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
0169-2070/

(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.

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

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.

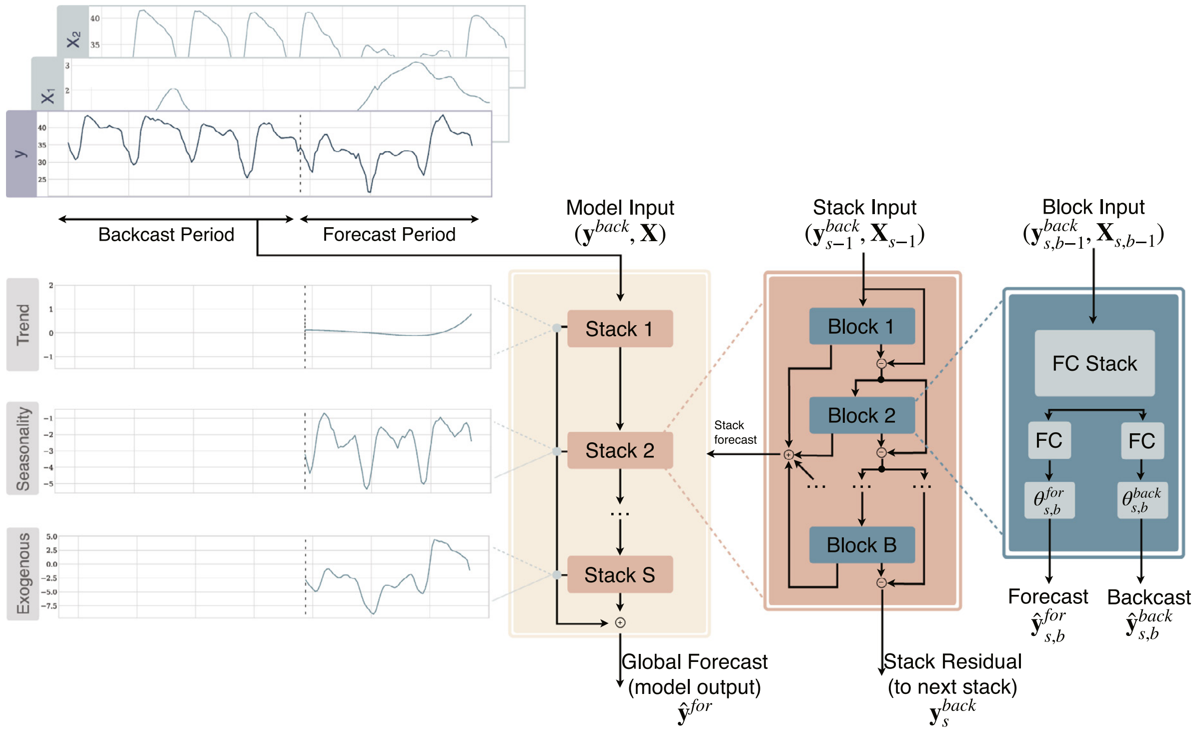


Fig. 1. The building blocks of NBEATSx are structured as a system of multilayer fully connected networks with ReLU-based nonlinearities. Blocks overlap using the doubly residual stacking principle for the backcast $\hat{\mathbf{y}}^{back}_{s,b}$ and forecast $\hat{\mathbf{y}}^{for}_{s,b}$ outputs of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

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

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

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

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:

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

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).

