Erasmus University Rotterdam

Erasmus School of Economics

Bachelor Thesis

Econometrics and Operational Research

Improving short-term load forecasting accuracy with novel hybrid models after multiple seasonal and trend decomposition

Author
Jill Langenberg
Student ID: 482410

Supervisor
Sebastiaan Vermeulen
Second Assessor
Jochem Oorschot

July 5, 2020

Abstract

This paper describes the improvement of the accuracy of short-term load forecasting through decomposition of the load time series and modeling each component separately. A novel Multiple Seasonal and Trend decomposition by LOESS algorithm is developed to decompose the time series into a base component and a weather-sensitive component. The base component is modeled with the double seasonal additive Holt Winter's method. The weather-sensitive component is modeled with linear regression, multivariate adaptive regression splines (MARS) and an artificial neural network. The forecasts from the three methods are each combined with the forecast from the double seasonal additive Holt Winter's method and form three different hybrid models. The hybrid model using MARS is the most accurate in forecasting short-term load. This novel hybrid model that combines double seasonal additive Holt Winter's and MARS could ensure that electrical companies are able to forecast their short-term load more accurately and reduce costs.

ERASMUS UNIVERSITY ROTTERDAM

Contents

1 Introduction

Accurate forecasts of load demand play an important role in the energy sector. The day-to-day operations of utility systems rely on short-term load forecasts, which is the prediction of the system load over an interval ranging from one day to two weeks (Wang, Zhang, & Chen, 2016). Market operators need these load forecasts to determine day-ahead market prices (Chen et al., 2009). Accurate forecasts are also needed to improve the economic efficiency of power system operation (Qiuyu et al., 2017). If an incorrect prediction is made, it can lead to either a load over- or underestimation. A load overestimation means an excess in supply and consequently more costs. In addition, producing more energy than needed leads to wasted use of fossil fuels and an unnecessary increase in emissions (Sarkodie & Strezov, 2018). Underestimation of the load is also costly, since more costly supplementary services are then needed to provide more electricity (Bianchi, Maiorino, Kampffmeyer, Rizzi, & Jenssen, 2017). Therefore, it is necessary that accurate forecasts of short-term load be made. Much research has already been conducted in the field of short-term load forecasting, but the forecast methods still require more accuracy.

Hourly load time series are influenced by multiple factors. For example, hourly load series contain a within-day and within-week seasonal cycle (Taylor, 2003), resulting in a double seasonal pattern in the short-term load. In addition, weather conditions play a role in the amount of short-term load (Son & Kim, 2017). Capturing all the different factors with one model is difficult. However, it may be possible to improve the accuracy of forecasts by combining multiple individual forecasts (Clemen, 1989). Extending this idea, a combination of different forecasting algorithms could be a solution to improve accuracy. If we decompose the data into multiple components we isolate the important features of the data into sub-series. We can then use different techniques to forecast each component. Such a combination of multiple forecast methods, resulting in a hybrid model, will capture the specific aspects of the data and improve the forecast accuracy (Theodosiou, 2011).

This gives rise to the research question of this paper: **'Which hybrid model is the most accurate for short-term load forecasting when using a decomposition-ensemble algorithm for the hourly load time series: the double seasonal additive Holt Winter's method combined with linear regression, with multivariate adaptive regression splines or with an artificial neural network?'** In order to answer this research question, we first need to know which decomposition algorithm works best for short-term hourly load data. After the hourly load data is decomposed with the chosen algorithm, a forecasting method must be chosen for each of the different components. Next, we must determine which variables are of interest for forecasting each component. Finally, we can answer the research question by comparing all the hybrid models based on different criteria, to check which hybrid model forecasts most accurately.

To provide an answer to our research question, we use data obtained from ENTSO-E, Power Statistics (2019) and KNMI, Koninklijk Nederlands Meterologisch Instituut (2020). The data sets contain the hourly load demand and hourly values of different weather variables in the Netherlands,

respectively, for the period April 30 2009 to April 30 2019. A period of ten years is chosen because the patterns in the load data remain similar over the years, not showing a lot of volatility. This is due to the fact that electricity is used on a daily basis and seen as an essential necessity in our life (Zohuri & McDaniel, 2019). The models are trained with a train data set and tested with a test data set. The train data set is equal to the first nine years of the complete data set and the test data set is equal to the last year. A longer period is used for the train data set to ensure that the models are trained properly and with enough data.

Several different methods are used and extended to answer the research question. First, an extended decomposition method based on the already existing Seasonal and Trend decomposition by LOESS method (STL) developed by Cleveland, Cleveland, McRae, & Terpenning (1990), is described in this paper. This novel decomposition method can take multiple seasonal cycles into account and is called Multiple Seasonal and Trend decomposition by LOESS (MSTL). After decomposing the load data into multiple components, each component can be forecasted with a different algorithm. The base component, containing the trend and seasonality components (Qiuyu et al., 2017), is forecasted with the double seasonal additive Holt Winter's method. This is an extension of the standard Holt Winter's method. The extension ensures that not one but two seasonal cycles can be described in an additive way. The other component, defined as the weather-sensitive component (Qiuyu et al., 2017), is forecasted by using three different methods: namely, Linear Regression (LR), Multivariate Adaptive Regression Splines (MARS) and an Artificial Neural Network (ANN). These three methods are also used by Nalcaci, Özmen, & Weber (2019) to model the electric load, while looking at the relationship between weather variables and electric load. The forecasts from the three methods are each combined with the forecast from the double seasonal additive Holt Winter's method and form three different hybrid models. The three hybrid models are compared with each other to see which hybrid model ensures a more accurate forecast of the hourly load.

From the three developed hybrid models, the hybrid model using MARS is the most accurate in forecasting the short-term load and the hybrid model using ANN is the least accurate. The hybrid model using MARS is slightly more accurate than the hybrid model using LR, however the differences are minor. The novel hybrid model that combines double seasonal additive Holt Winter's and MARS could ensure that electrical companies are able to forecast their short-term load more accurately and reduce costs.

A lot of literature is already available about forecasting short-term load, but not much research has been done on decomposing the load time series first and subsequently modeling the components separately. By extending an already existing decomposition method, MSTL is developed and we are able to take multiple seasonality cycles into account when decomposing the load time series. Thus, a more accurate representation of all the different components in hourly load data can be made. In addition, combining the double seasonal additive Holt Winter's method with either LR, MARS or ANN results in three novel hybrid models to model hourly load. This research thus provides added value to the already existing literature and may suggest further work for other researchers.

In this paper, Section 2 provides a literature review. This section elaborates on existing literature about short-term load forecasting and also shows which decomposition method can be used best. Section 3 describes the data used in this research. Next, Section 4 describes the different used models and evaluation criteria. Section 5 presents the results of the different models and compares them with each other. Finally, Section 6 draws a conclusion and answers the research question.

2 Literature Review

The hourly load time series is influenced by multiple factors. Firstly, we see that hourly load time series contain a within-day and within-week seasonal cycle (Taylor, 2003). Thus, we have a double seasonal pattern in the short-term load time series. In addition, weather conditions play a role in the amount of short-term load (Son & Kim, 2017). Developing a model that can accurately forecast all these different components is difficult and almost impossible, since every model has its shortcomings. A hybrid model, that uses different forecasting algorithms, could be a possible solution to overcome these shortcomings (Clemen, 1989). In order to be able to apply a hybrid model, typically the data is decomposed, then the hybrid model is applied and finally the forecasts are assembled again (X. Zhang & Wang, 2018).

Although the decomposition of the load data can be done in numerous ways, an extended version of the Seasonal and Trend decomposition by LOES (STL) is used to decompose the hourly data in this research. The classical approaches for decomposing a time series are the STL method, the X11 method and Seasonal Extraction in ARIMA Time Series (SEATS). Both the SEATS and X11 methods are suitable for only quarterly and monthly data (Dagum & Bianconcini, 2016). Since we use hourly data in this research, SEATS and X11 are not suitable for the data of this research. In addition, STL has a strong resilience to outliers in data, which results in a robust component sub-series (Theodosiou, 2011). This robustness can translate into more accurate forecasts. The STL method developed by Cleveland et al. (1990) decomposes a time series into three components: the trend, seasonal and remainder components. However, our hourly data contains the two aforementioned within-day and within-week seasonal cycle. Therefore, multiple STL (MSTL) needs to be developed in which the time series is decomposed into four components: trend, seasonal 1, seasonal 2 and remainder component. This paper describes this novel MSTL method.

Previous research by Qiuyu et al. (2017) shows a load decomposition based on the standard STL. They consider the trend and seasonal component as one component, the base component. The remainder component is seen as the weather-sensitive component. Both components can be estimated with different types of algorithms. This research follows the same approach as Qiuyu et al. (2017) and also considers the trend, seasonal 1 and seasonal 2 components as the base component and the remainder component as the weather-sensitive component.

Qiuyu et al. (2017) estimate the base component with the standard Holt Winter's method. The standard Holt Winter's method can model a seasonal pattern within a time series (Winters, 1960). It

combines exponential smoothing and state space methods. Since our load time series contains both a within-day and within-week seasonal cycle, the standard Holt Winter's method would not suffice. Taylor (2003) introduced a new method, in which a double seasonal multiplicative Holt Winter's method is suitable for a time series with two seasonal patterns. The double seasonal Holt Winter's method appears to outperform the standard Holt Winter's method when two seasonal cycles are present in a time series. Thus, the double seasonal Holt Winter's method is used for modeling the base component with two seasonal cycles. Only Taylor (2003) described the multiplicative version, whereas we need the double seasonal additive Holt Winter's method since we assume additive seasonality in our time series. In this research, we describe the additive version, a combination of the research done by Taylor (2003) and Qiuyu et al. (2017). This additive version, similar to the one described by Caiado (2010), is described in the methodology section of this paper.

Electricity consumption is also substantially affected by weather conditions (Hor, Watson, & Majithia, 2005). Both statistical and artificial intelligence based models have been developed to model the relationship between short-term load and exogenous weather variables. These methods include for example simple or multivariate linear regression, ARMA models, artificial neural networks and exponential smoothing (Singh, Khatoon, Muazzam, Chaturvedi, et al., 2012). Nalcaci et al. (2019) have used Linear Regression (LR), Multivariate Adaptive Regression Splines (MARS) and an Artificial Neural Network (ANN) to model the electric load, while looking at the relationship between weather variables and electric load. They found that the MARS model was the most accurate forecasting model. This paper partly replicates the research of Nalcaci et al. (2019) and therefore LR, MARS and ANN are used to model the weather-sensitive component.

In order to forecast the weather-sensitive component, different variables have to be chosen. Temperature is a common weather condition taken into consideration when forecasting the load. Friedrich & Afshari (2015) showed that using the four variables temperature, wind speed, humidity and global horizontal irradiation allowed for a more accurate load forecast than using only the temperature as variable. Therefore, these four weather variables are used in this research. Values of weather variables however, are not known in advance. Thus, they have to be forecasted in order to use them as exogenous variables when forecasting the short-term load. For this research however, the weather values are considered to be perfectly forecasted and no additional model is made for forecasting the weather variables.

