Energy Forecasting for the Global Energy Forecasting Competition 2014

Semester Project Report

Fabian Brix
MSc Candidate
School of Computer & Communication Sciences
Swiss Federal Institute of Technology Lausanne
fabian.brix@epfl.ch

Supervisor
Tri Kurniawan Wijaya
PhD Student
School of Computer & Communication Sciences
Swiss Federal Institute of Technology Lausanne
tri-kurniawan.wijaya@epfl.ch

Abstract—**In this report we summarize the findings rom the Semester Project "Energy Forecasting for the Global Energy Forecasting Competition 2014". In this project we create a simulation environment for the data made available through the load forecasting track of the competition. In our approach to forecasting we draw from recent publications in the field even if our final methods are not as sophisticated as the methods described therein. The student's interest in learning about forecasting methodologies with this project has been fulfilled.**

I. Introduction

A. GEFCom 2014

The Global Energy Forecasting Competition (GEFCom 2014) is the second edition of a competition first held on Kaggle in 2012 that attracted hundreds of participants contributing many novel ideas to the energy forecasting field. The second edition lasted from 08/15/2014 to 12/15/2014 and was sponsored by several IEEE bodies and the International Journal of Forecasting. It included four competition tracks: *electric load*, electricity price, wind power and solar power forecasting. This time around the different tracks were hosted on the CrowdANALYTIX community platform. During the competition the data was released on a rolling basis: after the release of the inital dataset, the organizers released the target values that had to be predicted the week before on a weekly rolling basis. In contrast to GEFCom 2012, GEFCom 2014 required the participants to explore *probabilistic forecasting*. In order to feature on the final leaderboard the contestants needed to submit entries for the tasks posed throughout the total forecasting horizon. In the course of this semester project we focused solely on the electric load forecasting track. The aim of this track was to forecast the probabilistic distribution, in quantiles, of the hourly loads of one energy utiliy. It had a forecasting horizon of one month and consisted of three trial periods and twelve competitive periods. In order to feature on the final leaderboard nine competitive submissions were required. The fact that the temperature data needed for the load forecast horizon was not provided meant that not only load, but also temperature had to be forecast. Participating in the competition was not prioritized after initial attempts, since there was a significant learning curve in learning how to approach a forecasting problem and setting up the experiments. Instead the focus was on comparison of the performance of different methods for temperature and load prediction on the whole dataset of the competition via cross validation.

B. Review of Energy Load Forecasting

In this section we are going to briefly, by no means exhaustively, introduce & review the area of energy load forecasting. Before we can start reviewing the field of study, we first have to clarify its actual meaning. The term load itself is ambiguous in the sense as to which energy entity it refers to [1]. In connection with forecasting "load" usually refers to demand in kW or energy in kWh. "Demand" is used to refer to the electric power that is delivered. "Energy" is the integral of power over time and therefore refers to total consumption. On hourly data, they are generally assumed to be the same.

Energy forecasting is also a prevalent term with many different meanings. We can refer to just "forecasting the energy (in kWh)" or whole range of wider applications "such as gas and electric load forecasting, renewable generation forecasting, price forecasting, demand response forecasting, outage forecasting, and so forth" [1]. Energy load forecasting or just load forecasting, as it is commonly referred to, is usually concerned with the prediction of hourly, daily, weekly, and annual values of the system demand and peak demand of an electric utility [2]. Such forecasts are sometimes categorized as short-term (up to 1 week), medium-term (1 week - 1 year) and long-term (> 1 year) forecasts, depending on the time horizon. In the load forecasting track of GEFCom 2014 we are concerned with forecasting the daily electricity load of a utility for a whole month for example. The task is therefore on the threshold between short-term and medium-term load forecasting.

Forecasting has an over 100-year-long history in the utility industry and is an integral part of planning of energy systems and their effective operation and maintenance. The reader is encouraged to consult [3] for a condensed version of the history of load forecasting. Since forecasting is necessary for planning the accuracy of forecasts have a major impact on electric utilities and their regulators [2]. In case of overestimation of future energy load, utility providers will operate too many units possibly driving energy demand and in case of long-term forecasts investment in the construction of new infrastructure can be wasted. Underestimation leads to unmet demand and systems that are vulnerable to crashes.

Electricity load follows a nonlinear, volatile pattern subject to several exogenous variables such as weather conditions, randomness in human behavior leading to randomness in demand and economic conditions and demographic changes. In GEFCom 2014 the exogenous variables are limited to weather conditions in the form of recorded temperature at several sites and calendar effects such as the effects of weekends and holidays on the electricity demand. The output of load forecasts can either be point forecasts or estimates of the probability distribution of values of future demand or load allowing for improved decision analysis, as required during GEFCom 2014. “In today’s competitive and dynamic environment, more and more decision making processes in the power and energy industry are relying on probabilistic forecasts. The applications of probabilistic energy forecasts spread across planning and operations of the entire energy value chain” (GEFCom organizers).

Due to the significance of load forecasting and the nature of the problem there are of course many different approaches to achieving good forecasts. Please refer to [4] Section 2.2 for a review of literature reviews of load forecasting if you are interested in evolution of methods. In this report we analyse the predictions produced by algorithms that are assumed to be capable of capturing the nonlinear dependencies between the exogeneous variables and the load such as general additive models, random forests and feedforward neural networks. We do not strive to implement any algorithms ourselves, but rather put to use the respective R packages available. We rely on insights from recent publications in the field of probabilistic forecasting [5] and semi-parametric additive models [2], [**?**] concerning both the use of models and features.

II. Dataset and Evaluation Metrics

A. Dataset

The dataset provided by GEFCom 2014 includes hourly historical load and temperature data for one utility in an undisclosed district, divided into zones, on the east coast of the United States of America. No further information as to the nature of the energy demand at the utility, be it domestic or industrial is given. The 25 weather stations in the dataset provide historical temperature for their respective zones. However, the load data consists only of the system level load in Megawatts (MW) and does not include the zonal level load series. Therefore, forecasts in the context of Smart Grid Technology are not required. The temperature data made available consists of 25 temperature series in Fahrenheit dating from 01/01/2001 1am to 12/01/2011 midnight. The load data of the utility is recorded starting from the 01/01/2005 at 1am with the same enddate.

