Universität Duisburg-Essen

Forecasting Study of the Energy Demand in Netherlands, Belgium and Luxembourg

Alex Amaguaya TU Dortmund
Anarghya Murthy TU Dortmund
Sebastian Reyes TU Dortmund

Advanced Forecasting in Energy Markets
Prof. Dr. Florian Ziel

24 February 2023

INTRODUCTION AND LITERATURE REVIEW

In this paper is presented the methodology of the construction of a 240 hour Energy Load forecasting problem, a special case in Short Term Load Forecasting problems. The main aim is to provide an overview of selected Short Term Load Forecasting methods used to solve the task at hand.

The task at hand is that a hypothetical forecast provider company wishes to switch to a 240 hour-ahead energy demand forecast for the countries of Belgium, Netherlands, and Luxembourg. To this goal, the company bought access to and used the information provided by two databases: the DWD-MOSMIX, historic weather data, and ENTSO-E transparency data on the electricity market. The ENTSO-E data contains the target variable, Actual Energy Load. The use of MOSMIX weather data is justified by its importance in older forecasting studies, such as Fidalgo et. al. (2007).

For this purpose, a prediction study that takes into account historical data for the year 2022 during 8am each day was designed. The models used to predict the energy demand would be compared with each other in order to choose the one that is less biased and more efficient. In addition to the comparison between different models, the results between different estimators to arrive at the best answer from the total set of predictions were considered.

DATASET

This section describes the datasets of the forecasting study, which were constructed from the data of three countries: Belgium, Netherlands, and Luxembourg. The forecasting study was constructed with information from two databases: MOSMIX and ENTSO-E. From these, information for the countries of Netherlands, Luxembourg, and Belgium were extracted to create data sets directly relevant for the forecast, i.e. EDAT and MET data sets.

The ENTSO-E database contains data on the historic actual load, the forecast objective, as well as the official ENTSO-E 240 hours-ahead load forecasts. The official ENTSO-E forecasts are used as a benchmark for our own model. The initial date for the data sets used from all countries was January 12, 2021, and the end date was November 28, 2022.

DWD's MOSMIX database contains data on weather related variables, including (but not limited to) Temperature, Wind Speed, Global Irradiance, Wind Direction, Effective Cloud Cover, etc. Temperature related variables are selected because the demand for heating increases with colder temperatures. Wind related variables have been selected because wind energy is the largest renewable source of energy in the Netherlands and Belgium. Global Irradiance is the total solar energy reaching the ground, which is also suspected to have a direct impact on energy demand.

In addition, some missing values were detected in the datasets and they were replaced using a spline interpolation.

The time series decomposition for all the three countries is similar. As an example, Figure 1 depicts the time series decomposition of Netherlands energy demand. This decomposition is composed of three parts, trend, seasonal, and random. The trend section shows that it is necessary to include a variable that capture the trend of the time series. The seasonal section shows that the time series has many multiple seasonalities, daily, weekly, monthly, etc. The multiple seasonalities are then validated with the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots.

