Udacity Machine Learning Nanodegree - Capstone Project Report

Fritz Arnold

email fritz.arnold@rwth-aachen.de

October 29, 2017

Abstract

Estimating future electricity prices is a critical economic success factor for energy market participants. In this project, a modular Day-Ahead electricity price forecasting approach is proposed. Feeding on historical price data, grid weather data and load forecasts, the model delivers precise estimations of hourly Day-Ahead electricity prices for the German bidding zone. A time series model, based on a combination of convolutional and recurrent neural networks, provides a predictive foundation. This basic estimation is enriched with predictions of wind and solar power generation in Germany, derived from a convolutional neural network feeding on multi-dimensional grid weather data. Together with a load estimation, provided by the German Transmission System Operators, all three input streams are aggregated in an Gradient Boosting Regressor to produce a final estimation for the electricity prices of the upcoming day. Evaluated on a full year performance basis, this approach yields competitive forecasting results when compared to other approaches in current literature.

Contents

1 Introduction and Problem Definition

This first section provides a brief overview of the project domain background, the problem definition and goals as well as methods for performance evaluation.

1.1 Domain Background

On the liberalized German energy market electricity can be traded on specialized energy exchanges. The primary organized electricity trading scheme is the so called *Day-Ahead market*. Trades on the Day-Ahead market contract the obligation to deliver or accept a certain quantity of energy during an one hour block of the next day. Contracts are determined in an anonymous auction: Market participants place bidding orders until 12:00 noon the day before delivery. Orders are defined by the amount of energy to be bought (sold) together with the maximum (minimum) accepted buying (selling) price and the hour of delivery. When the bidding period ends, a *Market Clearing Price* (MCP), also denoted as *uniform price*, is determined by the so called *Merit Order Model*. The merit order is based on the different marginal energy generation costs among generation technologies. Producing energy at selling prices below marginal generation costs over an extended period of time is causing economic damage. Hence, selling orders are generally priced at the marginal costs of the respective technology. The marginal generation costs of the last power plant that is needed to fulfill the current demand determines the market clearing price. This mechanism is depicted in Figure 1. The market clearing price is applied to all orders that are accepted by the merit order, independent of the offered quantity or original bidding price. Bids at lower prices are accepted, bids at higher prices rejected. Since renewable power generation technologies operate at nearly zero marginal generation costs, their production level determines the amount of other generation technologies that 'make it' into the merit order. High amounts of renewable generation generally lead to lower prices, since expensive generation is 'pushed out' of the merit order.

Due to the anonymous nature of the bidding process, Day-Ahead prices are unknown prior to market closing; orders of other market players are hidden until the auction process is finished and all accepted orders become binding. In order to optimize their bidding strategy, market participants are interested in estimating the market clearing price for all 24 hourly blocks of the next day. Precise knowledge of Day-Ahead prices is crucial for the economic success of a market actor.

For an in-depth review about common electricity market mechanisms, the motivation to forecast electricity prices and recent approaches and algorithms, the reader is referred to [19].

1.2 Available Input Data

Electricity prices are determined by (expected) supply and demand. Electricity demand generally has a strong time dependency, with peaks on evenings and weekends and lows during the night and weekly working hours. The supply side is characterized by the electricity generation mix, i.e. coal and nuclear thermal power plants, hydro power plants, and renewable energy generation (mainly wind and solar energy). In order to take all possible input factors into account, the following primary data is considered as an input for the forecasting model:

- Historical hourly Day-Ahead market clearing prices for the German bidding zone. These prices can be used to model short- and long-term temporal patterns in electricity prices. Historical prices can be obtained from

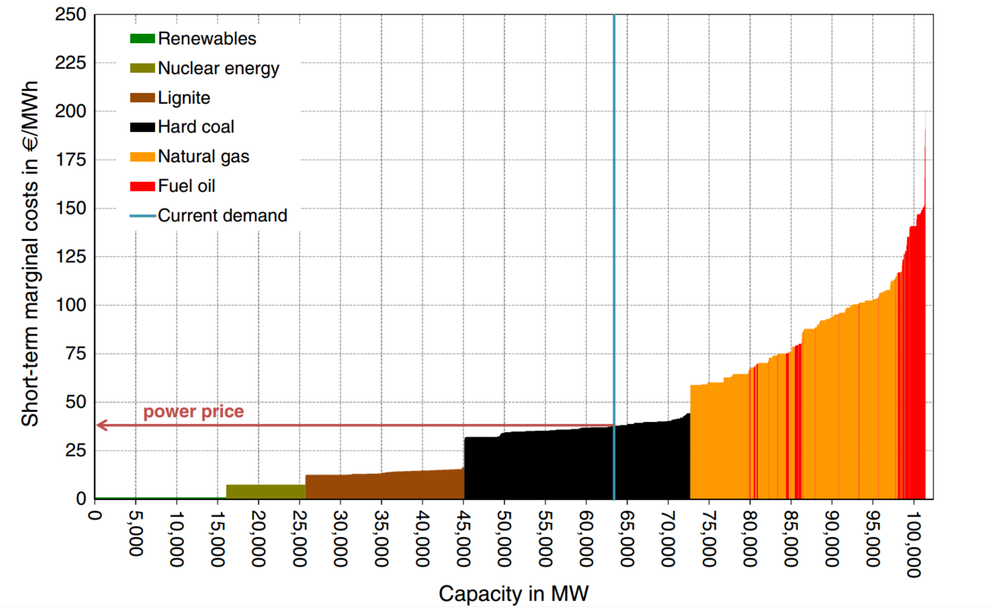

Figure 1: The Merit Order Curve. Taken from [6].

the EPEX SPOT website [9].