3 Data

For this research, we look at the short-term load of the Netherlands. ENTSO-E, Power Statistics (2019) has data of the hourly load demand (in MW) for each country in Europe, including the Netherlands. The data available for the Netherlands ranges from 01 January 2006, 00:00, to 30 April 2019, 21:00, and consists of 116830 observations. This data set does not contain any missing values.

The data from 30 April 2009, 22:00, to 30 April 2019, 21:00, is used in this research. This set consists of 87648 observations and covers exactly 10 years in total. The data of hourly load is not as volatile as data of financial assets for example. The patterns in the load data remain similar over the years, not showing much volatility. This is due to the fact that electricity is used on a daily basis and seen as an essential necessity in our lives (Zohuri & McDaniel, 2019). Throughout the period of ten years we do see an increasing trend. New households and companies are the reason of this increase. However, the trend does not alter the patterns and therefore the whole period of ten years can be used in this research. The choice to not use all data provided by ENTSO-E, Power Statistics (2019), is based on ensuring that the computation time of the model is not too large.

Both a training and testing period are needed. The training period consists of N observations and the testing period has T observations. For the training period we look at the period 30 April 2009, 22:00, to 30 April 2018, 21:00. This is equal to nine years and 78888 observations: N=78888. The training period has six outliers. At six different times the value of the load of the Netherlands was equal to 0 MW. This is not realistic and therefore considered to be an outlier. The values are removed and linear interpolation is done to obtain new values for these times. The testing period is equal to one year, with the period 30 April 2018, 22:00, to 30 April 2019, 21:00. This is equal to 8760 observations: T=8760.

Table 3.1 contains some relevant summary statistics for the hourly load and Figure 3.1 shows the hourly load over time for the whole data set. It is clear from this figure that seasonal cycles are present in the hourly load. Our load data shows additive seasonality. In addition, we see a small trend in the data. Figure 3.2 displays the seasonal cycles even better. This graph shows the hourly data of two weeks, for the period 01 February 2019 to 14 February 2019. Both the within-day and within-week seasonal cycles are clearly visible.

	Load (MW)	**Temp. (°C)**	**Wind speed (0.1 m/s)**	**Sunlight (%)**	**Humidity (%)**
Mean	12806,21	10,70	33,71	20,33	80,10
Max.	18620	35,4	150	100	100
Min.	7206	-18,8	0	0	17

Table 3.1: Summary statistics for the hourly load data and weather variables for the period 30 April 2009 to 30 April 2019.

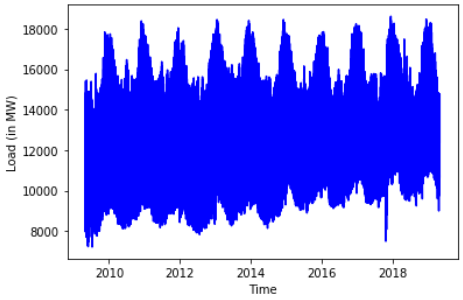


Figure 3.1: Hourly load data for the period 30 April 2009 to 30 April 2019.

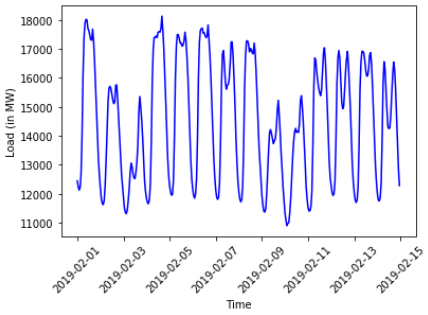


Figure 3.2: Hourly load data for the period 01 February 2019 to 14 February 2019.

For the weather conditions, KNMI, Koninklijk Nederlands Meterologisch Instituut (2020) reported hourly data for the temperature (in Celsius), average wind speed (in 0.1 metres per second), relative humidity (in percent) and amount of sunlight per hour (in percent). The data set for the weather conditions is available for the whole training and testing period and does not contain any missing values. The KNMI has 35 weather stations in the Netherlands, amongst which is weather station De Bilt. For general facts and figures about the weather of the Netherlands, weather station De Bilt is used. In this research, data from weather station De Bilt is also considered as representative for the whole of the Netherlands. Table 3.1 contains some relevant summary statistics for the weather variables, indicating that no outliers are present. Figure 7.4 in the Appendix displays the movement of the weather variables over time, also showing no outliers.

4 Methodology

Seasonal and Trend decomposition by LOESS (STL) decomposes a time series into a trend, seasonal and remainder component. STL has a simple design and can therefore be used easily for a large time series, without having a too large computational time (Cleveland et al., 1990). Since our data contains two seasonal patterns, multiple STL (MSTL) has to be developed where the time series is decomposed into four components: trend, seasonal 1, seasonal 2 and the remainder component. MSTL is described in Section 4.1. After decomposing the time series, we have a base and weather-sensitive component, as suggested by Qiuyu et al. (2017). The base component, consisting of the trend, seasonal 1 and seasonal 2 components, and the weather-sensitive component, equal to the remainder component, are both modeled separately. The base component is modeled by the double seasonal additive Holt Winter's method, described in Section 4.2. The weather-sensitive component is modeled in three different ways: with linear regression, multivariate adaptive regression splines and an artificial neural network, all described in Section 4.3. Since STL is an additive method, both modeled components can be added together to form the three final hybrid models. With the final hybrid models, forecasts can be made. Since it is useful for electrical companies to know 24 hours ahead what the load will be for their planning, we perform 24-hours ahead forecasts in our test

period. With the forecasts the evaluation criteria can be calculated, which are described in Section 4.4. Python (1991) is used for implementing all the different models.

4.1 Multiple Seasonal and Trend decomposition by LOESS

Cleveland et al. (1990) have developed the Seasonal and Trend decomposition by LOESS (STL) method, which is applicable to data with one seasonal pattern. In our data, however, we encounter two seasonal cycles, namely an inner-day and inner-week seasonal cycle. Therefore, we need a Multiple Seasonal and Trend decomposition by LOESS (MSTL), which decomposes the time series in a trend, seasonal 1, seasonal 2 and remainder component. To the best of our knowledge, no other research has been done on this topic yet. Below the novel method is described, based on an extension of the research of Cleveland et al. (1990).

MSTL is composed of an inner loop nested inside an outer loop, similar to STL. The inner loop consists of multiple seasonal and trend smoothing that updates the seasonal and trend components, respectively, in each pass. In total the inner loop is passed through $n_{(i)}$ times. The number of observations in the daily and weekly seasonal cycles are indicated with $n_{(d,c)}$ and $n_{(w,c)}$, respectively. This number also represents the total number of cycle-subseries. For example, the daily seasonal cycle consists of 24 observations, representing all hours in a day. All observations in the data set corresponding to the first hour of a day form a daily cycle-subseries. This is the case for all the 24 hours in a day and therefore we have in total 24, $n_{(d,c)}$, daily cycle-subseries.

Similar to the normal STL method, for MSTL the LOESS procedure is used for seasonal and trend smoothing. LOESS stands for locally weighted regression and is a generalization of the LOWESS procedure, which is an acronym for locally weighted scatterplot smoother (Jacoby, 2000). LOESS uses a smoothing parameter q for smoothing a component. For smoothing the daily and weekly cycle-subseries the smoothing parameters $q_{(d,s)}$ and $q_{(w,s)}$ are used, respectively. A low-pass filter is also applied with $q_{(l)}$ as smoothing parameter and the last smoothing parameter, $q_{(t)}$, is for smoothing the trend component.

When using LOESS, a neighborhood is defined for each data point x_t and the points in that neighborhood, $x_{i,t}$, are then weighted with neighborhood weights, $w_{i,t}$. These weights depend on the distance of the points in the neighborhood respective to the data points and are calculated with the tricube weight function. A polynomial of degree z is fitted to the neighborhood points. Either $z = 1$ or $z = 2$ can be used; the fitting is locally-linear or locally-quadratic, respectively. In this research we fit local-linear.

In the outer loop the inner loop is first walked through $n_{(i)}$ times and then the remainder component and robustness weights, ρ_t, are computed. The outer loop is done $n_{(o)}$ times. The inner loop provides us with estimates of the trend, T_t, daily seasonality, DS_t, and weekly seasonality component, WS_t. The remainder component is equal to $R_t = Y_t$ - T_t - DS_t - WS_t. The robustness weights are calculated based on this remainder component. These weights are used in the next run of the outer loop in the $n_{(i)}$ passes of the inner loop to reduce the influence of transient, aberrant

behaviour on the trend and seasonal components. The weights also reflect how extreme R_t is. An outlier in the data resulting in a large $|R_t|$ has a small or zero weight. The robustness weights at time t are equal to:

$$\rho_t = \frac{B(|R_t|)}{6 * median(|R_t|)}, \tag{1}$$

with $B(v)$ being equal to the bisquare weight function:

$$B(v) = \begin{cases} (1-v^2)^2 & \text{if } 0 \leq v < 1 \\ 0 & \text{if } v \geq 1 \end{cases} \tag{2}$$

The robustness weights ρ_t are multiplied with the neighbourhood weights $w_{i,t}$ when smoothing the daily seasonal, weekly seasonal and trend component in the inner loop.

Each pass of the outer loop consists of 3 steps, which results in the trend, daily seasonality, weekly seasonality and remainder component. The first step, which is divided into 6 sub-steps, represents the nested inner loop which is passed through $n_{(i)}$ times. Steps 2 and 3 refer to the remaining steps of the outer loop. For the initial pass through the nested inner loop the trend component is initialised as $T_t^{(0)} \equiv 0$. For the next pass through the inner loop, $T_t^{(j)}$ is equal to the found trend component in the previous inner loop. For the initial pass through the outer loop, the robustness weights are initialised as $\rho_t^{(0)} \equiv 1$. For the next pass through the outer loop, $\rho_t^{(k)}$ is equal to the found robustness weights in the previous outer loop.

For illustration, in the (k+1)st pass of the outer loop, the trend component, $T_t^{(k+1)}$, seasonal components, $DS_t^{(k+1)}$ and $WS_t^{(k+1)}$, and remainder component, $R_t^{(k+1)}$, for the time series Y_t with t=1 to N for the train period, or to T for the test period, are computed in the following way:

Step 1: Perform the nested inner loop $n_{(i)}$ times, resulting in the values for the trend, $T_t^{n_{(i)}} \equiv T_t^{(k+1)}$, daily seasonal, $DS_t^{n_{(i)}} \equiv DS_t^{(k+1)}$, and weekly seasonal component, $WS_t^{n_{(i)}} \equiv WS_t^{(k+1)}$. The j^{th} pass of the inner loop is carried out as follows:

I: Detrend each time series: Y_t - $T_t^{(j-1)}$.

II.1: Smooth each daily cycle-subseries of the detrended series, $n_{(d,c)}$ daily cycle-subseries in total, using LOESS with $q = q_{(d,s)}$, $z = 1$ and the neighborhood weights multiplied by the robustness weights $\rho_t^{(k)}$. The smoothed values for all daily cycle-subseries form the temporary daily seasonal series $DC_t^{(j)}$.

