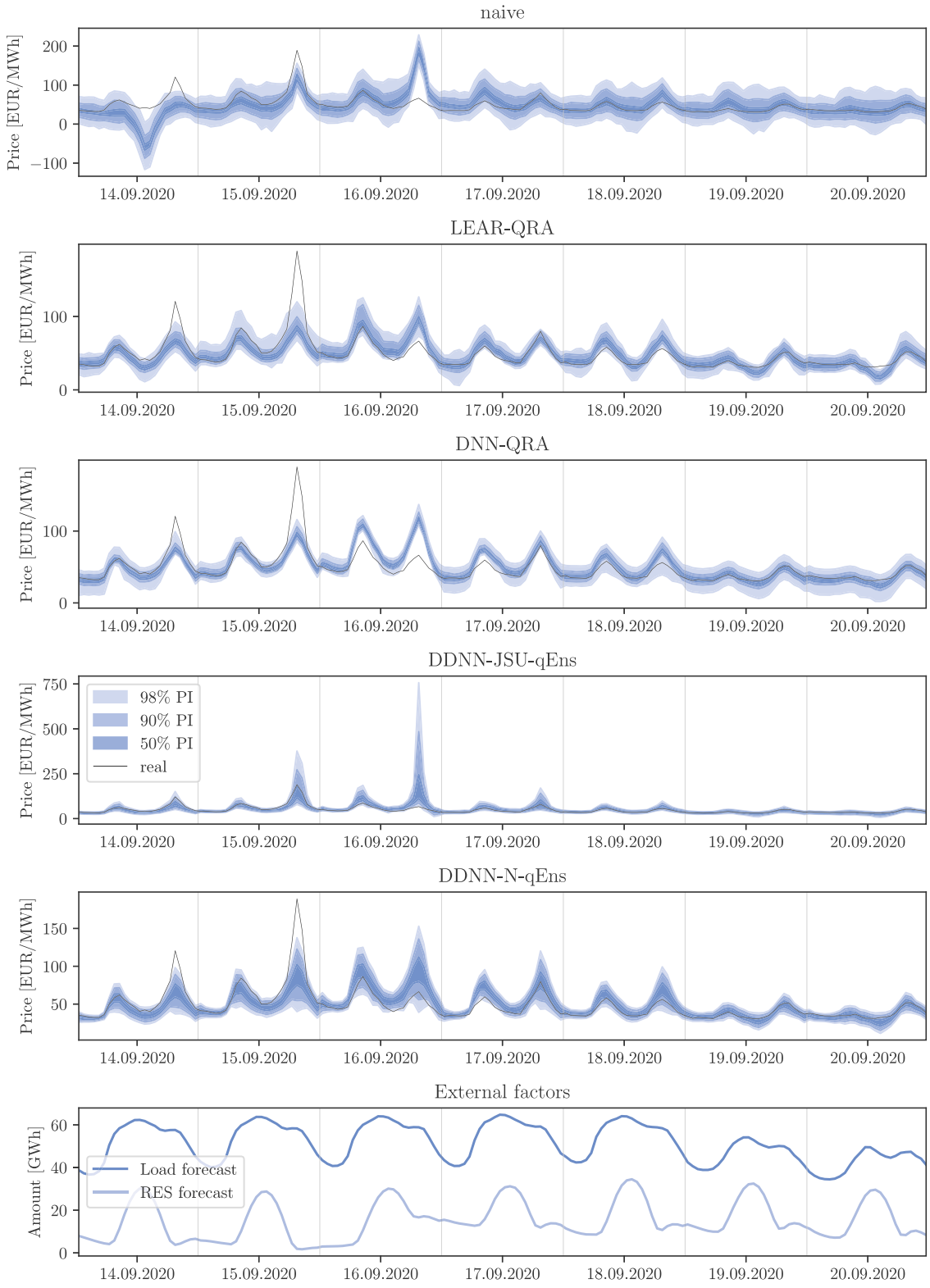


Fig. 6. *Top panels*: Visualization of 98%, 90% and 50% prediction intervals for five models and a week in September 2020. Quantile (horizontal) averaging with mean is depicted for the two DDNN models. The actual (real) prices are plotted in black. *Bottom panel*: Load and RES generation day-ahead forecasts for the same period.

outperformed by those of all other models. Conversely, the first column is black meaning that the naive forecasts do not outperform those of any other model.

We can observe in Fig. 7 that DDNN-JSU-qEns is the best model overall, with forecasts significantly better than from any other model (the last column has only green or yellow cells). Moreover, horizontal averaging (qEns) yields significantly better predictions than the vertical one (pEns) for both the normal and JSU distributions, while QRA models (both LEAR and DNN ones) produce significantly better forecasts than their QRM counterparts.

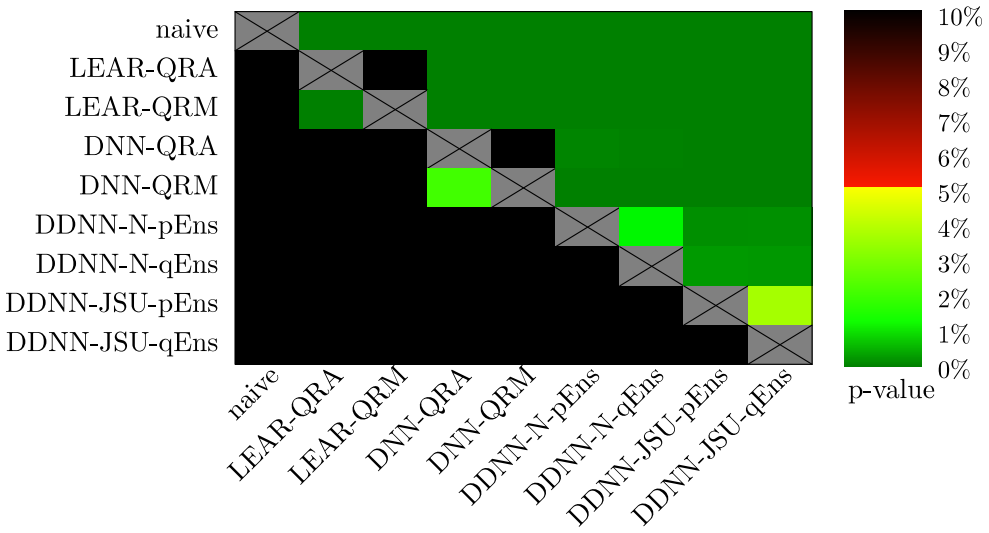


Fig. 7. Results of the Diebold-Mariano test for the CRPS loss. For each pair of models we perform two one-sided tests and use a heat map to indicate the range of the p-values. The closer they are to zero (→ dark green) the more significant is the difference between the forecasts of a model on the X-axis (better) and the forecasts of a model on the Y-axis (worse). Black color indicates p-values of 10% or more. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Total trading profits in EUR for the strategies defined in Section 5.2. Values in the top row correspond to the PI width used. The best result in each column is emphasized in bold.

	90%	80%	70%	60%	50%	$\text{unl}_{\text{median}}$
Fixed hours	8048					
LEAR-QRA	10 184	11 525	11 873	11 954	11 895	11 519
LEAR-QRM	10 230	11 329	11 680	11 584	11 493	**12 003**
DNN-QRA	9 995	11 182	11 449	11 505	11 301	11 533
DNN-QRM	10 106	11 235	11 320	11 310	11 303	11 699
DDNN-N-pEns	5 874	9 766	11 514	11 900	12 145	10 646
DDNN-N-qEns	6 587	10 187	11 488	11 999	12 110	10 723
DDNN-JSU-pEns	9 163	11 147	12 074	12 308	12 154	11 492
DDNN-JSU-qEns	**10 249**	**11 823**	**12 247**	**12 408**	**12 360**	11 750

Table 5
Per-transaction profits in EUR/MWh for the strategies defined in Section 5.2. Values in the top row correspond to the PI width used. The best result in each column is emphasized in bold.

	90%	80%	70%	60%	50%	$\text{unl}_{\text{median}}$
Naive	7.26					
LEAR-QRA	13.47	12.68	12.28	12.16	12.38	10.40
LEAR-QRM	13.13	12.57	**12.31**	**12.21**	**12.53**	**10.83**
DNN-QRA	13.77	12.26	11.54	11.39	11.37	10.41
DNN-QRM	13.69	11.88	11.57	11.36	11.46	10.56
DDNN-N-pEns	13.99	12.33	11.92	11.78	11.65	9.61
DDNN-N-qEns	13.33	11.93	11.82	11.74	11.64	9.68
DDNN-JSU-pEns	**14.92**	**12.96**	12.15	11.90	11.66	10.37
DDNN-JSU-qEns	14.04	12.45	12.05	12.02	11.79	10.60

5.3.4. Trading strategy

Lastly, the results for the trading strategies are presented in Tables 4 and 5. With respect to the total profit reported in Table 4, we can observe that DDNN-JSU-type models are among the best for all PI widths and the unlimited bids benchmark. Most notably, they are by far outperforming their normal distribution counterparts.

Since the trading results assume no transaction and operational costs, a much more important metric is the per-transaction profit reported in Table 5. Here, we can observe that DDNN-JSU-type models outperform alternative approaches, while still achieving a high total profit (as presented in Table 4). In particular, the best performing model yields 8% higher per-transaction profit than the best non-DDNN model (14.92 vs. 13.77 EUR/MWh; see the 90% PI column in Table 5). Moreover, probabilistic forecasts achieve higher per-transaction profits than the unlimited benchmark, regardless of the PI width, and the fixed hours benchmark is outperformed by every other forecasting method. Finally, note that quantile-based trading strategies built on probabilistic forecasts can achieve profits that are very close to the theoretical maximum — the "oracle" strategy, which always buys at the lowest price of the day and sells at the highest, yields a total profit of 13,587 EUR.

5.4. The need for multiple hyperparameter sets

Even though the hyperparameter optimization uses a repeated neural network training procedure to mimic the rolling calibration window setting used later for the evaluation, the optimal sets obtained using independent hyperparameter trials differ significantly. Moreover, all the optimal sets have a similar, i.e. within 2% difference, *in-sample* error metric — what is not reflected in the out-of-sample error obtained using this set. Here, the differences are much more prominent, up to 10%. The locality of the hyperparameter optimization is clearly visible in the optimal sets chosen in independent trials, despite most of the trials being stalled after around 1000 iterations. Fig. 8 shows choice frequency of the considered input features (i.e., the number of hyperparameter sets that uses a particular input variable), described in Section 4.1. All 3 considered neural network models are quite consistent when selecting the inputs. The most important ones are the prices of the previous day and two days ago, the current DA forecasts of load and RES, the previous day's DA forecasts of RES and the recent gas price. The least important are the further lags of prices and load forecast.

Besides the differences in the inputs chosen, hidden layer sizes are the most prominent, especially for the probabilistic networks. They found optima in both the larger and smaller networks, as shown in Table 6. For example, one of the probabilistic neural networks that used the JSU distribution uses 565 and 962 neurons in the hidden layers (amounting to over 540,000 weights just between the two hidden layers), whereas other had 940 and 58 (over 54,000 weights) or 123 and 668 (over 82,000 weights). Moreover, even the activation functions chosen were not unanimous, but `softplus` seems to be the best for the first hidden layer. We also observe that the dropout is almost never chosen, similarly to the regularization of the network weights.