The dataset acted both as training and validation for reasons already discussed (Section I-A). For the load forecasting track we are concerned with the dataset consists of 15 spreadhseets in the format of Comma-Seperated Values (CSV). The first spreadsheet contains data starting from 01/01/2001 at 1am up until midnight on the 10/01/2014 from when on the incremental spreadsheets released every week contain only one month of data.

The forecasts were required to be made starting from 10/01/2010 on a monthly rolling basis for 15 months. We evaluate our models on 14 months, because the last month of data was not provided. As mentioned in section I-A the nature of the data provided required the contestants to produce their own temperature forecasts for the month ahead in the dataset.

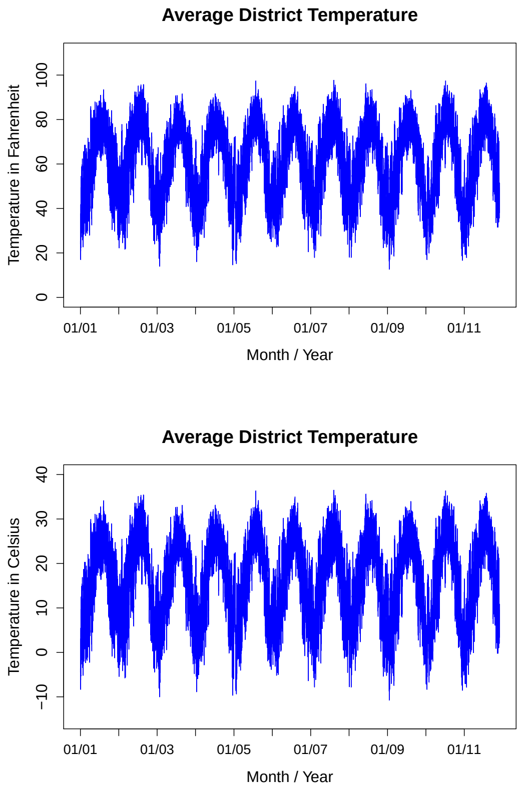


Fig. 1: Average district temperature over the whole given time period

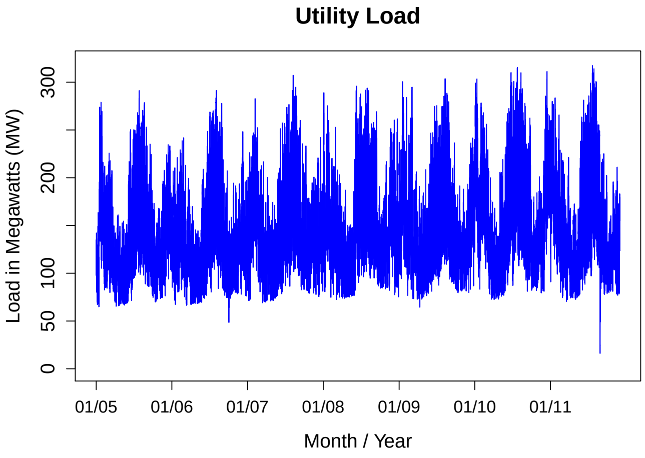


Fig. 2: Utility load of the whole given time period

B. Data Cleaning

An annoying feature of the dataset is that the timestamps are not saved in the international ISO 8601 standard, but as “MMddYYYY H:m” without leading zeros for both days and months. Fortunately, the dataset was provided continuously without gaps and therefore the problem could be easily solved by hard-coding the first and last datetimes and using these to generate the needed sequence of datetimes.

C. Temperature Preprocessing

We use the hourly average temperature of all weather stations as the basis series for our temperature forecasts (Figure 4a). In other words, we use the average district temperature to predict the district utility load.

In another approach we compute the cross-correlations for the 25 weather station (Figure 3). The high correlations suggest that, taking into account the inevitable inaccuracy of the temperature forecasts, station 1 can be used for forecasting.

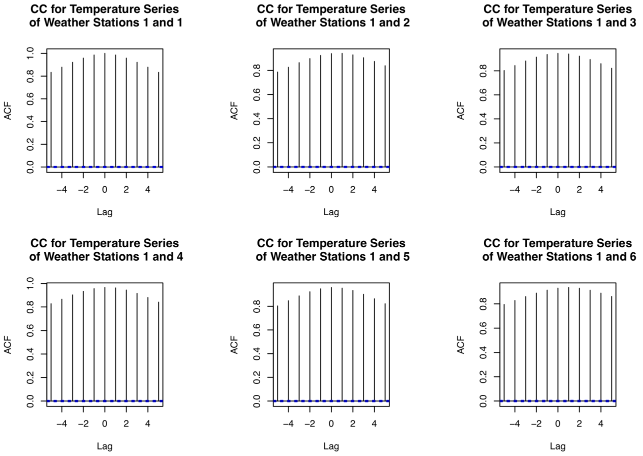


Fig. 3: Cross Correlation Plots of Temperature Station 1 Series with Series of Temperature Stations 1-6. Cross Correlations for Stations 7-25 omitted for convenience of display.

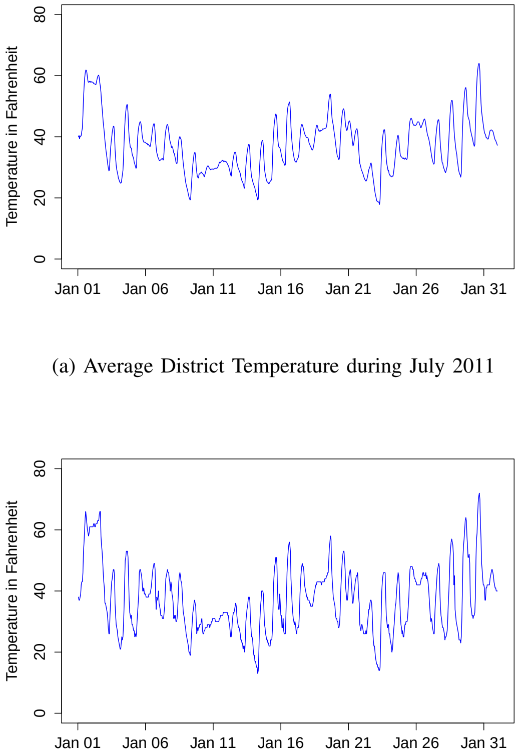
Furthermore we perform a principal component analysis (PCA) on the temperature series and evaluate the use of the first principal component that captures around 90% of the variance in the temperature (Section V).