II.2: Smooth each weekly cycle-subseries of the detrended series, $n_{(w,c)}$ weekly cycle-subseries in total, using LOESS with $q = q_{(w,s)}$, $z = 1$ and the neighborhood weights multiplied by the robustness weights $\rho_t^{(k)}$. The smoothed values for all weekly cycle-subseries form the temporary weekly seasonal series $WC_t^{(j)}$.

III: Apply a low-pass filter to both $DC_t^{(j)}$ and $WC_t^{(j)}$, which consists of multiple parts: a moving average of length $n_{(d,c)}$ or $n_{(w,c)}$, followed by another moving average of length

$n_{(d,c)}$ or $n_{(w,c)}$, followed by a moving average of length 3, respectively. Then a LOESS smoothing with $z = 1$ and $q = q_{(l)}$ is done, which results in outputs $DL_t^{(j)}$ and $WL_t^{(j)}$.
IV.1: Detrend the smoothed daily cycle-subseries, which results in the daily seasonal component: $DS_t^{(j)} = DC_t^{(j)}$ - $DL_t^{(j)}$.
IV.2: Detrend the smoothed weekly cycle-subseries, which results in the weekly seasonal component: $WS_t^{(j)} = WC_t^{(j)}$ - $WL_t^{(j)}$.
V: Deseasonalise the time series: Y_t - $DS_t^{(j)}$ - $WS_t^{(j)}$.
VI: Smooth the deseasonalised time series using LOESS with $q = q_{(t)}$, $z = 1$ and the neighborhood weights multiplied by the robustness weights $\rho_t^{(k)}$. These smoothed values are the trend component, $T_t^{(j)}$, of the jth loop.

Step 2: Calculate the remainder component: $R_t^{k+1} = Y_t$ - $T_t^{(k+1)}$ - $DS_t^{(k+1)}$ - $WS_t^{(k+1)}$.
Step 3: Calculate the robustness weights, $\rho_t^{(k+1)}$, based on R_t^{k+1}.

After $n_{(o)}$ iterations of the outer loop steps, including $n_{(i)}$ iterations nested within Step 1, we arrive at the estimates of the trend, daily seasonality, weekly seasonality and remainder components.

In summary, MSTL consists of an outer loop and a nested inner loop and within these loops 8 parameters are necessary with the following corresponding values:

- $n_{(d,c)} = 24$ (Number of observations in daily seasonal cycle)
- $n_{(w,c)} = 168$ (Number of observations in weekly seasonal cycle)
- $n_{(i)} = 1$ (Number of passes through the nested inner loop)
- $n_{(o)} = 5$ (Number of passes through the outer loop)
- $q_{(l)} = 0.5$ (Smoothing parameter for low-pass filter)
- $q_{(t)} = 0.6$ (Smoothing parameter for T_t)
- $q_{(d,s)} = 0.5$ (Smoothing parameter for DS_t)
- $q_{(w,s)} = 0.5$ (Smoothing parameter for WS_t)

The values of the parameters are chosen based on both Cleveland et al. (1990) his research and recommendations regarding the parameters, and an analysis of the data.

4.2 Modeling the base component: double seasonal additive Holt Winter's method

The double seasonal Holt Winter's method is used to forecast a time series with a double seasonal pattern, such as our base component. Taylor (2003) researched the double seasonal multiplicative Holt Winter's method. However, for this research we assume additive seasonality and thus need the double seasonal additive Holt Winter's method. Taylor (2003) suggested that the additive version can be developed in a similar way from the standard Holt–Winters method for additive

seasonality. This standard Holt Winters method for additive seasonality can be found in Qiuyu et al. (2017). Caiado (2010) has combined the researches of Taylor (2003) and Qiuyu et al. (2017) and developed the double seasonal additive Holt Winter's method with an extension to adjust for first-order autocorrelation. When only combining the researches of Taylor (2003) and Qiuyu et al. (2017) and not taking the extension for adjustment of first-order auto correlation as done by Caiado (2010) into account, the equations forming the double seasonal additive Holt Winter's method are equal to:

$$\textit{Level:} \quad L_t = \alpha(Y_{bc,t} - (D_{t-s_1} + W_{t-s_2})) + (1-\alpha)(L_{t-1} + T_{t-1}) \tag{3}$$

$$\textit{Trend:} \quad T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)T_{t-1} \tag{4}$$

$$\textit{Seasonality 1:} \quad D_t = \delta(Y_{bc,t} - (L_t + W_{t-s_2})) + (1-\delta)D_{t-s_1} \tag{5}$$

$$\textit{Seasonality 2:} \quad W_t = \omega(Y_{bc,t} - (L_t + D_{t-s_1})) + (1-\omega)W_{t-s_2} \tag{6}$$

$$\textit{Forecast:} \quad \hat{Y}_{bc,t+k} = L_t + k \cdot T_t + D_{t-s_1+k} + W_{t-s_2+k} \tag{7}$$

where α, γ, δ and ω are the smoothing parameters and $Y_{bc,t}$ is the value of the base component for time t. D_t and W_t are two separate seasonal indices for the daily and weekly seasonalities with a cycle of length s_1 and s_2, respectively. s_1 is equal to 24 and s_2 is equal to 168, representing the within-day and within-week cycle, respectively. In Equation 3, the level index at time t, L_t, is calculated by smoothing the current observation, $Y_{bc,t}$, to the sum of the within-day and within-week seasonality indices of the corresponding observations in the previous seasonal cycles, D_{t-s_1} and W_{t-s_2}, respectively. The trend at time t, T_t, is calculated in Equation 4 by smoothing the difference in the current and previous level, (L_t - L_{t-1}). In Equation 5, the index for the within-day seasonality, D_t, is calculated by smoothing $Y_{bc,t}$ to the sum of L_t and the within-week seasonal index in the previous seasonal cycle, W_{t-s_2}. In addition, in Equation 6, the index corresponding to the within-week seasonality, W_t, is calculated by smoothing $Y_{bc,t}$ to the sum of L_t and the within-day seasonal index of the previous seasonal cycle, D_{t-s_1}. Finally, the k-hours-ahead forecast is calculated in Equation 7.

The initial values, L_0, T_0, D_0 and W_0 are specified as described in Taylor (2003). The smoothing parameters are estimated by minimising the mean squared error (MSE) of the one-step-ahead forecast in the training set. The MSE is calculated as follows:

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(Y_{bc,t} - \hat{Y}_{bc,t})^2 \tag{8}$$

After minimising the MSE and receiving the estimates of the smoothing parameters, 24-hours ahead forecasts of the base component for the test period are done with these estimated smoothing parameters and Equations 3, 4, 5, 6 and 7.

4.3 Modeling the weather-sensitive component

To model the weather-sensitive component, Y_{wsc}, three methods are used: linear regression, multivariate adaptive regression splines and an artificial neural network. For each of these methods the same set of I explanatory variables are used: $\boldsymbol{X} = \{X_{avg.temp}, X_{avg.wind}, X_{humidity}, X_{sun}, X_{firstlag}\}$, in which each variable is an Nx1 vector for the train period and an Tx1 vector for the test period. $X_{firstlag}$ is equal to the first lag of the weather-sensitive component, $Y_{wsc,t-1}$. In this research we assume to have perfectly forecasted weather variables, thus each of these weather variables is known for the train and test period.

As mentioned earlier, we perform 24-hours ahead forecasts in our test period. The initial 24 observations of our test period are calculated, solely based on the last observation of the train period by performing a repeated 1-hour ahead forecast: With the last observation of the train period we forecast the first observation of the test period. Based on this forecast, we can forecast the second observation. We continue to do this until we have the forecast of the 24^{th} observation.

Thereafter, we can calculate the 24-hours ahead forecasts. For illustrating purposes, an example: at time t=1 we want to forecast the value of $Y_{wsc,t}$ at t=25 (a 24-hours ahead forecast). Since we take a lag of $Y_{wsc,t}$ as an explanatory variable, we need the value of $Y_{wsc,t}$ at t=24 in order to know the value of the forecast. Since we only know the values of $Y_{wsc,t}$ up until t =1, we need to forecast all values of $Y_{wsc,t}$ for times t=2 until t=25, all based solely on the value of $Y_{wsc,t}$ at t=1. This process is repeated for each time $t \in T$.

4.3.1 Linear regression

With linear regression a relationship between the dependent variable, the weather-sensitive component Y_{wsc}, and the explanatory variables is predicted. Since we have multiple explanatory variables, we need to use multivariate linear regression. The regression model for Y_{wsc} is:

$$Y_{wsc} = \boldsymbol{X}^T \beta + \epsilon, \tag{9}$$

where Y_{wsc} is equal to a Nx1 vector containing the hourly load for each time $t \in N$ and $\boldsymbol{X}$ is an IxN matrix with the I regressors. β is an Ix1 vector consisting of all coefficients and ϵ is a Nx1 vector with error terms. β is estimated with Ordinary Least Squares (OLS). The estimate of β in combination with the values of the explanatory variables $X_{avg.temp}$, $X_{avg.wind}$, $X_{humidity}$, X_{sun} and $X_{firstlag}$ are used to calculate the 24-hours ahead forecasts of the T observations in the test period.

4.3.2 Multivariate adaptive regression splines

The multivariate adaptive regression splines (MARS) method was introduced by Friedman (1991). MARS is a non-parametric and non-linear technique and used to estimate general functions of high-dimensional arguments. The training data set is partitioned into separate piecewise linear segments

with different gradients. The linear segments, splines, are connected together. The connection points between the splines are called knots. These splines are known as the basis functions, $B_i(x)$. The basis functions can have two forms. One possible form is:

$$c^+(x,\tau) = [x-\tau]_+, \qquad c^-(x,\tau) = [-x+\tau]_+, \qquad x,\tau \in \mathbb{R}. \tag{10}$$

The value of τ represents the value of the corresponding knot and the $+$ symbol indicates that only the positive parts are used, otherwise the function is equal to zero. These two functions can be called a reflected pair, illustrated in the appendix in Figure 7.3. The second possible form for a basis function is a product of two or more basis functions as defined in Equation 10. A basic MARS model for our weather-sensitive component then has the form:

$$Y_{wsc} = \theta_0 + \sum_{i=1}^{k} \theta_i B_i(\boldsymbol{X}) + \epsilon, \tag{11}$$

where B_i are the basis functions from Equation 10 or products of two or more such functions, θ_0 is the intercept and θ_i are the coefficients corresponding to the basis functions.

The MARS algorithm is the union of two sub-procedures, forward and backward stepwise algorithms. In the forward stepwise algorithm the knots are placed on random positions within the range of each explanatory variable, which defines a pair of basis functions. At each step in the algorithm, the model adapts the knot and its corresponding pair of basis functions to give the maximum reduction in sum-of-squares residual error. The process of adding basis functions continues until the maximum number is reached, K. This then results in over-fitting of the data and a very complicated model (W. Zhang & Goh, 2016).

The backward stepwise algorithm deletes the redundant basis functions, which contributes to the computational burden, without degrading the fit to the data. The MARS algorithm uses generalised cross-validation (GCV) for decreasing the computational burden (Nalcaci et al., 2019). GCV is defined as follows:

$$GCV(\mu) = \sum_{t=1}^{N} \frac{(Y_{wsc,t} - f_\mu(x_t))^2}{(1 - M(\mu)/N)^2}, \tag{12}$$

where μ is equal to the number of basis functions. The numerator is equal to the average-squared residual of the fit to the data and the denominator represents a penalty that accounts for the increased variance associated with increasing model complexity.