To conclude, we observe that there is a need for repeating the hyperparameter optimization process. Despite the robust optimization

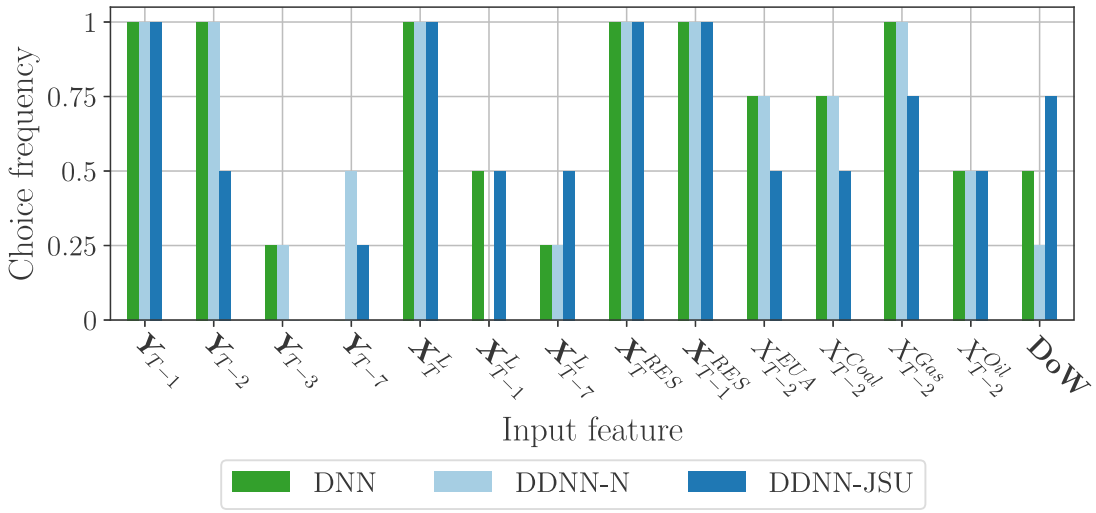


Fig. 8. The frequency with which (out of 4 hyperparameter optimization trials) an input group was chosen for inclusion in DNN models. The groups are described in detail in Section 4.1.

Table 6
Activation functions and number of neurons selected in each of the hyperparameter tunings.

Run		DNN				DDNN-N				DDNN-JSU			
		1	2	3	4	1	2	3	4	1	2	3	4
Layer 1	Activation	softplus	softplus	softplus	softplus	softplus	softplus	softplus	softplus	softplus	elu	softplus	softplus
	Neurons	906	912	979	965	948	266	542	110	565	940	243	123
Layer 2	Activation	softplus	softplus	relu	elu	elu	relu	softplus	relu	relu	elu	elu	elu
	Neurons	901	619	767	448	554	883	641	823	962	58	895	668

setting, the end results are vastly different — both in terms of the parameters chosen, and the out-of-sample error metrics. A form of the forecast combination is crucial for the outperformance of QR-based methods.

6. Conclusions

The paper proposes an application of distributional neural networks to probabilistic day-ahead electricity price forecasting and a simple, yet well-performing aggregation scheme for the distributional neural networks that stabilizes the predictions. Since probabilistic forecasting is the essence of risk management – Value-at-Risk (VaR) is nothing else but a quantile forecast – our study provides important implications for managing portfolios in the power sector.

Comparing the results with the literature approaches, we observe a strong performance of the neural networks — both the probabilistic forecasts from the proposed methods and from quantile regression applied to their point counterparts are significantly more accurate than the statistical-based combination of LEAR and quantile regression. The added complexity of the neural network having to model the distribution of the data, rather than just their expected values, proves effective, especially when the limitations incurred by the distribution choice are not too severe. However, while the DDNN approach does not add too much complexity on top of the point DNN counterpart, the performance of the model depends strongly on the distribution. As the optimal choice is likely related to the data that is modeled, one of the possible future research directions will be a study that tests a broader selection of distributions across multiple datasets, with various electricity usage patterns and generation mixes.

Interestingly, the benefit of using distributional neural networks is visible also when mean absolute errors of the median (50th percentile) forecasts are considered. The DDNN-JSU-Ens approach is the only one that outperforms the ensemble of point NNs in this regard.

CRediT authorship contribution statement

Grzegorz Marcjasz: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Michał Narajewski:** Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Rafał Weron:** Conceptualization, Methodology, Writing – review & editing. **Florian Ziel:** Conceptualization, Methodology, Writing – review & editing.

Acknowledgments

This research was partially supported by the Ministry of Science and Higher Education (MNiSW, Poland) through Diamond Grant No. 0219/DIA/2019/48 (to G.M.) and the National Science Center (NCN, Poland) through grant No. 2018/30/A/HS4/00444 (to R.W. and F.Z.).

References

Abadi, M., Agarwal, A., Barham, P., et al., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. URL: https://www.tensorflow.org/. Software available from tensorflow.org.

Abramova, E., Bunn, D., 2020. Forecasting the intra-day spread densities of electricity prices. Energies 13 (3), 687.

Afrasiabi, M., Mohammadi, M., Rastegar, M., Stankovic, L., Afrasiabi, S., Khazaei, M., 2020. Deep-based conditional probability density function forecasting of residential loads. IEEE Trans. Smart Grid 11 (4), 3646–3657.

Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2623–2631.

Barnes, E.A., Barnes, R.J., 2021. Controlled abstention neural networks for identifying skillful predictions for classification problems. J. Adv. Modelling Earth Syst. 13 (12), e2021MS002573.

Barnes, E.A., Barnes, R.J., Gordillo, N., 2021. Adding uncertainty to neural network regression tasks in the geosciences. arXiv:2109.07250.

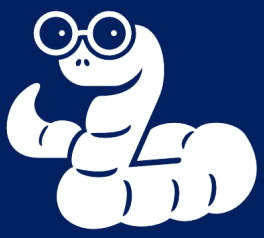
Berrisch, J., Ziel, F., 2022. CRPS learning. J. Econometrics http://dx.doi.org/10.1016/j.jeconom.2021.11.008.

Billé, A.G., Gianfreda, A., Del Grosso, F., Ravazzolo, F., 2022. Forecasting electricity prices with expert, linear, and nonlinear models. Int. J. Forecast. http://dx.doi.org/10.1016/j.ijforecast.2022.01.003.
Bordignon, S., Bunn, D.W., Lisi, F., Nan, F., 2013. Combining day-ahead forecasts for British electricity prices. Energy Econ. 35, 88–103.
Browell, J., Gilbert, C., 2022. Predicting electricity imbalance prices and volumes: Capabilities and opportunities. Energies 15 (10).
Brusaferri, A., Matteucci, M., Ramin, D., Spinelli, S., Vitali, A., 2020. Probabilistic day-ahead energy price forecast by a mixture density recurrent neural network. In: 2020 7th International Conference on Control, Decision and Information Technologies (CoDIT), Vol. 1. pp. 523–528.
Chollet, F., et al., 2015. Keras. https://keras.io.
Diebold, F.X., Mariano, R.S., 1995. Comparing predictive accuracy. J. Bus. Econ. Statist. 13 (3), 253–263.
Dudek, G., 2016. Multilayer perceptron for GEFCom2014 probabilistic electricity price forecasting. Int. J. Forecast. 32 (3), 1057–1060.
Efron, B., 1979. Bootstrap methods: Another look at the jackknife. Ann. Statist. 1–26.
ENTSOE, 2022. ENTSOE transparency. https://transparency.entsoe.eu/. Accessed: 2022-09-30.
Gianfreda, A., Bunn, D., 2018. A stochastic latent moment model for electricity price formation. Oper. Res. 66 (5), 1189–1203.
Gneiting, T., 2011. Quantiles as optimal point forecasts. Int. J. Forecast. 27 (2), 197–207.
Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., Hyndman, R.J., 2016. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. Int. J. Forecast. 32 (3), 896–913.
Hong, T., Pinson, P., Wang, Y., Weron, R., Yang, D., Zareipour, H., 2020. Energy forecasting: A review and outlook. IEEE Open Access J. Power Energy 7, 376–388.
Hubicka, K., Marcjasz, G., Weron, R., 2018. A note on averaging day-ahead electricity price forecasts across calibration windows. IEEE Trans. Sustain. Energy 10 (1), 321–323.
Jahangir, H., Tayarani, H., Baghali, S., Ahmadian, A., Elkamel, A., Golkar, M.A., Castilla, M., 2019. A novel electricity price forecasting approach based on dimension reduction strategy and rough artificial neural networks. IEEE Trans. Ind. Inform. 16 (4), 2369–2381.
Janczura, J., Wójcik, E., 2022. Dynamic short-term risk management strategies for the choice of electricity market based on probabilistic forecasts of profit and risk measures. The German and the Polish market case study. Energy Econ. 110, 106015.
Janke, T., Steinke, F., 2019. Forecasting the price distribution of continuous intraday electricity trading. Energies 12 (22), 4262.
Jędrzejewski, A., Lago, J., Marcjasz, G., Weron, R., 2022. Electricity price forecasting: The dawn of machine learning. IEEE Power Energy Mag. 20 (3), 24–31.
Johnson, N.L., 1949. Systems of frequency curves generated by methods of translation. Biometrika 36 (1/2), 149–176.
Karabiber, O.A., Xydis, G., 2019. Electricity price forecasting in the Danish day-ahead market using the TBATS, ANN and ARIMA methods. Energies 12 (5), 928.
Keles, D., Scelle, J., Paraschiv, F., Fichtner, W., 2016. Extended forecast methods for day-ahead electricity spot prices applying artificial neural networks. Appl. Energy 162, 218–230.
Kraft, E., Keles, D., Fichtner, W., 2020. Modeling of frequency containment reserve prices with econometrics and artificial intelligence. J. Forecast. 39 (8), 1179–1197.
Kupiec, P.H., 1995. Techniques for verifying the accuracy of risk measurement models. J. Deriv. 3 (2).
Lago, J., De Ridder, F., De Schutter, B., 2018. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. Appl. Energy 221, 386–405.
Lago, J., Marcjasz, G., De Schutter, B., Weron, R., 2021. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. Appl. Energy 293, 116983.
Luo, S., Weng, Y., 2019. A two-stage supervised learning approach for electricity price forecasting by leveraging different data sources. Appl. Energy 242, 1497–1512.
Maciejowska, K., 2020. Assessing the impact of renewable energy sources on the electricity price level and variability–A quantile regression approach. Energy Econ. 85, 104532.
Maciejowska, K., 2022. Portfolio management of a small RES utility with a structural vector autoregressive model of electricity markets in Germany. Oper. Res. Decis. 32 (4), 75–90.
Maciejowska, K., Nitka, W., Weron, T., 2021. Enhancing load, wind and solar generation for day-ahead forecasting of electricity prices. Energy Econ. 99, 105273.
Maciejowska, K., Nowotarski, J., 2016. A hybrid model for GEFCom2014 probabilistic electricity price forecasting. Int. J. Forecast. 32 (3), 1051–1056.
Maciejowska, K., Uniejewski, B., Weron, R., 2023. Forecasting electricity prices. In: Oxford Research Encyclopedia of Economics and Finance. Oxford University Press, http://dx.doi.org/10.1093/acrefore/9780190625979.013.667.
Marcjasz, G., Uniejewski, B., Weron, R., 2020. Probabilistic electricity price forecasting with NARX networks: Combine point or probabilistic forecasts? Int. J. Forecast. 36 (2), 466–479.
Mashlakov, A., Kuronen, T., Lensu, L., Kaarna, A., Honkapuro, S., 2021. Assessing the performance of deep learning models for multivariate probabilistic energy forecasting. Appl. Energy 285, 116405.
Massey, Jr., F.J., 1951. The Kolmogorov–Smirnov test for goodness of fit. J. Am. Statist. Assoc. 46 (253), 68–78.
Narajewski, M., Ziel, F., 2020a. Econometric modelling and forecasting of intraday electricity prices. J. Commod. Mark. 19, 100107.
Narajewski, M., Ziel, F., 2020b. Ensemble forecasting for intraday electricity prices: Simulating trajectories. Appl. Energy 279, 115801.
Narajewski, M., Ziel, F., 2022. Optimal bidding on hourly and quarter-hourly day-ahead electricity price auctions: trading large volumes of power with market impact and transaction costs. Energy Econ. 110, 105974.
Nix, D., Weigend, A., 1994. Estimating the mean and variance of the target probability distribution. In: Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94), Vol. 1. pp. 55–60.
Nowotarski, J., Weron, R., 2015. Computing electricity spot price prediction intervals using quantile regression and forecast averaging. Comput. Statist. 30 (3), 791–803.
Nowotarski, J., Weron, R., 2018. Recent advances in electricity price forecasting: A review of probabilistic forecasting. Renew. Sustain. Energy Rev. 81, 1548–1568.
Oksuz, I., Ugurlu, U., 2019. Neural network based model comparison for intraday electricity price forecasting. Energies 12 (23), 4557.
Olivares, K.G., Challu, C., Marcjasz, G., Weron, R., Dubrawski, A., 2023. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. Int. J. Forecast. 39, 884–900.
Oreshkin, B.N., Dudek, G., Pełka, P., Turkina, E., 2021. N-BEATS neural network for mid-term electricity load forecasting. Appl. Energy 293, 116918.
Patra, S., 2021. Revisiting value-at-risk and expected shortfall in oil markets under structural breaks: The role of fat-tailed distributions. Energy Econ. 101, 105452.
Petropoulos, F., Apiletti, D., Assimakopoulos, V., et al., 2022. Forecasting: theory and practice. Int. J. Forecast. 38 (3), 705–871.
Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T., 2020. Deepar: Probabilistic forecasting with autoregressive recurrent networks. Int. J. Forecast. 36 (3), 1181–1191.
Serafin, T., Uniejewski, B., Weron, R., 2019. Averaging predictive distributions across calibration windows for day-ahead electricity price forecasting. Energies 12 (13), 2561.
Sgarlato, R., Ziel, F., 2022. The role of weather predictions in electricity price forecasting beyond the day-ahead horizon. IEEE Trans. Power Syst. http://dx.doi.org/10.1109/TPWRS.2022.3180119.
Shabanzadeh, M., Sheikh-El-Eslami, M.-K., Haghifam, M.-R., 2017. An interactive cooperation model for neighboring virtual power plants. Appl. Energy 200, 273–289.
Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B Stat. Methodol. 58 (1), 267–288.
Uniejewski, B., 2023. Smoothing quantile regression averaging: A new approach to probabilistic forecasting of electricity prices. arXiv:2302.00411.
Uniejewski, B., Maciejowska, K., 2022. LASSO principal component averaging: A fully automated approach for point forecast pooling. Int. J. Forecast. http://dx.doi.org/10.1016/j.ijforecast.2022.09.004.
Uniejewski, B., Marcjasz, G., Weron, R., 2019. Understanding intraday electricity markets: Variable selection and very short-term price forecasting using LASSO. Int. J. Forecast. 35 (4), 1533–1547.
Viehmann, J., 2017. State of the german short-term power market. Z. Energ.wirtsch. 41 (2), 87–103.
Wan, C., Xu, Z., Wang, Y., Dong, Z.Y., Wong, K.P., 2013. A hybrid approach for probabilistic forecasting of electricity price. IEEE Trans. Smart Grid 5 (1), 463–470.
Weron, R., 2014. Electricity price forecasting: A review of the state-of-the-art with a look into the future. Int. J. Forecast. 30 (4), 1030–1081.
Weron, R., Ziel, F., 2019. Electricity price forecasting. In: Soytaş, U., Sarı, R. (Eds.), Routledge Handbook of Energy Economics. Routledge, pp. 506–521.
Williams, P.M., 1996. Using neural networks to model conditional multivariate densities. Neural Comput. 8 (4), 843–854.
Zahid, M., Ahmed, F., Javaid, N., Abbasi, R.A., Zainab Kazmi, H.S., Javaid, A., Bilal, M., Akbar, M., Ilahi, M., 2019. Electricity price and load forecasting using enhanced convolutional neural network and enhanced support vector regression in smart grids. Electronics 8 (2), 122.
Zhang, F., Fleyeh, H., Bales, C., 2020. A hybrid model based on bidirectional long short-term memory neural network and catboost for short-term electricity spot price forecasting. J. Oper. Res. Soc. 1–25.
Zhou, S., Zhou, L., Mao, M., Tai, H.-M., Wan, Y., 2019. An optimized heterogeneous structure LSTM network for electricity price forecasting. IEEE Access 7, 108161–108173.
Ziel, F., 2016. Forecasting electricity spot prices using lasso: On capturing the autoregressive intraday structure. IEEE Trans. Power Syst. 31 (6), 4977–4987.
Ziel, F., Steinert, R., 2018. Probabilistic mid-and long-term electricity price forecasting. Renew. Sustain. Energy Rev. 94, 251–266.
Ziel, F., Weron, R., 2018. Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks. Energy Econ. 70, 396–420.

