

# Short-term electricity load forecasting with special days: an analysis on parametric and non-parametric methods

Esra Erişen<sup>1</sup> · Cem Iyigun<sup>2</sup> · Fehmi Tanrısever<sup>3</sup>

© Springer Science+Business Media, LLC, part of Springer Nature 2017

**Abstract** Accurately forecasting electricity demand is a key business competency for firms in deregulated electricity markets. Market participants can reap significant financial benefits by improving their electricity load forecasts. Electricity load exhibits a complex time-series structure with nonlinear relationships among the variables. Hence, models with higher capabilities to capture such nonlinear relationships need to be developed and tested. In this paper, we present a parametric and a nonparametric method for short-term load forecasting, and compare the performances of these models for lead times ranging from 1 h to 1 week. In particular, we consider a modified version of the Holt-Winters double seasonal exponential smoothing (m-HWT) model and a nonlinear autoregressive with exogenous inputs (NARX) neural network model. Using hourly load data from the Dutch electricity grid, we carry out an extensive empirical study for five Dutch provinces. Our results indicate that NARX clearly outperforms m-HWT in 1-h-ahead forecasting. Additionally, our modification to HWT leads to a significant improvement in model accuracy especially for special days. Despite its simplicity, m-HWT outperforms NARX for 6- and 12-h-ahead forecasts in general; however, NARX performs better in 24-h-, 48-h- and 1-week-ahead forecasting. In addition, NARX provides drastically lower maximum errors compared to m-HWT, and also clearly outperforms m-HWT in forecasting for short holidays.

**Keywords** Short-term electricity load · Exponential smoothing · Neural networks · NARX · HWT

---

✉ Cem Iyigun  
iyigun@metu.edu.tr

<sup>1</sup> The Dow Chemical Company, Istanbul, Turkey

<sup>2</sup> Industrial Engineering Department, Middle East Technical University, Ankara, Turkey

<sup>3</sup> School of Business Administration, Bilkent University, Ankara, Turkey

# 1 Introduction

Generating accurate long- and short-term consumption forecasts is essential to managing power systems efficiently (Taylor 2003, 2012). Short-term forecasts refer to forecasts from one minute ahead up to several weeks ahead and are generally used for planning daily operations such as clearing electricity transactions, scheduling generation capacity and managing load flows (Kyriakides and Polycarpou 2007). Long-term forecasts usually refer to forecasts longer than one year ahead, which are typically used for capital budgeting decisions such as investing in new generation and transmission capacity (Tanrisever et al. 2013).

Since the deregulation of European electricity markets in the 1990s, the number of market participants has drastically increased, introducing competition to the market. Consequently, short-term electricity load forecasting has become a major issue for planning electricity transactions in real time (Taylor 2010a). Efficiently scheduling electricity transactions is crucial to providing an economical and reliable supply of energy, and is a key business competency of firms in electricity markets (Bianco et al. 2010).

Short-term electricity demand includes daily and weekly cycles. Multiple factors, such as the season and the time of the day, have complex and nonlinear effects on electricity load (Chen et al. 2001). These multiple seasonality and complex nonlinear relationships make it difficult to model electricity load with traditional regression models. In addition, consumer behavior significantly diverges from the regular pattern on certain days such as holidays, national days, and the days close to these special days. Therefore, we consider neural network modeling as a strong candidate for capturing the complex relationships between the input and output variables. However, because of the complexity of neural network modeling (Lee and Tong 2012) we question whether it is appropriate to use it for short-term load forecasting. As noted by Hippert et al. (2001), more research on the effectiveness of artificial neural networks (ANNs) for short-term load forecasting is needed.

In this paper, we develop and compare a neural-network-based method and a relatively simple time-series-based method for short-term electricity load forecasting. Regarding the former approach, we present a nonlinear autoregressive with exogenous inputs (NARX) neural network. To the best of our knowledge, this is one of the first papers providing a comprehensive analysis of NARX for short-term load forecasting with special days. As a computational-intelligence-based method, we expect NARX to be able to capture complex relationships between the input and output information through its network structure. The latter, relatively simpler, approach is based on the Holt-Winters exponential smoothing method (HWT), which incorporates an autoregressive error component and enables accommodation of both intraweek and intraday cycles as a parametric model. In this model, we propose a modified version of HWT method considering the special days by including a correction factor for calendar days to improve its performance for special days. We expect this modification (m-HWT) to enable the model to learn from its errors to increase its capability of estimating demand on special days.

We carry out an empirical analysis by using load data from five Dutch provinces from 1 January 2008 to 30 November 2012. The results indicate that the modification significantly improves the performance of the HWT method on special days, although NARX still performs better on these days, outperforming m-HWT in 1-h-, 1-day-, 2-day-, and 1-week-ahead forecasts. Despite its simple structure, m-HWT is superior for forecasting lead times that are not multiples of 24 h (e.g. 6 and 12 h). In terms of maximum error, however, NARX outperforms m-HWT predominantly.

The rest of the paper organized as follows: Sect. 2 presents a brief review of the literature on short-term electricity load forecasting and clarifies the contribution of the study. Sections 3 and 4 detail our modification of the Holt-Winters method (m-HWT) and explain our NARX model, respectively. Section 5 presents our empirical results and compares the performance of the two forecasting methods. In Sect. 6, we summarize our results.

## 2 Literature review and contribution

The methods used for electricity load forecasting vary from the simplest conventional models to complex neural network and fuzzy logic models. Hahn et al. (2009) classified forecasting methods into two main categories: (1) classical time-series- and regression-based methods and (2) artificial- and computational-intelligence-based methods. Hybrids of these two categories constitute a third class of methods.

Regression models are commonly used in electricity demand forecasting because of their ability to relate external variables to electricity load (Hahn et al. 2009). In addition to calendar variables, there are numerous other external factors, such as meteorological, social, and economic variables that affect electricity load. Relating these variables to electricity demand is extremely important for generating mid- and long-term load forecasts (Bianco et al. 2009); however, their effects are usually negligible in the short-term (Taylor et al. 2008).

Time-series models provide another common stream of approach for electricity load forecasting. Cancelo et al. (2008) used autoregressive moving average (ARMA) models, a simple form of the time-series approach, by dividing electricity data into its components for generating daily and hourly forecasts for multiple days ahead. Soares and Medeiros (2005) included seasonality of electricity load series in their model through a two-level seasonal autoregressive (AR) model for short-term load forecasting. Hagan and Behr (1987) argued that a simple polynomial regression analysis combined with a Box and Jenkins transfer function model can result in more accurate forecasts. Taylor (2003) modified the Holt-Winters exponential smoothing model to accommodate for the seasonality of electricity loads and generated 1-day-ahead forecasts with half-hour forecasting intervals. He found that the modified approach outperforms both the traditional Holt-Winters method and the multiplicative double seasonal autoregressive integrated moving average (ARIMA) model. Taylor also favored time-series methods in Taylor (2012) where different alterations of exponential smoothing methods are compared to generate short-term electricity load forecasts.

Recently, computational-intelligence-based models have received significant attention in the literature for electricity load forecasting. Al-Saba and El-Amin (1999) developed an ANN model for peak-load forecasting and compared the results to AR models using data from a Saudi Arabian utility company. They showed that ANNs provide accurate results for long-term electricity load forecasting. Connor et al. (1994) was one of the first studies to consider Nonlinear Autoregressive (NAR) neural network models for electricity load forecasting. They compared a NAR model to a recurrent NAR moving average and to a feed-forward NAR model utilizing synthetic data on the Puget Power Electric Demand time series. They emphasized the importance of input configuration while presenting the superior performance of the recurrent networks. Others followed by investigating NARX and NAR moving average with exogenous variables (NARMAX) methods in short-term load forecasting (Czernichow et al. 1995). That paper was mostly concerned with constructing a scheme for the moving average (MA) part of the method. Espinoza et al. (2007) carried out a Kernel-based NARX model identification study for lead times of 1 and 24 h. Using electricity load data from student

apartments, Varghese and Ashok (2012) compared the performances of a feed-forward back-propagation neural network, a NARX network and a radial basis function model.

Several papers have compared various computational-intelligence models to more conventional methods (Taylor et al. 2006; Varghese and Ashok 2012); however, the only study that has compared NARX to a simpler method is Elias et al. (2011) which set a linear regression model as a benchmark for evaluating NARX's performance in generating daily forecasts. In their NARX model, the authors included weather variables, holidays, and weekly and monthly seasonalities as exogenous variables. However, unlike our work, they did not distinguish between different special days, nor did they model the days close to special days.

Neural networks can approximate a large class of functions with a high degree of accuracy but they are often criticized due to their complex black box structure. However smoothing models are easier to implement and understand. In this paper, we compare a NARX network model with a modified HWT model to investigate whether NARX will improve forecasting accuracies and whether the improvement is worth the complexity. Our contribution to the literature with this study can be summarized as:

- Modifying HWT for special days for short-term load forecasting** We develop a modified version of Taylor's Holt-Winters exponential smoothing method (m-HWT) to consider the impact of special days in electricity load forecasting. The proposed model results in a dramatic improvement in forecasting accuracy on special days. In particular, for special days, our m-HWT improves the forecasting performance of the traditional HWT models Taylor's original model on special days by between 16 and 30%. We tested the performance of m-HWT over five datasets for six forecasting lead times.
- Positioning NARX for short-term load forecasting with the inclusion of special days** In the literature, computational-intelligence-based methods are usually used for long-term forecasting. Only a few papers have examined the effectiveness of NARX models for short-term load forecasting, and none of them applied NARX to more than a few datasets or a few different lead times. In that respect, the current study provides a comprehensive analysis of the NARX method for short-term electricity load forecasting by applying it to five datasets and generating forecasts for six lead times. In addition, unlike the existing literature, we explicitly include special days into the NARX model, which leads to a significant improvement.
- Comparing conventional and artificial intelligence methods** The literature's view on complex methods for short-term load forecasting is mixed. While Taylor (2003) finds that simple time-series-based methods are usually sufficient for short-term load forecasting, Kim (2013) argues that more advanced methods are needed to capture the complex nature of demand dynamics. In this paper, through a comprehensive empirical study, we observe that complex neural network models can significantly outperform time-series-based simple methods for certain forecasting lead times. In particular, we observe that neural network models provide much better performance in terms of maximum error. This result may have significant managerial implications when making financial hedging decisions in the short term.

### 3 Modified Holt-Winters exponential smoothing

Exponential smoothing is a fairly simple forecasting method suitable for univariate time-series data. Despite its simplicity, it is one of the most effective automatic forecasting methods.

It applies recursive updating schemes while smoothing and forecasting data. The formulation of the exponential smoothing method for  $k$ -step-ahead forecasting can be stated as:

$$\hat{y}_t(k) = \alpha y_t + (1 - \alpha) \hat{y}_{t-k}(k) \quad (3.1)$$

or equivalently in error correction form:

$$\hat{y}_t(k) = \hat{y}_{t-k}(k) + \alpha e_t, \quad (3.2)$$

$$e_t = y_t - \hat{y}_{t-k}(k), \quad (3.3)$$

where  $y_t$  is the observed time series,  $\hat{y}_t(k)$  is the  $k$ -step-ahead forecast made at time  $t$ ,  $\alpha$  is the smoothing factor and  $e_t$  is the  $k$ -step-ahead forecast error.

