International Journal of Forecasting 30 (2014) 382–394

Contents lists available at ScienceDirect

International Journal of Forecasting

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

A gradient boosting approach to the Kaggle load forecasting competition

Souhaib Ben Taieb [a,*], Rob J. Hyndman [b]

[a] Machine Learning Group, Department of Computer Science, Faculty of Sciences, Université Libre de Bruxelles, Belgium

[b] Department of Econometrics and Business Statistics, Monash University, Clayton, VIC 3800, Australia

ARTICLE INFO

Keywords:
Short-term load forecasting
Multi-step forecasting
Additive models
Gradient boosting
Machine learning
Kaggle competition

ABSTRACT

We describe and analyse the approach used by Team TinTin (Souhaib Ben Taieb and Rob J Hyndman) in the Load Forecasting track of the Kaggle Global Energy Forecasting Competition 2012. The competition involved a hierarchical load forecasting problem for a US utility with 20 geographical zones. The data available consisted of the hourly loads for the 20 zones and hourly temperatures from 11 weather stations, for four and a half years. For each zone, the hourly electricity loads for nine different weeks needed to be predicted without having the locations of either the zones or stations. We used separate models for each hourly period, with component-wise gradient boosting for estimating each model using univariate penalised regression splines as base learners. The models allow for the electricity demand changing with the time-of-year, day-of-week, time-of-day, and on public holidays, with the main predictors being current and past temperatures, and past demand. Team TinTin ranked fifth out of 105 participating teams.

1. Introduction

We participated in the Load Forecasting track of the Kaggle Global Energy Forecasting Competition 2012, organised by the IEEE working group on Energy Forecasting (WGEF) (Tao, Pierre, & Shu, 2014). Team TinTin (Souhaib Ben Taieb and Rob J Hyndman) ranked fifth out of 105 participating teams. The competition involved a hierarchical load forecasting problem. We were required to backcast and forecast hourly loads (in kW) for a US utility with 20 geographical zones. Thus, 21 separate time series needed to be backcast and forecast: the 20 zonal level series, and the aggregate series.

The electricity demand is subject to a range of factors, including weather conditions, calendar effects, economic activity, and electricity prices. However, we were required to use only temperatures and calendar information. The available data consisted of the hourly loads for the 20 zones, and hourly temperatures from 11 weather stations, from the first hour of 1 January 2004 to the sixth hour of 30 June 2008. We are not aware of the locations of the 20 zones and the 11 weather stations, and in particular, we do not know which stations are located in or near which zones. Consequently, our task was to find a model that can take temperature and calendar information as inputs, and predict electricity loads as the output.

We used a separate model for each hourly period, ending up with 24 different models for one day, since the electricity demand pattern changes through the day. Each hourly model is estimated using component-wise gradient boosting (Bühlmann & Yu, 2003), with univariate penalised regression splines (Eilers & Marx, 1996) as base learners, together with automatic variable selection during the fitting process. The models can be cast in the framework of additive models to allow nonparametric and non-linear terms. The models allow for the electricity demand to change with the time-of-year, day-of-week, time-of-day, and on public holidays. The main predictors were current and past temperatures (up to a week earlier) and past demand (up to a week earlier).

* Corresponding author.
E-mail address: sbentaie@ulb.ac.be (S. Ben Taieb).

0169-2070/$ – see front matter
http://dx.doi.org/10.1016/j.ijforecast.2013.07.005

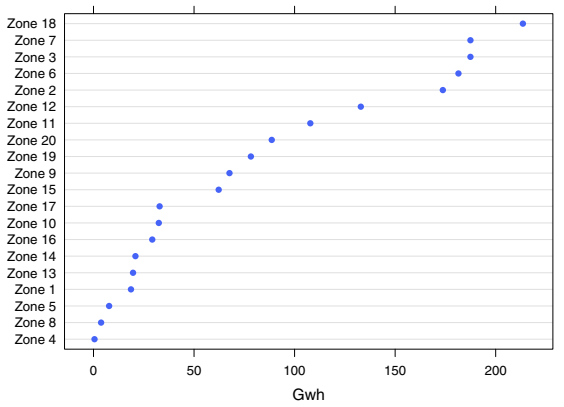

Fig. 1. Average demand for each zone.

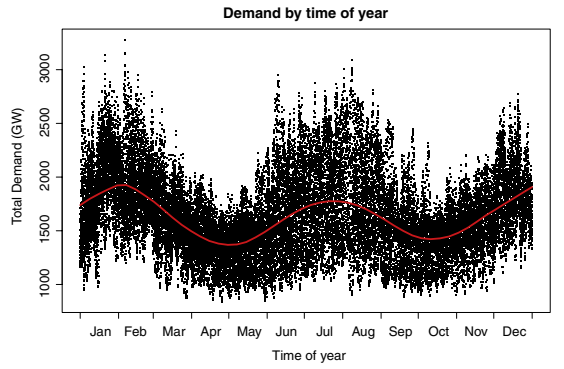

Fig. 2. Total demand plotted against the time of year. The smoothed mean demand is shown as a red line.

Our approach has been influenced by the work of Fan and Hyndman (2012), who developed semi-parametric additive models with penalised spline functions for forecasting the electricity demand in various Australian states.

The paper is organised as follows. The next section presents the data analysis and preprocessing we performed. Section 3 describes our forecasting methodology, clarifying the differences between the in-sample and out-of-sample weeks. Section 4 presents our model, together with the gradient boosting algorithm, and the proposed model is then analysed in Section 5. Finally, Section 6 provides some discussion and conclusions.

2. Data analysis and preprocessing

Fig. 1 shows the average demand for all zones during the period covered by the data. The average demand varied greatly across zones, with Zone 18 having the highest demand levels and Zone 4 the lowest. When exploring the data, we noticed that the data for Zones 3 and 7 are identical, and Zone 2 contains values that are exactly 92.68% of the demand values in Zones 3 and 7. Also, Zone 10 has a big jump in demand in the year 2008. Finally, Zone 9 contained very erratic demand patterns which did not seem to be in any way related to the temperature values.

Electricity demand is subject to a wide variety of exogenous variables, including calendar effects. Fig. 2 shows that there is a clear time-of-year effect in the demand data, with

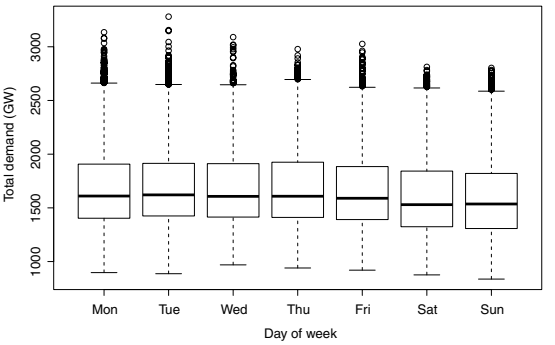

Fig. 3. Boxplots of the total demand by day of the week.

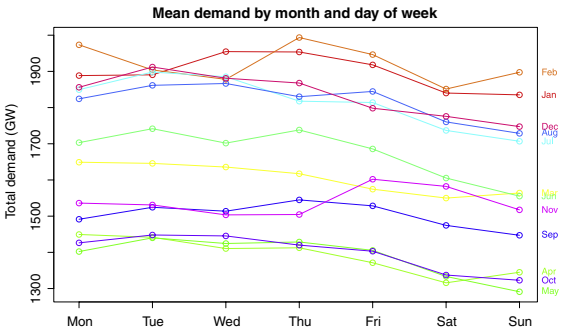

Fig. 4. Average total demand (GW) by month and by day of the week.

peaks in the mean demand around February and July, and troughs in April/May and October. In others words, winter and summer tend to show a high demand, while fall and spring have a lower demand.

Boxplots of the demand by day of the week are shown in Fig. 3. While the day-of-week effect is relatively small, there is a drop in demand for the weekends. The average demands for each month and each day of the week are plotted in Fig. 4. Because the day-of-week pattern is similar for all months, there is unlikely to be a strong interaction between day-of-week and time-of-year.