- Generation forecasts for the next day. Different electricity generation technologies have different generation costs. The respective share of generation technologies should have a major impact on the price level. Historical Day-Ahead generation forecasts can be obtained from the transparency platform of the European Network of Transmission System Operators (ENTSO-E) [8].

- Load forecasts for the next day. The load is equivalent to the electricity demand and therefore influences the electricity price. Historical load forecast data can be obtained from the ENTSO-E website [8].

- Historical load and generation data. Short- or long-term patterns in the energy generation mix might have an influence on the characteristics of supply and demand and thus the price. Historical load and generation data can be obtained from the ENTSO-E website [8].

Although all of the above information is potentially useful for price forecasting, not all of it is available before market closing. According to European jurisdiction [10], load forecast data has to be published two hours before the Day-Ahead market gate closure and is therefore available for the prediction. Generation forecasts have to be published until 18:00 Brussels time and might not be available for the prediction. Generation forecasts and actual generation levels are therefore only available under assumed *perfect information*. A model trained on these proxies constitutes a *benchmark* of the performance level that would be achievable with a certain model under perfect conditions.

However, some of these input features might be reproducible. Especially wind and solar power generation are strongly dependent on weather conditions. A model might incorporate weather variables (solar radiation, temperature, wind speed etc.) to predict actual renewable generation levels. Historical, geo-spatial grid weather data of air temperature, precipitation, direct and diffuse solar radiation, wind speed and wind direction in Germany is published by the German National Meteorological Service (DWD) in hourly resolution for a 5 by 5 km grid [7]. These historical *measurements* would have to be replaced by adequate *weather forecasts* in a real world deployment. They are only used here as a substitute because no large dataset of historical forecasts is available. It is discussed in the later course of that work why, under some basic assumptions, the prediction model derived from that data should be comparable to one based on actual weather forecasts.

2 Approach

In the following chapter, a problem definition and theoretical strategies for a solution of the underlying problem are given.

2.1 Problem Definition

The problem to be solved in this project is to forecast the hourly Day-Ahead market clearing price of every hour of the next day for the German bidding zone on the European Power Exchange (EPEXSPOT). The predictions for the day d have to be conducted before market closing time, in this case 12:00 noon of day $d-1$. The actual prediction has to rely solely on information that is available at that point of time to ensure applicability in a real world deployment. The learning task associated to this problem is that of a *regression*. Based on certain inputs, a continuous output, an electricity price, has to be predicted. The function mapping inputs to outputs has to be learned by one or multiple appropriate algorithms from example data. Regression belongs to the family of *supervised learning* tasks.

2.2 Solution Strategy

The final model used to solve the electricity price forecasting problem is composed of three modules. Each module tackles a specific learning task associated to the problem. All modules are part of a model pipeline that will output the actual price prediction.

Module I: The Time Series Model

Day-Ahead electricity prices show a multitude of temporal patterns. These patterns can be learned from historical data. The time series module takes historical input data of recent prices and generation levels prior to the forecasting day (the day of interest) and outputs an estimation for hourly prices based on learned temporal patterns. It should cover long term trends (monthly price levels), seasonalities (weekdays versus weekends) and recent price magnitudes (daily/weekly price development). The goal of this model is to produce a rather robust foundation for further refinement. It is expected that the predictions are able to cover typical daily price movements and the overall recent price level. A time series model will most likely fail to account for prices outside of the 'normal' price range. These exceptions, caused among other possible reasons by high levels of renewable generation, are subject to *future information* and therefore not inferable from pure historical data.

Module II: The Renewable Generation Model

The most obvious candidate among the possible model refinements mentioned above are future renewable generation levels, especially for wind and solar power generation. There are two main reason for that hypothesis: Firstly, renewable generation has a substantial influence on the merit order and therefore the Day-Ahead price, as shown above. Secondly, renewable generation is only dependent on weather conditions. While thermal power plant generation levels are adjusted based on complex economic decisions, photovoltaic generation is only dependent on the supply of sun (and other minor physical factors) while wind power is based on wind speed. Knowing these physical quantities with sufficient precisions should, in theory, enable a generation forecast, whereas human economic decision making seems to be a hopelessly complex concept to learn from sparse input data. Therefore, the second module focuses on the implementation of a model to predict wind and solar power generation levels for the forecasting day, based on weather data.

Module III: Aggregating all Inputs

Module I provides a first price estimation for every hour of the day of interest. Module II outputs expected solar and wind power generation for every hour. Together with hourly load predictions these inputs are aggregated in a regression model to produce a final hourly price estimation. It is expected that the additional information is able to improve the time series prediction. There will most likely be some prediction errors left since not all possible influencing factors have been included in the model.