The Holt-Winters method is an extension of exponential smoothing designed for series with trend and seasonality; therefore, it is also referred to as double exponential smoothing. The method is a robust and easy way of forecasting that works especially well with short-term sales and demand time-series data (Gelper et al. 2010). It models the data through a local mean, a local trend, and a local seasonal factor. There are two different formulations for multiplicative and additive seasonality. In this paper, we consider the HWT method with additive seasonality and without a trend term. As also noted in Taylor (2010b), including a trend term brings no improvements to forecast accuracy, as changes in demand level are not significant for short-term load forecasting. Taylor (2003) develops an extension of the regular HWT method to accommodate for the presence of two seasonal cycles, which is typical in electricity load data. That extension is presented in Eqs. (3.4)–(3.8).

$$\hat{y}_t^T(k) = l_t + d_{t-m_1+k_1} + w_{t-m_2+k_2} + \phi^k e_t^T, \quad (3.4)$$

$$e_t^T = y_t - \hat{y}_{t-k}^T(k), \quad (3.5)$$

$$l_t = l_{t-1} + \alpha e_t^T, \quad (3.6)$$

$$d_t = d_{t-m_1} + \delta e_t^T, \quad (3.7)$$

$$w_t = w_{t-m_2} + \omega e_t^T, \quad (3.8)$$

where  $\hat{y}_t^T(k)$  is Taylor's  $k$ -step-ahead forecast derived at time  $t$  and  $e_t^T$  stands for the error when demand in time  $t$  is forecasted with Taylor's adaption of HWT (Taylor 2010b).  $m_1$  and  $m_2$  are the number of periods in the first and second seasonal cycles, which correspond to daily and weekly cycles in our case.  $l_t$  is the smoothed level and  $d_t$  and  $w_t$  stand for seasonal indices for daily and weekly cycles, respectively. The smoothing parameters are denoted by  $\alpha$ ,  $\delta$ , and  $\omega$ ; and  $k_1 = [(k-1) \bmod m_1] + 1$  and  $k_2 = [(k-1) \bmod m_2] + 1$ . Including the autoregressive component  $\phi$  corrects for the first order residual autocorrelation and improves the forecast accuracies.

As presented above, Taylor's models do not consider special days. In practice, electricity load data consists of many special days, such as celebrations and national and religious holidays, which create a challenge for generating accurate forecasts. In this study, we modify Taylor's HWT to allow the model to learn from its previous errors on special days, which brings significant improvement to the model's performance. The updated model formulation is provided in (3.9)–(3.14):

$$\hat{y}_t(k) = l_t + d_{t-m_1+k_1} + w_{t-m_2+k_2} + \phi^k e_t + \left( \sum_{i \in S} s_{i,t+k} \frac{\sum_{\substack{j=t+k-(365+h)*24 \\ h \in \{-L, -L+1, \dots, L-1, L\}}}^L s_{i,j} \frac{e_j^T}{y_j}}{\sum_{\substack{j=t+k-(365+h)*24 \\ h \in \{-L, -L+1, \dots, L-1, L\}}}^L s_{i,j}} \right) \hat{y}_t^T(k) \quad (3.9)$$

where

$$e_t = y_t - \hat{y}_{t-k}(k), \quad (3.10)$$

$$e_t^T = y_t - \hat{y}_{t-k}^T(k), \quad (3.11)$$

$$l_t = l_{t-1} + \alpha e_t, \quad (3.12)$$

$$d_t = d_{t-m_1} + \delta e_t, \quad (3.13)$$

$$w_t = w_{t-m_2} + \omega e_t, \quad (3.14)$$

where  $s_{i,t}$  is a binary variable that is equal to 1 if  $t$  is a forecasting time interval on a special day of type  $i$  where  $i$  refers to type of the special day that time  $t+k$  belongs to. Here  $S$  refers to the set of special day types and  $e_j^T$  stands for the error when demand in time  $j$  is forecasted with Taylor's adaptation of HWT. In our modification, the model checks whether time  $t+k$  is a special day, and if it is, the model goes to previous year's data, checks a range of days ( $\pm L$ ) around that time  $t+k$ , to find a similar type of special day and updates the forecast by multiplying its forecast  $\hat{y}_t^T(k)$  by the percentage error of previous year's forecasting error on that special day,  $e_j^T/y_j$ .

For one-step-ahead forecasting, i.e., for  $k+1$ , the formulations in (3.9)–(3.14) are used, but for multi-step-ahead forecasts, (3.9) differs slightly. Taylor (2010b) formulized multi-step-ahead forecasts for  $1 < k \leq m_1$  as follows:

$$\hat{y}_t(k) = l_t + \frac{\alpha \phi (1 - \phi^{k-1})}{(1 - \phi)} e_t + d_{t-m_1+k_1} + w_{t-m_2+k_2} + \phi^k e_t, \quad (3.15)$$

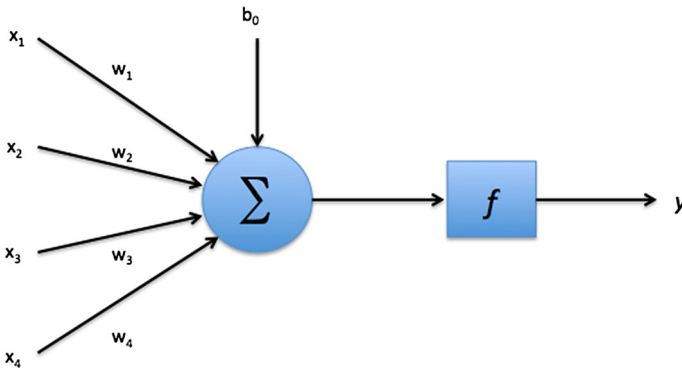
where the first two terms sum up to the expected value of the lagged smoothed level,  $l_t$ . For our case, we derived the formula for  $k > m_1$  as presented in (3.16):

$$\hat{y}_t(k) = l_t + \frac{\alpha}{(1 - \phi)} e_t + d_{t-m_1+k_1} + w_{t-m_2+k_2} + \phi^k e_t. \quad (3.16)$$

## 4 Nonlinear autoregressive with exogenous input neural networks

Artificial neural networks are highly interconnected simple processing units inspired by biological neural nets, which transmit signals via neurons and synapses. The method aims to capture complex relationships between input and output information with the network structure. The biggest advantage of ANNs compared to other computational methods is their capability of providing information about nonlinear and hidden patterns in the data. The advantage of ANNs lies in their execution; they implement linear discriminants in a space where inputs have been mapped nonlinearly. They also implement fairly simple algorithms where nonlinearity can be learned from training data (Duda et al. 1997).

Figure 1 illustrates the processing of a simple single-node (*a.k.a.* neuron) ANN structure. A node receives inputs  $x$ , multiplies each input by a weight  $w$ , adds a bias  $b_0$ , and



**Fig. 1** Structure of a single-node ANN

applies a transformation function  $f$  to generate an output  $y$ . Some of the commonly used transformations include log-sigmoid, hyperbolic tangent sigmoid, tansig transfer, and linear functions.

There are two types of neural networks with respect to the connections between neurons and the direction of data propagation: feed-forward and recurrent networks (Duda et al. 1997). Figure 2 presents a three-layered feed-forward neural network, where the data is received through the input layer, passed to the hidden layer, then transferred to the output layer. The term feed-forward refers to networks with interconnections that do not form any loops. There also exist recurrent or non-feed-forward networks, in which there are one or more loops of interconnections. In such networks, the input state is combined with the previous state activation through an additional weight layer (Bodén 2002; Catalão et al. 2007). An instance of recurrent networks is provided in Fig. 3.

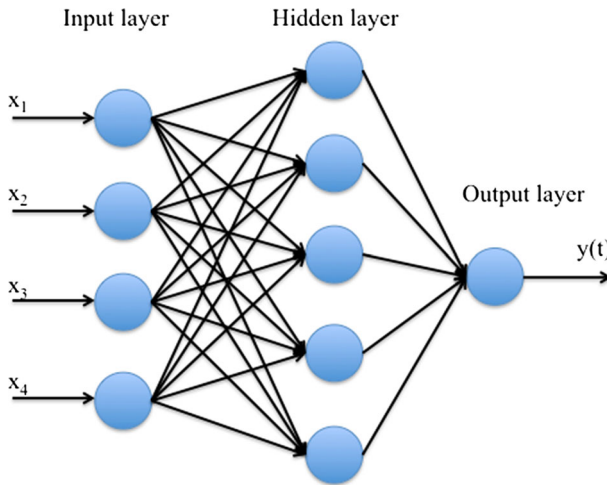
A NARX network is a type of recurrent dynamic neural network with feedback connections between the output and input layers. This type of network is specifically used for time-series forecasting. Another important property of NARX is that it allows exogenous inputs to become network inputs. It is derived from the autoregressive exogenous (ARX) model and can be mathematically stated as

$$\hat{y}_t = f(u_{t-D_u}, u_{t-1}, u_t; y_{t-D_y}, \dots, y_{t-1}), \quad (4.1)$$

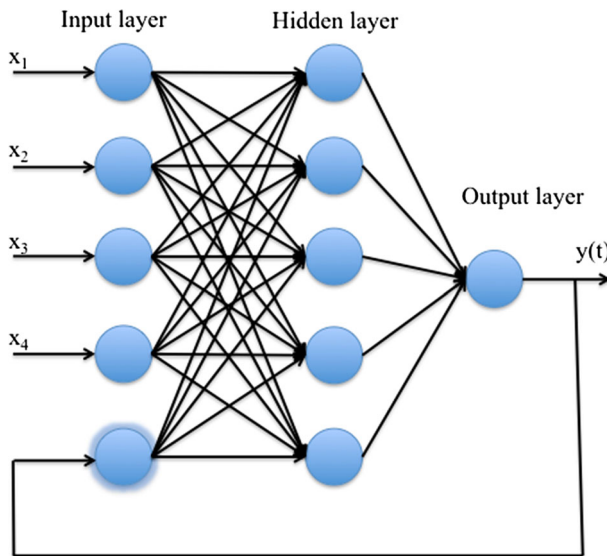
where  $u_t$  and  $y_t$  are inputs and outputs of the model at time  $t$ , and  $D_u$  and  $D_y$  are input and outputs delays. Delays represent the number of periods the inputs' and/or outputs' past values are fed back to the network. The nonlinear transformation function is denoted by  $f$ .

There are two types of NARX networks with respect to the information embedded into the feedback loop: open-loop and closed-loop networks. In open-loop networks, actual output values are fed back to the network. These networks are also called series-parallel (SP) mode networks. In the closed-loop architecture, the network's outputs (estimated values) are fed back to the network as inputs. These types of networks are also referred to as parallel (P) mode networks. The NARX model in (4.1) represents an SP architecture.