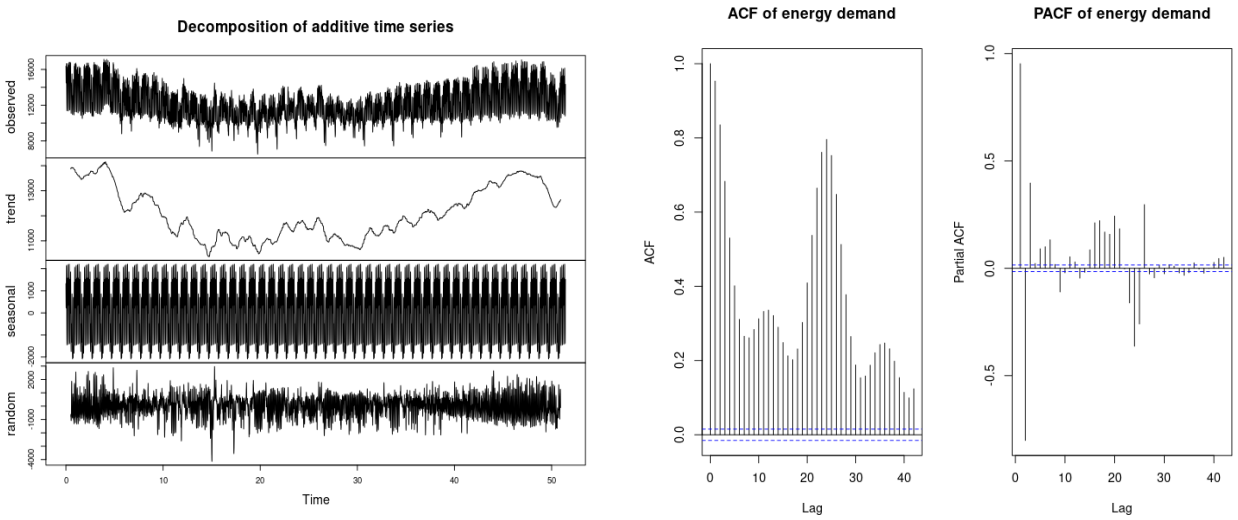

Figure 1. Time series decomposition of Netherlands energy demand during the first 51 weeks, and ACF & PACF.

Feature Engineering

This section covers the transformation and creation of the features that were used in the models. Many of these features were estimated using transformations, binary variables, or interactions between a set of variables. Variables were computed using the date information, for example, Hour of Day (HoD), Day of Week (DoW), Month of Year (MoY), Week of Year (WoY), Quarter of Year (QoY). Binary variables were created for each value in this group of variables. Likewise, some additional dummy variables were created as the first 6 hours of day (is_day_start), the last 6 hours of day (is_day_end), half a day before and one and a half days after the holiday dates (holidays_dummy, a binary variable for all holidays), saturday and sunday observations (weekend).

In addition, in order to capture the multiple seasonality of the time series, some Fourier terms were estimated using the cosine and sine function and were applied to the variables HoD, DoW, MoY, DoY, WoY, and QoY. The following formulas were used:

$$cosine.feature_i = cos(\frac{2*\pi*feature_i}{Total\ number\ of\ periods\ of\ the\ feature_i})$$

$$sine.feature_i = sin(\frac{2*\pi*feature_i}{Total\ number\ of\ periods\ of\ the\ feature_i})$$

On the other hand, some variables were created using the lags of the target variable. Also, a trend and some weather features were included for the experiments, temperature 2m above surface (TTT), and wind speed (FF). A spline interpolation was applied to the missing values of the weather features. Finally, a novel way of modeling the effect of holidays was estimated using some spline designs. After the estimation, 9 variables were created for each holiday.
Then, the interaction features by group were estimated, the first group consisted of HoD, DoW, MoY, QoY, weekend, and the second group was composed of WoY and holidays_dummy. Figure 2 shows the structure of the data gathering and a description of the groups of variables that were estimated and used in the models.

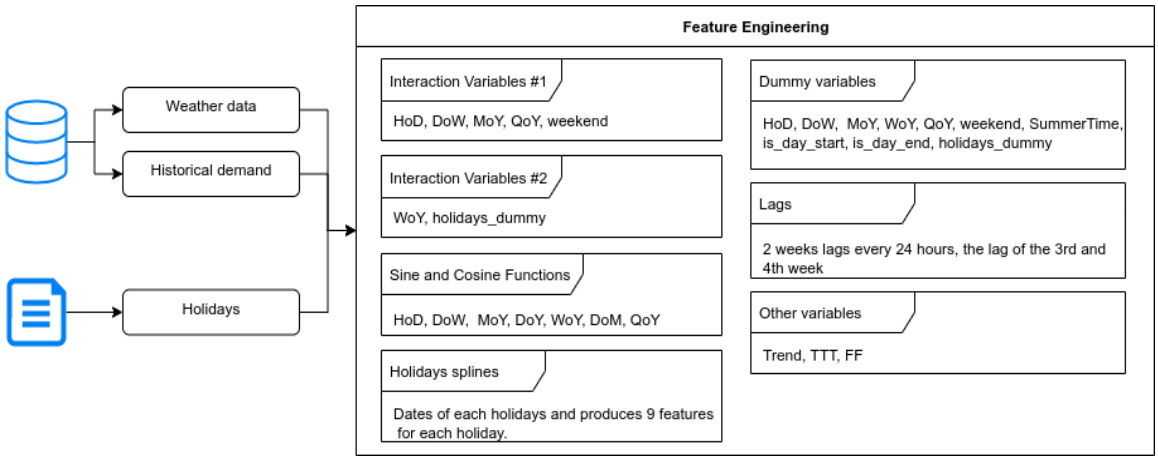

Figure 2. Data set construction and creation of group of features

METHODOLOGY

In the following section, the various Short Term Load Forecasting models relevant to the given task are explored, namely: Random Forests, Gradient Boosting, and Elastic Net approach. All models are benchmarked against an autoregressive model, as provided in the sample script. The model models were selected to provide an alternative to the linear estimation from the AR model. Consequently the calculation of the ensemble estimators that combine both linear and nonlinear models are made, in order to get better results.