We look at the way in which demand changes with the time of day in Figs. 5 and 6. Here, hour 0 corresponds to 12 am–1 am, hour 1 corresponds to 1 am–2 am, and so on. The night-time patterns for the two plots are similar, but there is a difference during the working hours (8 am–5 pm).

Fig. 7 shows the demand in Zone 18 plotted against the current temperature from station 9. There is a clear nonlinear relationship, indicating that the current temperature is an important predictor of demand. Demand is driven by air conditioning usage for temperatures above 20 °C, and by heating for temperatures below 15 °C. Similar but weaker relationships are seen in plots against lagged temperatures. Due to buildings' thermal inertia, it is important to consider lagged temperatures as well as current temperatures in any demand forecasting model. Fig. 8 shows the current demand plotted against lagged demand for different lags. There is a clear relationship between these variables due to the serial dependence within the demand time series.

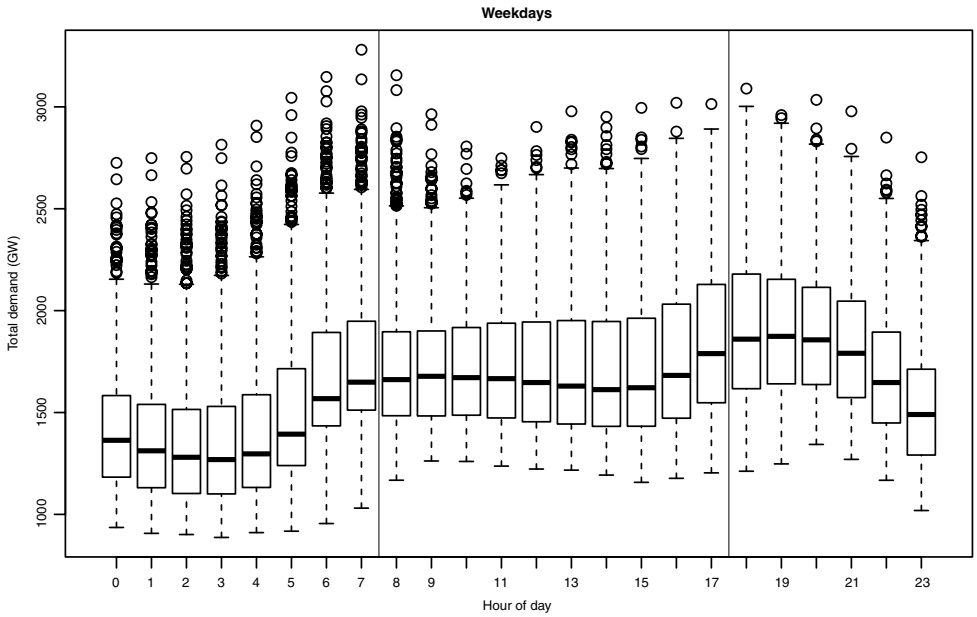

Fig. 5. Boxplots of demand by time of day for Monday–Friday.

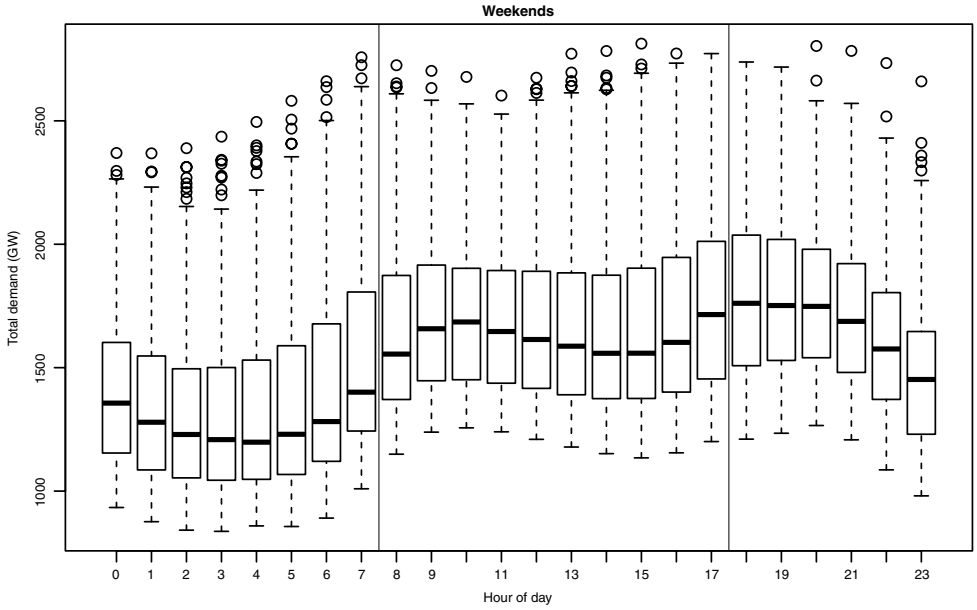

Fig. 6. Boxplots of demand by time of day for Saturday–Sunday.

Before developing any forecasting models, we pre-processed the data in order to avoid some potential problems. First, we removed leap days, to give 365 days for each year. This avoided problems with uneven seasonal periods, and resulted in only a small loss of information. Second, we took a log transformation of the demand. This is to stabilise the variance of the time series across time. There were also some outliers and unusual features in the data which we corrected before proceeding.

We identified some outliers in the temperature data for site 8, and replaced them with the data for the same period from the closest site in terms of Euclidean distance.

Zone 4 had some outliers in demand. We used Loess for fitting and then classified a point y_t as an outlier if $y_t - \hat{y}_t > \text{median}(y_t - \hat{y}_t) + k * \text{MAD}$, where $\hat{y}_t$ is the Loess fit of y_t, MAD is the mean absolute deviation and k is chosen so that the probability of an outlier is 0.002 under a normal distribution. Then, the outliers were replaced by the mean of the data.

For Zone 10, there was a big jump in demand in year 2008. We computed the means before and after the jump on the log-transformed data, then we took the difference and removed the jump in 2008. For the final forecasts, we restored the jump into the forecasts.

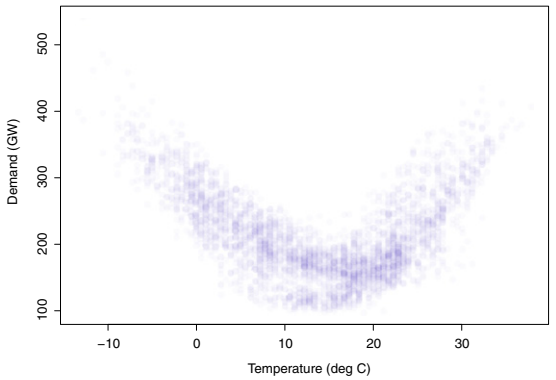

Fig. 7. Hourly demand (GW) plotted against temperature (degrees Celsius) for Zone 18 and station 9.

3. Forecasting methodology

The competition involved a hierarchical load forecasting problem with 20 zonal level series, plus the aggregate series. We used a bottom-up approach; that is, we forecast each zone independently and then took the sum of the 20 zonal forecasts to obtain forecasts for the aggregate. For each of the twenty zones, we were required to backcast the demand for eight in-sample weeks for which the temperatures at various sites were provided.

Because we do not know which temperature site corresponds to which zone, the temperatures from all of the sites are all potential predictors in each model. We used a testing week (the last week of the available data) to determine which sites to use for each zone. Fig. 9 gives the root mean squared errors (RMSE) obtained on the testing week using the real (in blue) and forecasted (in red) temperatures from the eleven sites for ten zones. We can see that the differences in forecasts obtained using the real temperatures (in blue) at different sites can be huge. Consequently, when forecasting the eight in-sample weeks, we use, for each zone, the site which minimises the error over the testing week.