After decreasing the computational burden using GCV, the MARS algorithm creates a model that consists of vital non-repetitive basis functions. The resulting model can handle both linear and non-linear behaviour (W. Zhang & Goh, 2016). After training the MARS model an output function is given in the form of Equation 11. With this MARS model we are able to forecast the T observations in the test period.

4.3.3 Artificial neural network

An artificial neural network (ANN) has a layered structure, consisting of different artificial neurons, called nodes. Every neural network has an input layer and an output layer and some also have hidden layers. All the layers are connected with one another and the connections contain weights. These weights can be seen as the coefficients that determine the influence of the regressors. The weights can adjust during the learning process of the neural network (Hassoun et al., 1995). This learning process is similar to fitting a model: statistical estimation of model parameters, namely the weights.

In our ANN the input layer consists of I nodes, I being equal to the number of explanatory variables. Thus, each node in the input layer corresponds to a regressor. Our output layer consists of one node, representing the dependent variable we are modeling. The hidden layer mainly serves as a connection between the inputs and output. One hidden layer is meant for simple tasks and adding too many hidden layers would increase the training time. One hidden layer seems enough for our task, therefore one hidden layer is used in this research for the ANN. The hidden layer consists of 12 nodes. The architecture of our used neural network is depicted in Figure 4.1, in which the dots in the hidden layer indicate the additional 7 nodes.

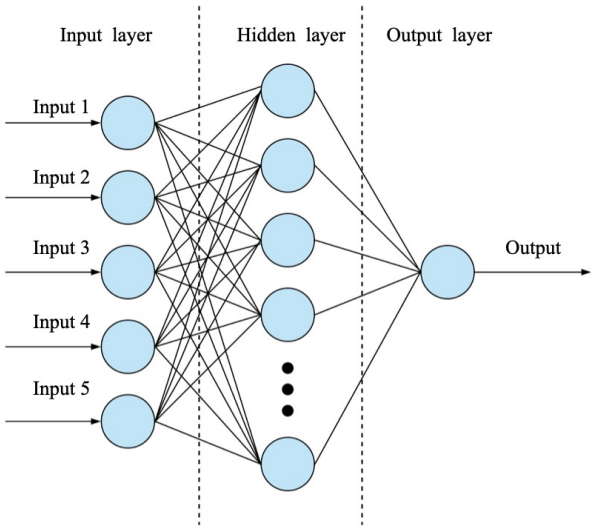


Figure 4.1: Architecture of the used artificial neural network.

This research will use the back-propagation algorithm, which is one of the most famous estimation algorithms to train multi-layered neural networks (Rumelhart et al., 1986), (Dillon et al., 1991). For both the hidden layer and output layer an activation function is needed. The activation functions are necessary to learn the complex patterns in the data and introduce non-linearity into the network.

In the nodes of the hidden layer we use the sigmoid function as activation function:

$$f(Z_i) = \frac{1}{1+e^{Z_i}}, \qquad \text{where } Z_i = \sum_{k=1}^{I} w_{i,k}x_k \text{ and } i \in \{1, ..., 12\}. \tag{13}$$

Z_i is equal to the input function for the i^{th} node in the hidden layer, in which x_k is equal to the k^{th} explanatory variable and $w_{i,k}$ is equal to the weight assigned to variable x_k in the i^{th} node of the hidden layer. For each node i in the hidden layer the value of $f(Z_i)$ is computed. Since we have a regression task where our output can be either negative or positive, our output layer uses the linear activation function:

$$g(Q) = Q = Y_{wsc}, \qquad \text{where } Q = \sum_{i=1}^{12} w_i f(Z_i). \tag{14}$$

In this equation w_i is equal to the weight assigned to the connection of the i^{th} node in the hidden layer to the output node. As a result, the output node gives the values of the weather-sensitive component, Y_{wsc}.

The back propagation algorithm results in a set of optimal weights which minimise the loss function. The most popular loss function for neural networks is the mean squared error (MSE), which is used for this ANN. The formula of the MSE is given in Equation 8, in which $Y_{bc,t}$ is now replaced by $Y_{wsc,t}$. Updating the weights, and with that minimising the MSE, is done using the mini-batch stochastic gradient descent, which is a typical choice when training neural networks (Ruder, 2016). The optimizer is the stochastic gradient descent (SGD) optimizer and the size of the mini-batch taken in this ANN is equal to 24. The process of updating the weights is repeated until all training inputs are exhausted (Khwaja et al., 2015). In order for the ANN to work we have to standardise the dependent and explanatory variables.

4.4 Comparing the models

After training all the models with the train data set, we forecast the values of the observations in the test period. These forecasts are then compared to the actual values of the test period using evaluation criteria. The model evaluation criteria used are the mean absolute percentage error (MAPE) defined in Equation 15, which is the most widely used error summary measure in electricity demand forecasting (Taylor, 2003), the root mean squared error (RMSE) defined in Equation 16, the R^2 defined in Heij et al. (2004) and the mean absolute error (MAE) defined in Equation 17.

$$MAPE = \frac{1}{T}\sum_{t=1}^{T}\left|\frac{Y_t - \hat{Y}_t}{Y_t}\right| \tag{15}$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(Y_t - \hat{Y}_t)^2} \tag{16}$$

$$MAE = \frac{1}{T}\sum_{t=1}^{T}|Y_t - \hat{Y}_t| \tag{17}$$

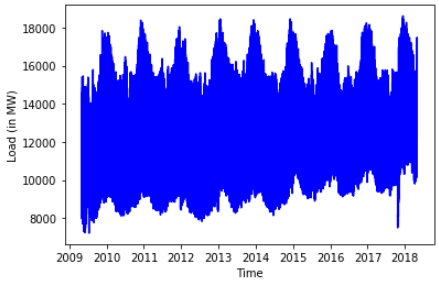
All these evaluation criteria tell us if a model is accurately forecasting the true values. A high R^2 value and low RMSE, MAPE and MAE values indicate an accurate model.

5 Results

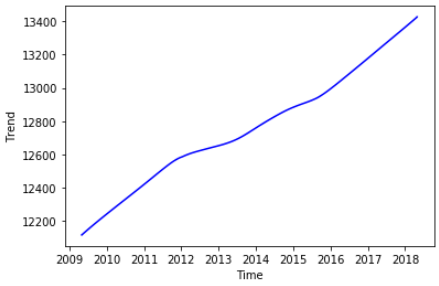
5.1 Decomposition of the load time series

After decomposing the load time series in the train period with MSTL we get four different components: the trend, daily seasonality, weekly seasonality and remainder component, displayed in Figure 5.1. As a reference the original load time series for the train period is also included. We observe an increasing trend in Figure 5.1b, indicating a load increase over time. This is due to increasing population and new technologies such as electric cars, smart grids and renewable energy production (Nalcaci et al., 2019). Figures 5.1c and 5.1d show the daily and weekly seasonality cycles, respectively. These two figures are both differently scaled than Figures 5.1a, 5.1b and 5.1e. This is due to the fact that we have many observations in our train period and therefore the graphs are very dense, not showing any clear seasonal pattern if not zoomed in. Figure 5.1c displays the graph of the daily seasonality component for a week. We can clearly see that there are indeed 7 recurring patterns visible, confirming the daily seasonality. Figure 5.1d also confirms the weekly seasonality. This figure displays the graph of the weekly seasonality component for 7 weeks, showcasing 7 recurring patterns. Figures 7.4a and 7.4b in the appendix show the dense graphs for the whole train period and indicate a nice smoothing. Figure 5.1e finally shows the remainder, thus the weather sensitive component for the train period. Figure 7.5 in the appendix displays the base component for the train period.

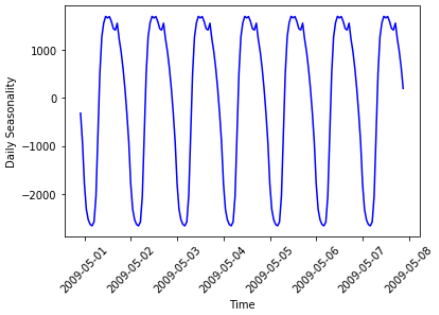
Decomposition of the load time series in the test period with MSTL results in the four components displayed in Figure 5.2. This figure also includes the original load time series for the test period, in Figure 5.2a, as a reference. Since we decompose a shorter time period, the trend looks different and catches on to smaller changes than in the train period. We can see, in Figure 5.2b, that around the winter months the load increases, which could be due to cold weather, and around spring a decline starts. Since our test period is equal to a year, the graph of the trend displays the yearly seasonality. Figures 5.2c and 5.2d show the daily and weekly seasonality cycles, respectively, on a different scale. With the different scale we can see that both are smoothed nicely and the recurring seasonal patterns are displayed clearly. Figures 7.4c and 7.4d in the appendix display the daily and seasonality components for the whole test period. Figure 5.2e finally shows the remainder, which is equal to the weather sensitive component. Figure 7.6 in the appendix displays the base component for the test period.



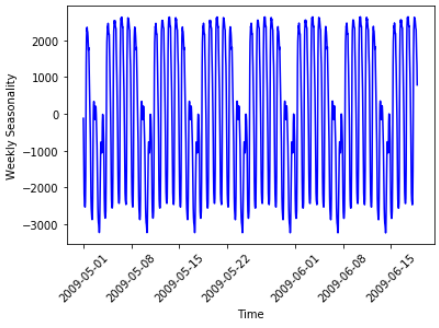
(a)



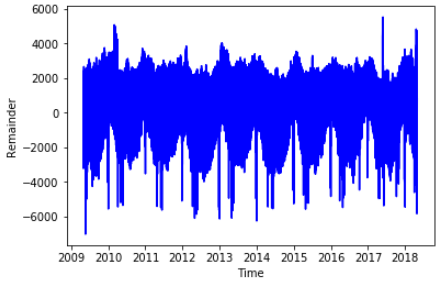
(b)



(c)

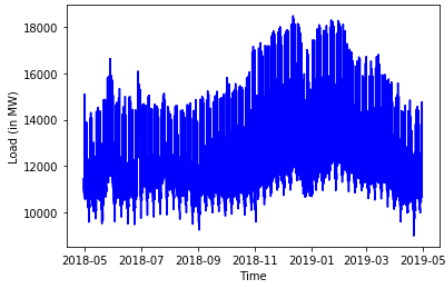


(d)

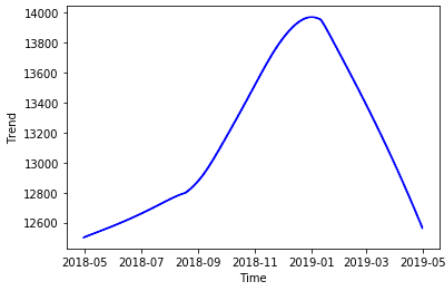


(e)

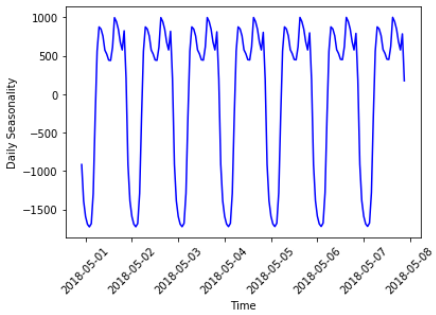
Figure 5.1: The original load time series for the train period plus all 4 components after decomposing the load time series.