D. Evaluation Metrics

The competition requires each of the participants to provide percentile forecasts, that is forecasts of the quantiles $\tau = 0.01, 0.02, \ldots, 0.99$ with natural lower and upper bounds. The Evaluation Metric employed to score the contestants’ submissionsis the tilted loss/error function also known as the pinball loss/error function. In the following paragraphs let y denote an observation and $\hat{y}$ denote a corresponding forecast while ξ is defined as the residual $y - \hat{y}$.

$$L_\tau(\xi) = \begin{cases} \tau\xi & \text{if } \xi \geq 0 \\ (\tau-1)\xi & \text{if } \xi < 0 \end{cases} \quad \text{where } \xi = (y - \hat{y})$$

To evaluate the full predictive densities, this score is then averaged over all target quantiles, from 0.01 to 0.99, for all time periods throughout the forecast horizon. The lower the pinball score, the better the forecast.



(a) Average District Temperature during July 2011

(b) Temperature of 1st Weather Station during July 2011

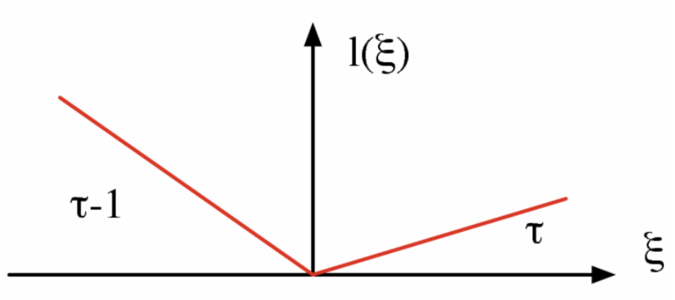


Fig. 5: The tilted loss function for 50th, $\tau = 0.5$, and 75th, $\tau = 0.75$, quantile [6].

In our own evaluation of forecasts, we further use some well-established point error metrics, for the simple reason that we generate our quantile predictions starting from a point prediction. The measures are the Mean Average Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE).

The MAE and RMSE measures are the two most common scale-dependent errors, meaning that the residuals ξ_i (for ith observations) are on the same scale as the data. Hence, MAE and RMSE are in units of Fahrenheit or Mega Watt for our data set.

$$\mathrm{MAE} = \mathbb{E}[\xi_i]$$

$$\mathrm{RMSE} = \sqrt{\mathbb{E}[\xi_i]}$$

As a percentage error the MAPE measure is scale-independent. In our case it is useful for giving an immediate sense of the relative scale of the error.

$$\mathrm{MAPE} = \mathbb{E}\left[\left|\frac{100\xi_i}{y_i}\right|\right]$$

The measure is undefined for $y_i = 0$. Fortunately, in our dataset all values are several integers larger than zero for both load and temperature series so that the measure is neither undefined or affected by extreme values.

III. Feature Selection

In the following paragraphs we explain the methodology we use to obtain our final feature selection for both temperature and load forecasting. The choice of features is limited to calendar features for temperature forecasting, because the only additional data allowed in the competition are Federal US holidays and the forecasting horizon is set to 1 month ahead.

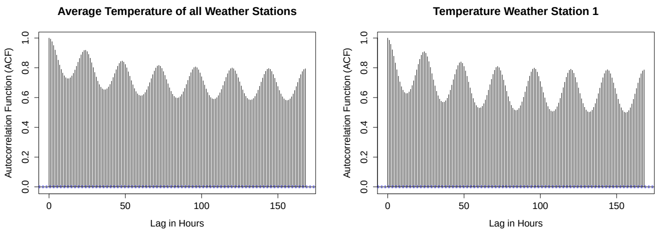
A. Time Lags

First of all, we want to find out which previous values can be useful in a prediction. Given a time horizon of 1 month we would like to forecast the whole month using only one model for reasons of convenience. This circumstance induces a time lag of minimum 28 days for the month of february. For simplicity of implementation we define the same minimum lag of 31 for every month of the year. In order to choose the lag variables we interpret the cross-correlation of observations y_i and y_{i-k}, for different hourly lags $k = 1, 2, ..,$ of both temperature and load time series in (Figure 6). This serial correlation of observations is known in statistics as autocorrelation and demonstrates the similarity of observations in a series as a function of the time lag between them. Using the empirical mean and standard deviation of our target series $Y = \{y_1, y_2, \ldots, y_i, \ldots, y_N\}$ we can compute the autocorrelation for lag k with:

$$\mathcal{R}(k) = \frac{\mathbb{E}\left[(y_i - \mu)(y_{i-k} - \mu)\right]}{\sigma^2}$$

As can be seen in both figure 6a and 6b the highest autocorrelations occur with a periodicity of 24 hours. Therefore we choose 24 hours as the base number for selecting the time lags of our predictions. The average temperature series in the former figure displays less variance in the correlations than the first station in the latter because the hourly temperature is less extreme due to the averaging over zones.

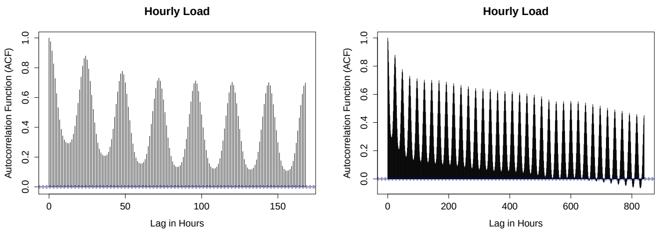
Figure 7 shows the autocorrelation function of the load series for different maximum lags. As can be seen the load correlations also have a periodicity of 24 hours (Figure 7a). The other figure 7b show that after an initial exponential decrease the correlations decrease near linearly. Since the type of day has an influence on the utility load (Section III-B), we set the basis for time lags to $7 \cdot 24$ hours for the load and the minimum time lag for monthly load forecasts therefore to 35 days.



(a) time lag: 7 days, 168 hours

(b) time lag: 7 days, 168 hours

Fig. 6: Autocorrelation function estimates for hourly average temperature of district and hourly temperature at station 1 for different maximum time lags.