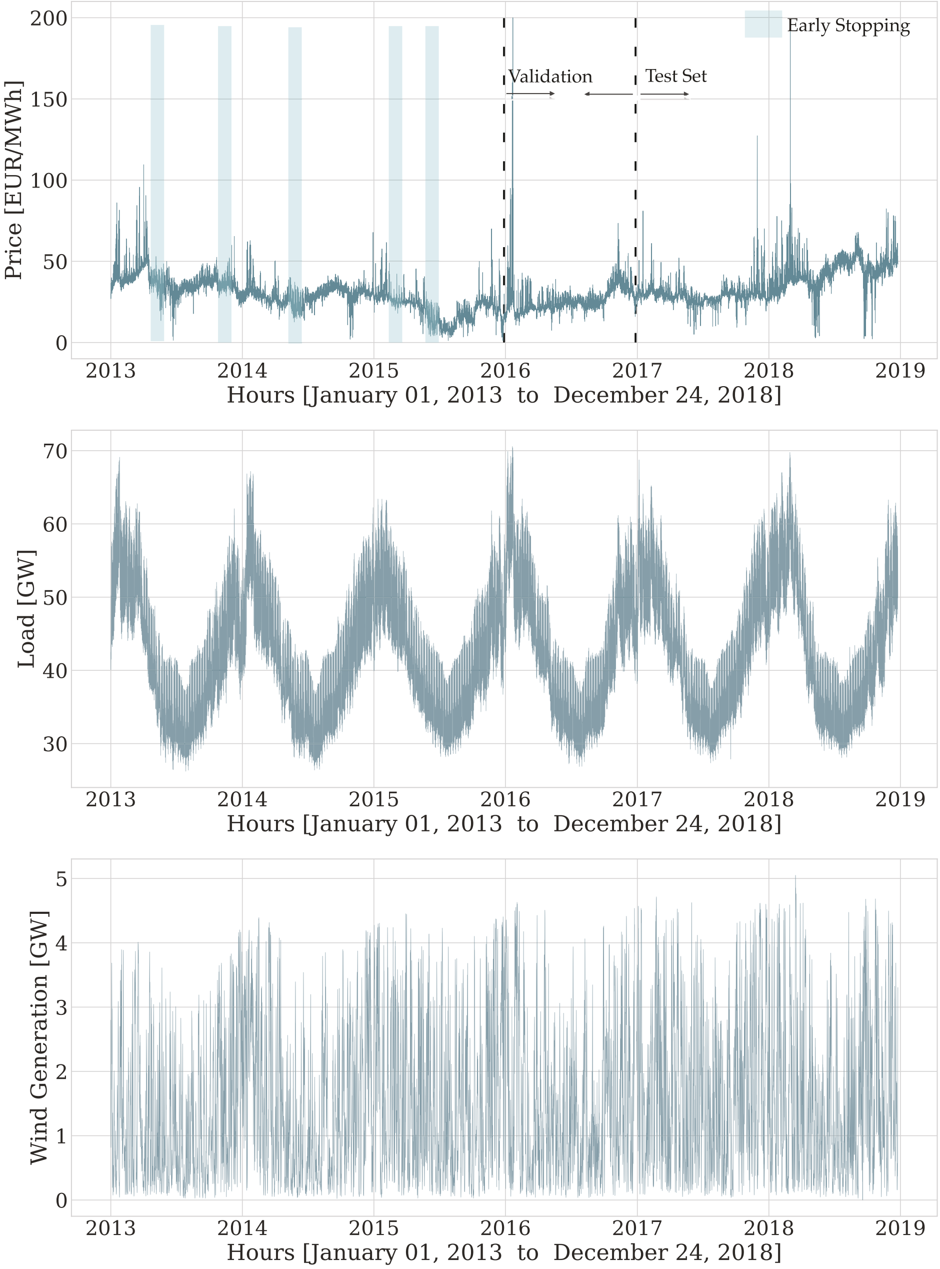


Fig. 2. The top panel shows the day-ahead electricity price time series for the Nord Pool (NP) market. The second and third panels show the day-ahead forecast for the system load and wind generation. The training data are composed of the first four years of each dataset. The validation set is the year 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:

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

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

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

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

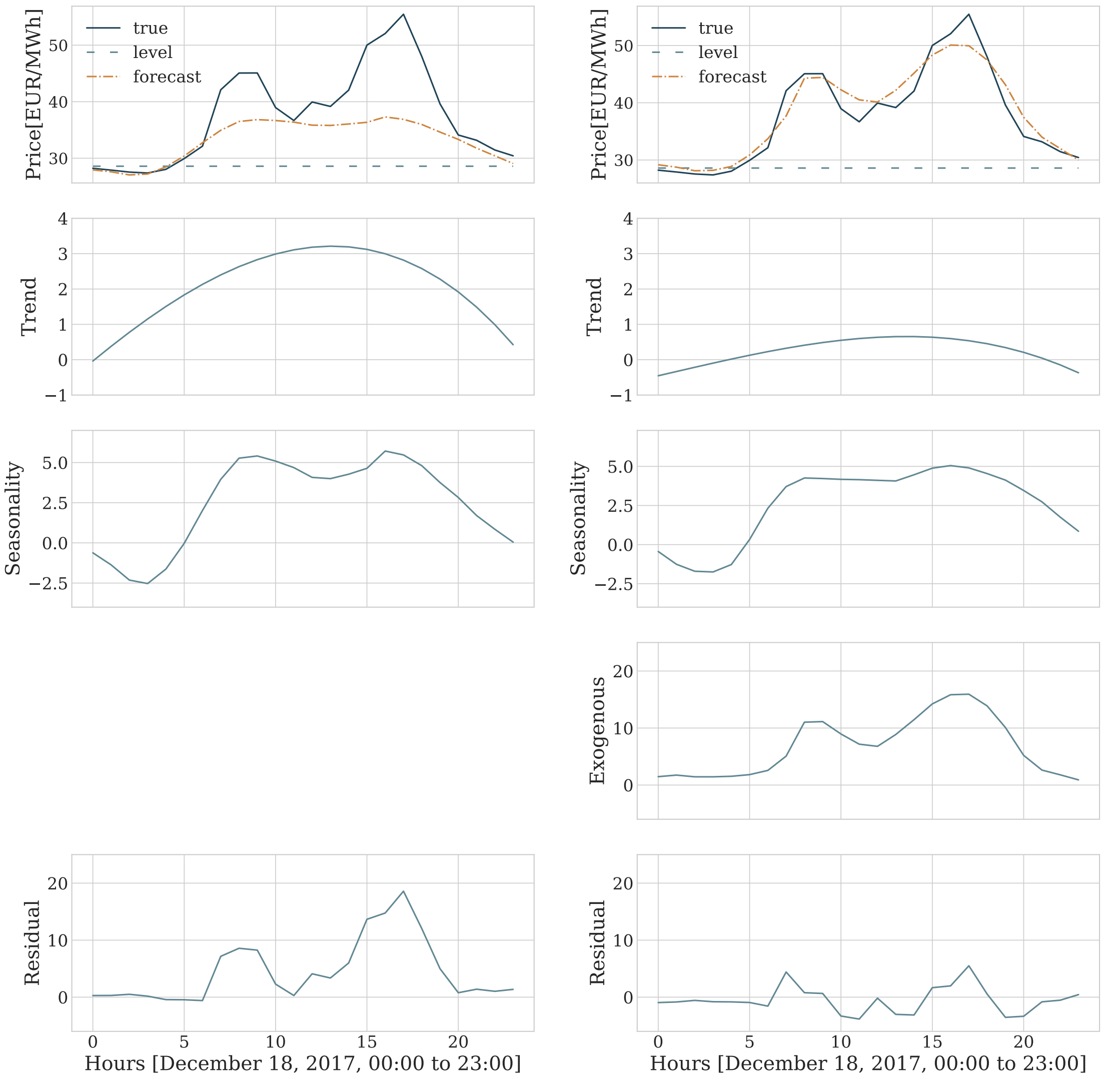
where $y_{d,h}$ and $\hat{y}_{d,h}$ are the actual value and the forecast of the time series at day d and hour h for our experiments given the two years of each test set $N_d = 728$.

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.



(a) NBEATS

(b) NBEATSx

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.	$N_h \in \{50, \dots, 500\}$
Exogenous Temp. convolution filter size (Equation 9)	$\{2, \dots, 10\}$
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 **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).