Paper 5

Trading on short-term path forecasts of intraday electricity prices. Part II – Distributional Deep Neural Networks

Grzegorz Marcjasz, Tomasz Serafin, Rafał Weron



WORKing papers in Management Science

WORMS/23/01

Trading on short-term path forecasts of intraday electricity prices. Part II – Distributional Deep Neural Networks

Grzegorz Marcjasz[1]
Tomasz Serafin[1]
Rafał Weron[1]

[1] Department of Operations Research and Business Intelligence, Wrocław University of Science and Technology, Poland

WORMS is a joint initiative of the Management Science departments of the Wrocław University of Science and Technology, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland

Trading on short-term path forecasts of intraday electricity prices. Part II – Distributional Deep Neural Networks

Grzegorz Marcjasz[a,1], Tomasz Serafin[a], Rafał Weron[a]

[a]*Department of Operations Research and Business Intelligence, Wrocław University of Science and Technology, 50-370 Wrocław, Poland*

Abstract

We propose a novel electricity price forecasting model tailored to intraday markets with continuous trading. It is based on distributional deep neural networks with Johnson SU distributed outputs. To demonstrate its usefulness, we introduce a realistic trading strategy for the economic evaluation of ensemble forecasts. Our approach takes into account forecast errors in wind generation for four German TSOs and uses the intraday market to resolve imbalances remaining after day-ahead bidding. We argue that the economic evaluation is crucial and provide evidence that the better performing methods in terms of statistical error metrics do not necessarily lead to higher trading profits.

Keywords: Intraday electricity market, Probabilistic forecast, Path forecast, Prediction bands, Trading strategy, Neural networks

1. Introduction

The European power trading landscape is undergoing significant changes as the generation from renewable energy sources (RES), such as wind and solar, continues to grow, accompanied by ongoing market integration and active demand-side management Grossi and Nan (2019); Maciejowska (2020). They make it more difficult to balance the supply and demand sides in the power system, mainly due to high uncertainty regarding the RES generation during the day-ahead (DA) auction. Therefore, we observe the shift towards shorter time horizons in electricity trading. The day-ahead market, which traditionally played a crucial role in electricity trading in Europe, is now slowly losing the market share to the intraday (ID) trading. Between years 2021 and 2022, the volume traded on European intraday markets (operated by EPEX) increased by 9%, while the day-ahead – decreased by 5% (EPEX, 2023).

This gradual change of focus is not yet visible in the electricity price forecasting (EPF) literature. The search of Scopus-indexed[1] articles reveals that only around 3% of EPF articles consider the topic of intraday electricity price forecasting.

Among the existing literature, researchers focus on few distinct topics. Kiesel and Paraschiv (2017) investigate the bidding behavior of German intraday electricity market participants and link the RES generation forecast errors to the electricity price changes. Narajewski and Ziel (2020a) and Marcjasz et al. (2020) focus on forecasting the ID3 index – the most commonly used proxy for the German intraday price (see Section 2). Janke and Steinke (2019) conduct a forecasting study with the focus on the quantiles of the price distribution for the last three hours of trading before the delivery. Linear regression models and an ensemble of neural networks are compared to several naive benchmarks. Narajewski and Ziel (2020b) and Serafin et al. (2022) propose ensemble forecasting methods for the continuous intraday markets, which in case of the latter paper, are used as a basis for a trading strategy which serves as a tool for the economic evaluation of electricity price forecasts. This particular direction is recently gaining attention of researchers and as Hong et al. (2020) and Maciejowska et al. (2023) argue, it is an important aspect of the model evaluation that at the same time is commonly overlooked in the literature.

In this paper, we address the aforementioned existing literature gap and extend the trading strategy proposed by Serafin et al. (2022) with a more realistic (from a perspective of a wind power plant owner) set of assumptions. More precisely, we consider wind generation forecast errors and use the intraday market to cover the imbalance left after the day-ahead bidding. Additionally we argue that the economic assessment of the forecast is the key factor in choosing the optimal approach from the perspective of the decision maker. Moreover, we propose a novel ensemble prediction model based on the well-performing machine learning approach of Marcjasz et al. (2023) and show that – albeit the results of the statistical evaluation are not unanimous – it is the best among tested methods in all trading simulations we performed.

The rest of this paper is structured as follows. In Section 2 we describe datasets used in this study. In Section 3 we provide the description of forecasting models while in Section 4 we introduce the "building blocks" that the models use – from point forecasting methods, through probabilistic and path trajectories to prediction band generation algorithm. In Section 5

*Corresponding author

Email address: grzegorz.marcjasz@pwr.edu.pl (Grzegorz Marcjasz)

[1]the EPF articles were queried using `TITLE-ABS-KEY(''electricity price*'' AND (''forecast*'' OR ''predict*'' ))` query, while the intraday EPF ones — `TITLE-ABS-KEY(''electricity price*'' AND (''forecast*'' OR ''predict*'' ) AND (''intraday'' OR ''intra-day''))`

we introduce trading strategies that are used for the economic evaluation of forecasts. Section 6 demonstrates the results of statistical and economic evaluation and provides a discussion on the applicability of both approaches. Lastly, Section 7 concludes the findings of this paper.

2. Data

2.1. Market description

Unlike the auction-based markets, the German intraday continuous trading does not have a single price for the product (i.e., for the delivery of a set amount of electricity over a given period). Instead, the price depends on the moment of entering the market – and as a result, we are presented with the price trajectory. The trajectory starts at 16:00 on the day preceding the delivery and ends 5 minutes before the delivery begins. The exchange lists three price indices: IDfull, ID1 and ID3, that are computed as a volume-weighted average price of transactions in the whole trading period, last hour before the delivery and last 3 hours before the delivery, respectively. While the indices are informative – they provide an approximation of the price via a single value – they do not present the whole information, especially regarding the trading opportunities.

We use the same dataset as Serafin et al. (2022) to facilitate the comparison. The dataset comprises the transaction data (price, volume and timestamp) for the hourly contracts on the German intraday electricity market covering period from 15.06.2017 to 29.09.2019. The first 364 days serve as an initial calibration window for the point forecasts, followed by three 91-day calibration periods: for the probabilistic forecast estimation based on the point ones (for the LASSO-based method), for the path forecasts and finally for the simultaneous coverage probability (see Section 5.3). This leaves a 200-day out-of-sample test period for the path forecasts. The data split is visualized in Figure 3 in Serafin et al. (2022).