For the eight in-sample weeks, demand data were available before and after the week, and temperature data were available during the week. The data following each of the in-sample weeks are also useful for predicting the demand values during the in-sample weeks (although this is not possible in real forecasting operations). In order to use these data, we fitted two forecasting models. The first model was estimated in the usual way, using data available up to the start of the in-sample week. The second model reversed the time ordering of the data, meaning that we were back-casting using data which were available after the end of the in-sample week. We expect the forecasts to do best at the beginning of the week, and the backcasts to do best at the end of the week, because they involve data which are closer to the days being predicted. Therefore, after estimating the two sets of models, we took a weighted combination of the two sets of forecasts in order to produce the final forecasts. More precisely, if we denote the forward forecasts by $\hat{y}^{(\mathrm{F})}_{t+h}$ and the backward forecasts by $\hat{y}^{(\mathrm{B})}_{t+h}$, with $h = [1, \ldots, 168]$, then the final forecasts are given by

$$\hat{y}_{t+h} = w_h \hat{y}^{(\mathrm{F})}_{t+h} + (1 - w_h)\hat{y}^{(\mathrm{B})}_{t+h},$$

where $w_h = \mathrm{sigmoid}(-7 + \frac{14*h}{168})$ and $\mathrm{sigmoid}(x) = 1/(1 + e^{-x})$ is the sigmoid function.

Forecasts were also required for one out-of-sample week without temperature data. In contrast to the in-sample weeks, temperatures were not provided for the out-of-sample week. Thus, we had to forecast the temperatures for this week, then use these when forecasting the demand during that week. We used the average temperature at the same period across the years as the forecast, as shown in Fig. 10. The differences between the out-of-sample forecasts obtained using different temperature sites are not as large as for the in-sample weeks, as can be seen from Fig. 9 (see the red bars). This is because

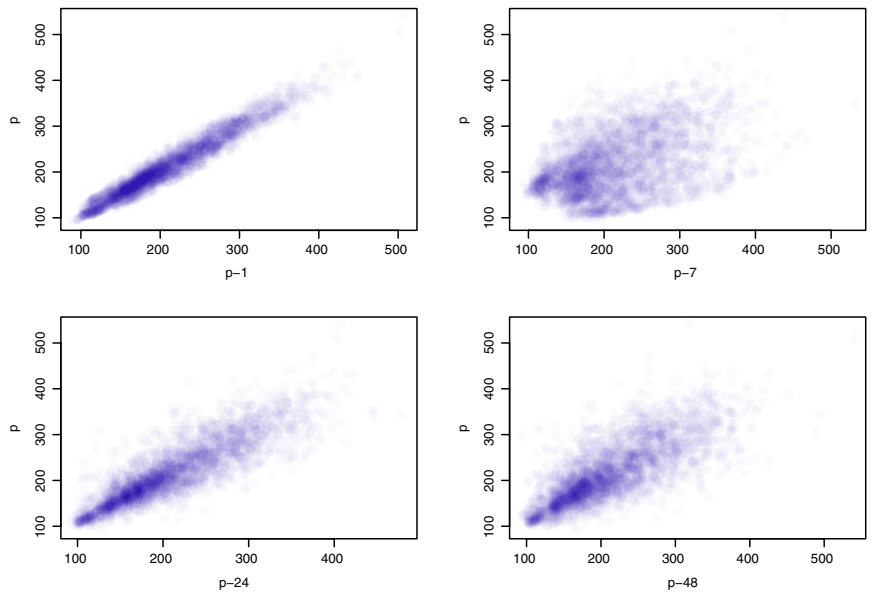

Fig. 8. Current demand plotted against lagged demand for different lags for Zone 18.

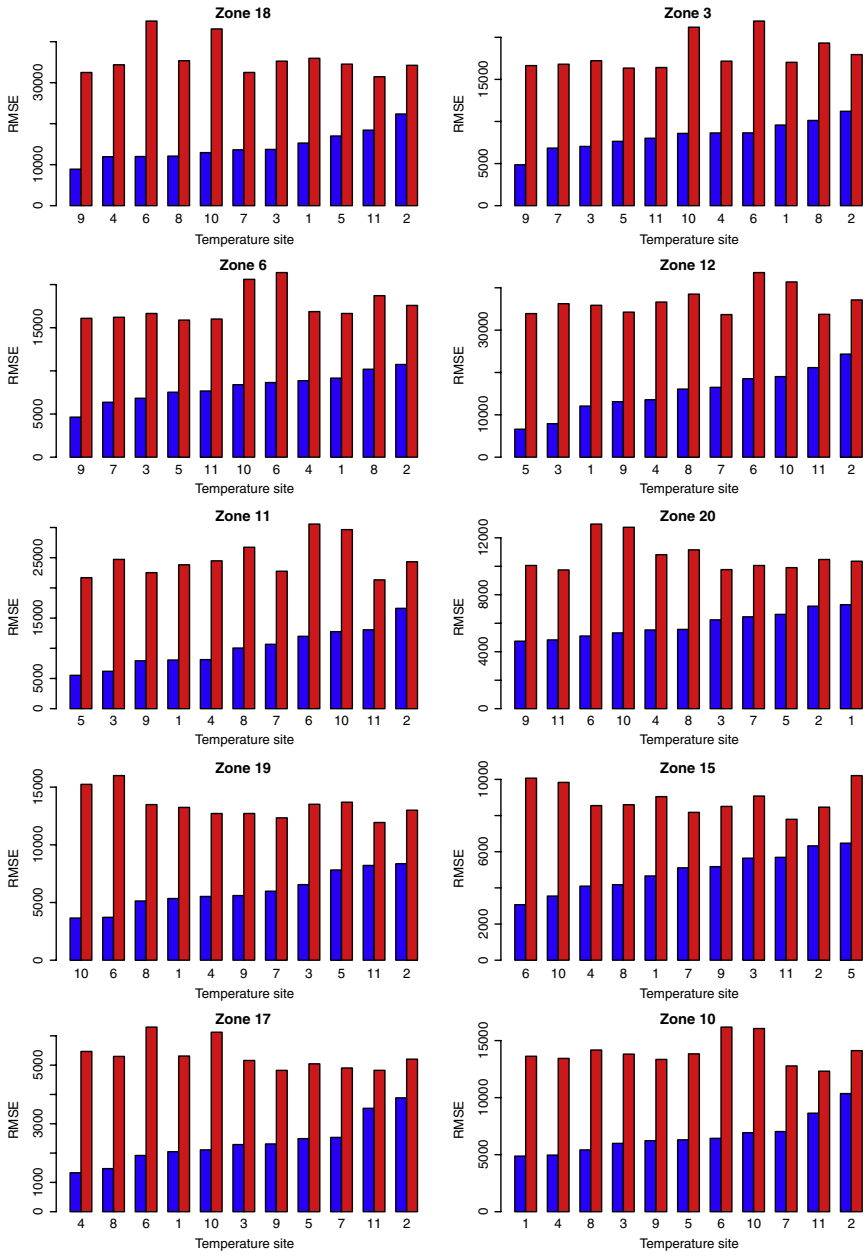

Fig. 9. Root mean square errors (RMSE) over the testing week using real (in blue) and forecasted (in red) temperatures. The sites are ranked according to the RMSE when using real temperatures. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

we do not have actual temperatures in that week, and our forecast temperatures have a smaller range than the actual temperatures. So, for out-of-sample forecasts, we average the forecasts obtained from the three best temperature sites when forecasting the demand, in order to reduce the variance of the forecasts. These sites are also shown in Fig. 9.

Since the demand patterns vary greatly during the year, and for computational convenience, we did not use all of the demand data for estimating our models. Instead, for each of the available years, we used only part of the data around the week to be forecast. More precisely, we computed the average temperature for the week to be forecast and, for each year, selected the 12 consecutive weeks around this week which have the closest average temperature. We then filtered the demand data by keeping these 12 consecutive weeks of each year. For the out-of-sample week, for which we do not have the average temperature for the week to be forecast, we performed the same calculations using the average temperature of the previous week.