2.3 Evaluation Metrics

The price forecasts are evaluated based on different performance metrics. To evaluate regression model performance oftentimes relative error measures are used. These measures are not really appropriate in this case. Generally speaking, an economic decision based on a Day-Ahead price prediction might lead to economic benefits or damage, depending on the forecast precision, or more precisely the absolute prediction error in €/MWh. The overall current level of true and predicted prices should play a minor role when evaluating the impact of the error (a loss of 1€ is a loss of 1€, independent of whether the current price level is at 10€/MWh or 50€/MWh). Hence, relative error measures are not really appropriate here. Therefore, the main evaluation metric is chosen to be the the mean absolute error between predicted and actual prices. Let $\hat{P}_i$ be the predicted market clearing price for an hourly time step i, P_i the actual market clearing price and n the number of time steps in the evaluation dataset. The Mean Absolute Error (MAE) of the prediction on the evaluation data can be defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{P}_i - P_i| \tag{1}$$

It can be helpful though to give some kind of worst-case prediction scenario. This can be measured by calculating the maximum prediction error (MPE).

$$MPE = max\Big\{|\hat{P}_i - P_i| \quad \forall i \in n\Big\} \tag{2}$$

In order to punish higher errors more than smaller ones, the Mean Squared Error (MSE) or the related Root Mean Squared Error (RMSE) can be applied.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{P}_i - P_i)^2 \tag{3}$$

$$RMSE = \sqrt{MSE} \tag{4}$$

In literature the Mean Absolute Percentage Error (MAPE) is often applied as a relative error measure for regression tasks. It is not used in this project due to its susceptibility for misleading implications. Imagine a true price level of 0.1€/MWh and a price prediction of 5.1€/MWh. The absolute error would be 5€/MWh, nothing to worry about too much. In this example case however, the Mean Absolute Percentage Error would be 5000%. Having only a few of those examples in the evaluation data would lead to a heavily distorted error metric.

2.4 Benchmark Model

In order to give an indication about the benefits of applying a certain model, a naive benchmark is defined. The simplest benchmark is to use the Day-Ahead prices of day $d-1$ as a forecast for the prices of day d. This can be extended to use the prices of the same day a week before. Due to the strong temporal patterns in electricity prices these benchmarks should yield a relatively good estimation. A more advanced option is to take average historical prices for every individual hour of similar days as a forecast. Here, the average price of the same hour of the same weekday of the last two weeks as well as the last four weeks is calculated as a benchmark prediction. Moreover, the model performance is compared to a multitude of results from current literature about Day-Ahead electricity price forecasting.

3 The Data

In the following chapter all available datasets are be described and explored thoroughly. These analyses are used as a first anchor for modeling decisions and an indication about possible upcoming challenges to be met.

3.1 Day-Ahead Electricity Prices

The major European energy exchange, the EPEX SPOT, publishes hourly Day-Ahead prices for the German bidding zone starting from July 20, 2007. All prices until August 31, 2017 have been extracted from the website. This makes up for a total of 88704 data points. No data points are missing. All prices are given in €/MWh. Table 1 shows the major statistical characteristics of the price data. Roughly half of the observed prices range between the 25% and

count	mean	std	min	25%	50%	75%	max
88704	41.30	21.35	-500.02	29.0	38.64	50.97	821.90

Table 1: General descriptive statistics of the Day-Ahead electricity price data in €/MWh.

75% quantile of 30€ and 51€/MWh, the average price is 42€/MWh. There are some significant extreme points present in the data, with a minimum observed price of -500€/MWh and a maximum of 822€/MWh. Figure 2 shows extreme price peaks in late 2007 and extreme price lows in late 2009. These extreme values are most likely due to market flaws in the early days of Day-Ahead trading. In recent years, the common price range got much smaller. This should be considered when scaling the data prior to training, especially if relative scaling is used. Outlier removal seems to be appropriate for electricity prices.

The common temporal patterns in Day-Ahead prices mentioned before are depicted in Figure 3. Each diagram shows the average value of all data points given a certain filter criterion. There are three major temporal patterns present in the data. The first being yearly trends with high prices in the winter and lower prices during the summer. Furthermore, prices are on average lower during weekends, compared to working days. Moreover, the course of electricity prices shows a clear daily pattern, with low prices at nights, peaks in the morning and evening hours and a drop during working hours. The requirements laid down on the time series model have to include the adequate representation of all three different temporal patterns at minimum.