In a neural network, the neurons in each layer are interconnected by modifiable weights. Transformation takes place only in the hidden and output layers; the input layer serves only for transferring the input. For a network that consists of  $p$  inputs, each hidden node  $j$  computes the weighted sum of its inputs, denoted by  $net_j$  as:



**Fig. 2** Example of three-layered feed-forward neural network structure



**Fig. 3** Example of three-layered recurrent neural network structure

$$net_j = \sum_{i=1}^p x_i w_{ji} + w_{j0}, \quad (4.2)$$

where  $i$  and  $j$  stand for the nodes in the input and output layers, respectively,  $x_i$  stands for any type of input (multivariate inputs or input from the feedback loop),  $w_{ji}$  represents the input weights to node  $j$ , and  $w_{j0}$  is the bias value at node  $j$ .

After computing the  $net$  value, a transfer function is used to generate the hidden node's output:



$$y_j = f(\text{net}_j). \quad (4.3)$$

Similarly, for all hidden layers and output layer units, *net* values are calculated as the weighted sum of the values received from the previous layer's nodes plus the bias value, and then the transformation function of the node is applied. The general representation of node activation of a three-layered network with  $k$ ,  $j$ , and  $m$  neurons in the hidden layers is

$$\hat{y}_t = f\left(\sum_{j=1}^{n_H} w_{kj} f\left(\sum_{i=1}^p w_{ji} x_i + w_{j0}\right) + w_{t0}\right), \quad (4.4)$$

where  $n_H$  is the number of hidden nodes in the output layer,  $w_{kj}$  is the interconnection weight from the hidden layer to the output layer,  $p$  is the number of nodes in the input layer and  $w_{j0}$  and  $w_{t0}$  are the bias values. Each node applies a transformation function to the weighted sum and transfers its product. It should be noted that for single-layer neural network with linear transfer functions in the output layer, the system could be interpreted as a linear regression model. Similarly, a network with logistic transfer functions is equivalent to logistic regression.

As a data-driven model, training is fundamental in neural network modeling. During the training phase, the network adjusts the weight and bias values to produce the best predictive results. One of the most popular methods for network training is the back-propagation algorithm, which is a natural extension of the least-mean-squares (LMS) method. The learning process starts with an untrained network, and a training dataset is fed to the input layer which passes through the network to result in an output value. The obtained value is compared to the target value, which is the actual output in the dataset. The difference corresponds to the error. In back propagation, the criterion function is some scalar function of the network's weights. With respect to the learning rate, the weights are adjusted to minimize the error given as

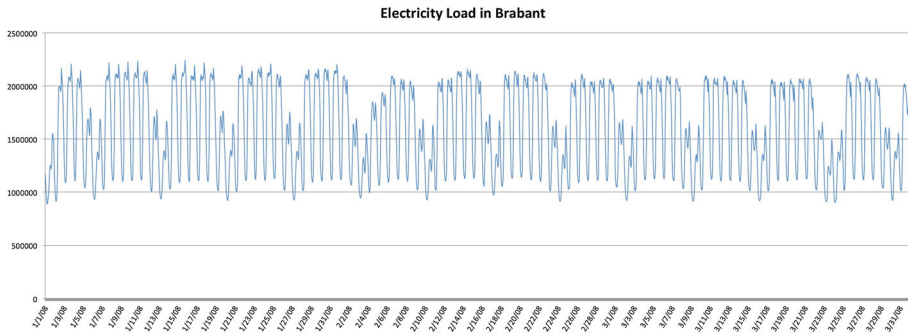
$$J(\mathbf{w}) = \frac{1}{2} \sum_{t=1}^r (y_t - \hat{y}_t)^2, \quad (4.5)$$

where  $\mathbf{w}$  stands for the vector of weights,  $r$  is the length of the network output vector,  $y_t$  is the target value, and  $\hat{y}_t$  is the estimated value. The back-propagation algorithm is based on the gradient descent algorithm, and the weights are updated in the direction of error reduction, starting from the initial values (Duda et al. 1997),

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu(k) \nabla J(\mathbf{w}(k)), \quad (4.6)$$

where  $\mu$  is the learning rate (taking a value between 0 and 1),  $\mathbf{w}(k)$  is the weight vector in iteration  $k$  and  $\nabla J(\mathbf{w}(k))$  is the gradient vector. Similar to the gradient descent procedure, the learning rate controls the amount of change in weights and bias values in each iteration  $k$ . Larger values can give a faster convergence to the minimum but may also produce oscillation (Bodén 2002).

Compared to other networks, NARX neural networks are more powerful, converge faster, and more easily generalizable. In this paper, we consider NARX networks with zero input delays and various outputs delays. Output delays are selected with respect to autocorrelation values and seasonalities present in the data, whereas the input delay is set to zero due to the structure of special day variables. The details are explained in Sect. 5.3.



**Fig. 4** Hourly electricity load in the Brabant dataset between January 2008 and March 2008

## 5 Numerical study

For our numerical analysis, we use a dataset of hourly electricity load levels of five Dutch provinces for a period of 256 weeks between 1 January 2008 and 30 November 2012. The dataset from each region is named for its region (Brabant, Noord, Friesland, Limburg and Maastricht). The approach and detailed results are presented through the Brabant and Noord dataset and all other results are given in the Electronic Companion. The electricity load data contains daily and weekly cycles, which can be clearly seen in Fig. 4. Load patterns are very similar on weekdays. On weekends, the load decreases significantly, but the pattern remains similar from weekend to weekend.

Another important element of the dataset is the calendar variables, which covers the special days noted above. Similar to Kim (2013) we analyzed the effect of these days with respect to their effects on different hours of the day and with respect to the variation in their effects from year to year. Consequently, similar to Tanrisever et al. (2013) we find that the following variables affect electricity load in the Netherlands:

- **School Holidays** In the Netherlands, schools are closed during public holidays, Christmas, on the May holiday, and for a spring break, summer break, and autumn break.
- **Bouwvak** The period during the summer when construction companies do not operate.
- **Liberation Day** Public holiday celebrated every five years.
- **Carnival** Three days of celebrations in the southern part of the Netherlands. This variable is excluded from the dataset of northern regions.
- **Christmas period** We observe deviations from the regular demand pattern not only on 25 December but also on Christmas Eve and Boxing Day. Hence we define these 3 days as the ‘Christmas period.’
- **New Year’s Eve** New Year’s Eve and New Year’s Day also experience a sharp decrease in electricity demand.
- **Queen’s/King’s Day** A public holiday to celebrate the queen or king’s birthday.
- **Easter, Ascension Day, Whit Monday** Religious holidays in the Netherlands.

It is important to note that different special days have different effects on electricity load throughout the day, depending on the region and the day of week.

In addition to the exact dates, special days also affect electricity load on days close to them. Electricity load before and after holidays tends to decrease for most of the special days. Therefore, similar to Tanrisever et al. (2013) we defined and added variables called ‘*Day before holiday*’ and ‘*Day after holiday*’ to the dataset. More importantly, we observe

<b>Initialization:</b>	Given two weeks' electricity load data, $c_1 = 24$ and $c_2 = 168$ , referring to daily and weekly cycles.
<b>Step 1</b>	Initialize the level $I_t$ by averaging load data over the two-week period.
<b>Step 2</b>	Initialize the daily seasonal state variable $d_t$ as:
Step 2.1	Compute $c_1$ point-centered moving averages.
Step 2.2	Calculate differences between actual data and moving average values that roughly give the seasonality effect.
Step 2.3	Calculate average differences for $c_1$ hours of the day over two weeks.
<b>Step 3</b>	Initialize the weekly seasonal index $w_t$ as:
Step 3.1	Compute $c_2$ point-centered moving averages;
Step 3.2	Calculate differences between actual data and moving average values that roughly give the seasonality effect;
Step 3.3	Calculate average differences for $c_2$ hours of the week over two weeks.

**Fig. 5** The algorithm for initializing the state variables

that the effect is even more significant when the day before or after a holiday falls on a Monday or Friday, due to the fact that on such days people are more willing or able to take 1 day off for a longer holiday. Hence, we added another variable called '**Bridge day**' to the dataset to denote these special days.

In our study, we only used historic data and calendar variables to forecast short-term electricity load. As Taylor (2010b) also pointed out, meteorological and economic variables also have distinct effects on electricity demand in the long term. However, for short-term forecasts, these variables can be excluded, as consumer adaptation to changes in these variables takes time (Taylor et al. 2006).

### 5.1 Modified Holt-Winters exponential smoothing

Considering the model specifications presented in (3.9)–(3.14), to model time series data with the HWT method, initial values need to be estimated for level and seasonal components and for the smoothing parameters. To initialize the state variables ( $I_t$ ,  $d_t$ ,  $w_t$ ) in (3.9)–(3.14), we use 2-week intervals that do not include any special days to prevent divergent observations from causing misleading fluctuations in initialization. The algorithm for initializing state variables is presented in Fig. 5.

In our model, lag value ( $\pm L$ ) is set as equal to 20 days; that is, we search for an interval of 40 days in the previous year to identify the effects of the respective special days. Last, the model parameters  $\alpha$ ,  $\delta$ ,  $\omega$ , and  $\phi$  in (3.9)–(3.14) are derived using a similar method to Taylor (2010b) and Engle and Manganelli (1999). Figure 6 shows the steps of the algorithm.

Different than Taylor (2010b)'s approach, in our models we derive the best parameters for each forecasting horizon. Table 1 presents the model parameters for the Brabant dataset (see "Appendix A" for the parameter values of the other regions).

After initializing the state variables and calibrating the models' parameters, we apply m-HWT modeling to the datasets of the different regions.

Iteration:	
<b>Step 1</b>	Generate $10^5$ vectors of four parameters that are uniformly distributed between 0 and 1.
<b>Step 2</b>	For every vector, compute sum of squared errors (SSE) of the training dataset.
<b>Step 3</b>	Define 10 vectors with the lowest SSEs as the set of possible model parameters.
<b>Step 4</b>	Generate all possible combinations of the selected 10 vectors.
<b>Step 5</b>	For every combination vector, compute sum of squared errors (SSE) of the training dataset.
<b>Step 6</b>	Assign elements of the vector with lowest SSE as model parameters.

**Fig. 6** The algorithm for deriving the model parameters  $\alpha$ ,  $\delta$ ,  $\omega$ , and  $\phi$

**Table 1** Model parameters for the Brabant dataset

Lead times (h)	$\alpha_{best}$	$\delta_{best}$	$\omega_{best}$	$\phi_{best}$
1	0.5487	0.1832	0.2658	0.3399
6	0.0377	0.2354	0.1590	0.7750
12	0.0145	0.1820	0.1769	0.8330
24	0.0005	0.2649	0.1066	0.9034
48	0.0007	0.0715	0.0768	0.7265
168	0.0001	0.0225	0.1713	0.8379

## 5.2 Nonlinear autoregressive with exogenous input neural networks

The NARX modeling consists of two stages: (1) training the model and (2) testing the model performance. Accordingly, we use 75% of our dataset for training (in-sample data), 15% for validation, and the remaining 15% for testing (out-of-sample data). In addition to the historical electricity load data, we define special days as binary variables and add them to the dataset.