The data contains the raw info for each of the executed transactions (timestamp, volume and price). To make the data better suited for modeing, we use an aggregated view of the market data. From the raw transaction data, we extract volume-weighted average prices (VWAPs) of the 15-minute timeframes that constitute the ID3 index, for a total of 12 subperiods. However, the first subperiod only considers 10 minutes of the data (in the modeling framework, we use information that is available 3 hours before the delivery, we allow 5 minutes for gathering the data and running the models) and the last two subperiods are ignored (as the last 30 minutes of trading is limited – only trades within the control zones are allowed). We therefore use 10 subperiods for evaluation of the strategy.

Having the VWAPs for the 10 subperiods $t_1, \ldots, t_{10}$, we can use them as an approximation of the price trajectory – and as Serafin et al. (2022) state – it also is more realistic for selling larger volumes than the prices of single transactions.

2.2. Exogenous data

Aside from the market data, we also have exogenous series that are used in the model. Firstly, we have German wind power generation data – hourly values describing the forecasted (day-ahead forecast) the generation and the series of actual (observed) values. We assume that the actual data is available with only a small delay (such data is publicly available, e.g. on the ENTSO-E platform). Secondly, we have similar data regarding the forecasted and actual load for Germany. For the depiction of the exogenous data (the day-ahead forecasts) we refer the reader to Figure 4 in Serafin et al. (2022).

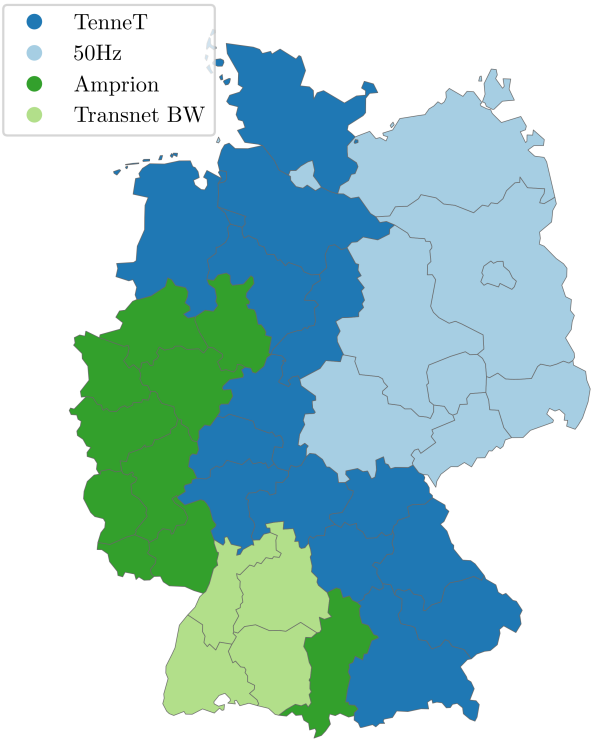


Figure 1: Map of Germany with the approximate geographic division to four zones covered by the service of respective TSOs.

2.3. Wind data for realistic strategy

The wind data described in the previous paragraph correspond to the nation-wide values. However, to better approximate what is the imbalance after the day-ahead bid and the update of the wind generation forecasts, we need to take the location of a power plant (as the wind gusts are not uniform over the whole country). We use the wind generation forecasts from the four German transmission system operators (TSOs): Amprion, 50Hertz, TenneT and Transnet-BW. For each zone, we have a set of two forecasts of the zonal wind generation, the day-ahead one (which is a basis for computing the volume sold on the day-ahead market) and the one closer to delivery (assumed to be equal to the generation), the difference of which needs to be purchased or sold on the intraday market (see Section 5.2.2).

3. Models

This section describes the three models we use for the generation of prediction bands – each comprises of various “building

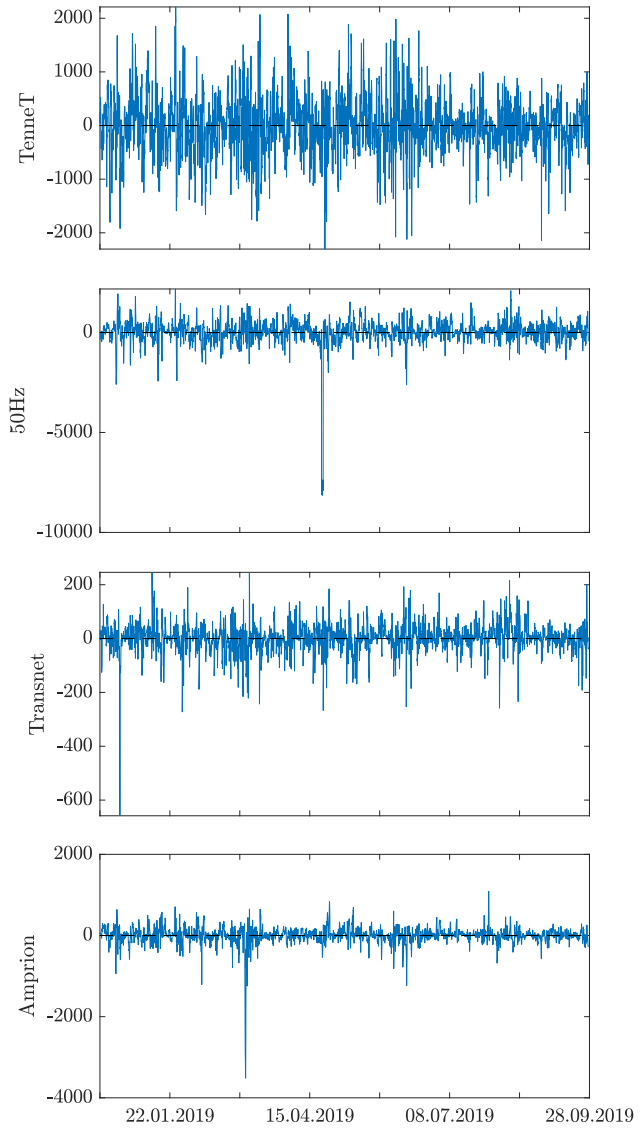


Figure 2: Hourly zonal imbalance plots for four German delivery zones.

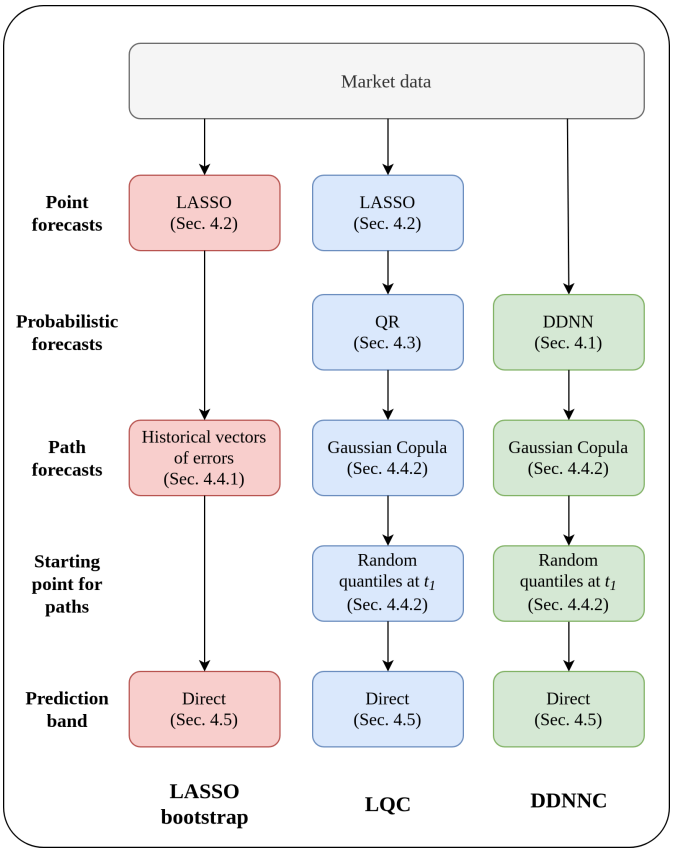


Figure 3: Flowchart presenting the "building blocks" of the forecasting approaches introduced in Section 3, based on the computational techniques described in Section 4.

blocks" (see Figure 3), however all of them use the same approach to obtain the prediction bands from the path forecasts – the Direct method described in Section 4.5. The respective steps are introduced in Section 4.

3.1. The DDNNC approach

The novel approach we introduce combines distributional neural networks and Gaussian copula-modeled temporal dependencies. As described in Section 4.1, the neural network outputs the probability density function directly – there is no intermediate point forecast created in the process. The steps for creating the path forecasts from the probabilistic one, choosing the starting point for the paths and prediction band computation are identical to the LQC method of Serafin et al. (2022). More specifically, first trajectory forecasts are computed, with temporal dependencies between the sub-periods modeled using a Gaussian copula. Next, vectors of innovations are affixed to randomly drawn values from the probabilistic forecast for t_1 sub-period and eventually, the Direct approach is used for computing the prediction bands.

3.2. The LQC approach

The so-called **LQC** approach proposed in Serafin et al. (2022) comprises three main parts: LASSO point forecasting model, quantile regression (to obtain probabilistic forecasts) and – like the DDNNC approach – copula-modeled structure of temporal dependencies. The QR is used to compute 99 percentile forecasts based on the point predictions. The 99 percentiles are linearly interpolated to obtain more granular quantiles, there is also extrapolation to the minimum and maximum prices for the extreme values.

3.3. LASSO bootstrap approach

Lastly, we use the better of the two point forecast-based methods described in Serafin et al. (2022): **LASSO bootstrap**. The approach uses LASSO point predictions as the base for obtaining price paths - it additionally samples vectors of historical point forecast errors to "correct" for the observed temporal dependency. This particular method proved to be an extremely well performing benchmark despite its simplicity.

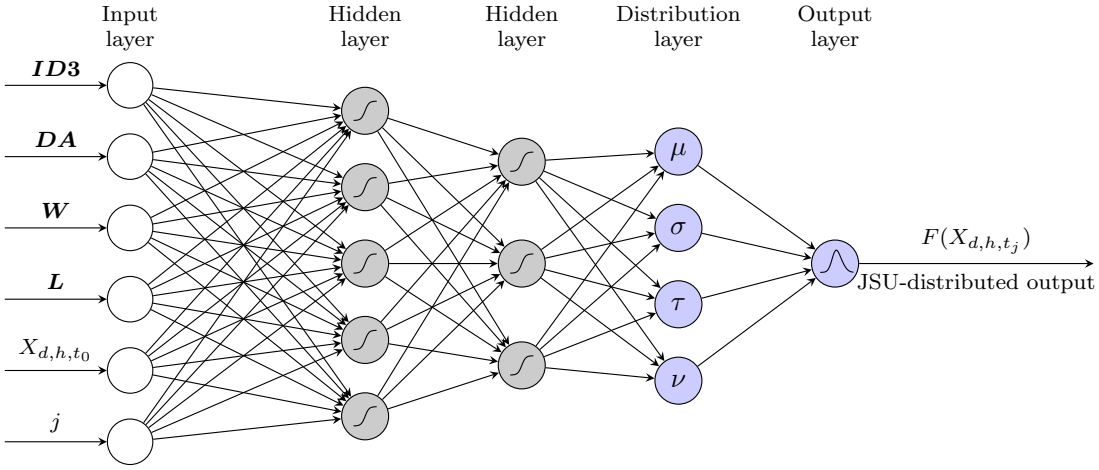


Figure 4: Visualization of the deep distributional neural network (DDNN) structure with the inputs similar to the LASSO defined in Eq. 1, and two hidden layers. The size and activation functions were chosen automatically in the hyper-parameter optimization study. For simplicity inputs in bold represent vectors of respective variables, i.e. $\boldsymbol{ID3}$ and $\boldsymbol{DA}$ represent lagged DA and ID3 values whereas $\boldsymbol{W}$ and $\boldsymbol{L}$ correspond to forecasted and actual values of load and wind generation used in in Eq. 1. Note that the neural network outputs the distribution of price for day d, hour h and j-th time period t_j, $j = 1 : 10$.