(a) time lag: 7 days, 168 hours

(b) time lag: 35 days, 840 hours

Fig. 7: Autocorrelation Function Estimates for hourly load data in MW for different maximum time lags.

B. Calendar Features

Figures 6 and 7 have shown both temperature and load series to possess daily frequency. We therefore extract the hour of the day from the datetimes and use it as a feature.

Figure 8 demonstrates the seasonal change of temperature and load throughout the year. In order to capture the seasonality in our models for both temperature and load, we use the time of the year as a feature. We indicate the progress of the current year with a continously growing sequence

$$\mathrm{toy} = \{\mathrm{toy}_1, \mathrm{toy}_2, \ldots, \mathrm{toy}_i, \ldots\}, \quad \mathrm{toy}_i \in [0, 1]$$

where the endpoints 0 and 1 only approximately, due to leap years, correspond to January 1st 1am and and subsequent midnight January 1st (Section II.B [7]). An alternative is to use a categorical variable indicating the present month.

The features we discussed so far are analyzed for both temperature and load forecasting later in this report (Section V). The nature of the load data allows us to proceed with the extraction of another feature, the day of the week as introduced by [8]. Energy consumption does not only depend on the climatic circumstances, but also directly on the calendar events, because consumption patterns vary between the different days of the week, weekends and holidays due to human activity (Section 2.6 [4]). As proposed in [9] section IV.A, we use different approaches of marking the days of the week, numbering the days of the week from 1 (Sunday) to 7 (Saturday) and assigning weekdays and weekend days with 1 and 2 respectively.

The only external data source allowed for the GEFCom load

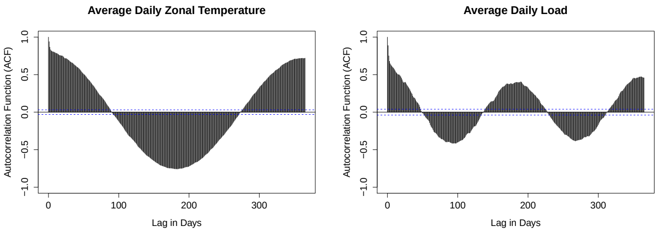


Fig. 8: Autocorrelation function estimates for daily averages of both hourly average temperature for the district and the load for a maximum lag of 1 year.

forecasting track are U.S. federal holidays for the time span of the provided data. This gives us the possibility to additionally detect holidays and assign them another integer (8) in the approach of consecutively numbering the days of the week.

Type	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Holiday
SDAYT	2	2	2	2	2	1	1	-
WDAYT	2	3	4	5	6	7	1	-
DAYT	2	3	4	5	6	7	1	8

TABLE I: Different assignments of integers for days of the week. SDAYT stands for simple day type (weekdays vs. weekends), WDAYT for week day type numbering all days of the week and DAYT includes holidays.

IV. Forecasting Methods & Methodology

In this section we briefly review the forecasting models used during this project. On the one hand we use a generalization of the Linear Model (LM), the generalized additive model (GAM), and on the other basic implementations of established machine learning methods such as neural networks (NN) and random forests (RF). For reasons of priority, we did not use our own implementations, but rather those that have been made available through R packages. With every method we create several models using different subsets of features for evaluation of both temperature and load forecasting (Section V).

A. Linear Model (LM)

We use the ordinary least squares (OLS) linear regression model as a benchmark method to interprete the prediction power of the other methods. OLS assumes the residuals to follow a normal distribution $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$. The linear predictor is therefore the sum of a linear combination of covariates and Gaussian noise and the predictor $y = \phi^T \mathbf{x} + \epsilon$. The vector of covariates $\mathbf{x}$ contains the pre-selected features, i.e. the temperature variable and the calendar effects, as well as the bias term x_0 and ϕ represents the vector of coefficients that are to be learnt with least squares approximation.

B. Generalized Additive Model (GAM)

The standard linear regression model we use is a generalized linear model with a gaussian distribution and an identity link function. For the general additive model we employ [10] we will not change the link function, since we are going to use a Gaussian distribution here as well. However, generalized additive models allow a generalization on top of that of the incorporation of other distributions as in Generalized Linear Models, that of *nonlinear predictors*. Hence, assuming a Gaussian error model the linear predictor turns into a nonlinear predictor with a combination of smooth functions of the predictor variables: $y = s_1(x_1) + s_2(x_2) + \cdots + s_p(x_p) + \epsilon$, where the $s(x_i)$ are smooth functions of the chosen features. Note that not all covariates need to be wrapped in smooth functions.

Since the nonlinear functions are smooth they can be estimated by penalized regression in a spline basis [**?**], [11]:

$$s_i(x) = \sum_{j=1}^{k_i} \beta_{i,j} \varphi_j^i(x)$$

Here k_i is the dimension of the spline basis and φ_j^q are the corresponding spline functions. The default splines for GAMs in the **mgcv** package are thin-plate splines [12] where $\varphi(x)$ is of the family of radial basis functions. The focus of this project was on forecasting workflow and feature selection, so that different spline specifications have not been tried out.

C. Neural Network (NN)

Neural networks are powerful machine learning algorithms. They exist in many varieties, with feedforward neural networks being the most basic. In our predictions we use a simple feedforward neural network with one hidden layer with a varying number of hidden units (from now on referred to as **hunits**) and a linear output unit. Such a network can be easily used for regression using the R package *nnet* [13].

D. Random Forest (RF)

Random forests [14] are a powerful ensemble method for capturing nonlinear dependencies that does not require significant tuning. They employ decision trees which are very popular in machine learning. To avoid the effects of overfitting a bunch of random decision trees are automatically. “Each simple tree gives its prediction, and these are then aggregated using the mean of the individual predictions. Each tree model assigns a class to the new vector of covariates, based on the recursive partitioning. Then, the prediction is the mean value of the responses that correspond to this class.” [**?**]. A decision tree can process continuous and discrete covariates, as is the case with our features, while providing a continuous response. Chapter 15 of [15] provides the reader with a detailed introduction to the subject. For simplicity, we use the *randomForest* R package [16] and try the algorithm for a varying number of trees. We will refer to this variable in the following as **ntrees**.