We observe that some special days have similar effects on electricity load, therefore these days are grouped together to decrease the dimensionality of the problem:

- Easter, Whit Monday, and Liberation Day
- Carnival
- Christmas Eve and New Year's Eve
- Queen's/King's Day
- Boxing Day and Christmas Day
- New Year's Day
- Ascension Day

In addition to the exact dates of special days, we also define and include 1 day prior to, 1 day after these days, and bridge days in the model, as noted above. Furthermore, we define *day of the week*, *hour of the day*, and *summertime* variables as inputs.

As described earlier, a neural network model consists of three types of layers: input, hidden, and output. The data is received through input layer and passed to hidden layer and transferred to output layer. Our **exogenous** input variables define a total of 13 nodes in the input layer, i.e., 7 variables for special days, 3 variables for the days close to special days and 3 calendar variables as explained above. The input layer also contains nodes of the

**Table 2** Model architectures of different forecasting lead times for the Brabant dataset

Lead times (h)	Hidden layer	Number of hidden nodes
1	1	30
6	1	35
12	1	75
24	1	35
48	1	30
168	1	35

feedback loop. Feedback delays are determined with respect to the autocorrelation values between different lags of infeed data. In this study, autocorrelation values above 0.8 are identified as highly correlated. Hence, among these lag values, considering the seasonalities, the reasonable ones are selected as feedback delays, which in our case are (in hours): electricity infeed 1, 2, 23, 24 (which is 1 day), 25, 168 (which is 1 week) and 169 h before. Therefore, together with these feedback delays, the input layer consists of **20 input nodes**. Obviously, the output layer contains only one node, which gives the electricity infeed forecast.

In the literature, it is well established that architectures with a single hidden layer are sufficient for addressing most forecasting problems, but at the expense of higher training times. Therefore, to keep the search for a model architecture to a reasonable limit, we consider single hidden-layer feed-forward networks as candidate model architectures. Next, we search for the best performing architectures for each forecasting lead time and region. We keep the number of hidden nodes between five and 80 (and consider networks with hidden nodes that are multiples of five). Networks with fewer than five hidden nodes are not usually capable of modeling and learning the data, and networks with more than 80 hidden nodes face the risk of overfitting.

Therefore, for all datasets and lead times, 16 different architectures are run with five different initializations. The best performing architectures for each lead time for the Brabant dataset are presented in Table 2. Next, for each forecasting lead time and region, we run the best performing architectures 10 times to find the best weight and bias values, and complete the network architecture. The NARX neural network architectures for other regions can be found in “Appendix B”.

The other important components of neural network models that impact the forecasting performance are the training algorithm and the transfer function. In this study, we use the Levenberg-Marquardt training method, which is a modification of the popular back-propagation algorithm. This method includes an approximation of Newton’s method, which is usually very efficient up to a few hundred nodes (Zhang et al. 1998). We use the tansig and purelin transfer functions in the hidden and output layers, respectively. We have also tested for the normality assumption on the residuals of the models. The experimental study had been conducted with the models that satisfy the normality assumption on the residuals.

### 5.3 Results

We first compare the performances of Taylor’s HWT and our m-HWT methods. Then, we compare NARX with these two methods. Post-sample accuracies are measured in terms of mean absolute percentage error (MAPE) and maximum percent error (MaxAPE). Post-

**Table 3** Performances of NARX, HWT and m-HWT for different day types in terms of MAPE for Brabant dataset

	Lead times												Avg				
	1	6	12	24	48	168	Testing (%)	Training (%)	Testing (%)	Training (%)	Testing (%)	Training (%)					
<b>Total</b>																	
HWT	1.59	1.44	1.81	2.76	2.64	3.18	2.86	4.01	3.45	4.74	4.13	3.05	2.72				
m-HWT	1.55	1.36	1.68	2.68	2.43	3.08	2.62	3.84	3.11	4.47	3.63	2.93	2.47				
NARX	0.71	0.75	2.09	2.46	2.29	1.84	2.06	2.52	2.75	2.89	2.85	2.04	2.19				
<b>Special</b>																	
HWT	2.01	1.55	2.94	2.26	3.34	5.30	4.11	6.85	5.18	7.87	5.90	4.89	3.72				
m-HWT	1.82	1.26	2.65	1.76	2.58	4.75	3.23	5.97	3.95	6.58	4.09	4.30	2.81				
NARX	0.80	0.74	2.16	2.35	2.42	2.18	2.28	3.01	3.08	3.57	3.25	2.36	2.35				
<b>Long holidays</b>																	
HWT	1.58	1.24	2.01	1.53	2.06	3.32	2.62	4.53	3.73	5.57	4.49	3.33	2.61				
m-HWT	1.52	1.13	2.00	1.48	2.02	3.22	2.44	4.13	3.08	4.71	3.27	3.10	2.24				
NARX	1.33	1.50	2.57	2.26	2.08	2.51	2.45	3.05	2.88	4.78	3.27	2.87	2.41				
<b>Short holidays</b>																	
HWT	4.10	4.97	7.49	10.49	17.82	14.98	21.00	18.19	21.58	19.09	21.84	12.51	16.28				
m-HWT	3.25	2.74	5.81	4.92	8.89	12.25	12.15	14.95	13.74	15.70	13.40	10.15	9.31				
NARX	0.76	0.70	2.17	2.37	2.45	2.19	2.28	3.04	3.10	3.56	3.25	2.36	2.36				
<b>Non-Spec</b>																	
HWT	1.48	1.40	1.80	1.65	2.37	2.58	2.39	3.22	2.79	3.86	3.46	2.54	2.34				
m-HWT	1.48	1.40	1.80	1.65	2.37	2.58	2.39	3.22	2.79	3.86	3.46	2.54	2.34				
NARX	0.67	0.75	2.08	2.50	2.07	1.72	1.94	2.35	2.60	2.65	2.65	1.92	2.11				

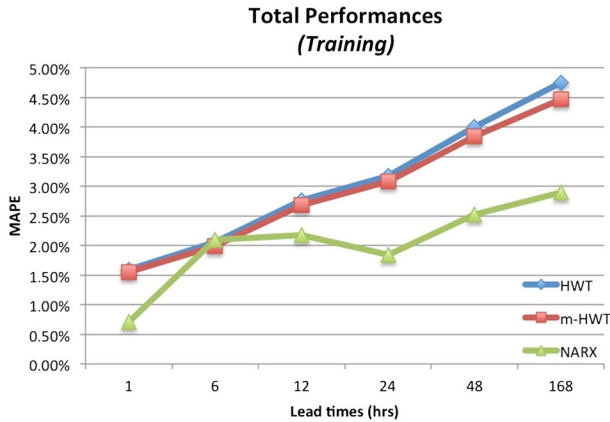


Fig. 7 Training performances for the Brabant dataset

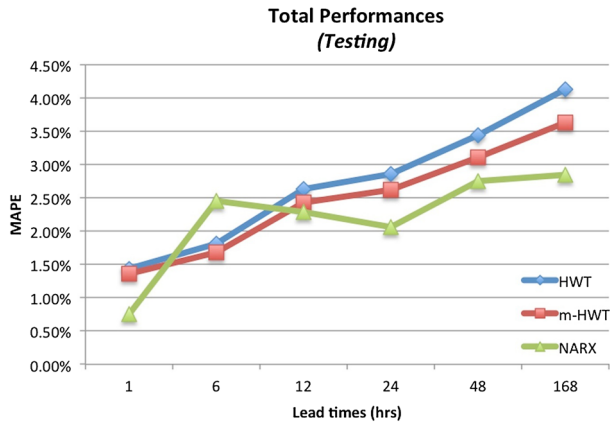
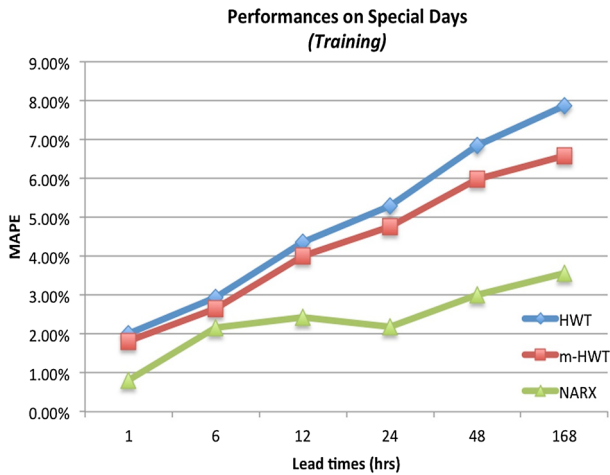


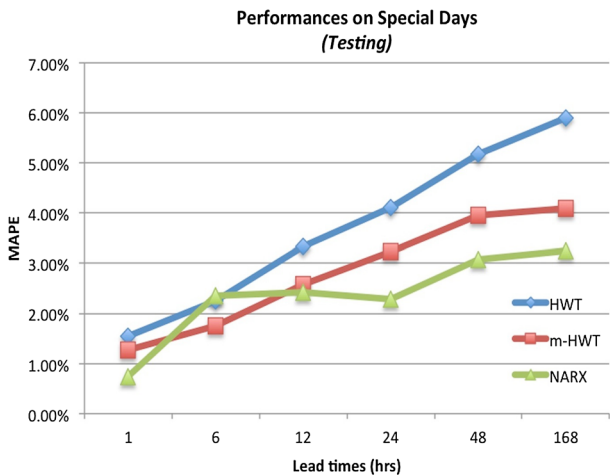
Fig. 8 Testing performances for the Brabant dataset

sample data accounts for 15% of the total dataset (corresponding to the period between 6 March 2012 and 30 November 2012). In addition, after carefully analysing the demand pattern on special days, we have grouped these days as: long holidays and short holidays, and separately reported the results for them. Long holidays last at least 1 week, and include school holidays and Bouwvak. Since these holidays are relatively long, they have a levelling effect on the load. On the other hand, short holidays (including Liberalization day, Carnival, Christmas, New Year’s Eve, New Year Holiday, Queens Day, Easter, Ascension Day, Whit Sunday and Monday and Good Friday) only take 1 to 3 days, and the demand during these days abruptly deviates from the usual pattern.

Table 3 presents the results for the Brabant dataset. We run the algorithms for different forecasting lead times ranging from 1 to 168 h, 1 week, (see the first line of Table 3) and report the methods’ performance. The first column lists three methods HWT, m-HWT and NARX. “Total” refers to the performance on the whole data, “Special” is for the performance on special days. Detailed analysis on special days is also given by grouping as “Long Holidays” and “Short Holidays”. Lastly the performances of the methods are separately reported for



**Fig. 9** Training performances on special days for the Brabant dataset



**Fig. 10** Testing performances on special days for the Brabant dataset

normal days titled as “Non-Spec”. We provide training and testing data performances for all cases.