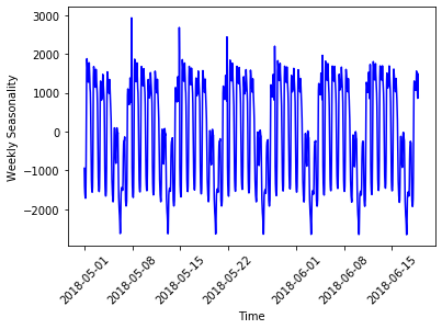
(a)



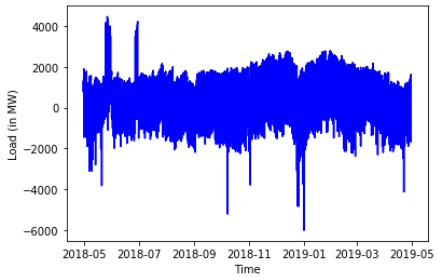
(b)



(c)



(d)



(e)

Figure 5.2: The original load time series for the test period plus all 4 components after decomposing the load time series.

5.2 Forecasting the base component

When training the double seasonal additive Holt Winter's model with the train data set, we minimise the MSE, which results in the estimates of the smoothing parameters: α, γ, δ and ω. The estimates are equal to: $[\hat{\alpha}, \hat{\gamma}, \hat{\delta}, \hat{\omega}] = [0.9999,\ 0.0001,\ 0.0001,\ 0.0000]$ and the minimised MSE has a value equal to 0.6954. The small value of $\hat{\gamma}$ seems reasonable since variation in the train period is mostly dominated by the seasonality cycles. The low values of the seasonal smoothing parameters, $\hat{\delta}$ and $\hat{\omega}$, indicate that the seasonal cycles do not change that much over time.

The estimates of the smoothing parameters, $\hat{\alpha}, \hat{\gamma}, \hat{\delta}$ and $\hat{\omega}$, and Equations 3, 4, 5, 6 and 7 together yield the 24-hours ahead forecasts of the base component for the test period. The evaluation criteria for the test period using double seasonal additive Holt Winter's method to forecast the base component are shown in Table 5.1. The high R^2 value and quite low MAPE value indicate that the model seems to be considerably accurate in forecasting the base component.

Evaluation Criteria	**Value**
RMSE	982.7158
R^2	0.8680
MAE	588.4492
MAPE	4.4105

Table 5.1: Evaluation criteria for the test period using double seasonal additive Holt Winter's method to forecast the base component.

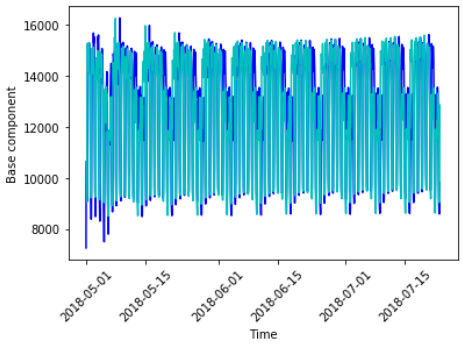


Figure 5.3: The cobalt blue graph represents the forecast and the turquoise graph the real value of the base component for the test period.

In Figure 5.3 the graph of the forecast of the base component based on double seasonal additive Holt Winter, in cobalt blue, is shown together with the graph of the actual value of the base component, in turquoise, for the first part of the test period. The graph for the whole test period is shown in Figure 7.7 in the appendix. At the beginning of the test period, the forecast seems to under-estimate the values of the base component more than further in the test period. It also seems as if the predictions accurately capture the patterns, but that they are delayed.

5.3 Forecasting the weather-sensitive component

Forecasting the weather-sensitive component, Y_{wsc}, for the test period with the trained models LR, MARS and ANN and comparing it with the true values of Y_{wsc} of the test period gave the different values of the evaluation criteria as shown in Table 5.2. Figures 7.8, 7.9 and 7.10 in the appendix show the graphs of the prediction of Y_{wsc} based on LR, MARS and ANN, respectively, compared to

the actual value of Y_{wsc} for the test period.

Evaluation Criteria	**LR**	**MARS**	**ANN**
RMSE	539.7838	538.1264	570.9048
R^2	0.8252	0.8262	0.8044
MAE	347.0190	343.3057	380.7116
MAPE	146.2716	146.5075	167.7300

Table 5.2: Values of the evaluation criteria when using linear regression, multivariate adaptive regression splines and an artificial neural network to forecast the values of the weather-sensitive component in the test period.

From Table 5.2 we see that ANN performs most poorly according to all evaluation criteria. It has the lowest R^2 value and the highest RMSE, MAE and MAPE value. MARS seems to forecast the weather-sensitive component the most accurate, since it has the highest R^2 value and the lowest RMSE and MAE value. However, the differences in the evaluation criteria for both MARS and LR are minor.

In Sections 5.3.1, 5.3.2 and 5.3.3 the output of training the LR, MARS and ANN models is given, respectively.

5.3.1 Linear regression

Training the LR model results in the following output function to estimate all values of the weather-sensitive component, Y_{wsc} for the test period:

$$Y_{wsc} = 549.039 - 2.520X_{avg.temp} - 6.811X_{avg.wind} - 6.490X_{sun} - 3.264X_{humidity} + 0.898X_{firstlag} + \epsilon, \tag{18}$$

in which all coefficients are significant, with a 5% significance level. R^2 is equal to 0.9220. All weather variables have a decreasing effect on Y_{wsc}, only the lag of Y_{wsc}, $X_{firstlag}$, has an increasing effect on Y_{wsc}.

5.3.2 Multivariate adaptive regression splines

The basis functions obtained from MARS are displayed in Table 5.3. All these basis functions appear in the final MARS model.

BF	**Formula**
BF_1	max{0, $X_{firstlag}$ + 13.680}
BF_2	max{0, -13.680 - $X_{firstlag}$}
BF_3	X_{sun}
BF_4	$X_{avg.temp}$

Table 5.3: Basis functions obtained from MARS.

The output function of MARS after training the model is equal to:

$$Y_{wsc} = 177.671 + 0.897BF_1 - 0.905BF_2 - 6.048BF_3 - 1.858BF_4 + \epsilon \tag{19}$$

in which all coefficients are significant, with a 5% significance level. The MSE is equal to 260881.3918, the value of GCV is equal to 260954.1616 and R^2 has the value 0.9214.

5.3.3 Artificial neural network

Since ANN is considered a 'black box', training the ANN does not give any output other than the scaled minimised MSE value, which is equal to 0.0676.

5.4 Comparison of all three hybrid models

Combining the forecasts of the base component, based on the double seasonal additive Holt Winter's method, with the forecasts of the weather-sensitive component, either based on LR, MARS or ANN, gives the 24-hours ahead forecast of the hourly load time series in the test period. The evaluation criteria for the three different hybrid models are displayed in Table 5.4.

Evaluation Criteria	**Values using LR**	**Values using MARS**	**Values using ANN**
RMSE	1176.0676	1175.1829	1185.8933
R^2	0.6500	0.6490	0.6319
MAE	768.3739	765.7104	786.3451
MAPE	6.0148	5.9927	6.1585

Table 5.4: Values of the evaluation criteria when using the three different hybrid models, double seasonal additive Holt Winter's method combined with either LR, MARS or ANN, to forecast the short-term load in the test period.

The hybrid model with ANN performs most poorly, with the lowest R^2 value and the highest RMSE, MAE and MAPE values. The hybrid models with either LR or MARS perform similarly. The hybrid model with MARS has a slightly smaller RMSE (0.08% smaller), MAE (0.35% smaller) and MAPE (0.37% smaller) value. The hybrid model with LR has an only slightly higher R^2 value (0.15% higher). These differences are minor. However, overall the hybrid model using MARS has the most accurate forecast for the short-term load. Figures 7.11, 7.12 and 7.13 display graphs of the load forecasts based on the hybrid model with LR, MARS and ANN, respectively, compared to the actual values of the load in the test period.

6 Conclusion

The goal of this research is to answer the research question: **'Which hybrid model is the most accurate for short-term load forecasting when using a decomposition-ensemble algorithm**

for the hourly load time series: the double seasonal additive Holt Winter's method combined with linear regression, with multivariate adaptive regression splines or with an artificial neural network?' First, a novel decomposition algorithm is developed to account for the multiple seasonal cycles in the hourly load time series. This novel decomposition algorithm is called Multiple Seasonal and Trend decomposition by LOESS (MSTL). After decomposing the time series into a base and weather-sensitive component, different algorithms are chosen to model each component. The double seasonal additive Holt Winter's method seemed the best fit for the base component, as suggested by Qiuyu et al. (2017). The weather-sensitive component is modeled with three different algorithms: linear regression (LR), multivariate adaptive regression splines (MARS) model and an artificial neural network (ANN).

The answer to the research question is that when using MSTL for the hourly load time series, the hybrid model using a combination of the double seasonal additive Holt Winter's method and MARS is more accurate than the hybrid model using either ANN or LR. The hybrid model using ANN is the least accurate and the hybrid models using either LR or MARS are more accurate. These two hybrid models perform similarly and although the differences are minor, the hybrid model using MARS is slightly more accurate. The novel, most accurate hybrid model using MARS may allow electrical companies to forecast the short-term load more accurately and incur fewer costs.

The outcome is in line with the findings of Nalcaci et al. (2019). They also found that MARS forecasted the short-term load more accurately than LR and ANN. This can be due to the fact that MARS is able to produce non-linear models for regression, whereas LR cannot. In addition, MARS outperforms LR in the context of being able to handle data of mixed type better and dealing better with irrelevant inputs. Furthermore, MARS has as the advantage that it is not a 'black box' like ANN. The output of MARS shows a relationship between the dependent and independent variables, whereas ANN does not show such an output.

Even though the hybrid model with MARS is the most accurate of the three hybrid models investigated, there is room for improvement of the model to achieve more accurate results. One possibility for an improved model is one that takes different variables into account when modeling the weather-sensitive component. Another possibility is taking more lags or different lags into account to help improve the accuracy of the model. A suggestion for further research is to investigate improving the model for the base component. As suggested by Caiado (2010) an adjustment of first-order auto correlation can be taken into account when using the double seasonal additive Holt Winter's method. In addition, different decomposition algorithms can be tested. Finally, different train and test periods can be used. In this research a large data set was used to train the model, but for short-term load forecasting a smaller data set to train the models may be sufficient.

By providing a novel decomposition method and hybrid models, this paper adds value to the already existing literature and therefore provides a basis for possible further research. The above mentioned possible extensions are beyond the scope of this research, but are strongly suggested as topics for further investigation on improving the accuracy of the short-term load.

References

Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., & Jenssen, R. (2017). An overview and comparative analysis of recurrent neural networks for short term load forecasting. *arXiv preprint:1705.04378*.