E. Time series cross-validation

After creating the features for the whole dataset, we run through the whole dataset, temperature and load respectively, in a rolling fashion to create our predictions. This approach can be described as “forecast evaluation with a rolling origin”. However, since, according to Hyndman [17], it is the natural and obvious analogue to leave-one-out cross-validation for cross-sectional data origins, we can call it "time series cross-validation".

Forecasts are created with two time horizons: 1 month and 1 week. The idea behind the weekly forecasts is to allow for shorter, but increasing time lags in week 1, 2, 3 and 4 of the month constituting the time horizon. The forecasts of week 1 to 4 are then combined with the rest of the forecast from the monthly forecasts.

F. *Creation of Quantiles*

We use the residuals $\xi = y - \hat{y}_i$ created during training to create the desired percentile forecast by offsetting the mean of the estimated gaussian distribution of residuals with the point forecasts produced by the mentioned methods and models.

V. Results

In this section we discuss the score of several configurations of the methods mentioned above with respect to the leaderboard of GEFCom 2014. We do this for different feature combinations. First, temperature forecasts are addressed and then load forecasting with and without temperature.

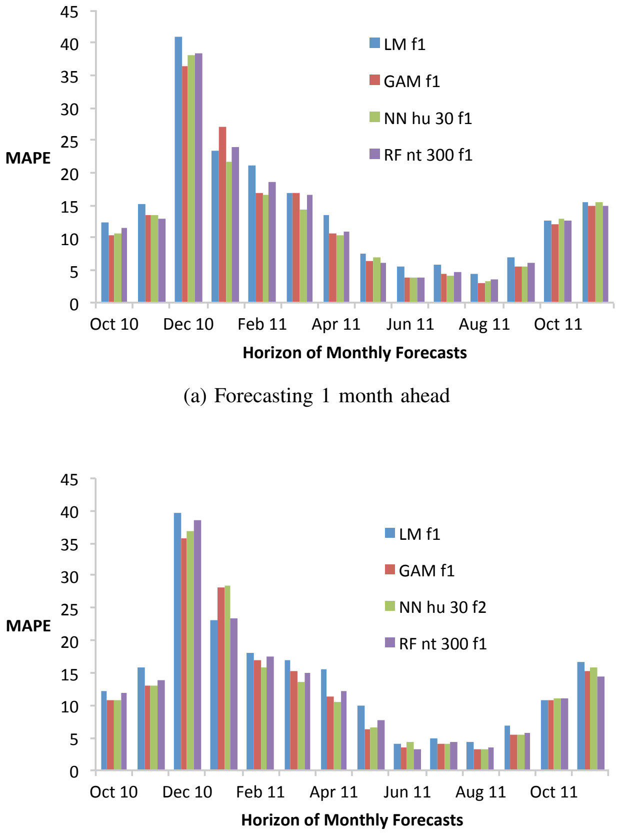
A. *Temperature*

In this section we display the results for temperature forecasting. Figure 9 is comparison of the best performing method configurations for monthly and weekly forecasting horizons. Over validation period the feedforward neural network with 30 hidden units (hu) has the best performance on average with two different models: with and without the use of a recent lag variable from before the forecasting horizon. The monthly and weekly MAPE score are 12.04% and 12.17% respectively. Significant difference A simpler way of trying to predict the temperature is to use the mean of the sequence of $k \cdot 365$ days, $k = 1, 2, \dots$ time lags applied to the temperature. However, the resulting score of 22.04% over the entire forecasting horizon is simply too high and not competitive with the method configurations shown in Figure 9.

B. *Load Modeling*

The focus of this project is on load forecasting, therefore more effort is made to explain the results for load forecasting. We are more interested in the temperature modeling affects the forecasts than how the temperature modeling itself can be significantly improved. In this section we show results for configurations of GAM, random forest and LM. LM serves the purpose of showing obvious effects of the temperature on the load forecasts, since its forecasts are very fast to simulate. Feedforward neural network load models are not discussed, because they did not perform well on the load compared to the other methods. Moreover, the forecasting tasks took a long time to run.

1) Effect of Temperature: To evaluate the effect of the temperature forecasts on the load prediction, we run our models with and without the temperature as a feature. Figure 10 shows the comparison of the MAPE of two different GAM models. Model "GAM f2" which does not include the average district temperature as a feature performs significantly worse. Over the whole cross-validation period it scores an average MAPE of 19.18 (1 month forecasting horizon), 18.25 (1 week forecasting horizon), compared to 16.33 and 15.83 respectively for the other model already shown in "GAM f1". Additionally to the features used for temperature modeling, these models now include the daytype as a categorical variable and the current, predicted temperature.



(a) Forecasting 1 month ahead

(b) Forecasting 1 week ahead 4 times

Fig. 9: MAPE scores for the best performing temperature method configurations. "f1" stands for "formula 1" which contains the most recent possible time lag variable, a time lag variable of 365 days, the time of year and the hour of the day. The feedforward neural network formula in subfigure 9b excludes the recent time lag variable.

2) Effect of Different Temperature Inputs: Now we have established that, despite their inaccuracy, the temperature forecasts do on average have a positive effect on our load forecasts. In the next step we explore if further the choice of temperature preprocessing affects the predictions. In order to achieve this, we run the same linear model for each of the different ways of selecting the temperature source discussed in section **??**. The temperature forecasts are performed with the GAM model from figure 9. The results strongly suggest that the choice of the temperature preprocessing, be it by selecting the first principal component of the temperature data, the average or just one station are negligible. We therefore proceed by to use the average of the temperatures at the different weather

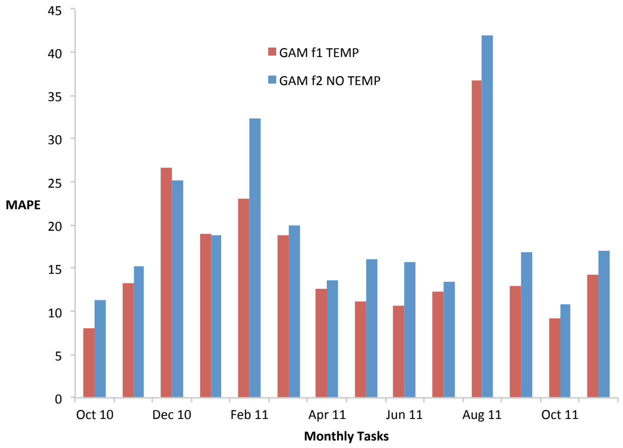


Fig. 10: MAPE score for models with and without temperature variable. The GAM model without the temperature feature, but otherwise identical, has a significantly higher MAPE. This shows that although the unsophisticated temperature forecast has a MAPE higher than 10% it can still have a positive effect on the load forecast.

stations.