4. Methodology

4.1. DDNN

Distributional deep neural networks are feed-forward networks that – compared to their point counterparts – differ only slightly in the structure (see Fig. 4) and are trained to minimize the log-likelihood instead of error metrics such as mean absolute error (Marcjasz et al., 2023). The network used in this study is a deep structure with outputs that create (via a four-part parameter layer corresponding to the four distribution parameters) an output in form of a Johnson SU distribution (Johnson, 1949). The model for predicting the distributions of VWAP for a given j-th subperiod ($j = 1, ..., 12$) for delivery at day d, hour h consists of the following 102 inputs:

1. 21 past ID3 index values (newest avaiable at the time of forecasting)
2. 25 day-ahead prices, ranging from day $d-1$, hour h to day d, hour h
3. the day-ahead forecast of 25 hourly values of wind generation and day-ahead load forecast (day $d-1$, hour h to day d, hour h)
4. the actual wind power production and observed load for the last observed hour (4 hours preceding the delivery) and hour h of day $d-1$
5. last VWA price of the 15-minute interval (period spanning from 3h15m to 3h before the delivery)
6. a multi-valued indicator variable corresponding to the modeled subperiod j

Note, that we do not use dummies corresponding to the weekday or hour of the day in the model in a fashion similar to the day-ahead models, as this information is strongly correlated with the day-ahead prices and load forecasts. The model, however uses a dummy to mark the training samples coming from j-th subperiod. The original formulation of the model proposed by Serafin et al. (2022) did not have this information, as 12 separate models were trained, one for each future horizon (see Section 4.2). Based on a limited numerical study, for this particular deep neural network model it is beneficial to use only one model that has a vastly larger set of training samples. Moreover, unlike for the LASSO model (that follows the original formulation of Serafin et al. (2022)), the input data is not preprocessed in any way (aside from the batch normalization applied in the neural network). The non-linear model structure is expected to fit well to the non-linear patterns in the data (Hill et al., 1994; Jędrzejewski et al., 2022).

The calibration window used in the neural network training was 364 days ($24 \cdot 12 \cdot 364$ samples), 20% of which were randomly left out as an unseen data for the validation. Whenever the forecast error on the validation set did not improve in the last 50 iterations over the whole dataset, the training is assumed to be finished (and the weights from iteration that yields the lowest validation error are restored). The process is called early stopping and is a common practice in the literature (Lago et al., 2018; Yao et al., 2007).

Due to the randomness of the neural network training process, the results of consecutive runs (training processes – the trained network is deterministic) vary – for a more robust performance, it is a standard practice to train multiple neural networks (in this case, using the same data) and treat their combined ouptut as the final outcome of the model (here, horizontal (qAve) averaging was used, following Marcjasz et al. (2023)). The results reported in this paper correspond to the ensemble of 5 identical (w.r.t. the structure and input data) neural networks trained separetely, using the same hyper-parameter set (see paragraph below). Also, Fig. 7 presents the impact and variance of the profit from the trading strategy depending on the size of the ensemble.

The aforementioned hyper-parameter set was chosen in an additional optimization study, in which only the initial calibration data (the first 364 days of the dataset) was used to avoid an *ex-post* optimization. The last 13 weeks (91 days) of that

were allocated as a hyper-parameter validation dataset. The process of hyper-parameter optimization, in simple terms, was an iterative training of neural networks using different hyper-parameter values and evaluating them on the hyper-parameter evaluation dataset. In more detail, for each hyper-parameter set, there were 7 neural networks trained – each was evaluated on only 13 of the 91 validation days. This was done to make the hyper-parameter optimization more robust, as it limits the impact of the randomness on the optimization process. The hyper-parameter sets were chosen using Tree-structured Parzen Estimator implemented in the optuna package for Python – so the consecutive trials were based on the history of values the tested so far. Finally, the network was trained five times independently (using a random starting points for the weights in the model) with the chosen hyper-parameter set – the predictons made using these five trained networks constitute an ensemble. During the hyper-parameter calibration, the following hyper-parameters were determined:

- activation function for both hidden layers, (independently) chosen from sigmoid, relu, elu, tanh, softplus and softmax
- the number of neurons in both hidden layers (independently) – an integer from 16 to 1024
- the initial learning rate for the ADAM algorithm (float ranging from 10^{-7} to 10^{-1})
- dropout application (yes/no) and, if yes, the dropout rate – float from 0.0 to 1.0

4.2. Generating LASSO point forecasts

The LASSO-estimated point forecasts were generated using the same model as in Serafin et al. (2022), with the baseline model for the VWA of hour h on day d is (for the j-th subperiod before the delivery) given by:

$$\begin{aligned} X_{d,h,t_j} = \beta_0 &+ \underbrace{\sum_{i=4}^{24}\beta_{i-3}\mathrm{ID3}_{d,h-i} + \sum_{i=0}^{24}\beta_{22+i}\mathrm{DA}_{d,h-i}}_{\text{past ID3 and past/forward-looking DA prices}} \\ &+ \underbrace{\sum_{i=0}^{24}\beta_{47+i}\widehat{W}_{d,h-i} + \beta_{72}W_{d,h-4} + \beta_{73}W_{d,h-24}}_{\text{wind generation forecasts and past values}} \\ &+ \underbrace{\sum_{i=0}^{24}\beta_{74+i}\widehat{L}_{d,h-i} + \beta_{99}L_{d,h-4} + \beta_{100}L_{d,h-24}}_{\text{load forecasts and past values}} \\ &+ \underbrace{\beta_{101}X_{d,h,t_0}}_{\text{last VWA price}} + \varepsilon_{d,h,t_j}, \end{aligned} \tag{1}$$

where $\mathrm{ID3}_{d,h}$ denotes the value of the ID3 price index for day d and hour h, $\mathrm{DA}_{d,h}$ is the day-ahead price for day d and hour h, $\widehat{W}_{d,h}$ and $W_{d,h}$ are, respectively, the day-ahead predicted and actual wind generation for day d and hour h, $\widehat{L}_{d,h}$ and $L_{d,h}$ are the day-ahead predicted and actual system-wide load for day d and hour h, respectively, and X_{d,h,t_0} is the last known VWA price, i.e., the VWA price of transactions between 3 hours and 15 minutes and 3 hours before the delivery. The last regressor is widely used in the literature on forecasting the ID3 index prices (Marcjasz et al., 2020; Narajewski and Ziel, 2020a). Note, that for the sake of simplicity the notation $\mathrm{ID3}_{d,h-i}$ refers to the ID3 index value i hours before the day d and hour h even though the $h - i$ might be negative.

Note, that the inputs are identical to the ones used in the DDNN methdod (Section 4.1), with the omission of the variable indicating the modeled subperiod j. Instead, 12 separate models are constructed, one for each day d, hour h and subperiod j – although the inputs are exactly the same for each j, the modeled dependencies can be different, as LASSO method automatically limits the impact of less relevant input values, effectively creating 12 (possibly) different models (constructed as subsets of the baseline model) for each day and hour. As in the original paper, the calibration window had length of 364 days.

However, unlike in the DDNN model, the input data series undergo a variance stabilizing transformation, following Serafin et al. (2022) description. Each input series is independently normalized by subtracting the in-sample median and dividing by the in-sample median absolute deviation adjusted by the 75-th percentile of the standard normal distribution. Finally, the *area hyperbolic sine* is applied as the so-called variance stabilizing transformation (Uniejewski et al., 2018). This allows the data to be better suited for the linear model (and normalize the variances of all input series, which is beneficial for the LASSO method).

The model is estimated using the LASSO operator (Tibshirani, 1996), that implicitly (via the regularization of the model's coefficients) selects only the relevant inputs (note, that this results in a set of 24 hourly models that possibly use a different information set). The regularization parameter is in this study chosen automatically from a set of 50 values (that are automatically computed) through a cross validation procedure with 3 folds. The method is implemented in *scikit-learn* library for Python (Pedregosa et al., 2011).

4.3. Computing quantile forecasts using LASSO point estimates

Having the point LASSO forecasts as described in the previous Section, we use quantile regression with 91-day calibration window to generate an approximation of the probabilistic forecast. For each of the percentiles, we estimate its based on the previous forecast values and the actual values. Since we do it separately for each of the 10 sub-periods, we might observe a so-called quantile crossing (i.e., non-monotonic approximation of the percentiles), we prevent it by sorting the percentile estimates, as suggested by Maciejowska and Nowotarski (2016), Serafin et al. (2019) and Serafin et al. (2022).

4.4. Generating path forecasts

The study uses two different schemes of obtaining the path forecast (which are later used to construct the prediction bands in Section 4.5; note however, that in general the path forecast is not required for the prediction band to be generated (see e.g.,

AQL method in Serafin et al. (2022)). One of the approaches is based on the point forecasts (Section 4.4.1) and one uses the probabilistic forecast (Section 4.4.2). Both methods introduce time-dependency in the generated scenarios based on the historical forecasts and realized actual price paths.

4.4.1. Historical point forecast errors

The first method utilizes the point forecasts from the LASSO model as the base for generated scenarios. The time-dependency between prices in consecutive time points is introduced by adding a vector of past errors of the LASSO point model to the forecast for day d in the following way:

$$\widetilde{X}_{d,h,t_j} = \hat{X}_{d,h,t_j} + \varepsilon_{d^*,h,t_j},$$

where $\hat{X}_{d,h,t_j}$ is the point forecast obtained using LASSO, $\varepsilon_{d^*,h,t_j} = \hat{X}_{d^*,h,t_j} - X_{d^*,h,t_j}$ and d^* is a randomly selected day from the past 182 days.

4.4.2. Gaussian Copula

The second approach generates price scenarios based on the forecasted quantiles (from the probabilistic model – LASSO with QRA or DDNN) while the time-dependency is modeled with the Gaussian copula, similarly to Serafin et al. (2022). Using 91-day rolling calibration window, we estimate Σ – the temporal correlation matrix of transformed quantile coverage errors of probabilistic forecasts. Later, we simulate the trajectories by selecting quantiles in consecutive periods that are inter-correlated based on the estimated Σ. For the more detailed description see Pinson et al. (2009), Gneiting et al. (2007) and Janke and Steinke (2020).

4.5. Determining prediction bands from path forecasts

Following Serafin et al. (2022), we construct the prediction bands from the pool of simulated trajectories in order to later use them for the trading strategies (see Section 5). Prediction bands, unlike a set of prediction intervals, take into consideration the temporal dependency of the price forecast evolution in consecutive time points. Each prediction band (upper or lower) is characterized by the *simultaneous coverage probability* (SCP), which is the probability that the whole price trajectory lies below (for upper) or above (for lower) the band. Note that in the strategies we use for the economic evaluation of path forecasts, we make a decision of either selling or purchasing the electricity and therefore, at the time of the decision, only upper (for selling) or lower (for buying) prediction band is taken into consideration.