Caiado, J. (2010). Performance of combined double seasonal univariate time series models for forecasting water demand. *Journal of Hydrologic Engineering*, *15*(3), 215–222.

Chen, Y., Luh, P. B., Guan, C., Zhao, Y., Michel, L. D., Coolbeth, M. A., ... Rourke, S. J. (2009). Short-term load forecasting: Similar day-based wavelet neural networks. *IEEE Transactions on Power Systems*, *25*(1), 322–330.

Clemen, R. T. (1989). Combining forecasts: A review and annotated bibliography. *International journal of forecasting*, *5*(4), 559–583.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition. *Journal of official statistics*, *6*(1), 3–73.

Dagum, E. B., & Bianconcini, S. (2016). *Seasonal adjustment methods and real time trend-cycle estimation*. Springer.

Dillon, T., Sestito, S., & Leung, S. (1991). Short term load forecasting using an adaptive neural network. *International Journal of Electrical Power & Energy Systems*, *13*(4), 186–192.

ENTSO-E, Power Statistics. (2019). *Monthly aggregated hourly load values by country, European data.* Retrieved from `https://www.entsoe.eu/data/power-stats/`

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The annals of statistics*, 1–67.

Friedrich, L., & Afshari, A. (2015). Short-term forecasting of the Abu Dhabi electricity load using multiple weather variables. *Energy Procedia*, *75*, 3014–3026.

Hassoun, M. H., et al. (1995). *Fundamentals of artificial neural networks*. MIT press.

Heij, C., de Boer, P., Franses, P. H., Kloek, T., van Dijk, H. K., et al. (2004). Econometric methods with applications in business and economics. In (p. 83). Oxford University Press.

Hor, C.-L., Watson, S. J., & Majithia, S. (2005). Analyzing the impact of weather variables on monthly electricity demand. *IEEE transactions on power systems*, *20*(4), 2078–2085.

Jacoby, W. G. (2000). Loess: a nonparametric, graphical tool for depicting relationships between variables. *Electoral Studies*, *19*(4), 577–613.

Khwaja, A., Naeem, M., Anpalagan, A., Venetsanopoulos, A., & Venkatesh, B. (2015). Improved short-term load forecasting using bagged neural networks. *Electric Power Systems Research*, *125*, 109–115.

KNMI, Koninklijk Nederlands Meterologisch Instituut. (2020). *Hourly weather data in the Netherlands.* Retrieved from `https://projects.knmi.nl/klimatologie/uurgegevens/selectie.cgi`

Nalcaci, G., Özmen, A., & Weber, G. W. (2019). Long-term load forecasting: models based on mars, ann and lr methods. *Central European Journal of Operations Research*, *27*(4), 1033–1049.

Python. (1991). Python: A dynamic, open source, programming language, version 3.7.6 64-bit. *Python Software Foundation*. (Available at python.com)

Qiuyu, L., Qiuna, C., Sijie, L., Yun, Y., Binjie, Y., Yang, W., & Xinsheng, Z. (2017). Short-term load forecasting based on load decomposition and numerical weather forecast. In *2017 IEEE conference on energy internet and energy system integration (ei2)* (pp. 1–5).

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Sarkodie, S. A., & Strezov, V. (2018). Assessment of contribution of Australia's energy production to CO_2 emissions and environmental degradation using statistical dynamic approach. *Science of the Total Environment*, *639*, 888–899.

Singh, A. K., Khatoon, S., Muazzam, M., Chaturvedi, D., et al. (2012). Load forecasting techniques and methodologies: A review. In *2012 2nd international conference on power, control and embedded systems* (pp. 1–10).

Son, H., & Kim, C. (2017). Short-term forecasting of electricity demand for the residential sector using weather and social variables. *Resources, conservation and recycling*, *123*, 200–207.

Taylor, J. W. (2003). Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society*, *54*(8), 799–805.

Theodosiou, M. (2011). Forecasting monthly and quarterly time series using stl decomposition. *International Journal of Forecasting*, *27*(4), 1178–1195.

Wang, L., Zhang, Z., & Chen, J. (2016). Short-term electricity price forecasting with stacked denoising autoencoders. *IEEE Transactions on Power Systems*, *32*(4), 2673–2681.

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management science*, *6*(3), 324–342.

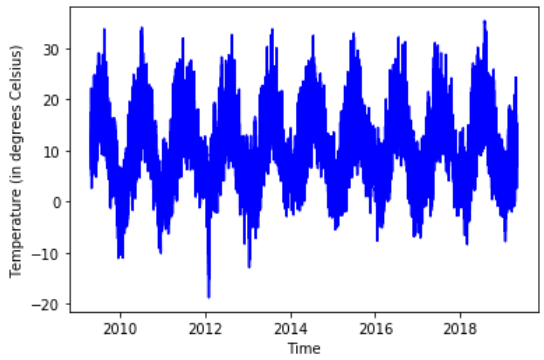
Zhang, W., & Goh, A. T. (2016). Multivariate adaptive regression splines and neural network models for prediction of pile drivability. *Geoscience Frontiers*, *7*(1), 45–52.

Zhang, X., & Wang, J. (2018). A novel decomposition-ensemble model for forecasting short-term load-time series with multiple seasonal patterns. *Applied Soft Computing*, *65*, 478–494.

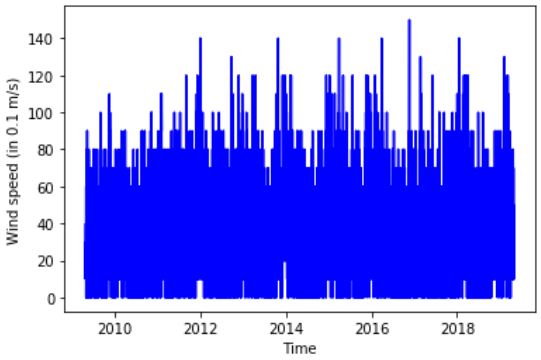
Zohuri, B., & McDaniel, P. (2019). The electricity: An essential necessity in our life. In *Advanced smaller modular reactors* (pp. 1–21). Springer.

7 Appendix

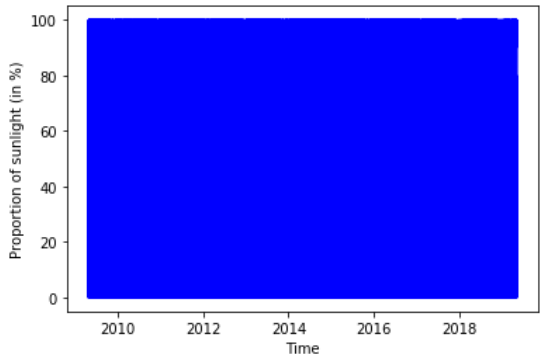
Weather variables



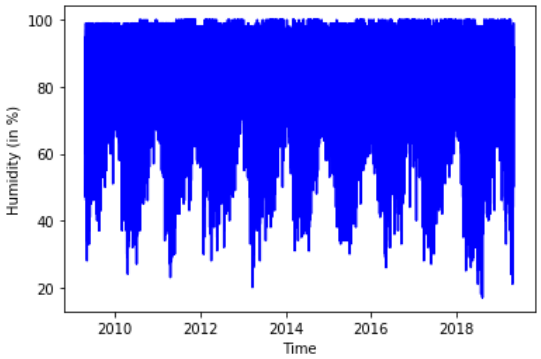
(a) Average temperature



(b) Average wind speed



(c) Amount of sunlight



(d) Humidity

Figure 7.1: Development of the hourly values of the weather variables in the period 30 April 2009 to 30 April 2019.

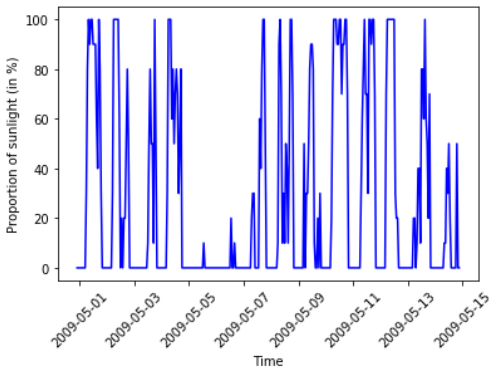


Figure 7.2: A zoom in of Figure 7.1c, displaying the development of the hourly values of the amount of sunlight more clearly for a smaller period: 01 May 2009 to 15 May 2009.

Multivariate Adaptive Regression Splines - Basis functions

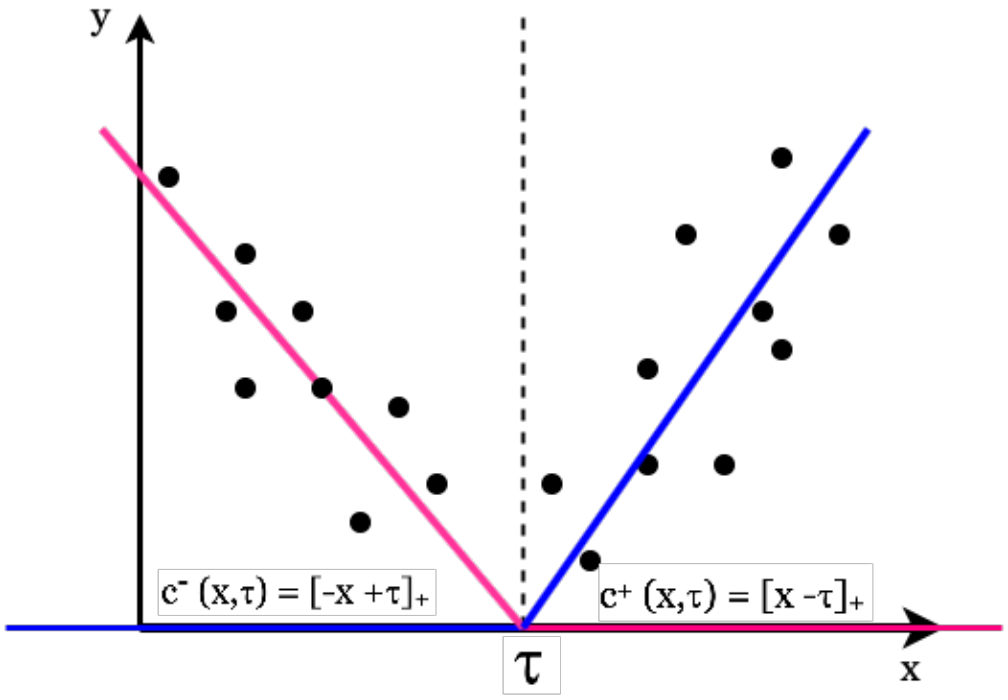
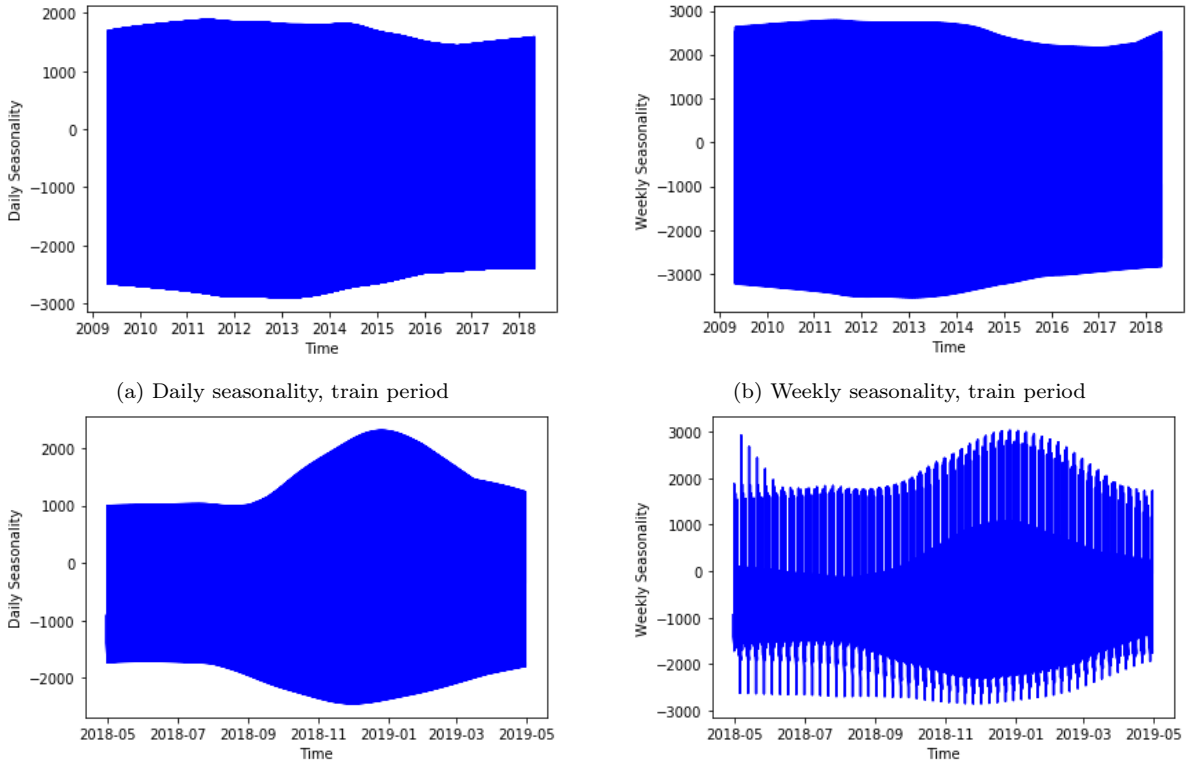