The above procedures were followed for all zones except zones 2, 7 and 9. Zones 3 and 7 contain identical data, and Zone 2 contains values that are exactly 92.68% of the demand values in Zones 3 and 7. Consequently, we do

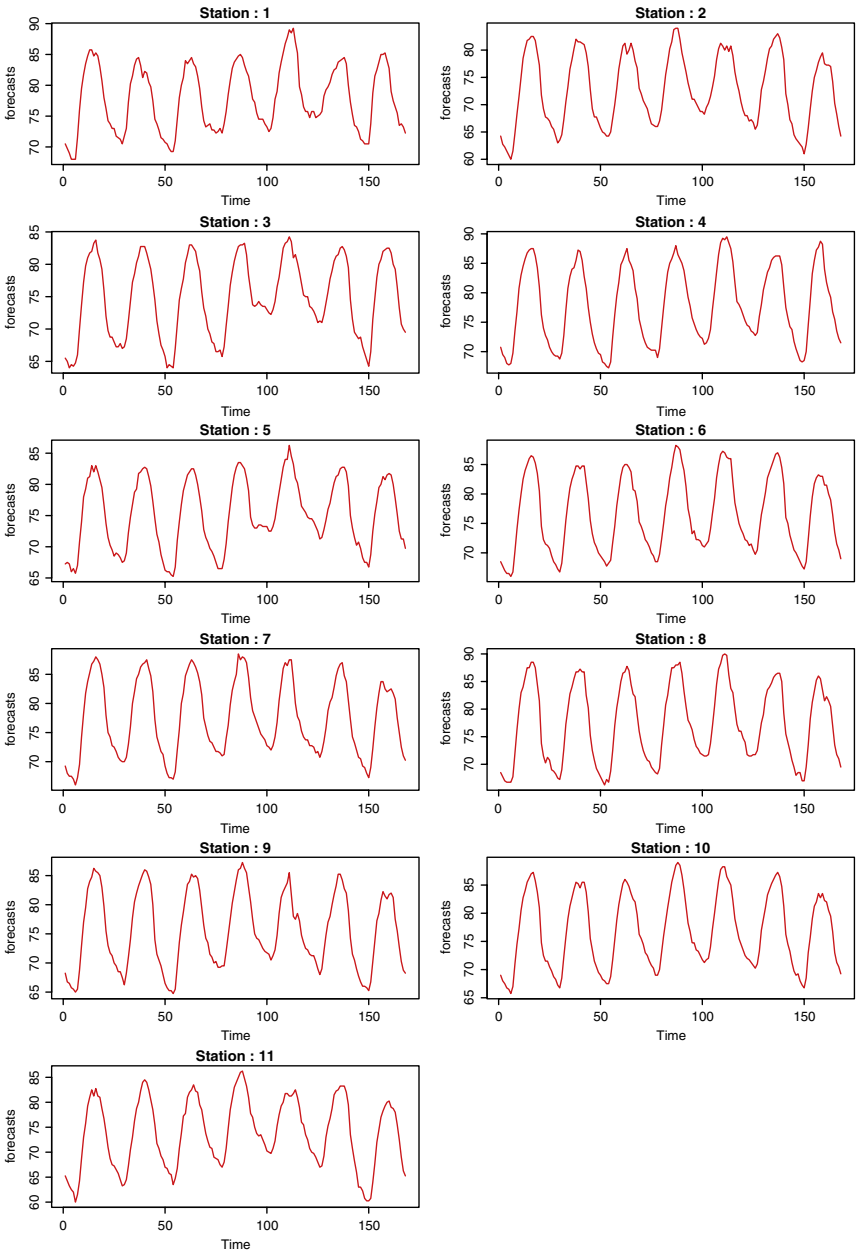

Fig. 10. Forecasts of temperature for the eleven stations.

not fit separate models for Zones 2 and 7; instead, we use the forecasts from Zone 3 to compute forecasts for Zones 2 and 7. For Zone 9, since we did not find any temperature-related patterns, we obtained forecasts for each period using the average of the same hour for every day of the week over the entire data set.

4. Forecasting model

One of the earliest electricity forecasting competitions was won by Ramanathan, Engle, Granger, Vahid, and Brace (1997) using a separate regression model for each hour of the day. This idea has since been used by Fan and Chen (2006), Fan and Hyndman (2012), Fay, Ringwood, Condon, and Kelly (2003) and McSharry, Bouwman, and Bloemhof (2005).

We follow the same approach and have a separate model for each hour of the day. These models are used recursively to produce the 168 forecasts required (24 h multiplied by 7 days). That is, we first produce the forecasts of the next day, then use them as inputs to produce forecasts for the day after. We also added the observations from the previous hour and the next hour in order to have more data. That is, when estimating the model for hour p, we used data from hours $p-1$, p and $p+1$.

We fit a model of this form for each hour of the day, for each zone, and for each of the nine weeks to be forecast. We

ended up with $24 \times 17 \times 9 = 3672$[1] models to estimate, with datasets of approximately 1000 observations and 43 input variables. The main predictors of the models are current and past temperatures (up to a week earlier) and past demand (up to a week earlier). In addition, the models also allow the demand to change with time-of-year, day-of-week, time-of-day, and on public holidays.

Each regression uses a nonparametric additive model with penalised regression splines. Component-wise gradient boosting is used to estimate each model, including variable selection during the fitting process. Both aspects are detailed in the following sections.

4.1. Non-parametric additive models

We use nonparametric additive models for forecasting the electricity demand. These models are in the regression framework, but have some non-linear relationships. In particular, the proposed models allow nonlinear and nonparametric terms using the framework of additive models (Hastie & Tibshirani, 1995).

The demand in hour p on day t is modelled using

$$y_{t,p} = c_p(t) + f_p(\boldsymbol{y}_{t,p}) + g_p(\boldsymbol{z}_{t,p}) + \varepsilon_t, \quad (1)$$

where

- $y_{t,p}$ denotes the demand on day t for hour p;
- $c_p(t)$ models all calendar effects (including time-of-year, day-of-week, holidays, etc.);
- $f_p(\boldsymbol{y}_{t,p})$ models the effects of recent demand variables, with $\boldsymbol{y}_{t,p}$ being a vector of past demands, prior to hour p on day t;
- $g_p(\boldsymbol{z}_{t,p})$ models the temperature effects, with $\boldsymbol{z}_{t,p}$ being a vector of recent temperatures variables at one of the temperature sites, prior to and including hour p on day t; and
- ε_t denotes the model error at time t.

We fit separate models of this form for each zone and for all in-sample and out-of-sample weeks.

4.1.1. Calendar effects

The calendar effects term, $c_p(t)$, includes annual, weekly and daily seasonal patterns, as well as public holidays.

- The day-of-week effect was modelled using a factor variable, which took a different value for each day of the week.
- The holiday effect was modelled using a factor variable, which took the value zero on non-work days,[2] some non-zero value on days before a non-work day, and a different value on days after a non-work day.
- The time-of-year effect was estimated using a simple first-order Fourier approximation (one sine variable and one cosine variable). A more complicated time-of-year effect was not necessary, as each model only included data within a 12 week period.

[1] There are only 17 zones to be modelled, because of the relationship between Zones 2, 3 and 7, and because we use a different approach for Zone 9.

[2] We used the US federal holidays listed on the US Office of Personnel Management website: http://www.opm.gov/Operating_Status_Schedules/fedhol/2008.asp.

4.1.2. Temperature effects

Due to thermal inertia in buildings, it is important to consider lagged temperatures as well as current temperatures in any demand forecasting model. The function $g_p(\boldsymbol{z}_{t,p})$ models the effects of recent temperatures on the aggregate demand, where $\boldsymbol{z}_{t,p}$ includes

- The current temperature, temperatures from the preceding 12 h, and temperatures for the equivalent hour on each of the previous two days;
- The minimum and maximum temperature over both the last 24 h and the previous day;
- The average temperatures for the previous day, the day preceding the previous day, and the last seven days.

Recall that, when forecasting the eight in-sample weeks, the temperatures from the site which gave the best forecasts (on the testing week) for each zone were used. When forecasting the out-of-sample week, the demand forecasts obtained using the best three temperature sites were averaged.