Elastic Net

One way to improve the prediction results of the AR model is through regularization, which seeks to decrease the variance of our estimation through a slight increase of the bias. Elastic Net allows to penalize at the same time the sum of squares of the coefficients and the absolute values of the coefficients. In other words, it is a combination of Ridge regression and Lasso regression, two techniques that seek to optimize the trade-off between variance and bias in order to minimize the error. In this way the model can better fit the phenomenon since it takes the best of the two techniques: Lasso tends to be good if there are a small number of parameters influencing the model, while Ridge performs well if most of the predictors affect the outcome. However it is hard to know in practice the situation of these parameters so through the combination of both methods we have a middle ground. The Elastic Net estimator works by combining both the ridge and lasso regressors. In the following equation, the coefficient alpha expresses the weight of each technique, taking a value equal to 1 for the lasso regression and 0 for the ridge.

$$L_{enet}(\widehat{\beta}) = \frac{\sum_{i=1}^{n}\left(y_i - x_i'\widehat{\beta}\right)^2}{2n} + \lambda\left(\overbrace{\frac{1-\alpha}{2}\sum_{j=1}^{m}\widehat{\beta}_j^{\,2}}^{\text{ridge}} + \overbrace{\alpha\sum_{j=1}^{m}|\widehat{\beta}_j|}^{\text{lasso}}\right)$$

Gradient Boosting

The second model used in the study is called Gradient Boosting, which is a family of algorithms used in both classification and regression based. It is based on the combination of weak learners -usually decision trees- to create a strong predictive model. The generation of weak decision trees is done sequentially, with each tree being created in a way that corrects the errors of the previous tree. The apprentices are usually "shallow trees", with few levels of depth (number of nodes). For their correct use, it is important to take into account the learning rate, which controls the degree of improvement of a tree with respect to the previous one. A small learning rate means a slower improvement but better adaptation to the data, which generally translates into improvements in the result at the cost of higher resource consumption. The advantages of gradient boosting are that it can be applied to regression and classification problems, it is a non-parametric method (so it is not necessary to comply with any specific distribution), its predictions are not greatly influenced by outliers and it has good scalability, so it can be applied to data sets with a large number of observations. That makes it a handy tool that can be applied for this specifical problem. The following equation represents the function $\widehat{f}(x)$ that approximates the output variable from the values of input variables by minimizing a loss function.

$$\widehat{f}(x) = \arg\min_{f(x)} \underbrace{E_x[\overbrace{E_y(\Psi[y, f(x)])}^{\text{expected y loss}}|x]}_{\text{expectation over the whole dataset}}$$

Random Forest

Random forest is an ensemble method that combines individual estimations of decision trees to obtain a final global result. The number of trees used is defined as a parameter that is adjusted by cross-validation, i.e. training and testing the model on fragments of our time series. The final result is defined through a bagging method that divides our time series into several randomly composed subsets of samples, hence the "random" of random forest. We then train the model on each subset and then combine the results with an aggregation rule. All this results in a robust model that is built from models that individually were not so robust, at the beginning. The advantages of the random forest is mainly the fact that it does not demand too many assumptions in the preparation of the data so we can use it with the time series of the databases we are working with. In addition, it allows us to handle many input variables and to identify the most significant ones. The following equation represents the mean square error that is minimized in each tree. Then the random forest is represented as the average over all of the decisions trees represented by the calculation of the feature of each tree f

$$MSE = \frac{1}{n}\sum_{i=1}^{N}(y_i - \mu)^2 \qquad RFf_i = \frac{\sum_{j \in all\ trees} normf_{i_{ij}}}{T}$$

Performance Metric

To compare the model results the Root Mean Squared Error (RMSE) was used. The errors in question are the calculated difference between the predicted energy demand (as forecasted by the model) and the actual demand (as visible in the ENTSO-E dataset). The advantage of RMSE (compared to e.g. MAE) is that the errors are squared before they are averaged, larger errors are given a bigger weight, which helps avoid distortions caused by large errors.