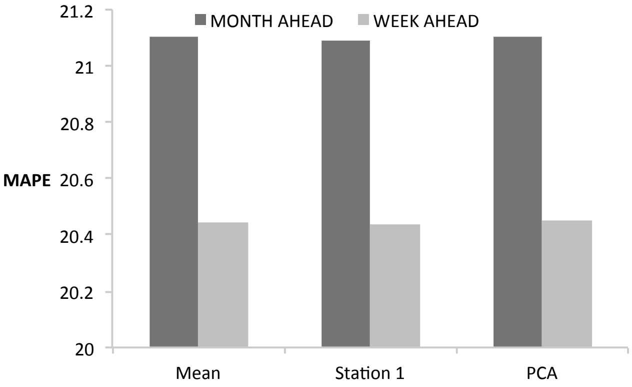


Fig. 11: MAPE and PINBALL for LM using different temperature inputs as feature. The other features are the type of day, the hour, time of year and the two lag variables.

3) Predicting Temperature for Load Training Period: Concerned that training the load models on the true temperature and then performing forecasts using the rather inaccurate temperature forecasts has adverse effects, we predict the temperature also for the training period of the load starting on 10/01/2006 and ending on 10/01/2010. On average the MAPE score increases with predicting the temperature for the load training period (**predtrain**) (Figure 12), however, the pinball error decreases. This is not only true for the GAM model depicted, but also for different configurations of the other forecasting methods mentioned.

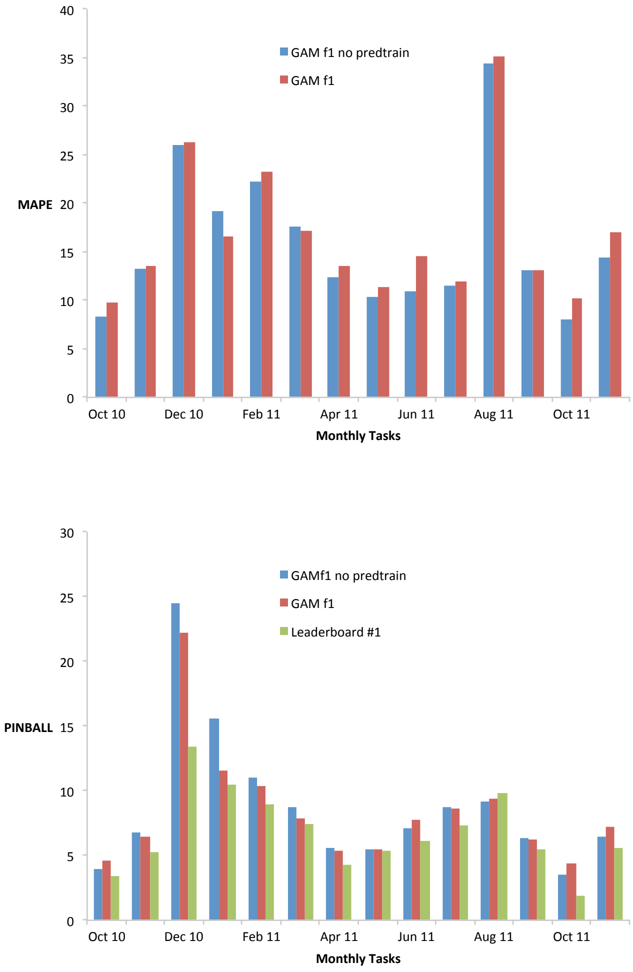
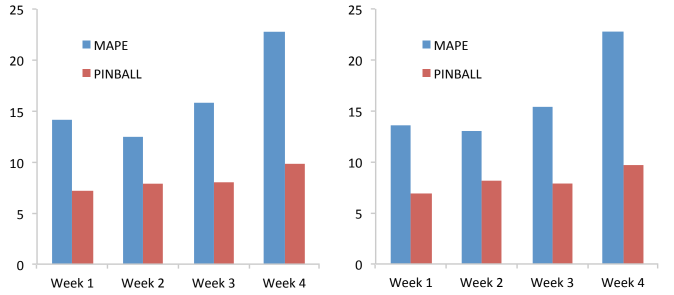


Fig. 12: MAPE and PINBALL scores for different temp models.

4) Effect of the Time Horizon: In section III-A we demonstrated the periodicity the serial correlation of the load data possesses. Using this characteristic we would like to improve the forecasts by predicting the first or only 4 weeks of the time horizon of a month, allowing us to use more recent lagged load values. Figure 13 shows how the MAPE and pinball evolve by week of a month that is to be predicted on average. We show two plots, one for a random forest model and one for a GAM model. These plots show that the errors are lower for the first weeks compared to the 4th week. For the effect the change of the time horizon has compared to the month ahead prediction please consult Figure 16. The simulation results suggest overall that the inclusion of a recent lag variable on top of a lagged value of 365 days has a positive effect on the model performace. Figure 14 shows both MAPE and pinball in separate plots for two GAM models with week ahead forecasting horizon including and excluding recent lagged values. However, it has to be noted that the overall improvement for both GAM and random forest models is around 0.1 for MAPE and even less for the pinball score.



(a) RF tree 100 model with yearly and recent lagged values.

(b) GAM formula 5 with yearly and recent lagged values.

Fig. 13: Scores by week 1-4 averaged over the monthly tasks making up the forecasting horizon.