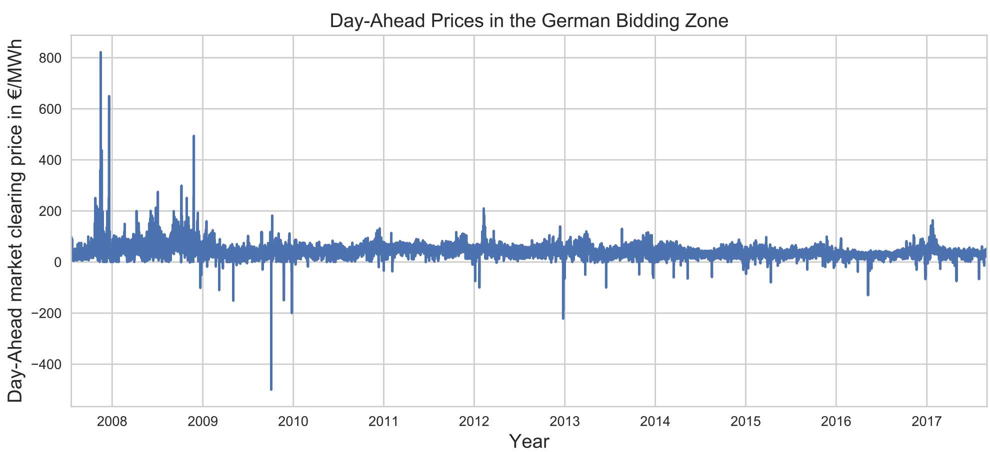

Figure 2: Day-Ahead electricity prices for the German bidding zone from 2007 to 2017.

3.2 Load and Generation

Load and generation data is available from January 1, 2015 until August 31, 2017. This data includes actual generation levels in MW for all major generation technologies, generation forecasts for solar and wind power generation, actual load and forecasted load in the German bidding zone in a *quarter hour* resolution. Accounting for occasional missing values in the different categories, the data includes between 92500 and 93500 data points in each category. Since the desired forecasting output, the price, is given in an hourly resolution, load and generation data has to be resampled in order to match the output frequency. Descriptive statistics for the true generation levels are given in Table 2, for the renewable generation forecasts in Table 3 and for true and forecasted aggregated load in Table 4.

Except for load forecasts, none of these features is actually available in a *real world application*. However, a

	biomass	brown coal	hard coal	wind offshore	pumped storage	solar	river hydro	wind onshore	nuclear	other
count	92642	92584	92797	93074	92992	92899	92961	93063	93114	92727
mean	4419	15029	9041	1307	1201	4333	4486	8420	8936	5165
std	349	2189	5074	1072	1464	6478	1437	6643	1591	2325
min	2778	5330	148	0	0	0	1791	163	3397	42
25%	4190	13921	4570	378	33	0	3324	3512	7720	3599
50%	4372	15265	8903	994	595	148	4289	6466	9073	4864
75%	4714	16558	13140	2167	1934	7162	5513	11378	10378	6255
max	5155	19230	21267	4332	9582	28527	13222	35745	11474	33409

Table 2: General descriptive statistics of hourly generation levels in MW in the German bidding zone by generation technology.

	solar forecast	wind offshore forecast	wind onshore forecast
count	93404	93308	93404
mean	4391	1338	8635
std	6534	1044	6729
min	0	4	187
25%	0	433	3660
50%	193	1070	6648
75%	7251	2162	11480
max	28310	4175	37338

Table 3: General descriptive statistics of forecasted hourly renewable generations levels in MW in the German bidding zone by generation technology.

	electric load forecast	actual electric load
count	92924	93396
mean	62069	62448
std	10804	11332
min	33802	30985
25%	53105	52940
50%	62013	62054
75%	71442	72492
max	85892	89063

Table 4: General descriptive statistics of forecasted and actual hourly load levels in MW in the German bidding zone.