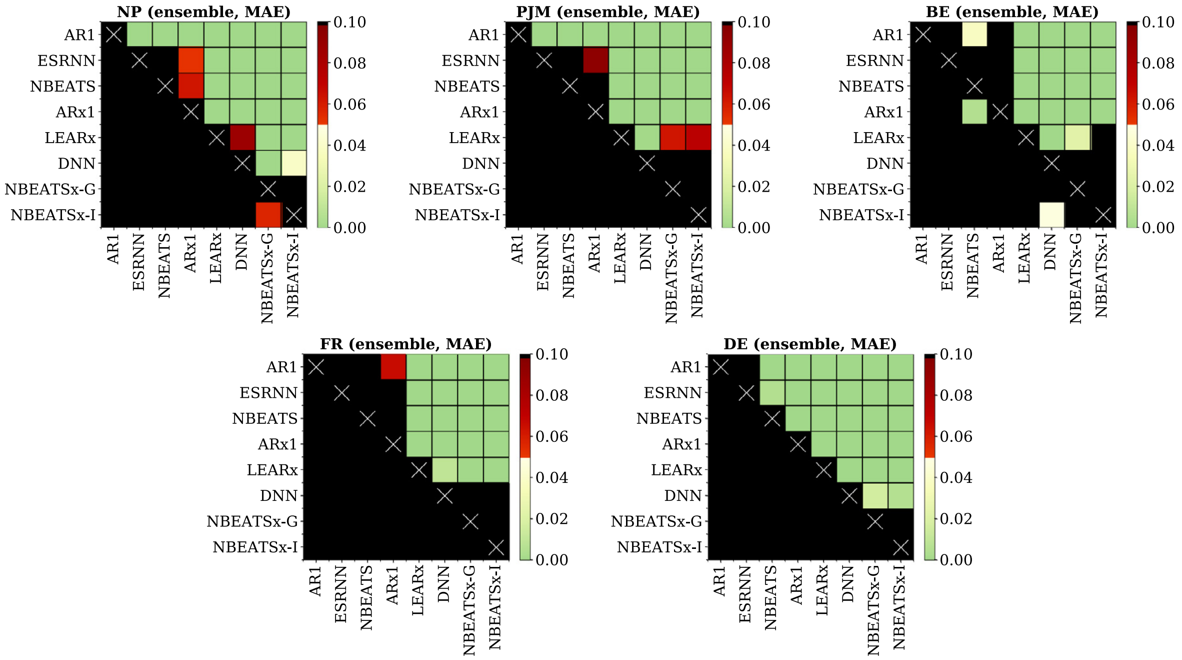


Fig. 4. Results of the Giacomini–White test for the day-ahead predictions with the mean absolute error (MAE) applied to pairs of the ensembled models on the five electricity markets datasets. Each grid represents one market. Each colored cell in a grid is plotted black, unless the predictions of the 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.