4.1.3. Lagged demand effects

We incorporate recent demand values into the model via the function $f_p(\boldsymbol{y}_{t,p})$, where $\boldsymbol{y}_{t,p}$ includes

- Lagged demand for each of the preceding 12 h, and for the equivalent hour in each of the previous two days. For example, when predicting the demand at 4 pm, we use the lagged demand from 4 am to 3 pm, as well as the lagged demand at 4 pm on the preceding two days.
- The minimum and maximum demand over the last 24 h.
- The average demand for the last seven days.

By doing this, the serial correlations within the demand time series can be captured within the model, and the variations in the demand level over the time can be embedded in the model as well.

Finally, we did not necessarily use all of the previous predictors in each model, but these were all candidate variables in our models. The process of selecting the variables is described in the next section, together with the gradient boosting algorithm.

4.2. Component-wise gradient boosting with penalised splines

Boosting is a prediction algorithm which stems from the machine learning literature, and is based on the idea of creating an accurate learner by combining many so-called "weak learners". Since its inception in 1990 (Freund, 1995; Freund & Schapire, 1996; Schapire, 1990), boosting has attracted a considerable amount of attention, due to its excellent prediction performance over a wide range of applications in both the machine learning and statistics literatures (Schapire & Freund, 2012). The improved performance of boosting seems to be associated with its resistance to overfitting, which is still under investigation (Mease, 2008; Mease & Wyner, 2008).

The gradient descent view of boosting (Friedman, 2001; Friedman & Hastie, 2000) has connected boosting with the more common optimisation view of statistical inference, while previous work had focused on bounding the generalisation error via the VC dimension and the distribution of

so-called margins (Schapire, Freund, Bartlett, & Lee, 1998). *Gradient boosting* (Friedman, 2001) interprets boosting as a method for function estimation, from the perspective of numerical optimisation in function space.

Bühlmann and Yu (2003) developed *component-wise gradient boosting* for handling high-dimensional regression problems by selecting one variable at each iteration of the boosting procedure. We refer to Bühlmann and Hothorn (2007) and Schapire and Freund (2012) for a general overview of boosting.

For all of our regression tasks, we used *component-wise gradient boosting* (Bühlmann, 2006; Bühlmann & Yu, 2003) with penalised regression splines (*P*-splines) (Eilers & Marx, 1996). By doing so, we take advantage of the good performance and the automatic variable selection of the boosting algorithm. In addition, *P*-splines allow us to have a smooth estimation of the demand. We now provide more details of the procedure we implemented.

Notice that Eq. (1) can be rewritten as

$$y_{t,p} = F_p(\boldsymbol{x}_t) + \varepsilon_{t,p},$$

where $\boldsymbol{x}_t = [t, \boldsymbol{y}_{t,p}, \boldsymbol{z}_{t,p}]$ contains all of the *potential* predictors to be considered in the model. Our goal is to estimate the function $F_p : \mathbb{R}^d \to \mathbb{R}$ for a given loss function. Since the forecasting accuracy for the competition was evaluated based on the *weighted root mean square error*, we used the quadratic loss function. The estimation of the function F_p based on a sample dataset $\{(y_{t,p}, \boldsymbol{x}_t)\}_{t=1}^T$ reduces to minimising

$$\hat{F}_p = \underset{F_p}{\operatorname{argmin}} \frac{1}{T} \sum_{t=1}^{T} (y_{t,p} - F_p(\boldsymbol{x}_t))^2.$$

Gradient boosting estimates F_p in a stagewise manner. We let $\hat{F}_p^{(m)}(\boldsymbol{x}_t)$ denote the estimation of F_p at the mth stage, where $m = 0, 1, \ldots, M$. The process begins with $\hat{F}_p^{(0)}(\boldsymbol{x}) = \bar{y}_p$, where $\bar{y}_p$ is the mean demand for hour p. Then, the model is updated using

$$\hat{F}_p^{(m)}(\boldsymbol{x}_t) = \hat{F}_p^{(m-1)}(\boldsymbol{x}_t) + \nu h_m(\boldsymbol{x}_t; \hat{\boldsymbol{\theta}}_m),$$

where $h_m(\boldsymbol{x}_t; \hat{\boldsymbol{\theta}}_m)$ is the weak learner estimate at the mth stage with parameters $\hat{\boldsymbol{\theta}}_m$, and $\nu \in [0, 1]$ is a shrinkage parameter. Gradient boosting with the quadratic loss is also called L_2 Boost (Bühlmann & Yu, 2003).

Let us note $\hat{h}_m(\boldsymbol{x}_t)$ as a shorthand for $h_m(\boldsymbol{x}_t; \hat{\boldsymbol{\theta}}_m)$. Given an estimation $\hat{F}_p^{(m-1)}$, each additional term $\hat{h}_m(\boldsymbol{x}_t)$ is obtained by computing the negative gradient

$$u_{t,p}^m = -\left.\frac{\frac{1}{2}\partial(y_{t,p} - F(\boldsymbol{x}_t))^2}{\partial F(\boldsymbol{x}_t)}\right|_{F(\boldsymbol{x})=\hat{F}_p^{(m-1)}(\boldsymbol{x})} = (y_{t,p} - \hat{F}_p^{(m-1)}(\boldsymbol{x}_t))$$

which gives the steepest descent direction. Then, a regression is applied on $\{u_{t,p}^m, \boldsymbol{x}_t\}_{t=1}^T$ by the weak learner, i.e.

$$\hat{\boldsymbol{\theta}}_m = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmin}} \sum_{t=1}^{T} [u_{t,p}^m - h_m(\boldsymbol{x}_t; \boldsymbol{\theta})]^2. \tag{2}$$

In other terms, $\hat{h}_m(\boldsymbol{x}_t)$ is selected so as to best predict the residuals from the previous model $\hat{F}_p^{(m-1)}(\boldsymbol{x}_t)$.

Finally, the solution is given by

$$\hat{F}_p(\boldsymbol{x}_t) = \hat{F}_p^{(M)}(\boldsymbol{x}_t) = \hat{h}_0(\boldsymbol{x}_t) + \sum_{m=1}^{M} \nu \hat{h}_m(\boldsymbol{x}_t), \tag{3}$$

where the estimation of $\hat{F}_p$ is improved continuously by an additional component (or boost) $\nu\hat{h}_m$ at stage m, and the hyperparameter M prevents overfitting by limiting the number of components.

From Eq. (3), we can see that the boosting procedure depends on two hyperparameters: ν, the shrinkage parameter, and M, the number of components (or number of stages). The value of ν affects the best value for M, i.e., decreasing the value of ν requires a higher value for M. Since they can both control the degree of fit, we should ideally find the best value for each of them by minimising some model selection criterion. However, Friedman (2001) showed that small values of ν are better, in that they lead to less overfitting of the boosting procedure. Hence, there is only one hyperparameter remaining for which the best value needs to be selected (Bühlmann & Yu, 2003).

In Eq. (2), the weak learner is estimating the model parameters using all of the predictors simultaneously. A better procedure can be used when there are many predictors: *component-wise gradient boosting* (Bühlmann, 2006; Bühlmann & Yu, 2003). The key idea is to use the weak learner with *one variable* at a time and select the one which makes the largest contribution to the fit. In other terms, Eq. (2) is replaced by the following procedure:

1. $\hat{\boldsymbol{\theta}}_m^{(k)} = \operatorname{argmin}_{\boldsymbol{\theta}\in\Theta} \sum_{t=1}^{T} [u_{t,p}^m - h_m(x_{kt}; \boldsymbol{\theta})]^2$, where x_{kt} is the kth variable of $\boldsymbol{x}_t$ and $k = 1, \ldots, d$.
2. $k_m = \operatorname{argmin}_{k\in 1,\ldots,d} \sum_{t=1}^{T} [u_{t,p}^m - h_m(x_{kt}; \hat{\boldsymbol{\theta}}_m^{(k)})]^2$.
3. Use $\hat{\boldsymbol{\theta}}_m^{(k_m)}$ at the mth stage.