Table 3 clearly shows that our m-HWT provides a significant improvement in forecasting performance over Taylor’s HWT. The improvement is around 9% for the Brabant dataset, and it is 8% on average for the five datasets we studied. As expected, the effect of modification is drastic on special days. For the Brabant dataset, the improvement for special days over Taylor’s HWT varies between 16 and 30% for different lead times. For short holidays, the improvement is even more significant; it reaches 53% in 6-h-ahead forecasting. Since HWT does not explicitly consider the demand pattern in these special days, it results in large errors (compared to NARX). m-HWT aims to correct this error by recognizing the sudden change in the data pattern during these days. In particular, m-HWT tries to identify similar



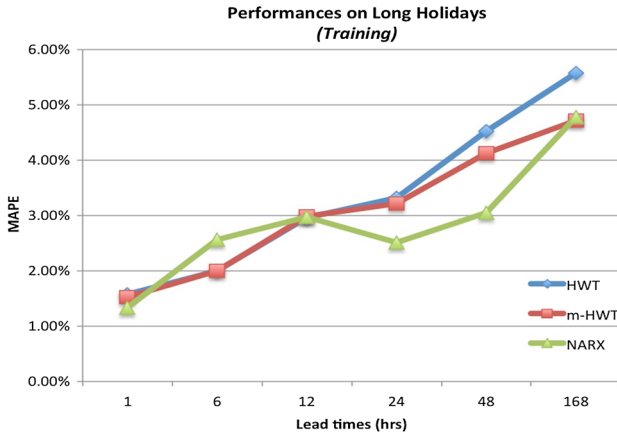


Fig. 11 Training performances on long holidays for the Brabant dataset

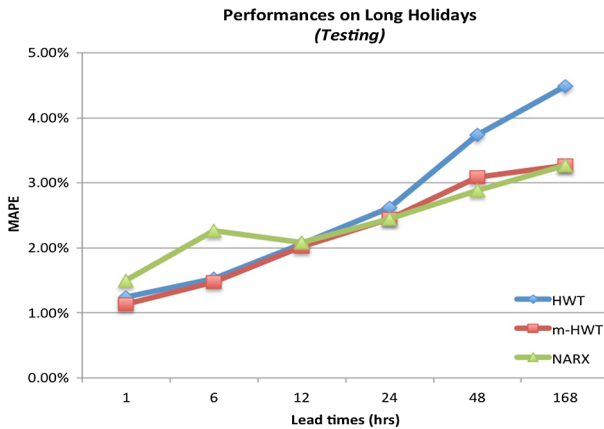
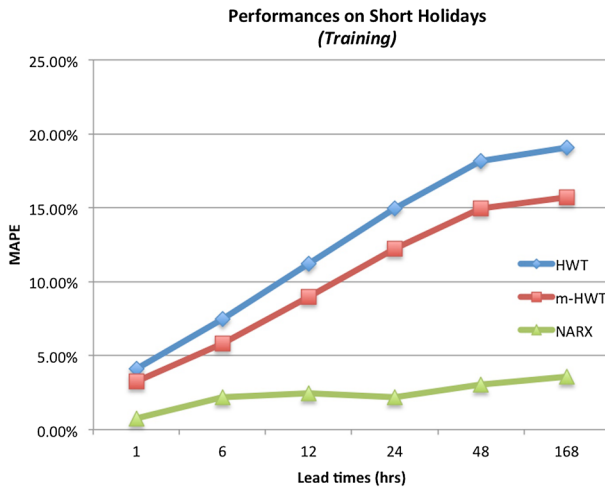


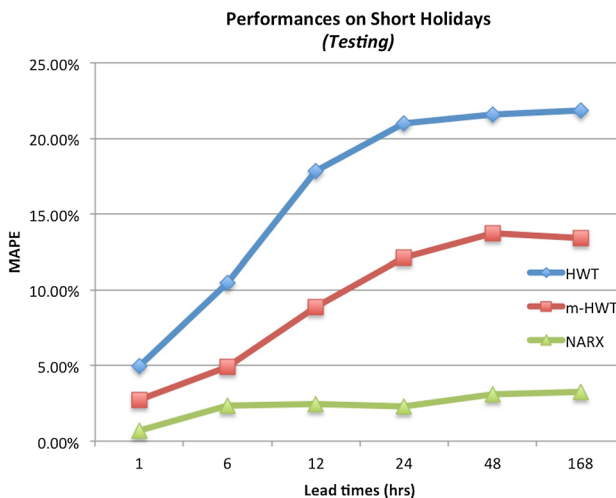
Fig. 12 Testing performances on long holidays for the Brabant dataset

short holidays in the past data and incorporate a correction factor based on these days. As a result, compared to HWT, m-HWT gives improved errors on these days. On the other hand, NARX makes the same correction in a more structured and advanced way by capturing the complex dynamics in the data during these special days. Hence, NARX provides a much better forecasting error on short holidays. For illustrative purposes, we plot the performance of the three methods for the Brabant dataset (both for training and testing data) in Figs. 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16. These figures show the MAPE of the methods for each lead time.

Figures 7 and 8 show that NARX performs better than m-HWT for almost all the forecasting lead times. Especially for 1-h-ahead forecasting, NARX is quite effective and provides a MAPE of as low as 0.71%. The m-HWT algorithm only slightly outperforms NARX for the 6-h-ahead forecast. Figures 9–14 present the forecasting performances on special days, long holidays, and short holidays. These figures reveal that NARX’s superior performance is mostly driven by its ability to forecast the load on short holidays; it performs drastically better than m-HWT and HWT on those days, while it is competitive with m-HWT on long holi-



**Fig. 13** Training performances on short holidays for the Brabant dataset



**Fig. 14** Testing performances on short holidays for the Brabant dataset

days. For normal days, NARX still outperforms m-HWT except for the 6-h-ahead forecast (Figs. 15, 16).

Next, in Table 4 we present our results for the Noord dataset. Similar to the Brabant dataset, m-HWT outperforms HWT, and NARX is more effective in forecasting short holidays and special days in general. As noted earlier, NARX performance deteriorates for the 6- and 12-h lead times. In addition, m-HWT improves HWT performance on average by eight percent, and NARX shows an improvement over HWT by approximately 10%. Regarding short holidays, m-HWT performs 40% better than HWT on average, whereas for NARX the improvement is 83%. Similar to the Brabant dataset, Figs. 17–26 illustrate the performance of the three methods for the Noord region. The

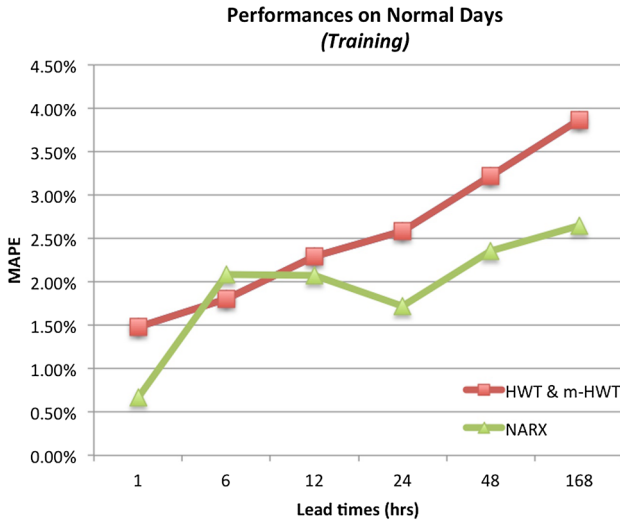


Fig. 15 Training performances on normal days for the Brabant dataset

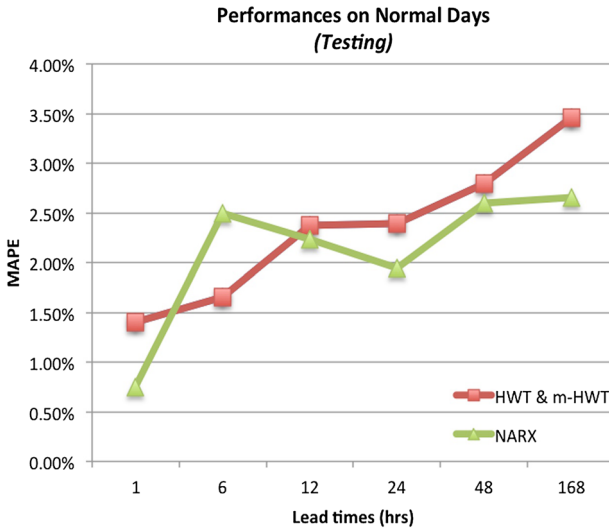


Fig. 16 Testing performances on normal days for the Brabant dataset

results are materially similar for the other three regions (Limburg, Maastricht and Friesland) and are given in “Appendix C”. Counter to the intuition; we observe that NARX outperforms m-HWT for 1 h ahead forecasts. This is because including special days to m-HWT introduces sudden shocks to the model whose effect is carried over multiple periods increasing the forecasting error for very short-term forecasts, such as 1-h ahead forecasting.

In general, for the Noord dataset, we observe that NARX is superior to HWT and m-HWT for short-term load forecasting, especially during short holidays. However, NARX’s

**Table 4** Performances of NARX, HWT and m-HWT for different day types in terms of MAPE for Noord dataset

	Lead times										Avg									
	1	6	12	24	48	168	Testing (%)	Training (%)	Testing (%)	Training (%)										
<b>Total</b>																				
HWT	1.55	1.42	1.91	2.82	2.78	3.10	2.96	3.71	3.20	4.90	4.71	3.00	2.83							
m-HWT	1.53	1.34	1.79	2.75	2.61	3.01	2.75	3.59	2.95	4.67	4.12	2.90	2.59							
NARX	0.85	0.90	3.02	2.87	3.13	2.07	2.36	2.38	2.56	2.99	2.97	2.36	2.54							
<b>Special</b>																				
HWT	2.03	1.53	2.99	4.86	4.10	5.49	4.69	6.63	5.27	7.82	6.82	4.97	4.16							
m-HWT	1.90	1.25	2.77	4.55	3.47	5.11	3.91	6.04	4.31	6.73	4.66	4.52	3.28							
NARX	0.97	0.95	3.29	3.18	3.10	2.53	2.49	2.89	2.79	3.78	3.81	2.77	2.72							
<b>Long Holidays</b>																				
HWT	1.74	1.25	2.36	3.70	2.90	4.17	3.34	5.20	3.93	6.35	5.50	3.92	3.13							
m-HWT	1.71	1.14	2.36	3.72	2.84	4.16	3.23	5.04	3.59	5.84	3.84	3.81	2.74							
NARX	1.85	1.81	4.17	4.13	2.66	3.76	2.14	3.78	1.98	4.98	3.00	3.78	2.41							
<b>Short Holidays</b>																				
HWT	4.28	4.72	7.98	14.00	17.90	15.86	20.28	17.90	20.78	19.35	22.08	13.23	15.95							
m-HWT	3.35	2.51	6.01	11.07	10.75	12.55	11.76	13.98	12.63	13.81	14.03	10.13	9.46							
NARX	0.93	0.89	3.29	3.18	3.12	2.54	2.53	2.91	2.85	3.78	3.87	2.77	2.74							
<b>Non-Spec</b>																				
HWT	1.44	1.40	1.60	2.26	2.39	2.45	2.43	2.91	2.55	4.09	4.03	2.46	2.42							
m-HWT	1.44	1.40	1.60	2.26	2.39	2.45	2.43	2.91	2.55	4.09	4.03	2.46	2.42							
NARX	0.81	0.89	2.92	2.76	3.17	1.90	2.30	2.19	2.60	2.72	2.60	2.22	2.49							