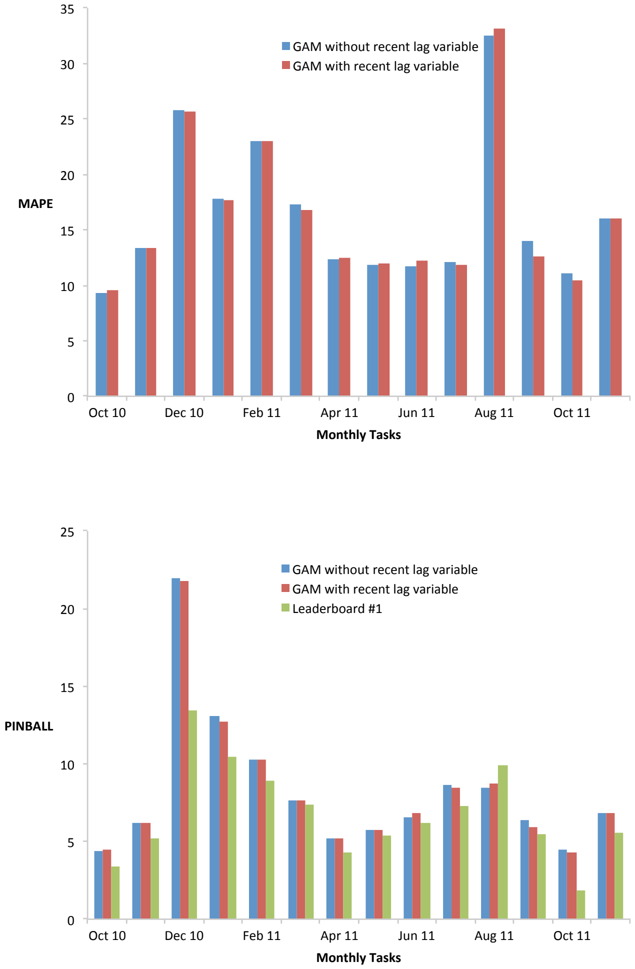


Fig. 14: MAPE and Pinball scores on monthly tasks for GAM models with and without recent lagged values. The temperature model used is the GAM f1 model shown in 9

5) Effect of different Daytype Assignments: After having dealt with the lag variables, we briefly address the different assignments of day types. Figure 15 shows that the different assignments have practically no effect on the MAPE and pinball when implemented in different GAM models. This makes sense because when probing the data no weekday versus weekend patterns or patterns for federal holidays seem to catch the eye. The load data provided is that of one utility and there is no information as to the proportion of domestic to industrial consumption. In other words: we have no clue as to where exactly the energy is from and what it is used for, rendering analysis in this respect difficult.

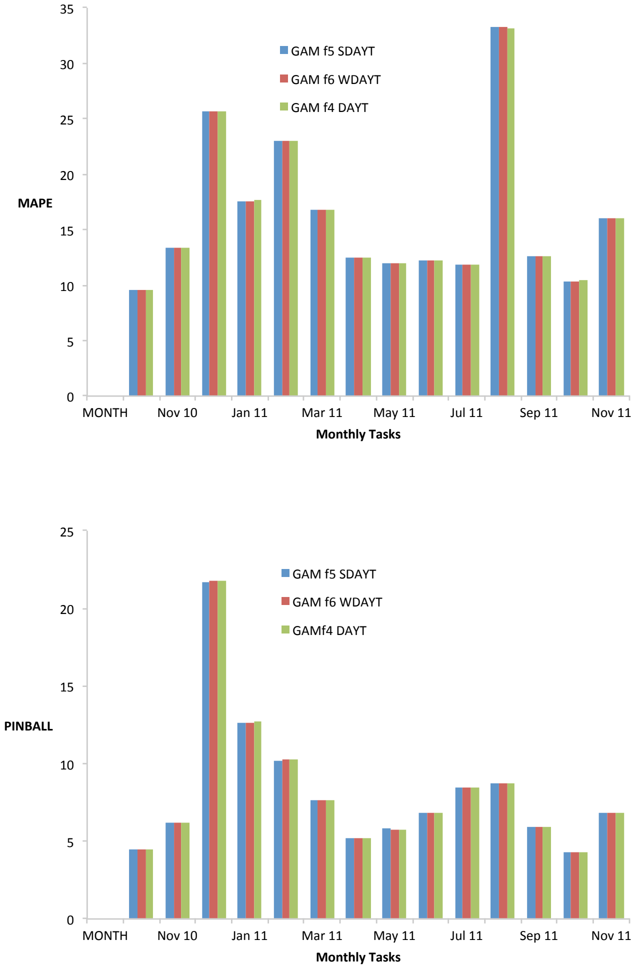


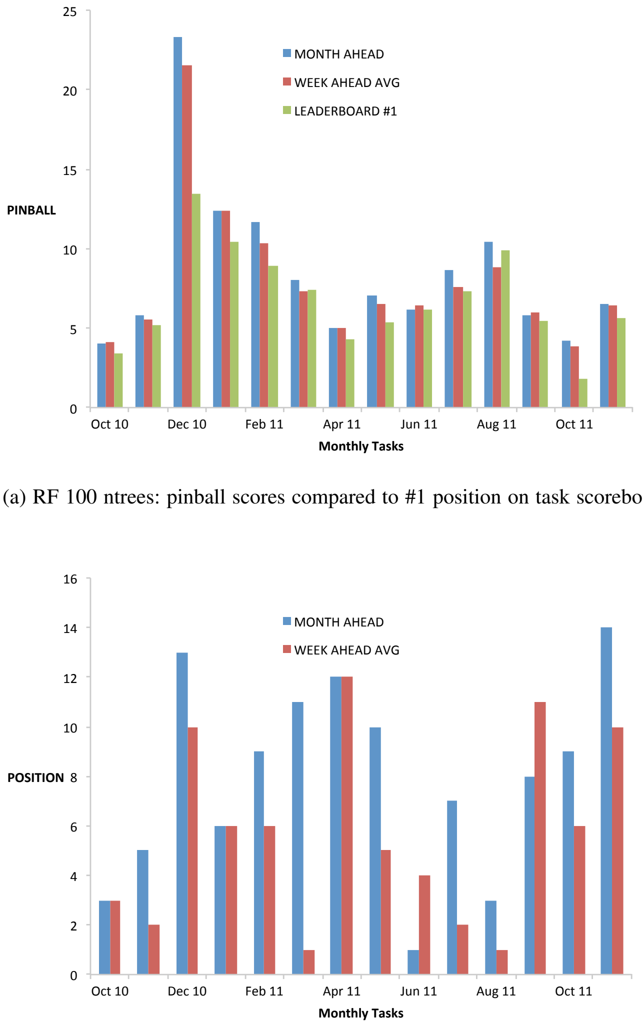
Fig. 15: MAPE and PINBALL scores for different GAM models with monthly forecasting.