More formally, the SCP for the upper prediction band B^U_{d,h,t_j} can be written as:

$$\mathbb{P}\left(X_{d,h,t_j} \leq B^U_{d,h,t_j}, \forall_j\right) = \text{SCP},$$

while for the lower B^L_{d,h,t_j}:

$$\mathbb{P}\left(B^L_{d,h,t_j} \leq X_{d,h,t_j}, \forall_j\right) = \text{SCP}.$$

The algorithm we employ for the construction of the prediction bands is similar to the one proposed by Staszewska (2007). Since the simultaneous coverage property requires the price paths to respect the prediction band in each time point, the procedure comes down to rejecting the forecasted trajectories containing extreme points (maximum values for upper and minimum values for lower prediction band) from the whole simulated sample until SCP % of trajectories remain. Then, the prediction band is created by selecting the maximum (or minimum) values of the remaining paths at each consecutive time point. For the reference see both panels of Figure 5 – light-gray dotted lines represent rejected trajectories, dark-gray solid lines the remaining trajectories, while the solid red line depicts the derived prediction band.

4.6. Evaluation of path forecasts

The path forecasts in this paper are evaluated twofold: first based on the statistical measures and later in context of the economic measures. The statistical evaluation is the standard literature approach for the ranking the accuracy of forecasting methodologies (Hyndman and Koehler, 2006; Maciejowska and Nowotarski, 2016; Makridakis et al., 2018). However, the statistical evaluation might not always be straightforward. Lago et al. (2021) note that the relative accuracy of different models might change when we consider various error metrics and suggest to report multiple well-defined error measures, suitable for the type of data (e.g., in case of electricity prices that can have close to 0 or even negative values, percentage errors lead to incorrect conclusions). Therefore, in this paper we use three well-known scoring metrics suitable for the evaluation of path forecasts: Energy Score, Variogram Score and Dawid-Sebastiani Score (Scheuerer and Hamill, 2015). As Scheuerer and Hamill points out, each of these have its shortcomings in sensitivity to certain types of the forecast biases (see Sections 4.6.1–4.6.3).

Moreover, in practice a manager has to make one decision – and multiple sources (error measures) might point to different suggested actions (Kolassa, 2020). Moreover, the optimal choice should be determined by the expectations of the decision-maker (for example maximization of the profits or reduction of the risk). However, the statistical evaluation does not provide the necessary information since there is no clear relation between the error measures and the expected outcome of the decision (such as profit or VaR maximization), making it unclear if the accuracy of better methods (with regards to the statistical error metrics) corresponds in practice to improved financial results.

Hence, there is a need for a more universal evaluation, ideally one that addresses the aforementioned issues, for example a market simulation that uses the forecasts as an automatic decision support system (Janczura and Wójcik, 2022; Kath and Ziel, 2018; Maciejowska et al., 2019; Serafin et al., 2022; Uniejewski, 2023). In this paper, we propose a market simulation approach based on a simple trading strategy to determine whether the best forecast in terms of the statistical measures would be also a top performer in the context of economic evaluation from the perspective of the power producer.

4.6.1. Energy Score

The *energy score* is defined by Gneiting and Raftery (2007):

$$\mathrm{ES}_{d,h} = \frac{1}{M}\sum_{i=1}^{M}\left\|\widetilde{X}^i_{d,h} - X_{d,h}\right\|_2 - \frac{1}{M(M-1)}\sum_{i=1}^{M-1}\sum_{l=i+1}^{M}\left\|\widetilde{X}^i_{d,h} - \widetilde{X}^l_{d,h}\right\|_2, \tag{2}$$

where $\widetilde{X}^i_{d,h} = \left(\widetilde{X}^i_{d,h,t_1}, \ldots, \widetilde{X}^i_{d,h,t_{10}}\right)$ is the i-th path forecast for day d and hour h, $X_{d,h}$ is the corresponding actual VWA price path and M is the number of generated paths, see Section 4.4 for details. The energy score is a strictly proper scoring rule and a useful tool for evaluating forecasts, including ensemble forecasts, as it generalizes the continuous ranked probability score (CRPS; Hersbach (2000)). However, it has been observed that the energy score may lack sensitivity to misspecifications in the correlations between different components (Pinson and Girard, 2012; Pinson and Tastu, 2013).

4.6.2. Dawid-Sebastiani score

The *Dawid-Sebastiani score* – a multivariate scoring rule based on the mean vector and covariance matrix of the predictive distribution (Dawid and Sebastiani (1999)) – is defined by:

$$\mathrm{DSS}_{d,h} = \ln\left(\det\left(\mathbf{S}_{d,h}\right)\right) + \mathbf{K}^T\mathbf{S}^{-1}_{d,h}\mathbf{K} \tag{3}$$

where $\mathbf{K}_{d,h} = (K_{d,h,t_1}, \ldots, K_{d,h,t_{10}})$ is a vector of 10 differences, each taking the form of:

$$K_{d,h,t_j} = X_{d,h,t_j} - \frac{1}{M}\sum_{i=1}^{M}\widetilde{X}^i_{d,h,t_j}$$

and $\mathbf{S}_{d,h}$ is the covariance matrix estimated from the simulated scenarios. This scoring rule corresponds to the logarithmic score for multivariate Gaussian predictive distributions and remains a proper scoring rule for a broader class of probability distributions. However, Scheuerer and Hamill (2015) argue that the score calculation is very sensitive to the small sample size, hence it is not always a good choice for ensemble forecast evaluation (see e.g., Table 2 in Feldmann et al. (2015)). Note, that in case of the forecasting exercise considered in this paper, the ensemble size is large enough for the score to be applicable.

4.6.3. Variogram score

Lastly, we use the variogram score which has been proposed as an alternative proper scoring rule by Scheuerer and Hamill (2015). The *variogram score* of order p (VS-p) is defined by:

$$\mathrm{VS}_{d,h} = \sum_{i=1}^{10}\sum_{j=1}^{10} w_{i,j}\left(\left|X_{d,h,t_i} - X_{d,h,t_j}\right|^p - \frac{1}{M}\sum_{l=1}^{M}\left|\widetilde{X}^l_{d,h,t_i} - \widetilde{X}^l_{d,h,t_j}\right|^p\right)^2, \tag{4}$$

where $w_{i,j} = \frac{1}{100}$. This scoring rule has been shown to be more discriminative in context of misspecifications in the correlations structure of ensemble forecasts than two metrics described earlier. However, the types of biases and misspecifications are unknown in the forecasting task, and different values of the p parameter yield a scoring rule that is sensitive to different types of errors (for details see Scheuerer and Hamill (2015)). Therefore, the optimal value is not known in advance – we use two recommended values ($p \in \{0.5, 1\}$) here, and come to a completely different conclusions between the two. Not knowing the source of the errors, we are unable to discern a better model using the variogram score.

5. Trading strategies

In order to evaluate the path forecasts in terms of the economic results, we use and extend the prediction band-based trading strategy proposed by Serafin et al. (2022). The original approach assumes the position of energy producer that owns intermittent renewable energy sources or manages multiple such sources own by different entities (similarly to Li and Park (2018) or Kath et al. (2020)). It simulates a surplus of 1MWh of electricity sold in the intraday market each hour of the day. Our first extension is assuming that the decision maker, instead of excess generation, faces a deficit of 1MWh of electricity which has to be covered on the short-term market for each hour. This strategy provides a different view on the challenges posed by the renewable generation sources – and the combination of both sides, which is the second extension we propose, allow for a realistic evaluation of the daily operations of RES producer.

The third trading strategy mimics the actual uncertainty of the wind power generation (forecasted day before the delivery) and better relates to the challenges of the day-to-day operations faced by a RES producing company. We use the data from 4 German TSOs (see 1) that contain two wind generation forecasts: day ahead $\widehat{W}^{\mathrm{DA}}_{d,h}$ and intraday $\widehat{W}^{\mathrm{ID}}_{d,h}$ (see Section 2.3). We assume that the energy producer have an installed capacity of roughly $\omega = 1\%$ (for the Transnet-BW zone) or $\omega = 0.1\%$ (for the remaining three zones) of the total wind power capacity in the respective zone. Based on the forecasts, the decision-maker submits offers to sell $\omega\widehat{W}^{\mathrm{DA}}_{d,h}$ MWh of electricity on the day-ahead market and then has to balance his/her position on the intraday market based on the updated value of the generation forecast $\omega\widehat{W}^{\mathrm{ID}}_{d,h}$.

For each hour we compute:

$$\Delta_{d,h} = \omega\widehat{W}^{\mathrm{ID}}_{d,h} - \omega\widehat{W}^{\mathrm{DA}}_{d,h}. \tag{5}$$

$\left|\Delta_{d,h}\right|$ represents the volume that needs to be sold ($\Delta_{d,h} > 0$) or purchased ($\Delta_{d,h} < 0$) during the intraday market continuous trading.

In all cases, we assume that the impact of our trades on the prices on the intraday market is negligible and ignore the transaction costs. The problem can be then summarized as finding the optimal time to enter the market for each individual hourly delivery period.

5.1. Naive strategies

Following Serafin et al. (2022), use three naive strategies that are not based on any generated forecasts. In the first strategy, $\mathrm{Naive}_{\mathrm{first}}$, the market participant always enters the market during the first period t_0. The second strategy, $\mathrm{Naive}_{\mathrm{last}}$, involves

taking the required position in the trading period closest to the delivery, t_{10}. The last ($\text{Naive}_{\text{avg}}$) strategy assumes that the total traded volume is split into 10 evenly-sized transactions throughout all periods $t_1 \ldots, t_{10}$.

5.2. Prediction band-based strategies

Having the prediction band generated based on the path forecasts, we use it as a time-varying price level of the recommended limit order (placed every 15 minutes). More precisely, the points from the prediction band (upper or lower depending on the trade direction) correspond to the prices of the limit orders (buy or sell) placed on the market in the consecutive 15-minute subperiods until one of the orders is filled. If none of the limit orders gets filled, we assume that the electricity is sold at the last VWA price, as in the $\text{Naive}_{\text{last}}$ strategy.

5.2.1. Fixed-volume sell/buy strategies

Serafin et al. (2022) introduces a novel strategy that uses prediction bands generated from forecasted price paths to support the decision-making of the company managing the renewable energy sources. It was assumed that the producer had to sell the excess of the electricity generation over the day-ahead bid, with a fixed volume order of 1MWh placed on the market each hour of each day. However, the stochastic RES generation might force the decision-maker to purchase the electricity instead. In this study, we address that and provide the results of not only always selling the electricity on the intraday market, but also always buying the same amount (since the optimal points of entry are different for both sides of the trade, the problems are similar, but with a different solution; see Fig. 5).

5.2.2. Realistic market simulation

This study, aside from considering separate perspectives of both the buyer and the seller, proposes a new, realistic market simulation, in which we assume the position of the decision-maker in a wind power plant. Each day, the manager offers 100% of the forecasted electricity generation for each hour of the next day (based on the day-ahead generation forecast). However closer to the delivery, a new, more precise forecast is available – and there will be a surplus or a shortage of electricity generated versus the day-ahead offer. Like in the fixed-volume strategies (see Section 5.2.1), the decision-maker needs to therefore balance it on the intraday market and the problem becomes an optimization of the time to enter the market. The main advantage of such an approach is getting rid of the unrealistic assumption that the balancing volume and direction are constant across all hours – here, we implicitly consider the correlation between the change of the wind generation forecast (day-ahead versus closer to the delivery) and the volume and direction of the balancing transaction. See Section 5 and Eqn. (5) for more details.