Figure 7.3: A pair of basis functions with knot τ: basis function $[-x + \tau]_+$ in pink and basis function $[x - \tau]_+$ in blue, together they form a 'reflected pair' (Nalcaci et al., 2019).

Dense graphs of daily and weekly seasonality components



(a) Daily seasonality, train period

(b) Weekly seasonality, train period

(c) Daily seasonality, test period

(d) Weekly seasonality, test period

Figure 7.4: The daily and weekly seasonality components for the whole train and test period.

Base Component

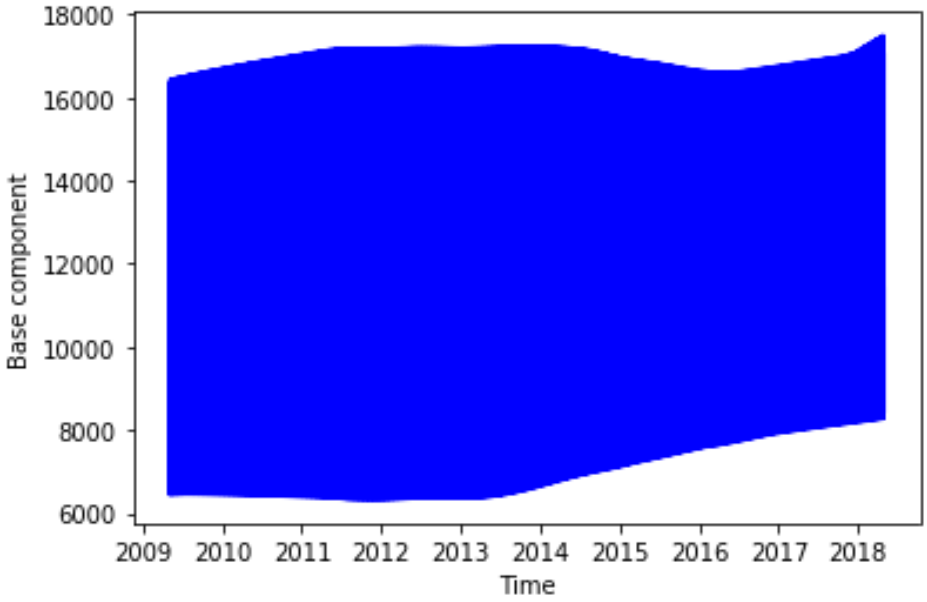


Figure 7.5: The base component of the train period.

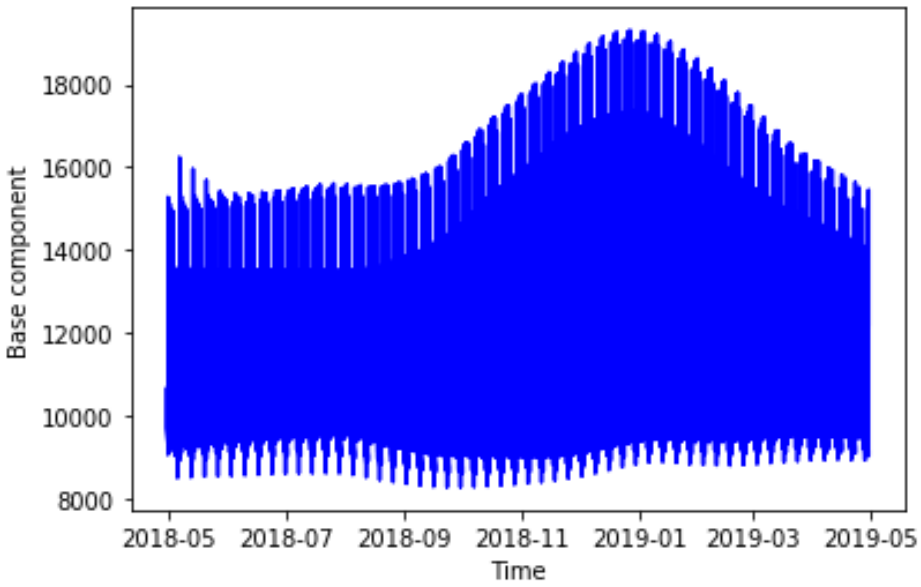


Figure 7.6: The base component of the test period.

Forecast of the base component

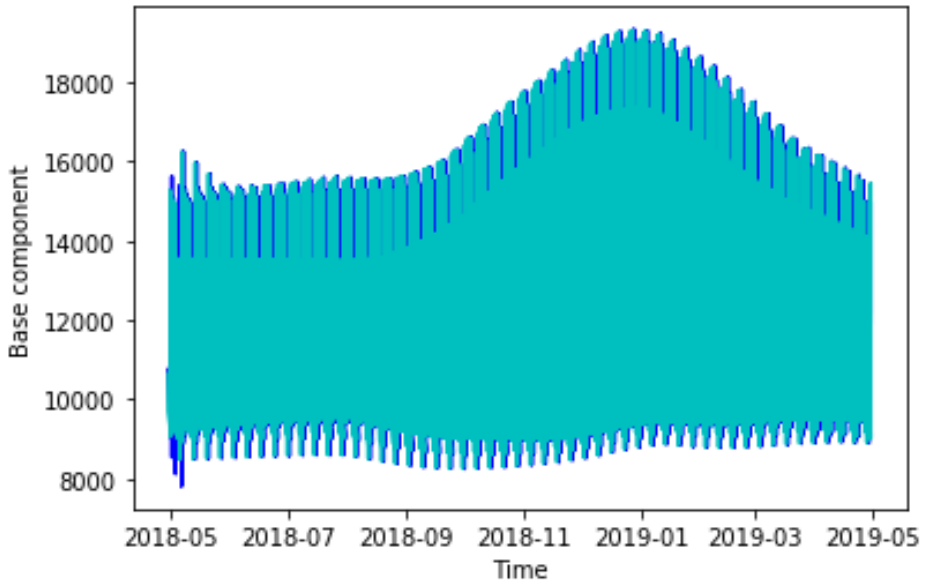


Figure 7.7: The cobalt blue graph equals the forecast based on double seasonal additive Holt Winter's method and the turquoise graph displays the real value of the base component for the test period.

Forecasts of the weather-sensitive component

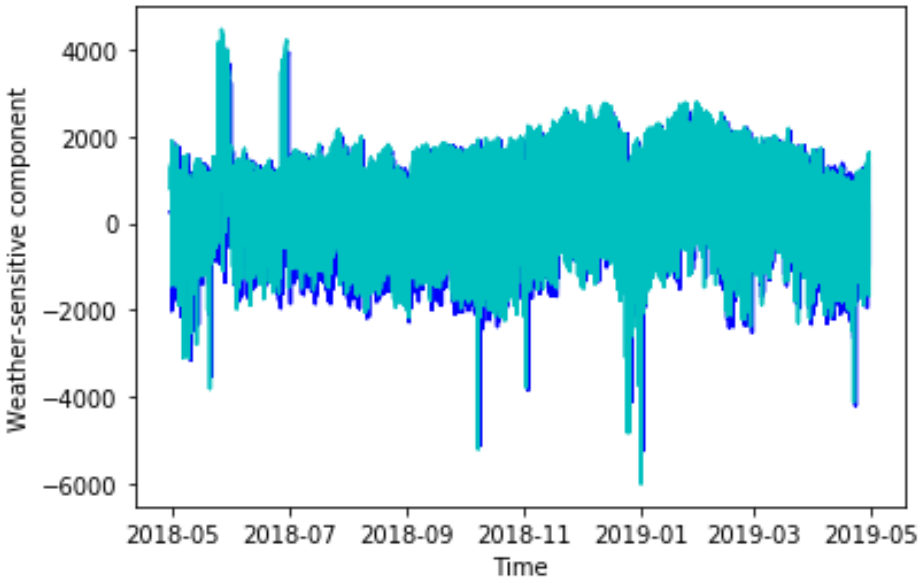


Figure 7.8: The cobalt blue graph equals the forecast of the weather-sensitive component based on LR and the turquoise graph displays the real value of the weather-sensitive component for the test period.