Experiments Pipeline

A pipeline was developed for the selection of the best model, and included three stages, the training, validation and test stage. To carry out this process, a matrix was created and separated into batches in order to save the predictions of the days ahead (~20 days per batch). The data available up to the first period of each batch was used for the training of each model, and the models were retrained according to each time horizon. This process was replicated for *N* batches and the figure 3 shows an example of this.

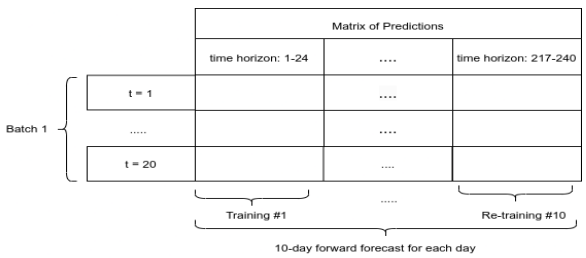

Figure 3. Data batch splitting

After the data batch splitting, the data available up to the first observation of each batch was separated into training and validation data sets. The validation data was composed with the last 20 observations of the available data, and the rest belonged to the training data. The last observations were selected as the validation component because sequential data were being handled. The random selection of the validation sample was not appropriate for time series. Then, the model was trained with some combinations of hyperparameters and the best one was selected based on the performance metric in the validation data (Figure 4). This pipeline was applied for the Elastic Net, Gradient Boosting and Random Forest algorithms.

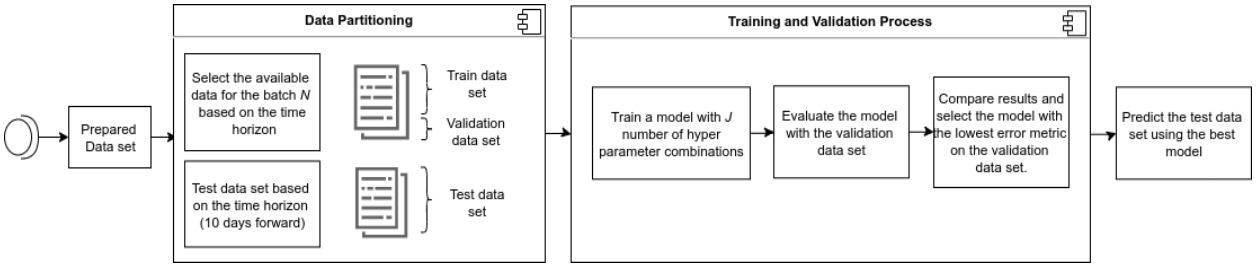

Figure 4. Experiment Pipeline

On the other hand, the AR model was only trained once for each batch and the best one was selected using the lowest AIC metric. Then, the model was used to predict 10 days forward for each day.

RESULTS

This section compared the performance between benchmark models, AR model and bench, with ElasticNet, Gradient Boosting, Random Forest algorithm and some ensemble methods. The ensemble models were estimated as the mean of the prediction between two or three of the previous models. Random Forest and Gradient Boosting were estimated using the Lightgbm package, while Elastic Net used the SGD package. Due to their fast estimation with multiple variables, these packages were used for the experiments. Table 1 shows the best models by country according to RMSE. In addition, it's worth mentioning that for each country a different number of variables were used due to the number of holidays that each country has. For Netherlands, Belgium and Luxembourg, 955, 957 and 966 variables were used, respectively.

Netherlands		Belgium		Luxembourg	
Model	**RMSE**	**Model**	**RMSE**	**Model**	**RMSE**
Ensemble between ElasticNet, Random Forest and AR	924,20	Bench	322,22	Ensemble between Random Forest and AR	66,68
Ensemble between Random Forest and AR	931,31	Ensemble between Random Forest and AR	525,40	Random Forest	68,44
Ensemble between ElasticNet, and Random Forest	943,27	Random Forest	528,80	Ensemble between Random Forest, and Gradient Boosting	68,96