Then, the final solution is expressed as

$$\hat{F}_p(\boldsymbol{x}_t) = \hat{h}_0(x_{k_0 t}) + \sum_{m=1}^{M} \nu \hat{h}_m(x_{k_m t}), \tag{4}$$

where $\hat{h}_m$ is a function of the k_mth predictor.

In our implementation, we used *P*-splines with 20 equally spaced knots and four degrees of freedom for the weak learner $h_m(x_{k_m t})$ terms. For the hyperparameter values, we set the value of ν to 0.15 and the maximum number of components (or stages) M to 500. Our implementation of the model depended on the *mboost* package for R (Hothorn, Bühlmann, Kneib, Schmid, & Hofner, 2012). We used the `gamboost` function with the following values for the `mboost_control` parameters: `nu=0.15` and `mstop=500`. For the base learners, we used the `bbs` function for numerical variables and the `bols` function for factor variables. Finally, we select the best number of stages $M (M \in \{1, \ldots, \texttt{mstop}\})$ using the `cvrisk` function with 5-fold cross-validation.

5. Model analysis

In this section, we analyse and interpret the results of the proposed forecasting model, in order to shed light on its attractive features.

Table 1
Description of all potential predictors. Forecasts are from demand$[t-1]$ to demand$[t+h-1]$, where $h \in \{1, \ldots, 24\}$.

Id	Variable	Id	Variable	Id	Variable	Id	Variable
1	Day of the week	12	demand$[t-8]$	23	temp.$[t+h-2]$	34	temp.$[t+h-13]$
2	Holiday	13	demand$[t-9]$	24	temp.$[t+h-3]$	35	temp.$[t+h-25]$
3	Time of year (sin)	14	demand$[t-10]$	25	temp.$[t+h-4]$	36	temp.$[t+h-49]$
4	Time of year (cos)	15	demand$[t-11]$	26	temp.$[t+h-5]$	37	min. temp. (prev 1)
5	demand$[t-1]$	16	demand$[t-12]$	27	temp.$[t+h-6]$	38	min. temp. (prev 2)
6	demand$[t-2]$	17	demand$[t+h-25]$	28	temp.$[t+h-7]$	39	max. temp. (prev 1)
7	demand$[t-3]$	18	demand$[t+h-49]$	29	temp.$[t+h-8]$	40	max. temp. (prev 2)
8	demand$[t-4]$	19	min. demand (prev 1)	30	temp.$[t+h-9]$	41	avg. temp. (prev 1–7)
9	demand$[t-5]$	20	max. demand (prev 1)	31	temp.$[t+h-10]$	42	avg. temp. (prev 1)
10	demand$[t-6]$	21	avg. demand (prev 1–7)	32	temp.$[t+h-11]$	43	avg. temp. (prev 3)
11	demand$[t-7]$	22	temp.$[t+h-1]$	33	temp.$[t+h-12]$	–	–

Fig. 11 gives the root mean squared errors (RMSE) obtained for the different zones on the testing week. The zones with larger errors are those with higher average demands (see Fig. 1). This suggests that zones with high average demands will carry more weight in the final error.

Fig. 12 gives the true hourly demand together with the fitted values for an increasing number of boosting iterations. Recall that gradient boosting is a stagewise fitting procedure which depends on a hyperparameter M (the number of boosting iterations), as per Eq. (4). We can see that the first iterations significantly reduce the error of the model, while the final iterations contribute to the model less and less. This confirms the theoretical analysis performed by Bühlmann and Yu (2003), who prove that the bias decreases exponentially and the variance increases with exponentially diminishing terms as the number of boosting iterations M increases.

One of the most attractive features of the componentwise boosting algorithm is the automatic variable selection induced by the procedure at each iteration. That is, among all of the potential predictors which are given in Table 1, few will be selected to contribute to each hourly model. See Section 4.1 for a more detailed description of the different predictors.

Note that the final solution of the boosting procedure, given in Eq. (4), can be rewritten as

$$\hat{F}_p(\boldsymbol{x}_t) = \sum_{m=0}^{M} \nu \hat{h}_m(x_{k_m t}) = \sum_{k\in\{1,\ldots,d\}} \underbrace{\sum_{\{m:k_m=k\}} \nu \hat{h}_m(x_{k_m t})}_{\hat{H}(x_{kt})}$$
$$= \sum_{k\in\{1,\ldots,d\}} \hat{H}(x_{kt}),$$

where $\hat{H}(x_{k_m t})$ is the relative contribution of the variable k to the final model and d is the number of initial predictors (the dimensionality of $\boldsymbol{x}_t$).

Let us define the model without the effect of predictor j as

$$\hat{F}_p^{(-j)}(\boldsymbol{x}_t) = \sum_{k\in\{1,\ldots,d\}\setminus\{j\}} \hat{H}(x_{kt}),$$

and the corresponding squared error as

$$E^{(-j)} = \sum_{t=1}^{T} (y_t - F_p^{(-j)}(\boldsymbol{x}_t))^2.$$

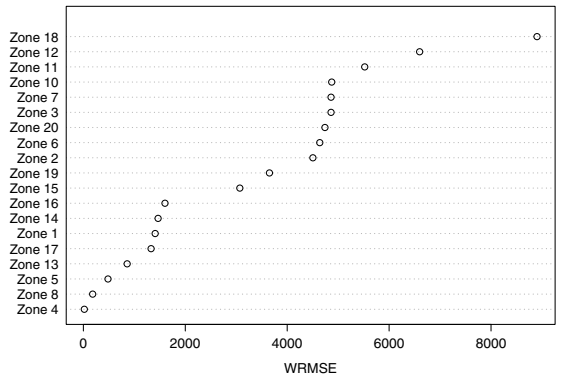

Fig. 11. Root mean squared error (RMSE) obtained for each zone on the testing week.

We then define the relative importance of a predictor j as

$$I_j = \frac{E^{(-j)} - E}{E},$$

where E is the squared error of the final model with all of the selected variables. In other words, the predictors which increase the error most after removing their relative effects are the most influential variables, *given* the other predictors.

Fig. 13 shows the ten variables which have the greatest influence (according to I_j) on the demand at different hours of the day. We assign the importance value $I_j* = 100$ to the most influential variable, and the values of the others are scaled accordingly, as per Friedman and Hastie (2000). To assist in visualisation, variables belonging to the same effect are plotted with the same color. That is, demand variables are colored in green, temperature variables in blue and calendar variables in red.

We can see that variables from all of the different effects are selected: calendar effects (in red), temperature effects (in blue) and lagged demand effects (in green). The importance of these different effects has been shown in Section 2. We have seen the different calendar effects in Figs. 2–6. The clear dependence between demand and temperature was illustrated in Fig. 7. Finally, Fig. 8 has shown that there is a clear dependence between the actual demand and previous lagged demand variables.