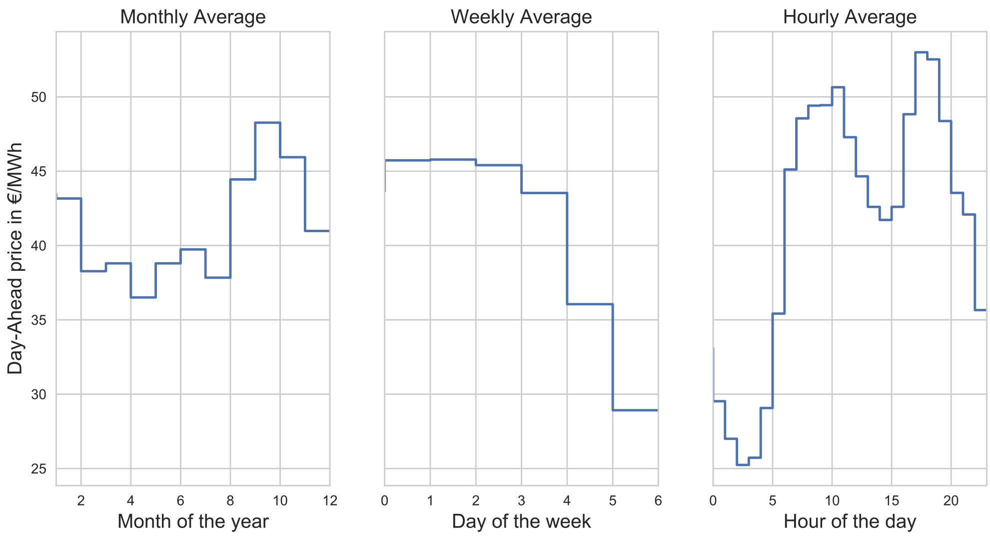

Figure 3: Monthly, weekly and hourly average Day-Ahead prices from 2007 to 2017

proven dependency between this data and the actual Day-Ahead price would motivate further modeling efforts to *replicate* this input data with appropriate models. At this point in time it is unknown whether actual or forecasted load and generation levels are better input variables for a price prediction. This decision is by no means obvious or trivial. Day-Ahead prices are based on bidding behavior, which in turn is based on *expectations* of market participants. It could very well be, that the forecasts match these expectations much better than the true, realized generation levels, making them a better basis for price prediction. This has to be clarified in the further course of this project. As mentioned above, this project focuses on load forecasts and renewable generation levels. Since the latter ones are not available in an application scenario, they have to be predicted by an additional model.

3.3 Grid Weather Data

Recently, the German National Weather Service published data from a *regional reanalysis* project for the year 2015. This dataset includes weather data for diffuse and direct solar radiation, precipitation levels, as well as zonal and meridional wind velocity for Europe. The data is organized in a grid of 5km by 5km patches and contains weather variable measurements for every patch for every hour in the year 2015. This data can be used to predict renewable generation levels in Germany. Since only weather conditions in Germany are relevant for that task, a subgrid spanning the national borders is cut out from the original data. This grid contains 145 latitude steps and 110 longitude steps, resulting in a two dimensional matrix with 15950 data points per hour per weather variable. Incorporating all five weather variables for all 8760 hours of the year 2015 makes up for a total of 698.61 million single data points. An exemplary visualization for the weather data is given in Figure 4. This plot shows national solar radiation levels for a certain hour in September 2015 in Germany from the grid data used in this work.

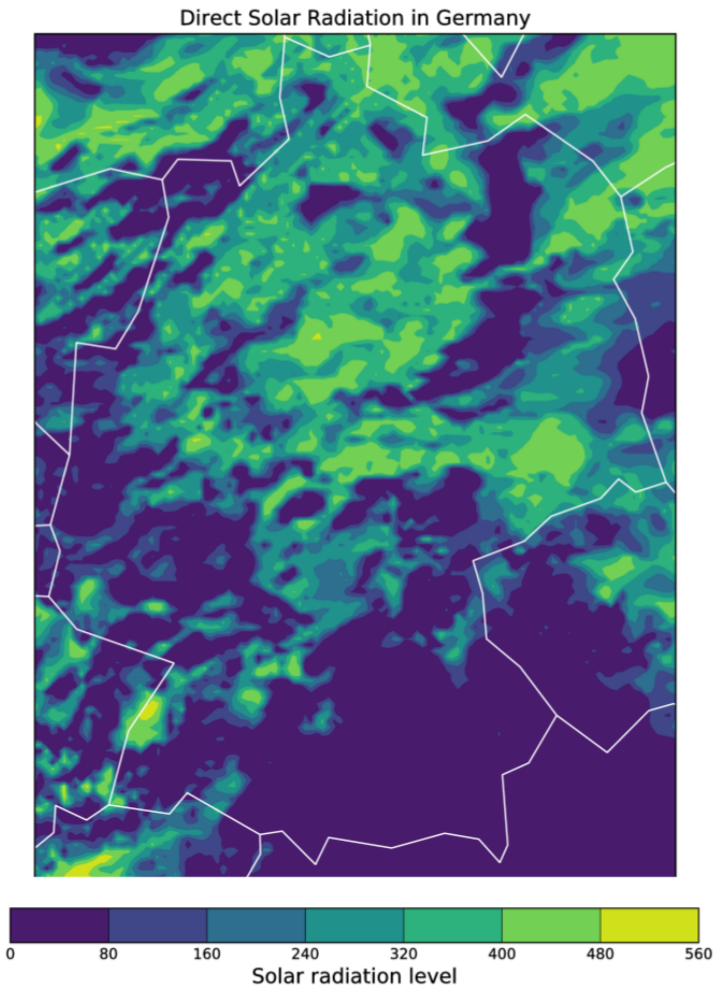

Figure 4: Direct solar radiation levels in Germany for September 3, 2015 at noon.

Of course, measured weather data is not available in a real world deployment either. What is available, are *weather forecasts* for the different variables. The DWD publishes 45 hour grid weather forecasts for all variables on a daily basis. Unfortunately, only the forecast for the current day is available at a time. Using these forecasts instead of historical data would have required the manual collection of forecasts over a whole year. Fortunately, a model that is fit on measured weather data, thereby modeling the basic physical principles behind the generation technology, should be able to predict generation levels for forecasted weather variables with the same precision. In the end, only the quality of the forecasts itself determines the precision of the overall prediction. It could turn out, that the generation level predictions based on weather forecasts actually provide an even more valuable input to a price prediction model, since the market players that determine the prices are relying on forecasts too when making their bidding decisions. This assumption only requires a certain common agreement or *authority* from which a majority of market participants is receiving their weather forecasts. The DWD is definitely a strong candidate for this authority. These assumptions however have to be proven empirically in the future by gathering weather forecasts for the DWD over a longer time horizon in order to test them on the model trained on weather measurements.