5.2.3. Ex-ante selection of the simultaneous coverage probability

As described in 4.5, we can derive a prediction band (upper or lower – depending on the trading direction) from a collection of path forecasts. However, we need to first specify SCP (simultaneous coverage probability), and its optimal value will vary – both in time and depending on whether we buy or sell. Following the methodology of Serafin et al. (2022), we leave out a 91-day long rolling calibration window to fit the optimal (most profitable) SCP. The selection is done independently for both the upper and the lower bands, based on the subset of hours for which the respective band type was used for trading (upper for selling and lower for buying). In order to confirm the validity of our approach, we compared the results of the automatic choice of SCP with the ex-post selected values for one of the German zones in Figure 6 – as can be seen, the automatic approach (red surface) yields profit very close to the optimal *ex-post* choices. Therefore, the results section of this paper will concentrate on the auto-SCP methods only.

5.3. Crystal-ball strategies

It is worth noting, that in the context of the proposed strategies there is a maximum and a minimum possible profit that can be extracted from trading activities. Therefore, we, following Serafin et al. (2022), introduce two additional reference strategies: Ref_{max} and Ref_{min}, which always enter the market in the subperiods guaranteeing the best and the worst execution prices, respectively. These strategies can be treated as a baseline for the economic evaluation of other methods and additionally they provide a reference point. Given that the realistic market simulation will have different volumes traded for different zones, we can't compare the raw profits between them. Hence, we define the *fraction of realized trading potential* (FRTP) – a metric allowing for the explicit and qualitative comparison of the results, computed as follows.

$$\text{FRTP}_{\text{method}} = 100\% \cdot \frac{\text{Profit}_{\text{method}} - \text{Ref}_{\text{min}}}{\text{Ref}_{\text{max}} - \text{Ref}_{\text{min}}}, \tag{6}$$

where $\text{Profit}_{\text{method}}$ corresponds to the sum of hourly profits of the trading strategy using the model's forecasts on the 200-day test period.

6. Results

As demonstrated in the literature (Kolassa (2020)), the selection of the "best" model (based on statistical evaluation) heavily relies on the choice of evaluation measure. Consequently, in the subsequent section, we will present the outcomes of both statistical and economic evaluation of the generated path forecasts. This approach aims to provide a comprehensive assessment of the forecasted data, taking into account not only statistical accuracy but also its relation to the economic performance.

6.1. Statistical measures

We evaluate the quality of forecasted price paths using three statistical measures especially suitable for this purpose: Energy Score (Gneiting and Raftery (2007), see Section 4.6.1), variogram score (Scheuerer and Hamill (2015), see Section 4.6.3) and Dawid-Sebastiani score (Dawid and Sebastiani (1999), see Section 4.6.2)). Results, calculated on the last 200 days

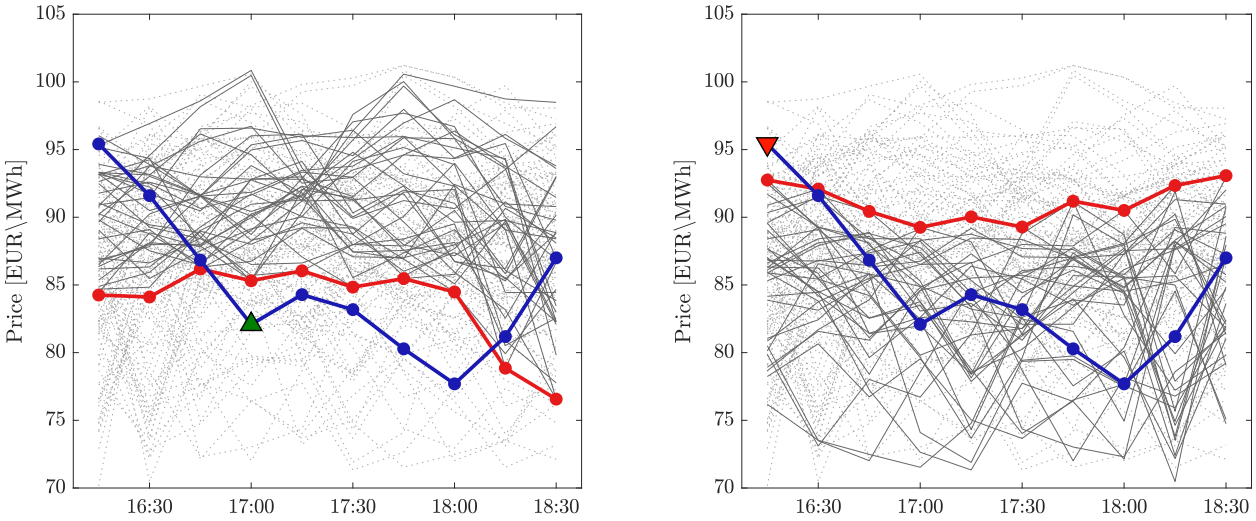


Figure 5: Exemplary trading situations based on lower (*left* panel) and upper (*right* panel) prediction bands with SCP 40%, derived from the same set of simulated trajectories. Green and red triangles mark the moments and prices of filled buy and sell orders, respectively.

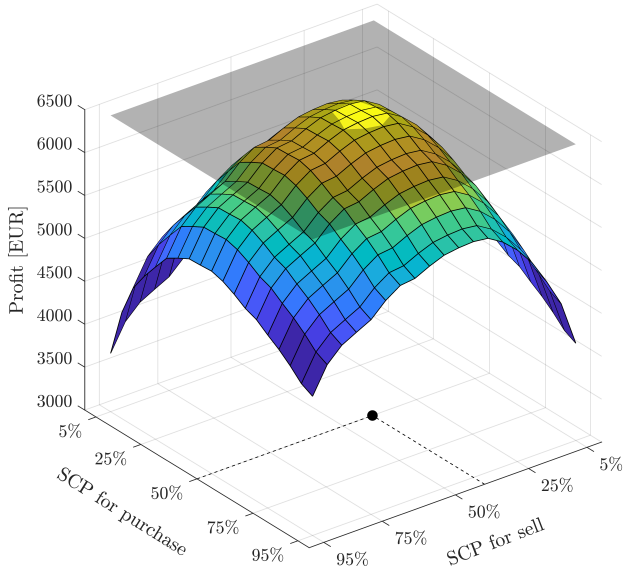


Figure 6: Mesh plot of profits from the realistic trading strategy (Section 5.2.2) for the TenneT zone, based on the trajectories from the DDNNC model. Parallel plane represents the profit from the automated selection of SCP.

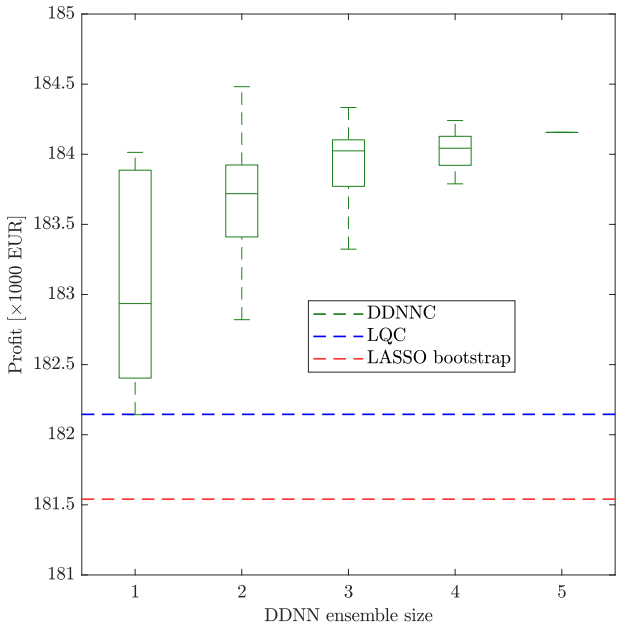


Figure 7: Profits from the fixed-volume selling strategy for different DDNN ensemble sizes. Boxplots were created based on all possible combination of a given number of forecasts from the pool of 5. For reference, solid lines correspond to LQC and LASSO bootstrap profits.

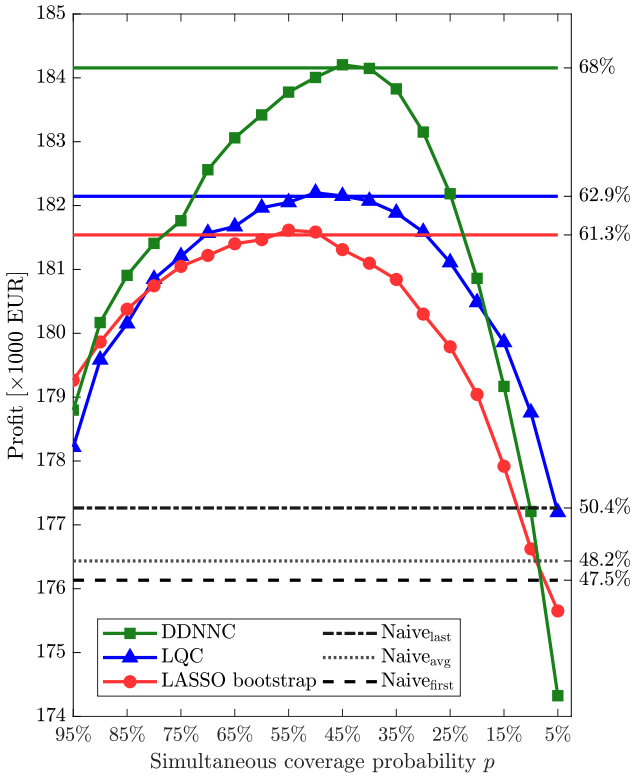


Figure 8: Profits from the fixed-volume selling strategy. On the left y-axis there is a nominal profit, whereas on the right – FRTP (see Section 5.3).

of the out-of-sample period, are presented in Table 1 and divided into peak (hours 8 to 19) and off-peak (remaining twelve hours of the day) periods. It is evident that various evaluation measures identify distinct top-performing models. Interestingly, in over half of the test cases the simplest considered approach (LASSO-bootstrap) exhibits the best performance – for both peak and off-peak test periods. The DDNNC approach outperforms its LASSO-QR-based counterpart with regards to almost every considered measure. The good performance of LASSO-bootstrap most probably stems from a completely different construction than two other models – it uses actual historical price evolution directly instead of estimating the dependency structure. Overall, it is difficult to ultimately pick the best model based on the presented results, the only clear conclusion is that DDNNC outperforms LQC. These results confirm that statistical evaluation of forecasts might not provide universal conclusions. In the next section, we present the results of the economic evaluation of path forecasts using trading strategies from Section 5.

6.2. Trading profits

Firstly, we will discuss the profits of the fixed-volume strategy (to provide a comparison to the original results published by Serafin et al. (2022)). In Figure 8 we can see the profits of a fixed-volume strategy that exclusively involves selling electricity on the market. Note, that this Figure is comparable with Fig. 10 of Serafin et al. (2022) – the Naive and LASSO-based methods are identical. The newly-proposed DDNNC model outperforms all remaining approaches, with the profit of the auto-SCP strategy higher than LQC by ca. 2000 EUR. This amount translates to approximately 5 percentage point improvement over the LQC model in context of the maximum profit achievable from forecasting for this particular strategy. For the sake of clarity, we omit the corresponding plot for the strategy that involves buying electricity – in this particular case, the DDNNC also performs better than other approaches with the profit higher than LQC by ca. 1500 EUR. Interestingly, the LASSO bootstrap performs better than LQC in this case (by almost 1000 EUR). Note, that although the averaging of multiple DDNN probabilistic forecasts is crucial in achieving such results, Figure 7 shows that even in the worst-case scenario (i.e., using only a single realization with the lowest out-of-sample profit) in the fixed-volume sell strategy, DDNN performs comparably to the LQC approach.