For the first horizon ($h = 1$), we see that the most important variable is the current demand (variable 5). This is not surprising, since the demand at hour $p+1$ is highly

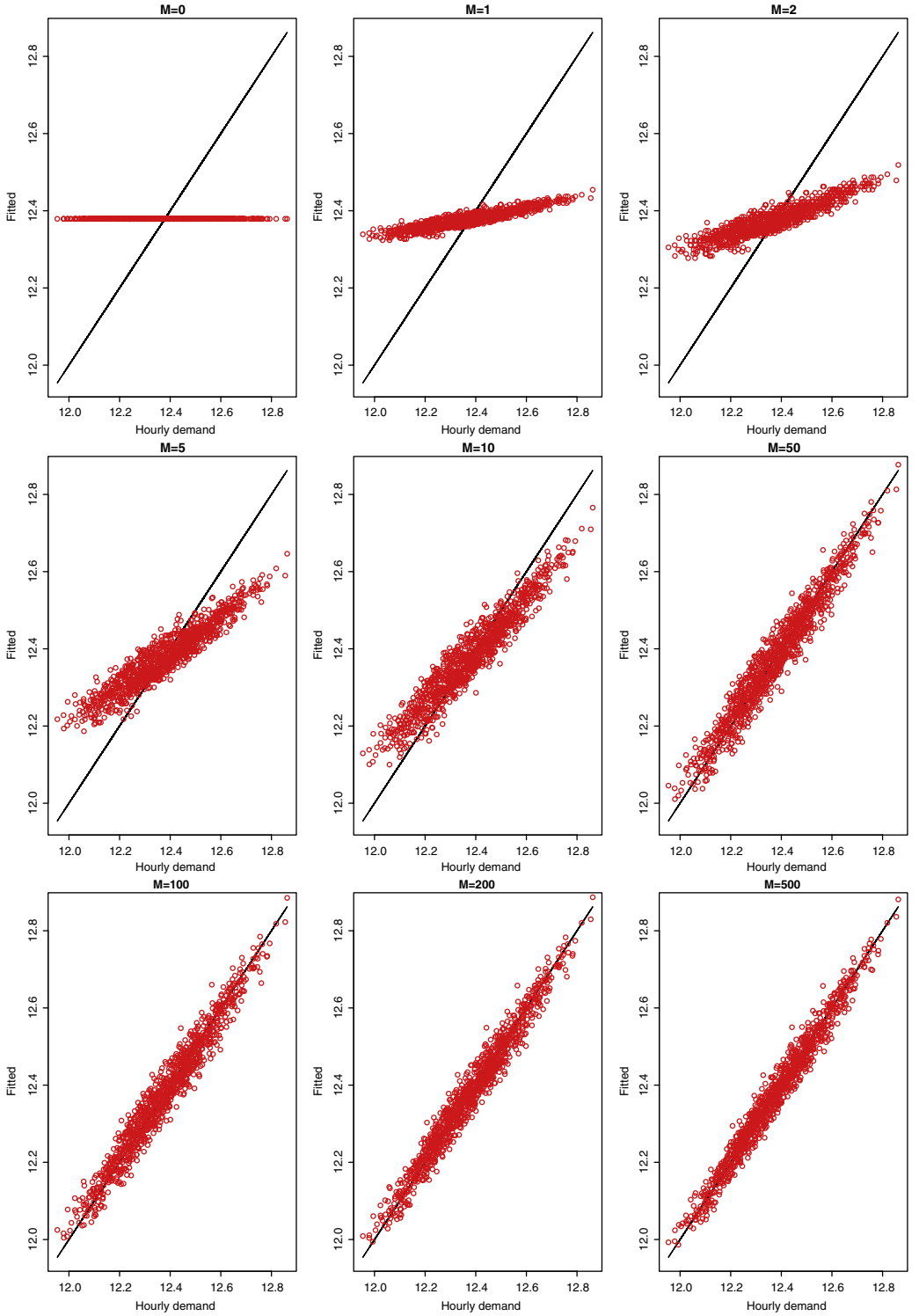

Fig. 12. $M = 500$ is the best in terms of cross-validation.

dependent on the demand at hour p. However, as one moves away from the starting point (i.e. $h = 2$–3), other variables, such as the demand at the equivalent hour for the previous day (variable 17), become important and the current demand loses importance.

During working hours (10:00–19:00 or $h = 4$–13), temperature variables (in blue) dominate demand variables (in green). In fact, the most important variables include the current temperature and some of the corresponding lagged temperatures (variables 22–25). During that period of the

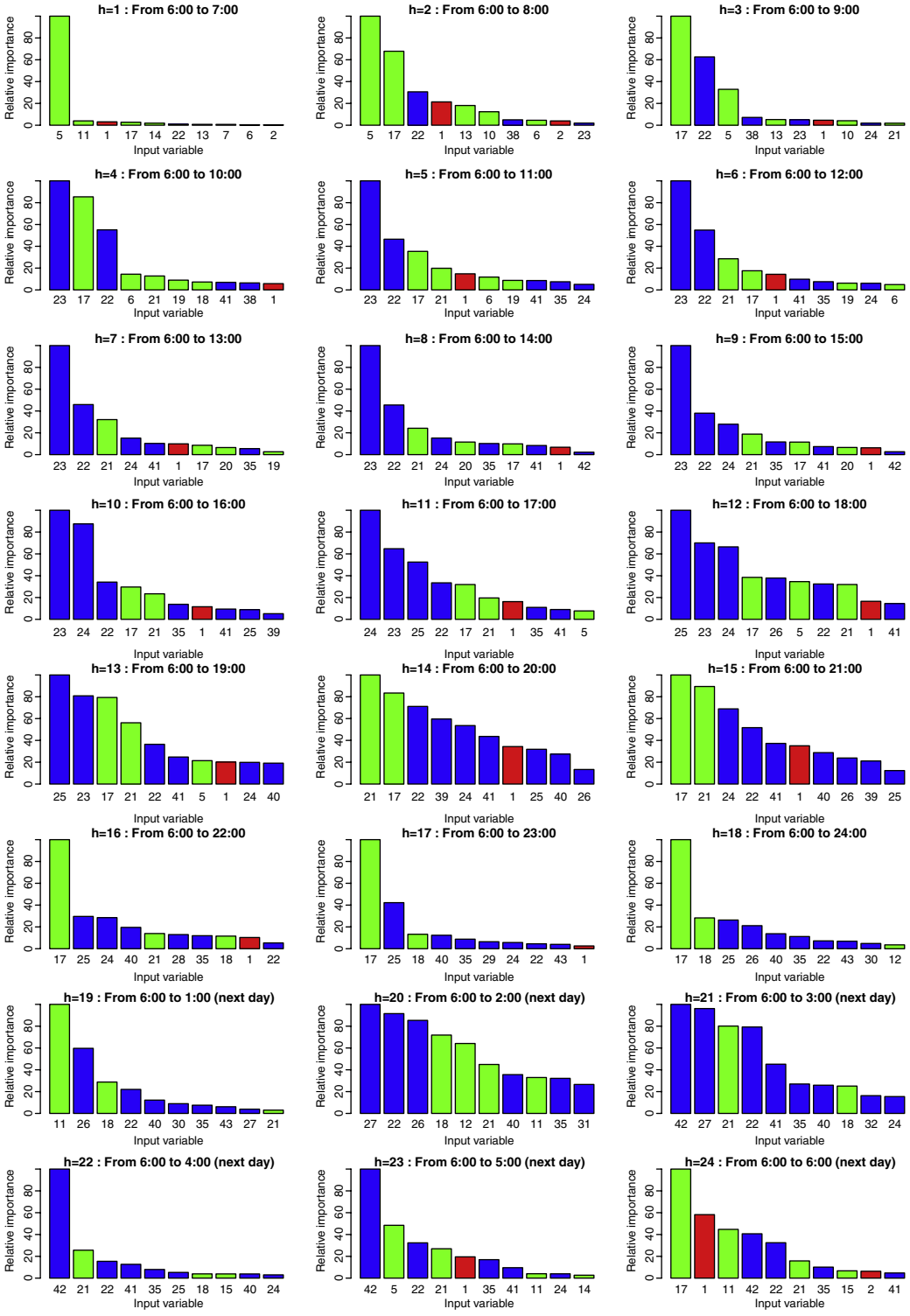

Fig. 13. Relative importances of the five first variables for the demand for different times of the day. Demand variables are colored in green, temperature variables in blue, and calendar variables in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

day, we can also see that the variable 17 is gaining importance with the horizon, with the highest importance at 19:00.