4 Algorithms, Implementation and Results

The following chapter presents the algorithms, model design decisions, data preprocessing, evaluation and improvement steps and the final results for every module defined in Section 2.2. All steps are evaluated against the benchmark models presented above. Benchmark performance on a test set spanning all price data of the year 2017 is given in Table 5. The best benchmark model is able to achieve a MAE of 8.53€/MWh.

name	mean absolute error
’last day’	8.53
’last week’	9.14
’two week average’	8.98
’four week average’	9.19

Table 5: Mean absolute error of the benchmark models for 2017 Day-Ahead prices in €/MWh.

4.1 Module I: The Time Series Model

There are numerous time series modeling approaches known from literature. Among the most popular are the Autoregressive Moving Average (ARMA) models and their task specific variations Autoregressive Integrated Moving Average (ARIMA) and Seasonal Autoregressive Integrated Moving Average (SARIMA). Recently, more complex approaches like Support Vector Regression or Artificial Neural Networks have been applied to the field of time series forecasting. In this project an Artifical Neural Network (ANN) is applied. The model includes convolutional layers for short term trends and input space reduction, recurrent layers in the form of Long Short Term Memory (LSTM) cells for sequence based trend identification and densely connected Linear Perceptron layers for output refinement.

Recurrent Neural Networks (RNN) make use of *sequence* data to consider input features within the context of other input features. Not only the feature itself, but also the history of previous features in a sequence is evaluated in order to infer the current model output. Such a sequence could for example be a vector of time series data, in this case prices, with a specified temporal order. The prediction of the price for the next hour not only depends on a set of previous prices but rather on the *order* of prices in previous hours and their magnitudes. LSTM networks are special cases of RNN, in which information can be stored, updated or forgot by means of a *cell state*. This allows LSTM networks to make use of high dimensional input spaces and learn *long term dependencies*. These attributes make LSTM networks suitable for time series prediction, in which future values of a quantity have to be inferred from historical inputs of that same quantity. A very comprehensive introduction to RNN and LSTM in particular is given in [14].

In case multiple input features are available, e.g. historical generation levels for certain technologies and historical prices, a multi-dimensional time-series model has to be applied. In a recent paper on time series forecasting [12], the authors propose a convolutional layer prior to a LSTM layer to incorporate a multi-dimensional feature space. This idea is be picked up in this work and extended by using the convolutional layer for additional reduction of the temporal input dimension. Not only might a convolutional layer include multiple features simultaneously, it is also able to consider multiple consecutive historical time steps at once. Convolutional neural networks (CNN) make use of filters, defined by a certain width, height and a fixed set of weights that span a 'window' of features. The filter is 'slided' over the input data in multiple convolutional steps. The amount of new features that is considered in each step is called a *stride*. In each step, the current inputs are weighted with the filter weights and the output is projected to a *feature map*. For further explanation about the main concepts of CNN, the reader is referred to [13].

If only one feature, say prices, is considered, the filter will only have a length (because the input will only be a vector). If more features are added, e.g. historical generation, the filter grows in width for each additional feature. The input will now be a matrix, where each column constitutes a feature and each row a timestep. Choosing appropriate filter dimensions and stride values, the convolutional layer can reduce the time dimension in the input space by mapping multiple original time steps to one single node in the feature map. If the input, for example, spans every hour of a full week, in this case 168 hours, choosing a filter height of 24 and a stride of 24 will effectively aggregate the input to a feature map of length 7, where each node constitutes the aggregated features of one full day from the input vector, weighted by the filter weights. This reduced input can be fed into the LSTM layer for sequence prediction. The idea behind that design is to assist the LSTM layer in considering longer input sequences at once without suffering from too many input features.

A first version of the time series model only considers price data. Additional features, in this case historical generation levels for different technologies, are added in a different step. A vector of 168 historical prices prior to the prediction day is chosen as an input for the model. The input range thereby covers all hourly prices of the last week. It is fed into a convolutional layer with kernel height and a stride of 24. This results in a feature map of length 7 for each filter applied to the input. This design is intentional: Each node in the feature map of a filter is a representation for prices from a whole day of the input week in consecutive order. The convolutional layer feeds into a LSTM layer, which is followed by a fully connected layer. The fully connected layer always contains 24 nodes, one node for every hour of the day of interest. Thereby, the model will always output predictions for every single hour of the upcoming day *at once*. The temporal structure of inputs and outputs is shown in Figure 5.

This data structure requires a specific preprocessing of the data. One sample contains a vector of historical input prices of the last seven days, while the output for a sample ('label') is a vector of length 24. This structure is applied to every day from the input dataset, except the first 168 entries for which no historical data is available. This reduces the original 88704 samples to 3696 samples, each representing one full forecasting day.