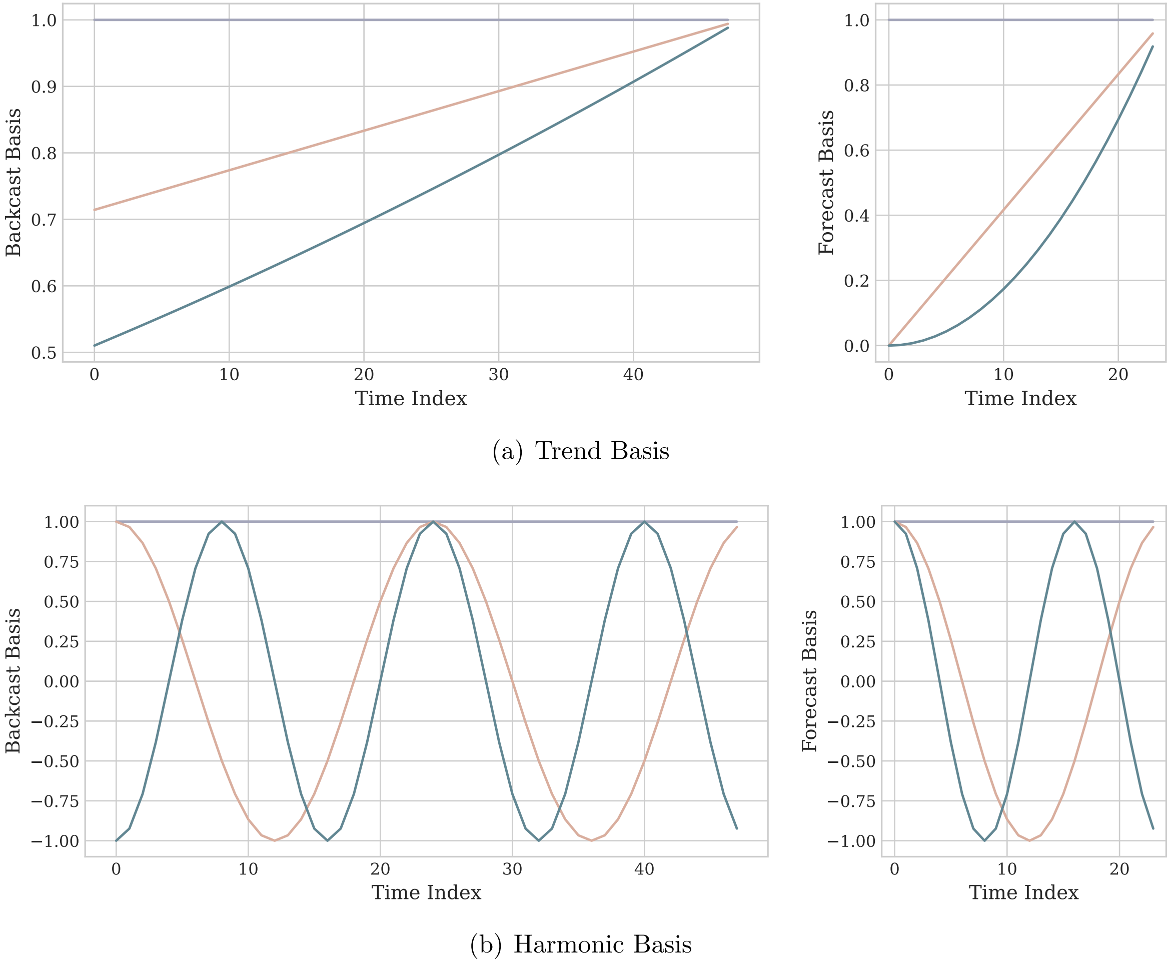
The training time of the *recalibration phase* of NBEATSx remains efficient, as it still trains in 75 and 81 s, increasing by 30 s on the relatively simple DNN. The computational time of the prediction remains within milliseconds. Finally the *hyperparameter optimization* scales linearly with respect 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.



(a) Trend Basis

(b) Harmonic Basis

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.).

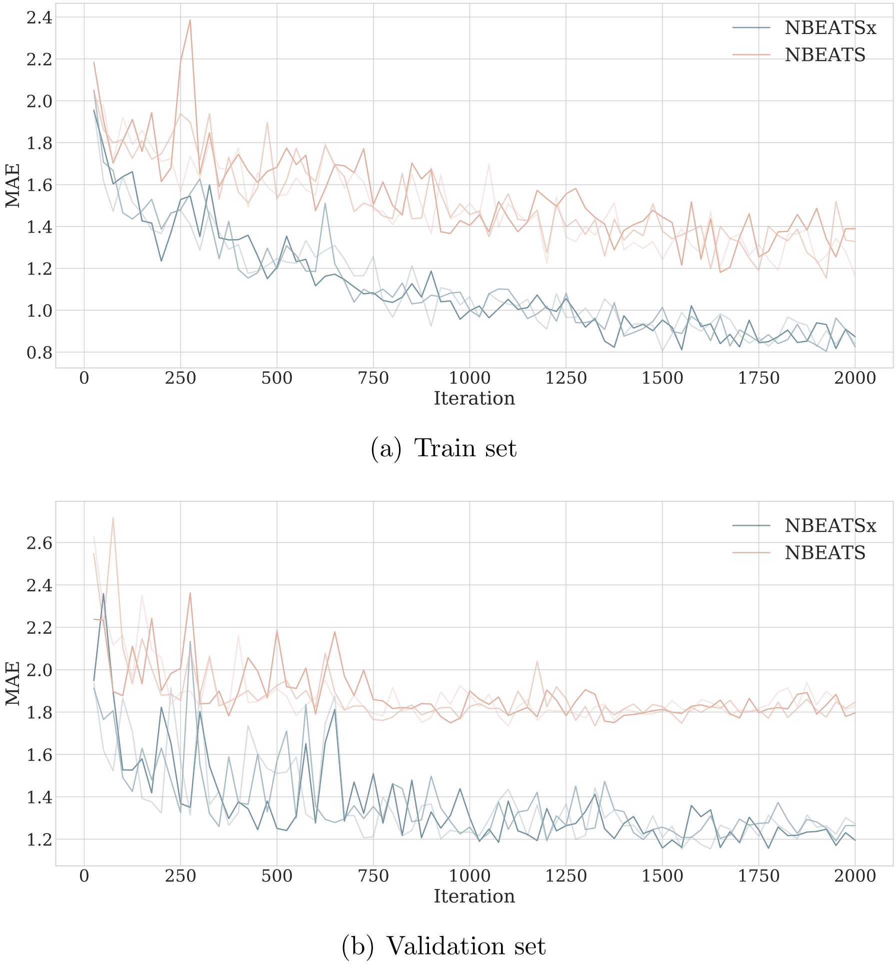
	LEARx	DNN	NBEATSx-G	NBEATSx-I
Recalibration	18.57	50.65	75.02	81.61
Prediction	0.0032	0.0041	0.0048	0.0054