For the last hours of the day (20:00–24:00 or $h = 14$–18), temperature variables become the most influential variables. For 20:00 and 21:00, variables 17 and 21 are both

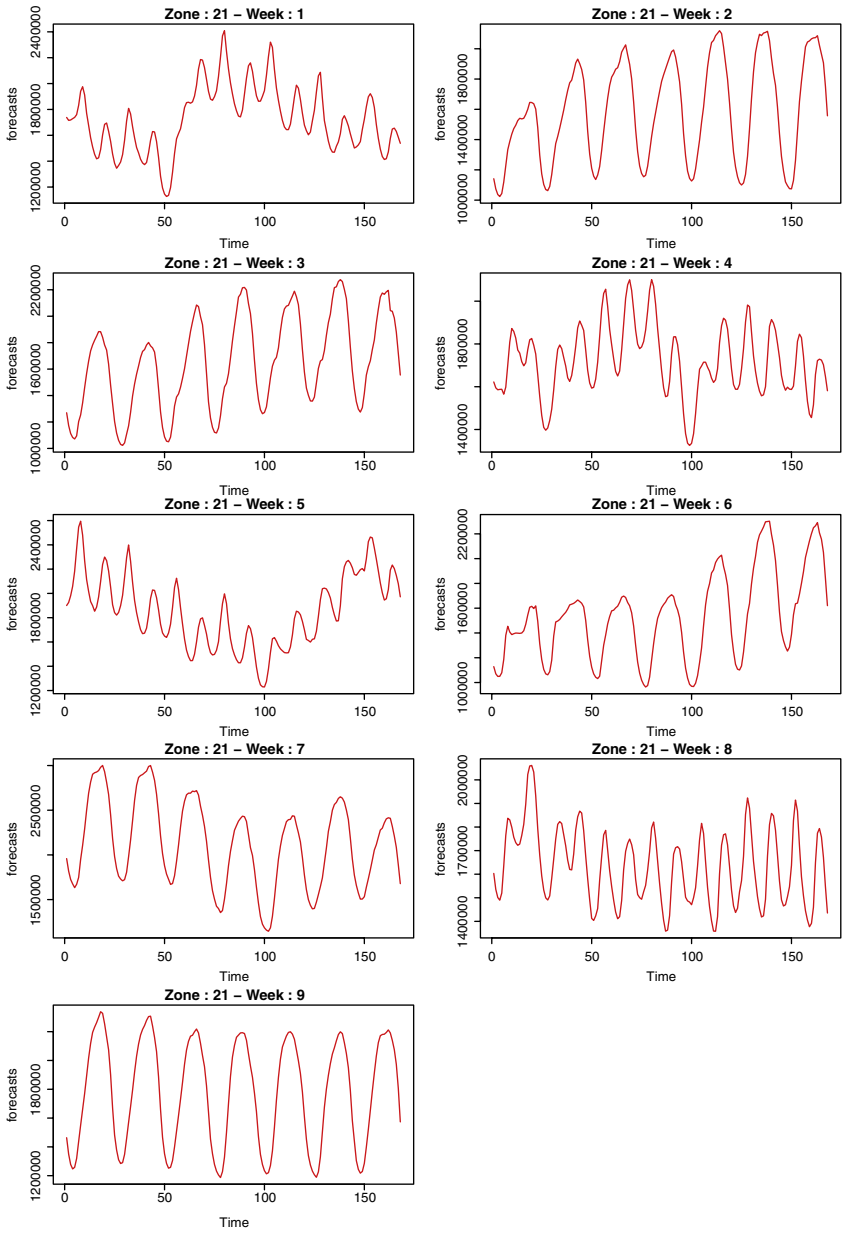

Fig. 14. Forecasts for Zone 21 for the eight in-sample weeks and the out-of-sample week.

highly important, while for the remaining hours of the day, only variable 17 remains a very influential variable.

For the first hours of the next day (1:00–05:00), temperature variables gain more importance, with a new variable appearing between 03:00 and 05:00, namely the average temperature of the previous day (variable 42).

Finally, at 06:00, we see that variable 17 is again the most influential variable, together with the day of the week.

The analysis of Fig. 13 has shown that the relative importances of the different variables and effects change with the time of the day. This shows that the dependence between the demand and the different variables considered changes with the forecasting horizon, making electricity load forecasting a challenging statistical problem.

For illustrative purposes, Fig. 14 gives the forecasts of the aggregate series (i.e., the sum of the forecasts over all zones) for the eight in-sample weeks and the out-of-sample week.

6. Conclusion

Our entry ranked fifth out of 105 participating teams. This suggests that our modelling strategy is competitive with the other models used to forecast the electricity demand. We have identified several aspects that help make our modelling strategy successful.

First, as in any prediction task, data analysis allowed us to identify and clean the data from any corrupted

information, enabling a better model performance. The data analysis step was also important for the identification of useful variables to use in the model.

Second, we used different models, including different effects, for each hour of the day, in order to model the demand patterns which are changing throughout the day.

Third, each hourly model allowed both nonlinear and nonparametric terms to be included in the final model. This gave the model a great deal of flexibility, and avoided making too many assumptions about the data generating process.

Finally, gradient boosting has often been shown to be an effective prediction algorithm for both classification and regression tasks. By selecting the number of components to be included in the model, we can easily control the so-called bias–variance trade-off in the estimation. In addition, component-wise gradient boosting increases the attractiveness of boosting by adding automatic variable selection during the fitting process.

The Kaggle load forecasting competition was a challenging prediction task which required several statistical problems to be solved, such as data cleaning, variable selection, regression, and multi-step time series forecasting.

References

Bühlmann, P. (2006). Boosting for high-dimensional linear models. *Annals of Statistics*, *34*(2), 559–583.

Bühlmann, P., & Hothorn, T. (2007). Boosting algorithms: regularization, prediction and model fitting. *Statistical Science*, *22*(4), 477–505.

Bühlmann, P., & Yu, B. (2003). Boosting with the L_2 loss: regression and classification. *Journal of the American Statistical Association*, *98*, 324–339.

Eilers, P. H. C., & Marx, B. D. (1996). Flexible smoothing with *B*-splines and penalties. *Statistical Science*, *11*(2), 89–121.

Fan, S., & Chen, L. (2006). Short-term load forecasting based on an adaptive hybrid method. *IEEE Transactions on Power Systems*, *21*(1), 392–401.

Fan, S., & Hyndman, R. J. (2012). Short-term load forecasting based on a semi-parametric additive model. *IEEE Transactions on Power Systems*, *27*(1), 134–141.

Fay, D., Ringwood, J. V., Condon, M., & Kelly, M. (2003). 24-h electrical load data—a sequential or partitioned time series? *Neurocomputing*, *55*(3–4), 469–498.

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and Computation*, 1–50.

Freund, Y., & Schapire, R. R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the 13th international conference on machine learning* (pp. 148–156).

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232.

Friedman, J., & Hastie, T. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *Annals of Statistics*, *28*(2), 337–407.

Hastie, T. J., & Tibshirani, R. (1995). *Generalized additive models*. London: Chapman & Hall/CRC.

Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M., & Hofner, B. (2012). *mboost: model-based boosting*. R package version 2.1-3. http://cran.r-project.org/package=mboost.

McSharry, P. E., Bouwman, S., & Bloemhof, G. (2005). Probabilistic forecasts of the magnitude and timing of peak electricity demand. *IEEE Transactions on Power Systems*, *20*, 1166–1172.

Mease, D. (2008). Evidence contrary to the statistical view of boosting: a rejoinder to responses. *Journal of Machine Learning Research*, *9*, 195–201.

Mease, D., & Wyner, A. (2008). Evidence contrary to the statistical view of boosting. *The Journal of Machine Learning Research*, *9*, 131–156.

Ramanathan, R., Engle, R. F., Granger, C. W. J., Vahid, F., & Brace, C. (1997). Short-run forecasts of electricity loads and peaks. *International Journal of Forecasting*, *13*, 161–174.

Schapire, R. (1990). The strength of weak learnability. *Machine Learning*, *5*(2), 197–227.

Schapire, R. E., & Freund, Y. (2012). *Boosting: foundations and algorithms*. The MIT Press.

Schapire, R. E., Freund, Y., Bartlett, P., & Lee, W. S. (1998). Boosting the margin: a new explanation for the effectiveness of voting methods. *Annals of Statistics*, *26*(5), 1651–1686.

Hong, T., Pinson, P., & Fan, S. (2014). Global energy forecasting competition 2012. *International Journal of Forecasting*, *30*(2), 357–363.