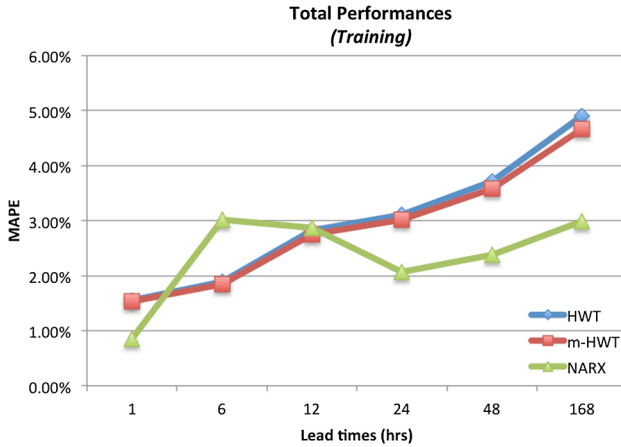


Fig. 17 Training performances for the Noord dataset

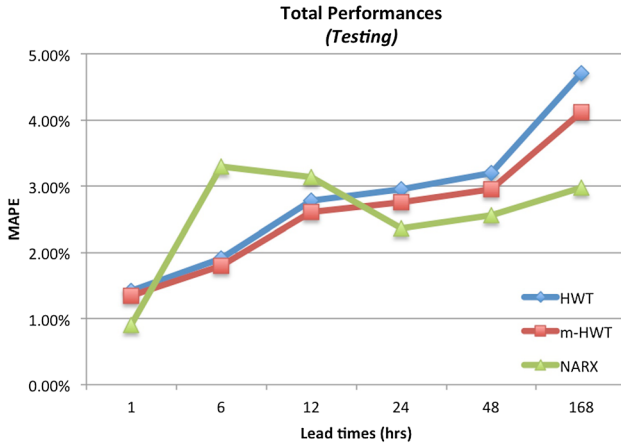


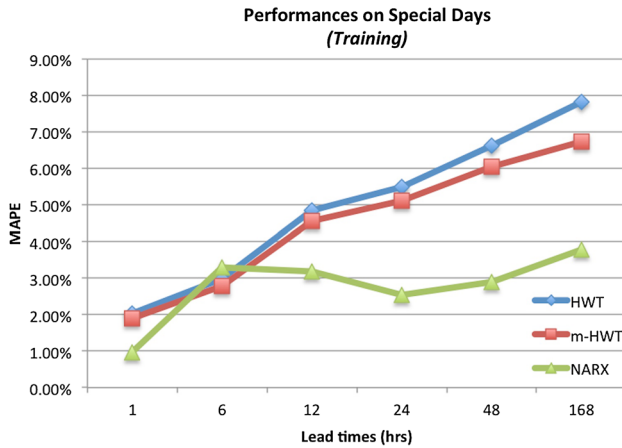
Fig. 18 Testing performances for the Noord dataset

performance deteriorates for lead times that are not multiples of 24 h (1 day). In addition, m-HWT significantly improves the performance of HWT, but in general NARX is more effective in forecasting special days. For forecasting the load on long holidays, however, m-HWT is competitive with NARX.

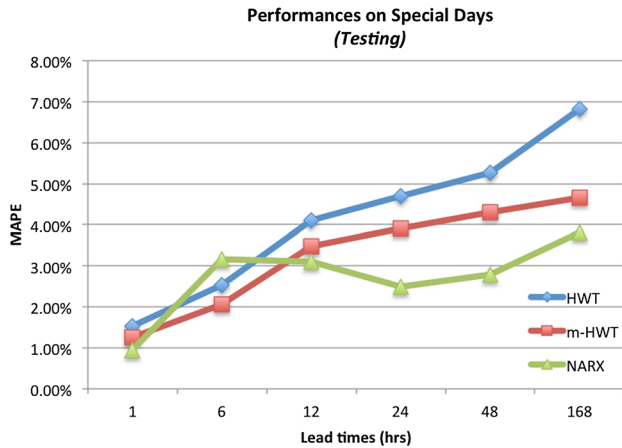
We also carried out a stability analysis to evaluate whether NARX’s performance is stable in the training and testing datasets alike. The stability values for each dataset are presented in Table 5, where being close to 1 refers to a more stable model.

Another performance measure we consider in this study is MaxAPE. In practice, this measure is of significant managerial importance due to risk management and hedging reasons. The maximum errors of the models for each region are presented in Table 6.

In terms of MaxAPE, NARX performs better than both HWT methods for all forecasting lead times and regions. The only exceptions are the 6-h-ahead forecasts for the Limburg and Noord datasets, due to the aforementioned loss of NARX accuracy for 6- and 12-h-ahead



**Fig. 19** Training performances on special days for the Noord dataset



**Fig. 20** Testing performances on special days for the Noord dataset

forecasting. NARX proves to be a good fit for market parties who would like to avoid large risks. Furthermore, m-HWT also decreases MaxAPE values up to 40% compared to Taylor's errors. In addition to the low MAPE values for 1-h-ahead forecasts, NARX also gives very competitive MaxAPE values (below 17%) for 1-h-ahead forecasts.

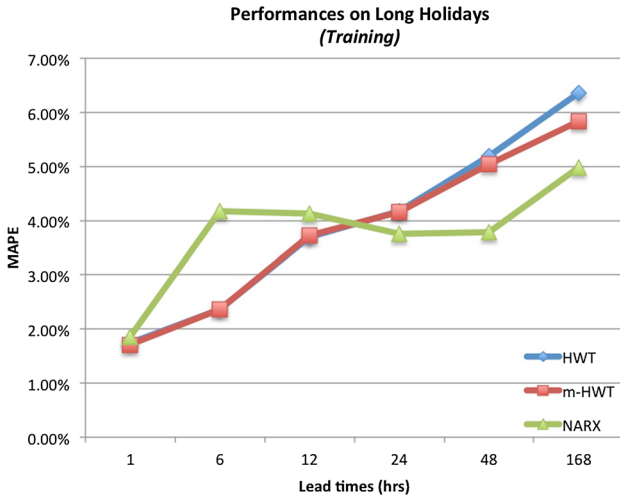


Fig. 21 Training performances on long holidays for the Noord dataset

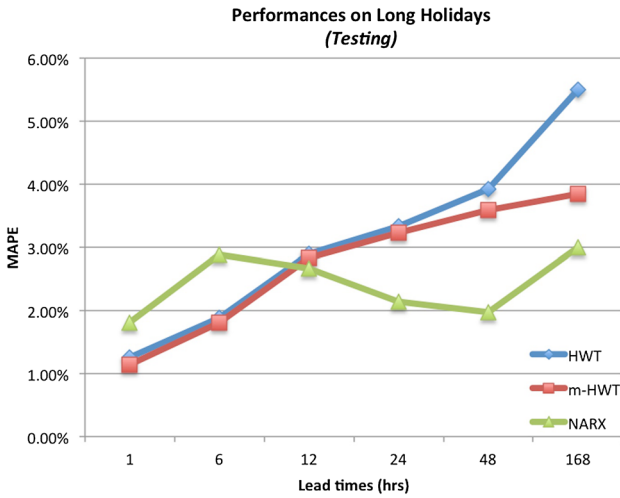
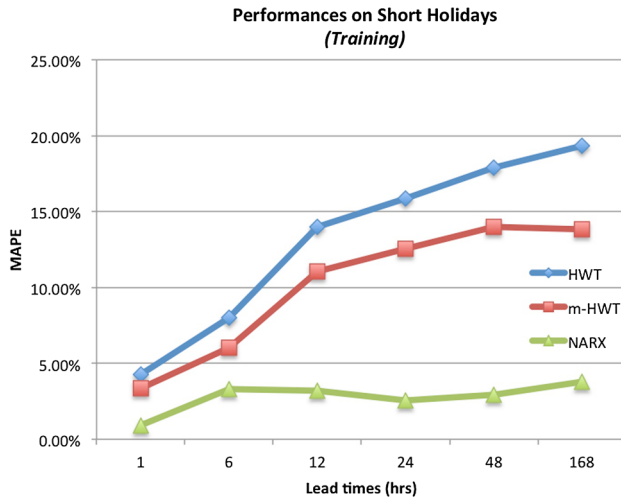
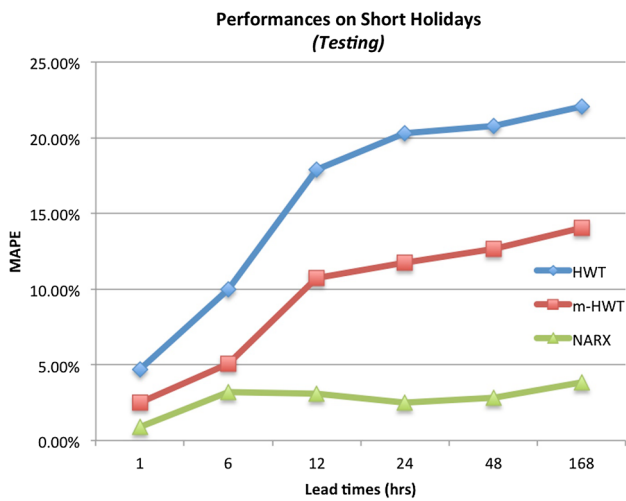