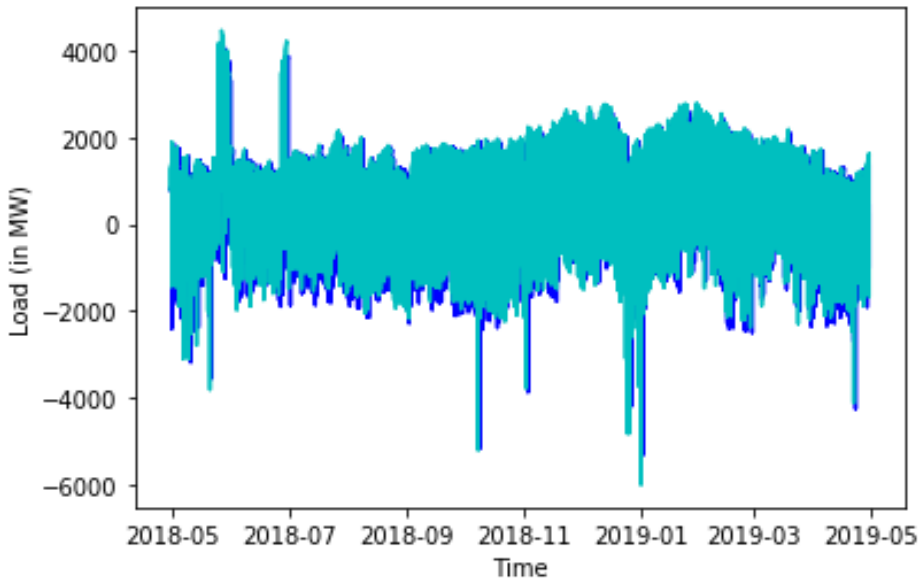


Figure 7.9: The cobalt blue graph equals the forecast of the weather-sensitive component based on MARS and the turquoise graph displays the real value of the weather-sensitive component for the test period.

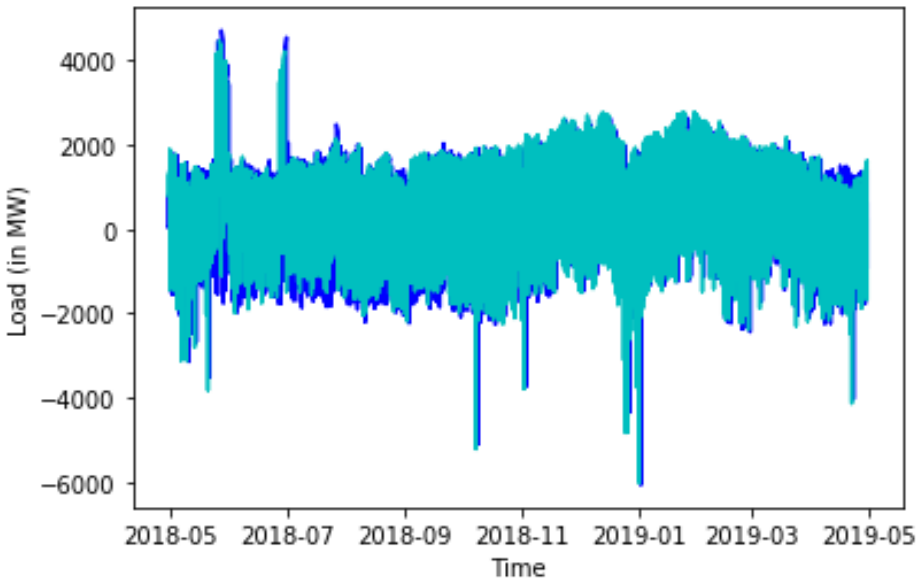


Figure 7.10: The cobalt blue graph equals the forecast of the weather-sensitive component based on ANN and the turquoise graph displays the real value of the weather-sensitive component for the test period.

Forecasts of the short-term load

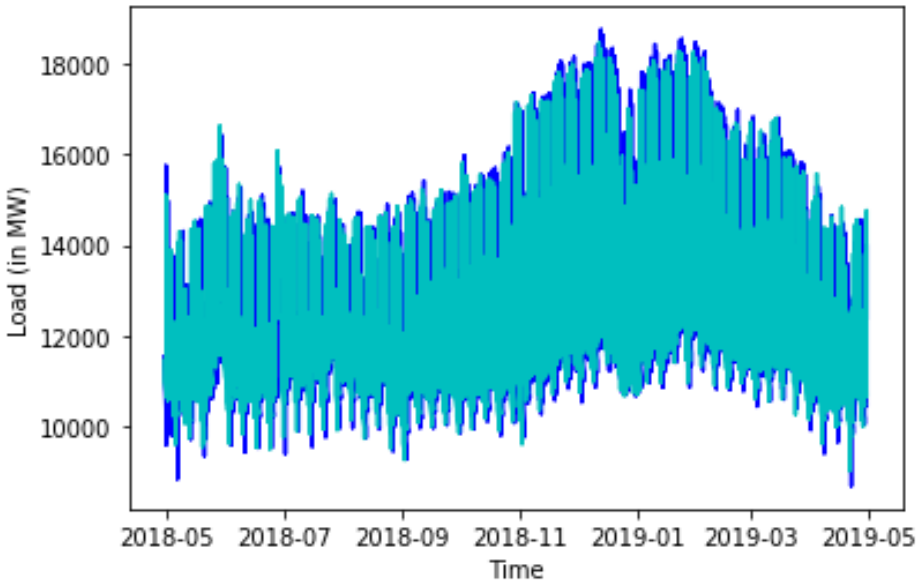


Figure 7.11: The cobalt blue graph displays the forecast of the short-term load based on the hybrid model with LR and the turquoise graph displays the real value of the short-term load for the test period.

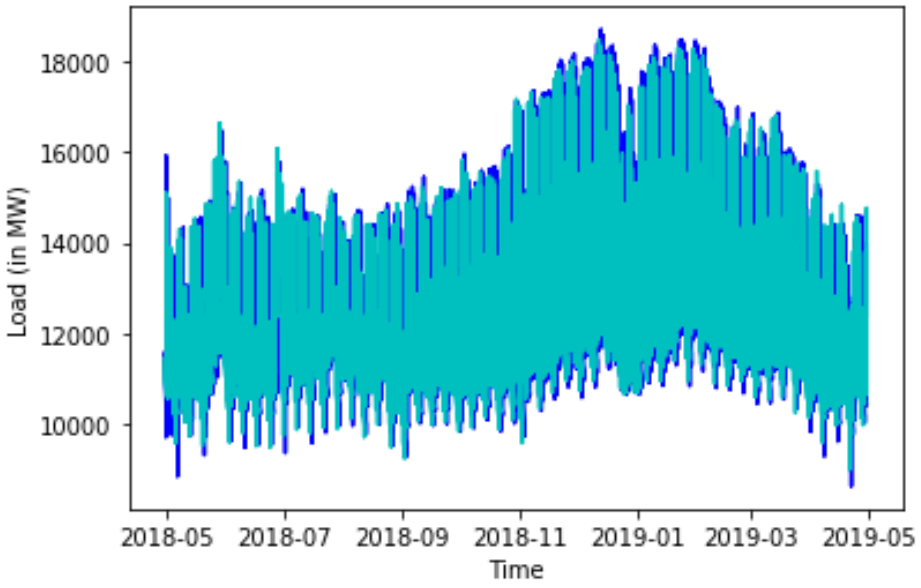


Figure 7.12: The cobalt blue graph displays the forecast of the short-term load based on the hybrid model with MARS and the turquoise graph displays the real value of the short-term load for the test period.

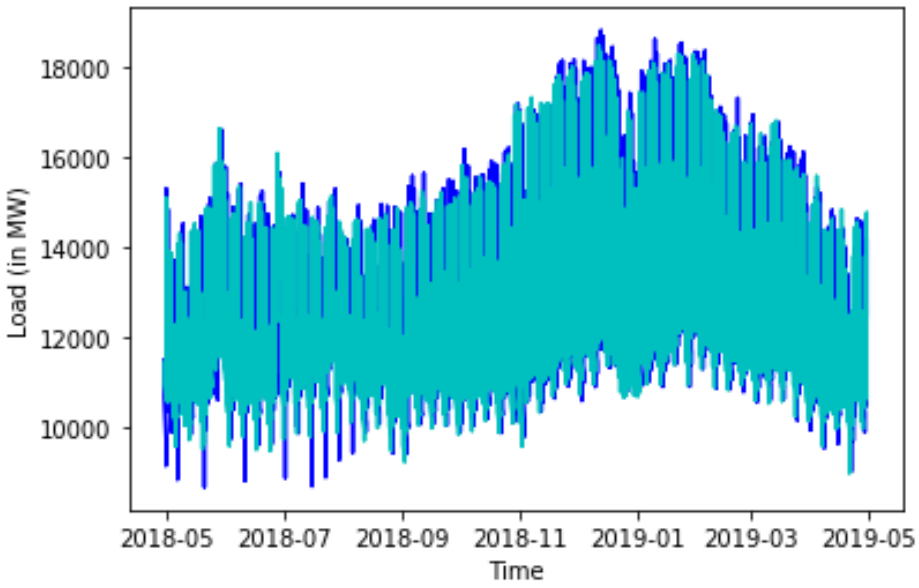


Figure 7.13: The cobalt blue graph displays the forecast of the short-term load based on the hybrid model with ANN and the turquoise graph displays the real value of the short-term load for the test period.

Programming code

Three main scripts of programming code are used in this research: *'Total Function MSTL + DHW + LR'*, *'Total Function MSTL + DHW + MARS'* and *'Total Function MSTL + DHW + ANN'*. All three scripts are built the same way: the first lines contain the code for the MSTL algorithm. This algorithm is applied to the train and test data set, resulting in the base and weather-sensitive component for the train and test period.

Next, a function representing the double seasonal additive Holt Winter's method is defined and used with the train data set to estimate the parameters needed for forecasting the base component for the test period. The base component for the test period is then forecasted. The evaluation criteria to compare the forecast of the base component and the real value of the base component are then defined and computed.

Following, the models for the weather-sensitive component are coded. This is where the three scripts differ: *'Total Function MSTL + DHW + LR'* contains the code for modeling the weather-sensitive component with linear regression (LR), *'Total Function MSTL + DHW + MARS'* contains the code for modeling the weather-sensitive component with multivariate adaptive regression splines (MARS) and *'Total Function MSTL + DHW + ANN'* contains the code for modeling the weather-sensitive component with an artificial neural network (ANN). After coding the models LR, MARS and ANN in each script and training each model, the forecasts of the weather-sensitive component for the test period are made and evaluation criteria for each model are computed.

Every script ends with combining the forecasts of the base and weather-sensitive component, by adding them up. This combination forms the forecast of the original hourly load time series. The forecasts are compared to the real values of the original hourly load time series by calculating the evaluation criteria.

In addition, five scripts are added: *'MSTL function'*, *'Double Seasonal Additive Holt Winters'*, *'Linear Regression function'*, *'Multivariate Adaptive Regression Splines'* and *'Artificial Neural Network'*. *'MSTL function'* is the script that defines the function describing the (k+1)st pass through the outer loop. *'Double Seasonal Additive Holt Winters'* is the script describing the training of the double seasonal additive Holt Winter's method and the 24-hours ahead forecasting of the base component. *'Linear Regression function'* is the script describing the training of the linear regression model and the 24-hours ahead forecasting of the weather-sensitive component. *'Multivariate Adaptive Regression Splines'* is the script describing the training of the multivariate adaptive regression splines model and the 24-hours ahead forecasting of the weather-sensitive component. Finally, *'Artificial Neural Network'* is the script describing the building and training of the artificial neural network and the 24-hours ahead forecasting of the weather-sensitive component.

These five scripts are incorporated into the aforementioned three main scripts. They are only added for clarity and the possibility to look at a particular algorithm more closely.