Secondly, we will discuss the results of the novel realistic strategy. In Table 2, we present the minimum and maximum possible profits (Ref_{min} and Ref_{max}, respectively; see Section 5.3) alongside the FRTP defined in Eqn. (6), which described the percentage of the maximum possible gain achieved by the respective model. Note, that all Naive strategies, in general achieve FRTP of ca. 50% – further emphasizing the viability of the benchmarking approach (the Naive methods can be compared to a coin toss). Here, the DDNNC outperforms other methods in every case, and LASSO-based approaches trade places for the second result. Moreover, the outperformance is significant – DDNNC is better than the second best approach by 1.6 to 4.4 percentage points, with FRTP ranging from 65.4% to 67.3%.

7. Conclusions

In this paper we addressed an existing literature gap regarding evaluation of the electricity price path forecasts for the German intraday market. Firstly, we have used multiple scoring rules for the statistical evaluation of simulated price trajectories. Secondly, we proposed an extension of the simple trading strategy of Serafin et al. (2022), as a more realistic way for the economic assessment of ensemble forecasts. We make an important argument, that statistical and economical evaluation might lead to contrary conclusions regarding the best-performing-model selection. Moreover, we argue that from the practical perspective of the decision maker, the latter approach provides a clear outlook on the performance of the proposed models.

Additionally, we propose a novel path forecasting methodology that uses deep distributional neural networks of Marcjasz et al. (2023) as a replacement for the point and probabilistic forecasting steps in the LQC approach. This machine learning approach performs the best among the considered methods, significantly improving upon the results of the LQC method with regards to almost every metric.

Our results show that for the German intraday market, three statistical evaluation metrics: energy score, Dawid-Sebastiani score and Variogram Score were unable to discern the best model unanimously – depending on the metric, LASSO bootstrap approach traded the first place with DDNNC. On the other

Table 1: The results of the statistical evaluation of trajectory forecasts. The best results in each category (metric and daytime) are marked with bold.

	ES		DSS		VS-0.5		VS-1	
	peak	off-peak	peak	off-peak	peak	off-peak	peak	off-peak
DDNNC	8.46	6.67	**32.93**	30.55	0.53	0.43	**25.47**	**15.35**
LQC	9.33	7.02	33.47	29.81	0.58	0.45	37.05	16.42
LASSO-bootstrap	**6.21**	**5.17**	42.20	**29.71**	**0.50**	**0.42**	37.72	28.04

Table 2: Profits of automated strategies for different zones. The bold results are the best in each row. The results from each model represents the FRTP (see Section 5.3)

	Ref_{min} [EUR]	Ref_{max} [EUR]	**DDNNC** [%]	**LQC** [%]	**LASSOb** [%]	$\textbf{Naive}_{\text{first}}$ [%]	$\textbf{Naive}_{\text{last}}$ [%]	$\textbf{Naive}_{\text{avg}}$ [%]
TenneT	-2110	10685	**66.9**	62.3	63.8	49.9	49.7	50.2
50Hz	-7824	4906	**67.3**	63.3	61.6	48.1	51.4	49.8
Transnet	-9834	1952	**65.4**	62.2	63.8	49.3	50.1	49.6
Amprion	-3593	200	**65.6**	61.2	60.5	53.6	45.4	50.7

hand, a market simulation (in both the simpler and the more realistic form) always favored DDNNC in our testing – this results holds both for the whole Germany and also the four zones that we used for the evaluation. This implicates that the DDNNC model outperforms the other approaches in the context of easily quantifiable (and universal) economic measures.

Moreover, we further justify the attractiveness of the economic evaluation framework of Serafin et al. (2022) and its applicability to more realistic trading simulations. It provides a well-defined measure of the potential economic impact of forecast quality improvement (as we do know the minimum and maximum possible profits) and all naive methods are comparable to a coin toss – they achieve ca. 50% of the FRTP.

Acknowledgments

This work was partially supported by the Ministry of Science and Higher Education (MNiSW, Poland) through Diamond Grants No. 0009/DIA/2020/49 (to T.S.) and no. 0219/DIA/2019/48 (to G.M.) and the National Science Center (NCN, Poland) through grant No. 2018/30/A/HS4/00444 (to R.W.).

Author contributions

T.S., R.W. – Conceptualization; G.M., T.S. – Investigation; G.M., T.S. – Software; R.W. – Validation; T.S. – Writing, original draft; R.W. – Writing, review & editing.

References

Dawid, A.P., Sebastiani, P., 1999. Coherent dispersion criteria for optimal experimental design. Annals of Statistics , 65–81.

ENTSO-E, 2022. Entso-e transparency platform. `https://transparency.entsoe.eu`. Accessed: 2023-02-27.

EPEX, 2023. Annual report 2022. `https://www.eex.com/fileadmin/Global/News/Group/News/20230124_EEX_Group_Annual_Volume_Report.pdf`. Date accessed: 03.08.2023.

Feldmann, K., Scheuerer, M., Thorarinsdottir, T.L., 2015. Spatial postprocessing of ensemble forecasts for temperature using nonhomogeneous gaussian regression. Monthly Weather Review 143, 955–971.

Gneiting, T., Balabdaoui, F., Raftery, A., 2007. Probabilistic forecasts, calibration and sharpness. Journal of the Royal Statistical Society B 69, 243–268.

Gneiting, T., Raftery, A., 2007. Strictly proper scoring rules, prediction, and estimation. Journal of the American Statistical Association 102, 359–378.

Grossi, L., Nan, F., 2019. Robust forecasting of electricity prices: Simulations, models and the impact of renewable sources. Technological Forecasting and Social Change 141, 305–318.

Hersbach, H., 2000. Decomposition of the continuous ranked probability score for ensemble prediction systems. Weather and Forecasting 15, 559–570.

Hill, T., Marquez, L., O'Connor, M., Remus, W., 1994. Artificial neural network models for forecasting and decision making. International Journal of Forecasting 10, 5–15.

Hong, T., Pinson, P., Wang, Y., Weron, R., Yang, D., Zareipour, H., 2020. Energy forecasting: A review and outlook. IEEE Open Access Journal of Power and Energy 7, 376–388.

Hyndman, R.J., Koehler, A.B., 2006. Another look at measures of forecast accuracy. International journal of forecasting 22, 679–688.

Janczura, J., Wójcik, E., 2022. Dynamic short-term risk management strategies for the choice of electricity market based on probabilistic forecasts of profit and risk measures. the german and the polish market case study. Energy Economics 110, 106015.

Janke, T., Steinke, F., 2019. Forecasting the price distribution of continuous intraday electricity trading. Energies 12, 4262.

Janke, T., Steinke, F., 2020. Probabilistic multivariate electricity price forecasting using implicit generative ensemble post-processing, in: Proceedings of the International Conference on Probabilistic Methods Applied to Power Systems – PMAPS 2020, p. 9183687.

Jędrzejewski, A., Lago, J., Marcjasz, G., Weron, R., 2022. Electricity price forecasting: The dawn of machine learning. IEEE Power and Energy Magazine 20, 24–31.

Johnson, N.L., 1949. Systems of frequency curves generated by methods of translation. Biometrika 36, 149–176.

Kath, C., Nitka, W., Serafin, T., Weron, T., Zaleski, P., Weron, R., 2020. Balancing generation from renewable energy sources: Profitability of an energy trader. Energies 13, 205.

Kath, C., Ziel, F., 2018. The value of forecasts: Quantifying the economic gains of accurate quarter-hourly electricity price forecasts. Energy Economics 76, 411–423.

Kiesel, R., Paraschiv, F., 2017. Econometric analysis of 15-minute intraday electricity prices. Energy Economics 64, 77–90.

Kolassa, S., 2020. Why the "best" point forecast depends on the error or accuracy measure. International Journal of Forecasting 36, 208–211.

Lago, J., De Ridder, F., De Schutter, B., 2018. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. Applied Energy 221, 386–405.

Lago, J., Marcjasz, G., De Schutter, B., Weron, R., 2021. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. Applied Energy 293, 116983.

Li, S., Park, C.S., 2018. Wind power bidding strategy in the short-term elec-

tricity market. Energy Economics 75, 336–344.
Maciejowska, K., 2020. Assessing the impact of renewable energy sources on the electricity price level and variability – a quantile regression approach. Energy Economics 85, 104532.
Maciejowska, K., Nitka, W., Weron, T., 2019. Day-ahead vs. intraday – forecasting the price spread to maximize economic benefits. Energies 12, 631.
Maciejowska, K., Nowotarski, J., 2016. A hybrid model for GEFCom2014 probabilistic electricity price forecasting. International Journal of Forecasting 32, 1051–1056.
Maciejowska, K., Uniejewski, B., Weron, R., 2023. Forecasting electricity prices. URL: https://oxfordre.com/economics/view/10.1093/acrefore/9780190625979.001.0001/acrefore-9780190625979-e-667, doi:10.1093/acrefore/9780190625979.013.667.
Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2018. The m4 competition: Results, findings, conclusion and way forward. International Journal of Forecasting 34, 802–808.
Marcjasz, G., Narajewski, M., Weron, R., Ziel, F., 2023. Distributional neural networks for electricity price forecasting. Energy Economics , 106843URL: https://www.sciencedirect.com/science/article/pii/S0140988323003419, doi:https://doi.org/10.1016/j.eneco.2023.106843.
Marcjasz, G., Uniejewski, B., Weron, R., 2020. Beating the naïve – combining LASSO with naïve intraday electricity price forecasts. Energies 13, 1667.
Narajewski, M., Ziel, F., 2020a. Econometric modelling and forecasting of intraday electricity prices. Journal of Commodity Markets 19, 100107.
Narajewski, M., Ziel, F., 2020b. Ensemble forecasting for intraday electricity prices: Simulating trajectories. Applied Energy 279, 115801.
Pedregosa, F., Varoquaux, G., Gramfort, A., et al., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.
Pinson, P., Girard, R., 2012. Evaluating the quality of scenarios of short-term wind power generation. Applied Energy 96, 12–20.
Pinson, P., Madsen, H., Nielsen, H.A., Papaefthymiou, G., Klöckl, B., 2009. From probabilistic forecasts to statistical scenarios of short-term wind power production. Wind Energy 12, 51–62.
Pinson, P., Tastu, J., 2013. Discrimination ability of the energy score. DTU Informatics.
Scheuerer, M., Hamill, T.M., 2015. Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities. Monthly Weather Review 143, 1321–1334.
Serafin, T., Marcjasz, G., Weron, R., 2022. Trading on short-term path forecasts of intraday electricity prices. Energy Economics 112, 106125.
Serafin, T., Uniejewski, B., Weron, R., 2019. Averaging predictive distributions across calibration windows for day-ahead electricity price forecasting. Energies 12, 256.
Staszewska, A., 2007. Representing uncertainty about response paths: The use of heuristic optimisation methods. Computational Statistics & Data Analysis 52, 121–132.
Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society B 58, 267–288.
Uniejewski, B., 2023. Smoothing quantile regression averaging: A new approach to probabilistic forecasting of electricity prices arXiv:2302.00411.
Uniejewski, B., Weron, R., Ziel, F., 2018. Variance stabilizing transformations for electricity spot price forecasting. IEEE Transactions on Power Systems 33, 2219–2229.
Yao, Y., Rosasco, L., Caponnetto, A., 2007. On early stopping in gradient descent learning. Constructive Approximation 26, 289–315.