Fig. 22 Testing performances on long holidays for the Noord dataset



**Fig. 23** Training performances on short holidays for Noord dataset



**Fig. 24** Testing performances on short holidays for the Noord dataset



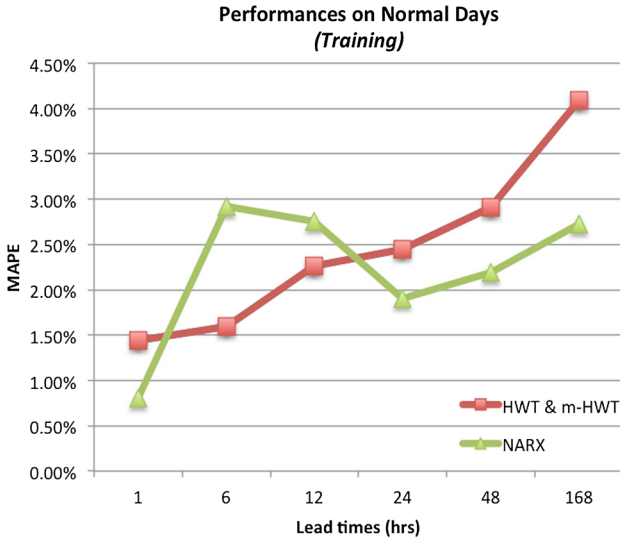


Fig. 25 Training performances on normal days for the Noord dataset

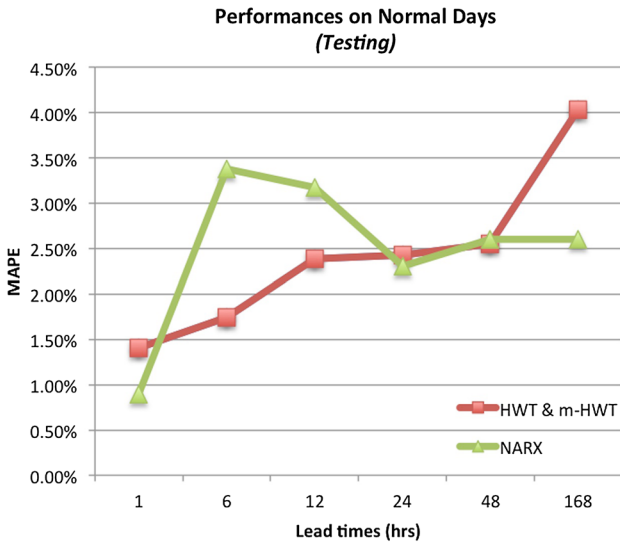


Fig. 26 Testing performances on normal days for the Noord dataset

**Table 5** Stability analysis of NARX for each region and lead time

	Brabant (%)	Limburg (%)	Maastricht (%)	Friesland (%)	Noord (%)
Lead times					
1	94.67	91.86	93.81	95.20	94.44
6	84.96	91.61	84.75	83.82	91.52
12	94.76	96.15	87.58	88.71	91.69
24	89.32	89.76	91.51	99.30	87.71
48	91.64	88.92	94.35	95.47	92.97
168	98.62	93.14	99.40	87.58	99.33

**Table 6** MaxAPE for each region and lead time

Lead times (h)	Brabant			Noord		
	NARX (%)	HWT (%)	m-HWT (%)	NARX (%)	HWT (%)	m-HWT (%)
1	9.04	25.88	15.90	11.22	24.63	19.51
6	24.03	33.93	24.54	34.33	28.64	28.74
12	17.94	56.04	46.78	26.01	53.32	50.58
24	15.26	60.81	48.67	21.05	55.22	52.41
48	33.23	59.68	48.88	41.58	55.69	54.11
168	28.83	59.39	51.19	33.18	53.92	48.30

Lead times (h)	Limburg			Maastricht			Friesland		
	NARX (%)	HWT (%)	m-HWT (%)	NARX (%)	HWT (%)	m-HWT (%)	NARX (%)	HWT (%)	m-HWT (%)
1	9.67	28.42	21.41	11.65	27.18	23.89	16.61	31.56	18.96
6	29.88	32.19	22.78	31.58	35.51	31.78	31.29	41.64	38.88
12	29.22	50.35	36.24	24.57	58.87	47.59	31.72	68.43	57.58
24	35.61	51.31	37.04	15.50	64.05	53.11	22.29	69.55	58.46
48	16.87	50.76	39.51	29.68	62.45	53.88	31.11	68.23	65.40
168	18.80	48.80	37.03	23.49	63.30	44.92	16.80	67.60	64.14

## 6 Conclusion

In this paper, we compare a modification of the exponential smoothing method (m-HWT) and a nonlinear autoregressive exogenous input neural network model (NARX) for short-term electricity load forecasting. We also compare our models with Taylor's HWT adaptation. One of the main motivations of this research is to examine whether complex artificial-intelligence-based methods such as NARX may perform better than simple time-series-based approaches for short-term electricity load forecasting.

Our findings indicate that NARX is significantly more accurate than time-series-based methods for 1-h-ahead forecasting. The MAPE values of NARX for 1-h-ahead forecasting ranges between only 0.75 and 1.25% for all regions we tested. As a computational-intelligence-based model, we show that NARX is very effective in capturing complex and nonlinear effects of special days in electricity infeed forecasting. We observe that NARX

is drastically superior to m-HWT in terms of MaxAPE performance for all forecasting lead times. We believe that this finding will make NARX very attractive to risk-averse decision makers in electricity markets.

Finally, our modification to Taylor's HWT method relaxes Taylor's no-special-day assumption and significantly improves forecast accuracy. Despite its simplicity, m-HWT usually outperforms NARX for 6- and 12-h-ahead forecasts. We believe that our proposed methods can provide significant financial savings to parties who are in need of accurate short-term electricity load forecasts.

In this paper, we have implemented our modified HWT and NARX on the electricity load data from the Dutch market to test the performance of the proposed approaches. Although electricity demand patterns may differ across countries, they share some fundamental similarities, and hence our results and analysis can be fine-tuned to work with demand data from other countries as well. For example, most of the special days we consider for the Dutch market are not applicable to the Turkish market. Nevertheless, in Turkey there are other special days including religious and national holidays, which have very similar demand patterns to the Dutch special days. For example, the drop in demand during the Republic Day in Turkey is very similar to the one during Independence Day in the Netherlands. Hence, to work with a different demand dataset one can simply calibrate our models with the new dataset's special days. Overall, the algorithms provided in this paper can be easily generalized to electricity markets in other countries by appropriately replacing the existing special days in the current models with the ones special to other electricity datasets.

We believe that other more advanced forecasting methods such as multivariate adaptive regression splines and quantile regression can be explored for electricity load forecasting (Koc and Iyigun 2014). The former is a non-parametric methodology that includes an extension of recursive partitioning that uses linear functions for local fit, and it can be successful in capturing nonlinear relationships in electricity load data.

## Appendix A: M-HWT model parameters

See Tables 7, 8, 9, 10.

**Table 7** Parameters for m-HWT models of Maastricht data set

Lead times (h)	$\alpha_{best}$	$\delta_{best}$	$\omega_{best}$	$\phi_{best}$
1	0.6099	0.1877	0.2070	0.2839
6	0.0528	0.1888	0.1300	0.7814
12	0.0262	0.1800	0.0894	0.8017
24	0.0043	0.2087	0.0729	0.7990
48	0.0004	0.0905	0.0442	0.9183
168	0.0001	0.0068	0.2103	0.9753

**Table 8** Parameters for m-HWT models of Limburg data set

Lead times (h)	$\alpha_{best}$	$\delta_{best}$	$\omega_{best}$	$\phi_{best}$
1	0.5623	0.1912	0.2168	0.3157
6	0.0433	0.1927	0.1305	0.7990
12	0.0203	0.2036	0.0735	0.8581
24	0.0023	0.2744	0.0943	0.9309
48	0.0035	0.0635	0.0719	0.9214
168	0.0009	0.0499	0.1648	0.8870

**Table 9** Parameters for m-HWT models of Friesland data set

Lead times (h)	$\alpha_{best}$	$\delta_{best}$	$\omega_{best}$	$\phi_{best}$
1	0.6107	0.1966	0.2430	0.2738
6	0.0262	0.1614	0.1662	0.8236
12	0.0115	0.2000	0.1213	0.7941
24	0.0006	0.2378	0.1423	0.8906
48	0.0035	0.0635	0.0719	0.9214
168	0.0001	0.0947	0.2340	0.9477

**Table 10** Parameters for m-HWT models of Noord data set

Lead times (h)	$\alpha_{best}$	$\delta_{best}$	$\omega_{best}$	$\phi_{best}$
1	0.5358	0.2287	0.2070	0.3493
6	0.0357	0.1884	0.1572	0.7887
12	0.0124	0.2078	0.1110	0.8236
24	0.0043	0.2087	0.0729	0.7990
48	0.0001	0.1435	0.0571	0.9396
168	0.0004	0.0068	0.2103	0.9753

## Appendix B: NARX architectures

See Tables 11, 12, 13.

**Table 11** Maastricht data set model architectures for different forecasting horizons

Lead times (h)	Hidden layer	Number of hidden nodes
1	1	25
6	1	60
12	1	60
24	1	50
48	1	20
168	1	20

**Table 12** Limburg data set model architectures for different forecasting horizons

Lead times (h)	Hidden layer	Number of hidden nodes
1	1	20
6	1	30
12	1	70
24	1	55
48	1	20
168	1	30

**Table 13** Friesland data set model architectures for different forecasting horizons

Lead times (h)	Hidden layer	Number of hidden nodes
1	1	5
6	1	30
12	1	20
24	1	30
48	1	30
168	1	25

## Appendix C: Model performance for different datasets

See Tables 14, 15, 16.

**Table 14** Performances of NARX, HWT and m-HWT for different day types in terms of MAPE for Limburg data set

	Lead times												Avg				
	1	6	12	24	48	168	Testing (%)	Training (%)	Testing (%)	Training (%)	Testing (%)	Training (%)					
<b>Total</b>																	
HWT	2.12	1.95	2.06	2.57	2.06	3.71	2.49	3.03	2.65	4.64	3.27	4.75	3.76	3.47	2.70		
m-HWT	2.06	1.82	1.96	2.57	1.96	3.67	2.32	2.92	2.43	4.53	2.98	4.48	3.39	3.37	2.48		
NARX	0.79	0.86	2.86	2.62	2.86	2.25	2.34	1.84	2.05	2.97	3.34	2.85	3.06	2.22	2.42		
<b>Special</b>																	
HWT	3.14	2.21	2.32	3.38	2.32	5.30	3.18	5.25	3.72	7.52	4.71	8.55	5.80	5.52	3.66		
m-HWT	2.78	1.70	1.94	3.31	1.94	5.04	2.56	4.61	2.89	6.75	3.63	7.18	4.45	4.95	2.86		
NARX	0.86	0.80	2.89	2.81	2.89	2.50	2.41	2.22	2.21	3.56	3.78	3.46	3.10	2.57	2.53		
<b>Long holidays</b>																	
HWT	2.25	1.61	1.78	2.65	1.78	3.91	2.17	3.49	2.51	5.29	3.50	6.28	4.80	3.98	2.73		
m-HWT	2.17	1.49	1.75	2.73	1.75	3.89	2.12	3.36	2.36	5.04	3.01	5.54	3.87	3.79	2.43		
NARX	1.26	1.64	2.97	3.57	2.97	3.29	2.45	2.74	2.67	3.68	2.74	4.20	2.24	3.12	2.45		
<b>Short holidays</b>																	
HWT	7.52	9.00	8.49	6.95	8.49	12.08	14.63	13.86	17.46	18.38	18.51	19.62	17.11	13.07	14.20		
m-HWT	5.72	4.15	4.15	6.16	4.15	10.66	7.48	10.74	8.95	15.10	10.73	15.19	11.12	10.59	7.76		
NARX	0.85	0.79	2.89	2.81	2.89	2.51	2.42	2.23	2.20	3.59	3.86	3.47	3.16	2.58	2.55		
<b>Non-Spec</b>																	
HWT	1.87	1.86	1.96	2.37	1.96	3.30	2.22	2.45	2.25	3.92	2.73	3.76	2.99	2.95	2.33		
m-HWT	1.87	1.86	1.96	2.37	1.96	3.30	2.22	2.45	2.25	3.92	2.73	3.76	2.99	2.95	2.33		
NARX	0.77	0.88	2.85	2.55	2.85	2.17	2.32	1.70	1.98	2.75	3.15	2.64	3.00	2.10	2.36		