6) Best Performancing Method Configuration: The best results obtained after simulating a large number of temperature and load forecast combinations and method configurations are for a random forest model with 100 trees (no daytype, week ahead forecasting, prediction of temperature for load training) using a neural network with 15 hidden units for the temperature forecast (1 month forecast horizon). The average for the entire evaluation period MAPE and pinball scores are 16.094 and 7.999 respectively. Using the pinball errors displayed in figure 16a we can now compute a hypothetical position on the final leaderboard of the GEFCom2014 competition using the mentioned model configuration. The rules state: "The final score is the trimmed mean of the weekly scores. The highest and lowest weekly scores are discarded. The participants are required to submit entries for at least 9 out of 12 weeks during the scoring period to be eligible for a position in the final leaderboard." Excluding the first three months (Oct, Nov, Dec) that were the trial period and the last month for which the real data has not been published, we can compute the trimmed mean excluding the best and worst pinball scores of 11 tasks. The hypothetical score thus is **7.16**. This score would have resulted in a third place on the final leaderboard, beaten by 6.93 (#1) and 7.10 (#2). The positions on the scoreboards for the individual tasks are displayed in figure 16b comparing the different prediction horizons. The average position for the task leaderboards would be 5.64. However, not every contestant reached a position equally high up the scoreboard every week allowing for our improved hypothetical position on the final leaderboard.

VI. Conclusion

In this report we show that competitive scores with respect to the GEFCom 2014 leaderboard can be achieved using relatively simple models. Computing competition scores in hindsight is not the "real deal", however, since the student started with little knowledge of forecasting methodologies it is motivating to see what is achievable. The results are striking, because more time was spent on defining different combinations of smoothing functions for the GAM models than on any on either the neural network or the random forest models. However, these methods performed best on temperature and load forecasting respectively. All that was needed was to set the **ntrees** parameter or hidden units parameter and plugin a group of features. Grid search on the **ntrees** parameter may yield an optimal chosing in between the values 50, 100, 150 that were specified for load forecasting. Preferable about the generalized additive model is that it allows for relationships to be defined by the user, which are then either validated or not through training a model. This gives it more explanarotary power than running a more "blackbox" method like a neural network or random forest.
The results are by no means perfect, much can surely still be improved with deeper data exploration and more time dedicated to the study of papers and theory of the forecasting methods that were used. A next stop could be to combine the GAM and random forest models to produce ensemble forecasts and evaluate these. Further, residual analysis may uncover patterns in the data. The student needed quite some time to get acquainted with the forecasting workflow and the problem of having two linked forecasting tasks, so having a working pipeline was the priority.

In summary, the project has been very enriching since the student knew hardly anything about the forecasting workflow before the commencement of this project and feels he has been given a good introduction to the subject. The student would like to sincerely thank Tri Wijaya and Matteo Vasirani for sharing their knowledge and valuable time.



(a) RF 100 ntrees: pinball scores compared to #1 position on task scoreboard.

(b) RF 100 ntrees: individual task scoreboard positions

Fig. 16: Pinball and hypothetical positions for the individual tasks (including trials Oct - Dec 2010) for a random forest with 100 trees. The model does not use a day type, yet does include a variable for recent lags. The method use for the temperature prediction is a neural network with month ahead forecasting.

References

[1] T. Hong, "Load, Demand, Energy and Power." [Online]. Available: http://blog.drhongtao.com/2014/09/load-demand-energy-power.html

[2] S. Fan, S. Member, and R. J. Hyndman, "Short-Term Load Forecasting Based on a Semi-Parametric Additive Model," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 134–141, 2012.

[3] T. A. O. Hong, "Energy Forecasting : Past , Present , and Future," *International Journal of Applied Forecasting*, pp. 43–49.

[4] T. Hong, "Short Term Electric Load Forecasting," Ph.D. dissertation, North Carolina State University.

[5] T. Hong, P. Pinson, and S. Fan, "Global Energy Forecasting Competition 2012," *International Journal of Forecasting*, vol. 30,

no. 2, pp. 357–363, Apr. 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0169207013000745

[6] I. Takeuchi, Q. V. Le, T. D. Sears, and A. J. Smola, “Nonparametric Quantile Estimation,” *Journal of Machine Learning Research Nonparamteric Quantile Estimation*, 2005.

[7] Y. Goude, R. Nedellec, and N. Kong, “Local Short and Middle Term Electricity Load Forecasting With Semi-Parametric Additive Models,” *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 440–446, Jan. 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6684602

[8] T. Hong, M. Gui, M. E. Baran, S. Member, and H. L. Willis, “Modeling and Forecasting Hourly Electric Load by Multiple Linear Regression with Interactions,” p. 4, 2010.

[9] R. J. Hyndman and S. Fan, “Density Forecasting for Long-Term Peak Electricity Demand,” *IEEE Transactions on Power SystemsHyndman, R. J., & Fan, S. (2010). Density Forecasting for Long-Term Peak Electricity Demand. IEEE Transactions on Power Systems, 25(2), 1142âĂŞ1153. doi:10.1109/TPWRS.2009.2036017*, vol. 25, no. 2, pp. 1142–1153, May 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5345698

[10] S. N. Wood, *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC, 2006.

[11] M. Clark, “ADDITIVE MODELS.” [Online]. Available: http://www3.nd.edu/~mclark19/learn/GAMS.pdf

[12] S. N. Wood, “Thin-plate regression splines,” *Journal of the Royal Statistical Society (B)*, vol. 65, no. 1, pp. 95–114, 2003.

[13] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*, 4th ed. New York: Springer, 2002. [Online]. Available: http://www.stats.ox.ac.uk/pub/MASS4

[14] L. Breiman, *Random Forests*, 2001.

[15] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[16] A. Liaw and M. Wiener, “Classification and Regression by randomForest.” *R News: The Newsletter of the R Project*, vol. 2, no. 3, pp. 18–22, 2002. [Online]. Available: http://cran.r-project.org/doc/Rnews/

[17] R. J. Hyndman, “Why every statistician should know about cross-validation.” [Online]. Available: http://robjhyndman.com/hyndsight/crossvalidation/