It is common to scale input data for Neural Networks to a range of 0 to 1. Day-Ahead prices include some significant outliers. These outliers distort the data scaling, because the extreme values define 0 and 1 equivalents. Since 'usual' samples only cover a very limited range compared to these outliers, the majority of the data would be squeezed to a range of very limited variation by the scaling procedure. Therefore, outliers in the dataset are replaced (not removed, to keep the time series structure intact). An outlier is defined as a price below the 25% or above the 75% quantile value minus or plus three times the inter-quantile range of said quartiles. The outliers

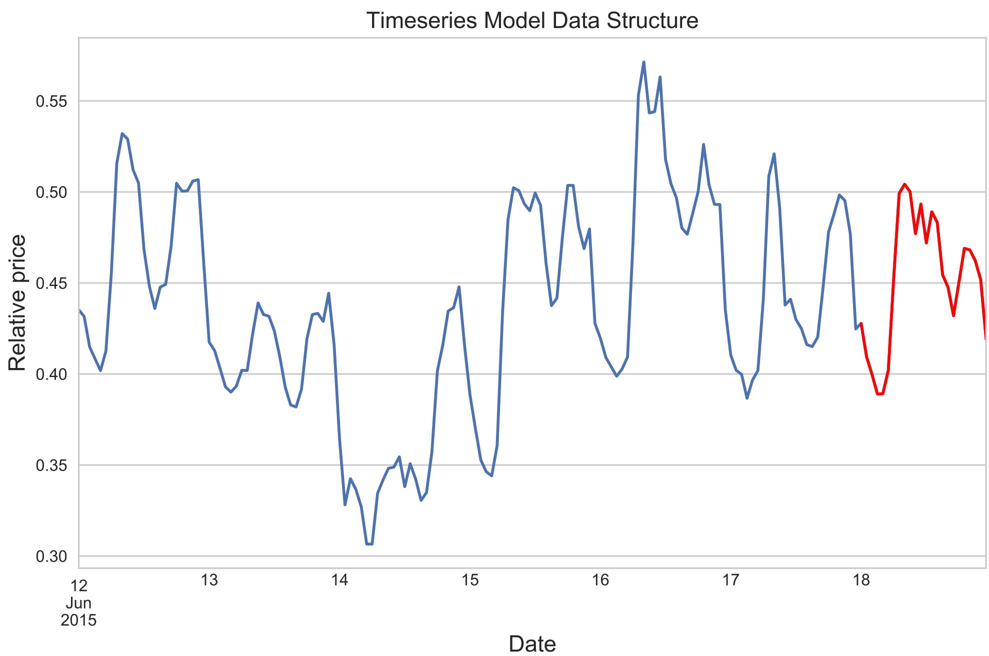

Figure 5: Time series model data dimensions. The blue line represents the historical prices used as input features for the model. The red line shows the model output, spanning all 24 hours of the prediction day.

are replaced by the respective thresholds. This data is scaled by applying a min-max scaling procedure, defined in Equation 5.

$$\hat{x}_i = \frac{x_i - min(x)}{max(x) - min(x)}. \tag{5}$$

The model is implemented using the Keras python library [5] and the tensorflow [1] backend. The mean squared error is chosen as a loss function and the adam algorithm as the optimizer. The model is trained for 200 epochs with a batch size of 50. All data prior to 2017 is used as training data, prices from 2017 constitute the training set. The training data is further split into a training and a validation set, whereby the validation set has a size of 20% of the original training set and is used to evaluate the model in between training epochs. The model state with the best validation performance is picked as the final model to be applied to the test set. Figure 6 shows the predictive performance for an exemplary week from the test set, Figure 7 the long term performance on the full test set.

Considering both results jointly shows, that the model performs quite well in covering long term trends (high, volatile price levels in winter vs lower prices in summer), the weekly seasonality of work days versus weekends, and the daily pattern of hourly price movements. Extreme values, like price peaks, are not covered very well by the model. Apart from that, there are some major errors over the course of single days. The model achieves a MAE of 5.89€/MWh, compared to the best benchmark score of 8.52€/MWh. This is a substantial improvement over the naive benchmark results and shows the validity of the chosen model architecture. In total, the model yields a solid

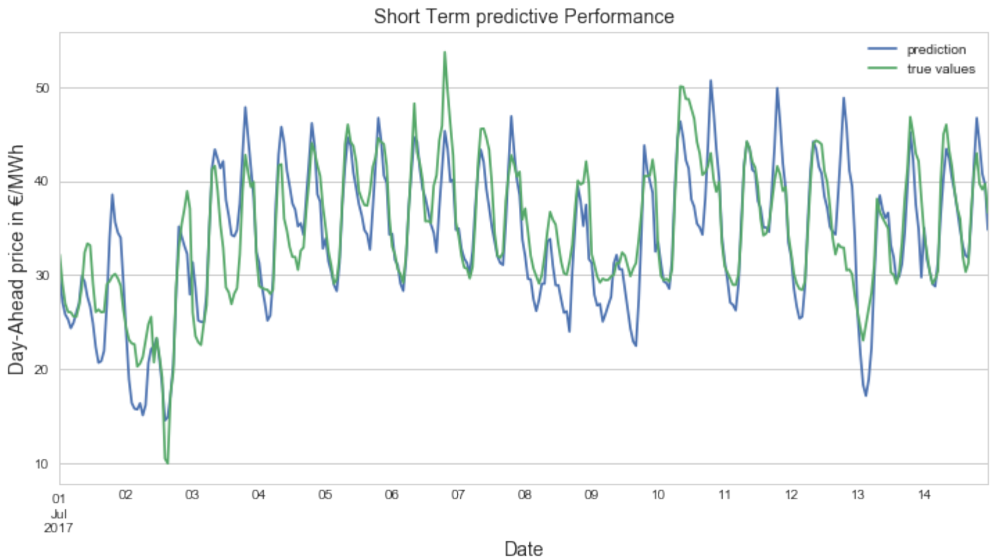

Figure 6: Predicted and true prices for two exemplary weeks from the test set.