**Table 15** Performances of NARX, HWT and m-HWT for different day types in terms of MAPE for Friesland data set

	Lead times												Avg				
	1	6	12	24	48	168	Testing (%)	Training (%)	Testing (%)	Training (%)	Testing (%)	Training (%)					
<b>Total</b>																	
HWT	1.96	1.81	2.64	3.42	3.15	3.45	3.15	3.85	3.45	3.86	3.50	3.01					
m-HWT	1.94	1.75	2.52	3.38	2.94	3.74	3.19	3.49	3.53	3.49	2.81						
NARX	1.19	1.25	3.46	3.38	3.81	2.87	2.85	2.95	3.09	3.30	2.70	2.96					
<b>Special</b>																	
HWT	2.28	1.85	3.44	5.67	4.74	5.97	4.95	6.83	5.36	7.82	5.66	4.42					
m-HWT	2.17	1.62	2.99	5.49	3.96	5.65	4.13	6.23	4.39	7.02	5.55	3.67					
NARX	1.35	1.40	3.26	3.83	3.71	3.35	3.00	3.56	3.56	3.47	3.13	3.07					
<b>Long holidays</b>																	
HWT	1.81	1.53	2.63	4.20	3.19	4.33	3.24	5.02	3.60	5.95	4.34	3.15					
m-HWT	1.82	1.44	2.60	4.33	3.08	4.38	3.15	4.93	3.32	5.75	4.53	2.91					
NARX	2.49	2.50	3.18	4.94	3.71	4.84	2.37	4.44	3.07	4.64	4.20	3.02					
<b>Short holidays</b>																	
HWT	5.23	5.47	12.71	17.23	22.62	18.87	24.63	21.14	25.58	22.55	15.92	18.97					
m-HWT	4.35	3.63	7.45	14.63	14.02	15.63	15.49	16.45	16.77	16.98	13.56	12.44					
NARX	1.28	1.34	3.26	3.82	3.71	3.34	3.05	3.58	3.60	3.46	3.12	3.08					
<b>Non-Spec</b>																	
HWT	1.87	1.80	2.42	2.83	2.70	2.80	2.63	3.09	2.89	3.81	2.95	2.60					
m-HWT	1.87	1.80	2.42	2.83	2.70	2.80	2.63	3.09	2.89	3.81	2.95	2.60					
NARX	1.13	1.18	3.55	3.21	3.84	2.71	2.76	2.74	2.90	2.70	2.55	2.90					

**Table 16** Performances of NARX, HWT and m-HWT for different day types in terms of MAPE for Maastricht data set

	Lead times												Avg				
	1	6	12	24	48	168	Testing (%)	Training (%)	Testing (%)	Training (%)	Testing (%)	Training (%)					
<b>Total</b>																	
HWT	1.86	1.75	2.36	2.23	3.52	3.23	3.90	3.47	4.67	3.90	3.90	4.86	5.41	4.86	3.62	3.24	
m-HWT	1.82	1.69	2.30	2.10	3.43	3.05	3.76	3.20	4.47	3.46	3.46	4.05	5.11	4.05	3.48	2.93	
NARX	1.06	1.13	2.89	3.41	2.89	3.30	2.37	2.59	3.17	3.36	3.36	3.35	3.33	3.35	2.62	2.86	
<b>Special</b>																	
HWT	2.35	1.82	3.51	2.63	5.69	4.03	6.93	4.89	8.56	5.99	5.99	7.18	9.68	7.18	6.12	4.42	
m-HWT	2.14	1.58	3.17	2.19	5.16	3.35	6.08	3.93	7.26	4.40	4.40	4.25	8.01	4.25	5.30	3.28	
NARX	1.13	1.18	3.19	3.61	3.20	3.62	2.84	3.10	3.79	3.84	3.84	4.02	4.21	4.02	3.06	3.23	
<b>Long holidays</b>																	
HWT	1.89	1.56	2.57	2.00	4.02	2.83	4.75	3.35	5.89	4.45	4.45	5.62	6.94	5.62	4.34	3.30	
m-HWT	1.83	1.48	2.54	1.96	3.96	2.77	4.54	3.15	5.40	3.58	3.58	3.48	6.02	3.48	4.05	2.74	
NARX	1.46	1.65	3.85	3.35	3.64	3.34	3.34	2.79	4.15	2.64	2.64	2.75	4.80	2.75	3.54	2.75	
<b>Short holidays</b>																	
HWT	4.60	4.75	8.07	9.77	13.82	17.56	17.60	22.30	21.62	23.47	23.47	24.82	23.09	24.82	14.80	17.11	
m-HWT	3.66	2.65	6.27	4.82	11.03	9.92	13.58	12.74	16.36	13.71	13.71	13.01	17.70	13.01	11.43	9.48	
NARX	1.11	1.16	3.21	3.63	3.22	3.63	2.86	3.13	3.82	3.94	3.94	4.11	4.22	4.11	3.07	3.27	
<b>Non-Spec</b>																	
HWT	1.74	1.73	2.07	2.07	2.97	2.93	3.13	2.93	3.70	3.11	3.11	3.98	4.32	3.98	2.99	2.79	
m-HWT	1.74	1.73	2.07	2.07	2.97	2.93	3.13	2.93	3.70	3.11	3.11	3.98	4.32	3.98	2.99	2.79	
NARX	1.04	1.10	2.79	3.33	2.78	3.20	2.20	2.37	2.94	3.09	3.09	3.08	3.02	3.08	2.46	2.70	



## References

- Al-Saba, T., & El-Amin, I. (1999). Artificial neural networks as applied to long-term demand forecasting. *Artificial Intelligence in Engineering*, 13, 189–197.
- Bianco, V., Manca, O., & Nardini, S. (2009). Electricity consumption forecasting in Italy using linear regression models. *Energy*, 34, 1413–1421.
- Bianco, V., Manca, O., Nardini, S., & Minea, A. A. (2010). Analysis and forecasting of nonresidential electricity consumption in Romania. *Applied Energy*, 87(11), 3584–3590.
- Bodén, M. (2002). A guide to recurrent neural networks and backpropagation. In the Dallas Project, SICS Technical Report, 2002, 03.
- Cancelo, J. R., Espasa, A., & Grafe, R. (2008). Forecasting the electricity load from one day to one week ahead for the Spanish system operator. *International Journal of Forecasting*, 24(4), 588–602.
- Catalão, J., Mariano, S., & Mendes, V. (2007). Short-term electricity prices forecasting in competitive market: A neural network approach. *Electric Power Systems Research*, 77(10), 1297–1304.
- Chen, H., Cañizares, C. A., & Singh, A. (2001). ANN-based short-term load forecasting in electricity markets. In: *Power engineering society winter meeting* (Vol. 2, pp. 411–415). Columbus, OH: IEEE.
- Connor, J. T., Martin, R. D., & Atlas, L. E. (1994). Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, 5(2), 240–54.
- Czernichow, T., Germond, A., Dorizzi, B., & Caire, P. (1995). Improving recurrent network load forecasting. In: *Proceedings of the IEEE international conference ICNN '95* (pp. 899–904).
- Duda, R. O., Hart, P. E., & Stork, D. G. (1997). *Pattern classification*. New York: Wiley.
- Elias, R. S., Fang, L., & Wahab, M. I. M. (2011). Electricity load forecasting based on weather variables and seasonalities: A neural network approach. In *8th international conference on service systems and service management (ICSSSM)* (pp. 1–6). IEEE.
- Engle, R., & Manganelli, S. (1999). *CAViaR: Conditional value at risk by quantile regression*. Cambridge, MA: National Bureau of Economic Research.
- Espinoza, M., Suykens, J. A. K., Belmans, R., & De Moor, B. (2007). Using kernel-based modeling for nonlinear system identification. *IEEE Control Systems Magazine*, 43–57.
- Gelper, S., Fried, R., & Croux, C. (2010). Robust forecasting with exponential and Holt-Winters smoothing. *Journal of Forecasting*, 300, 285–300.
- Hagan, M. T., & Behr, S. M. (1987). The time series approach to short term load forecasting. *IEEE Transactions on Power Systems*, 2(3), 785–791.
- Hahn, H., Meyer-Nieberg, S., & Pickl, S. (2009). Electric load forecasting methods: Tools for decision making. *European Journal of Operational Research*, 199(3), 902–907.
- Hippert, H. S., Pedreira, C. E., & Souza, R. C. (2001). Neural networks for short-term load forecasting?: A review and evaluation. *IEEE Transactions on Power Systems*, 16(1), 44–55.
- Kim, M. S. (2013). Modeling special-day effects for forecasting intraday electricity demand. *European Journal of Operational Research*, 230(1), 170–180.
- Koc, E., & Iyigun, C. (2014). Restructuring forward step of MARS algorithm using a new knot selection procedure based on a mapping approach. *Journal of Global Optimization*, 60(1), 79–102.
- Kyriakides, E., & Polycarpou, M. (2007). Short-term electricity load forecasting: A tutorial. *Trends in Neural Computation, Studies in Computational Intelligence*, 34, 319–418.
- Lee, Y.-S., & Tong, L.-I. (2012). Forecasting nonlinear time series of energy consumption using a hybrid dynamic model. *Applied Energy*, 94, 251–256.
- Soares, L. J., & Medeiros, M. C. (2005). Modelling and forecasting short-term electricity load: A two step methodology (No. 495).
- Tanrisever, F., Derinkuyu, K., & Heeren, M. (2013). Forecasting electricity infeed for distribution system networks: An analysis of the Dutch case. *Energy*, 58, 247–257.
- Taylor, J. W. (2003). Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society*, 54(8), 799–805.
- Taylor, J. W. (2010a). Triple seasonal methods for short-term electricity demand forecasting. *European Journal of Operational Research*, 204(1), 139–152.
- Taylor, J. W. (2010b). Exponentially weighted methods for forecasting intraday time series with multiple seasonal cycles. *International Journal of Forecasting*, 26(4), 627–646.
- Taylor, J. W. (2012). Short-term load forecasting with exponentially weighted methods. *IEEE Transactions on Power Systems*, 27(1), 458–464.
- Taylor, J. W., De Menezes, L. M., & McSharry, P. E. (2006). A comparison of univariate methods for forecasting electricity demand up to a day ahead. *International Journal of Forecasting*, 22(1), 1–16.
- Taylor, J. W., Mcsharry, P. E., & Member, S. (2008). Short-term load forecasting methods? An evaluation based on European data. *IEEE Transactions on Power Systems*, 22(4), 2213–2219.

- Varghese, S. S., & Ashok, S. (2012). Performance comparison of ANN models for short-term load forecasting. In *International conference on electrical engineering and computer science* (pp. 97–102).
- Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks? The state of the art. *International Journal of Forecasting*, *14*, 35–62.