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).

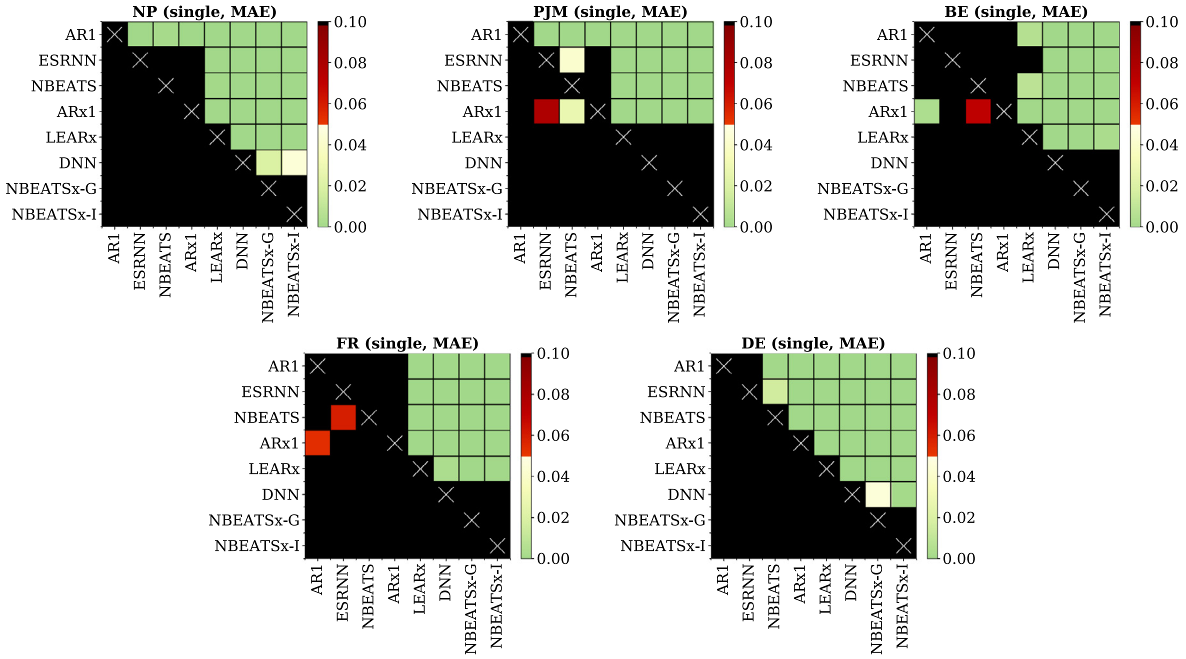


Fig. A.3. Results of the Giacomini–White test for the day-ahead predictions with the mean absolute error (MAE) applied to pairs of the single models on the five electricity markets datasets. Each grid represents one market. Each colored cell in a grid is plotted black, unless the predictions of the model 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.