Ensemble between ElasticNet, Gradient Boosting, Random Forest and AR	962,94	Ensemble between ElasticNet, Random Forest and AR	541,48	Ensemble between Gradient Boosting and AR	69,93
Ensemble between ElasticNet and AR	977,56	Ensemble between ElasticNet, and Random Forest	561,66	AR	71,81
Random Forest	982,02	Ensemble between ElasticNet, Gradient Boosting, Random Forest and AR	581,44	Gradient Boosting	74,53
Ensemble between ElasticNet, Gradient Boosting and AR	1.002,69	Ensemble between ElasticNet and AR	605,49	Ensemble between ElasticNet, Gradient Boosting, Random Forest and AR	140,88
AR	1.033,28	AR	625,00	Ensemble between ElasticNet, Random Forest and AR	181,40
ElasticNet	1.063,77	Ensemble between Random Forest, and Gradient Boosting	629,35	Ensemble between ElasticNet, Gradient Boosting and AR	182,01
Ensemble between ElasticNet, and Gradient Boosting	1.063,80	Ensemble between ElasticNet, Gradient Boosting and AR	636,62	Ensemble between ElasticNet, and Random Forest	266,89
Ensemble between Random Forest, and Gradient Boosting	1.068,89	Ensemble between Gradient Boosting and AR	669,91	Ensemble between ElasticNet and AR	267,17
Ensemble between Gradient Boosting and AR	1.070,22	Ensemble between ElasticNet, and Gradient Boosting	709,83	Ensemble between ElasticNet, and Gradient Boosting	267,41
Bench	1.235,93	ElasticNet	723,44	ElasticNet	531,27
Gradient Boosting	1.297,25	Gradient Boosting	845,85	Bench	5.351,84

Table 1. Average RMSE for all the countries

In addition, Table 2 shows the main hyperparameters used during the training of each of the algorithms and the duration time of the training and predicting process for each country.

Algorithm	Gradient Boosting	Random Forest	Elastic Net	AR
Hyper parameters	max_depth: 8-9, num_leaves: 23-24, min_sum_hessian_in_leaf: 34-35, num_iterations: 29-31, lambda_l1: 0.12, lambda_l2: 0.08, learning_rate: 0.38-0.39	max_depth: 5-12, num_leaves: 20-27, num_iterations: 40-50, lambda_l1 and lambda_l2: 0.01-0.1, learning_rate: 0.18-0.3	lambda1: 20-23, lambda2: 0.98-1	order.max: 672
Duration time	NE: 7,11 min., BE: 7,20 min., LU: 7,54 min.	NE: 10,03 min., BE: 10,4 min., LU: 10,51 min.	NE: 12,43 min., BE: 12,75 min., LU: 12,99 min.	NE: 13 sec., BE: 14,09 sec., LU: 29,44 sec.

Table 2. Hyper parameters used for the training stage and duration time

The previous results show that the ensemble and Random Forest methods obtained the lowest error in each country. On the other hand, the AR model was the fastest model for the training and prediction process (on average 19 seconds), and in second and third place were Gradient Boosting and Random forest (on average 7,28 and 10,31 minutes, respectively).

CONCLUSIONS

The results of the experiments showed that even though the tree methods have a longer training time (relative to AR models), they were able to outperform the traditional models. Possible improvements to increase the performance of the models were detected. One of them is the detection of trend change points or structural breaks in the time series. There are some packages that help with this problem, but they take a long time as the number of observations increases. In addition, the Facebook Prophet algorithm was used during the experiments, but the training process takes a long time and was discarded. Also, the comparisons were made using the mean squared error as opposed to the mean absolute error because the former is more robust towards outliers due to the squaring of the values in the calculation.

REFERENCES

Natekin, Alexey & Knoll, Alois. (2013). Gradient Boosting Machines, A Tutorial. Frontiers in neurorobotics. 7. 21. 10.3389/fnbot.2013.00021.

Januschowski, T., Wang, Y., Torkkola, K., Erkkilä, T., Hasson, H., & Gasthaus, J. (2022). Forecasting with trees. *International Journal of Forecasting*, *38*(4), 1473-1481. https://doi.org/10.1016/j.ijforecast.2021.10.004

Chaturvedi, S., Rajasekar, E., Natarajan, S., & McCullen, N. (2022). A comparative assessment of SARIMA, LSTM RNN and Fb Prophet models to forecast total and peak monthly energy demand for India. *Energy Policy*, *168*, 113097. https://doi.org/10.1016/j.enpol.2022.113097