predictive foundation for all major temporal patterns in the price data. In order to successfully cover short term deviations and 'unusual' price developments reliably, more than plain historical data is needed.

Model Parametrization

The first modeling approach is refined and parameterized in the following steps. At first, different layer sizes are applied. The convolutional layer is varied between 32, 64 and 128 nodes. The LSTM layer between 16, 32 and 64 nodes. None of the combinations yields significantly better results that the first attempt with 64 and 32 nodes for the layers. Inspired by [12], a linear bypass is added to the model. This bypass takes all 168 inputs directly into a densely connected layer of 64 nodes and passes them forward to a layer of 24 densely connected nodes. Their output is added to the output vector of the original 'branch' of the network to form a final output. This architecture is found to deliver more reliable results and therefore chosen as the final model architecture.

In order to find the optimal combination of hyperparameters for the model, a 10-fold cross validation grid search is conducted. This experiment applies a set of distinct parameter permutations to ten different, random splits of the data into training and validation set. The parametrization with the best average result is chosen to be optimal. It is important to train and evaluate the model on multiple folds of data. Training is subject to significant stochastic influences, like weight initialization and training data batching, which makes the same model vary in between training sessions substantially. In order to mitigate the effect of randomness, the model is evaluated on the basis of multiple training sessions, i.e. its *average performance*. The parameters tested in the grid search are the dropout regularization [18] probability for each layer, chosen from $\{0.1, 0.2\}$, the recurrent activation function of the LSTM cells, $\{linear, relu\}$, and the convolutional activation function, $\{linear, relu\}$. Furthermore, L1-L2 regularization, also known as elastic net regularization, is applied to the linear bypass layer. The penalization parameter for the regularizer is chosen from $\{0, 0.01, 0.02\}$, where a value of 0 refers to 'no regularization'. This

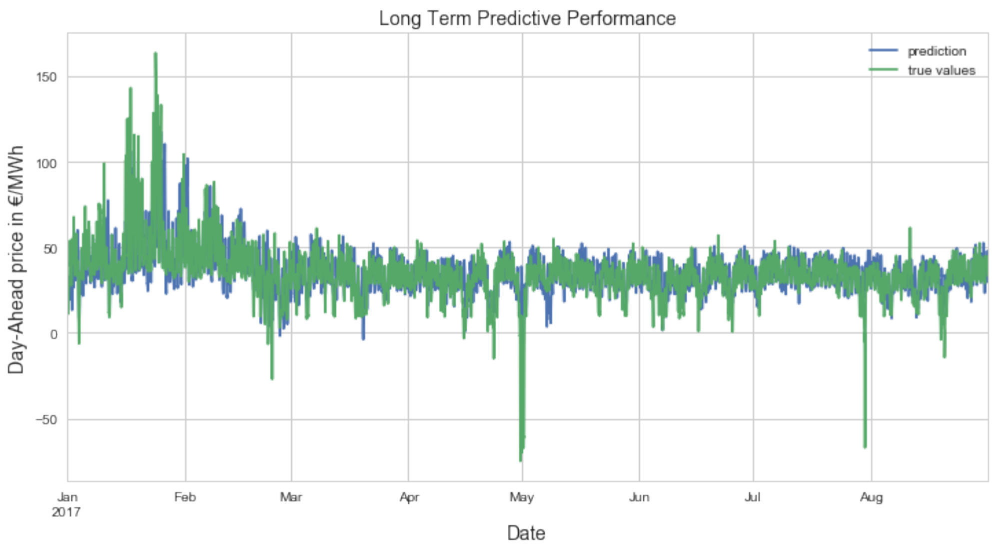

Figure 7: Predicted and true prices for the full test set.

makes up for a total of 24 permutations with ten folds per permutation, resulting in 240 full training iterations. The best parametrization is chosen based on the best average MAE over all folds.

Best results are achieved with a linear convolutional activation function, a ReLU LSTM recurrent activation function, a dropout chance of 0.1 and a elastic net regularization weight of 0.01, based on the average cross validation MAE. The fully parameterized model is trained again on the full training data which results in a final model MAE of 5.47€/MWh on the test set, a significant improvement over the non-optimized first modeling attempt.

Multi-Variate Time Series Model

In a next step, the purely price-based model is enriched with historical generation data. By adding columns to the input matrix (one column per feature) and additional width to the convolutional filter, the model is able to consider multiple features in parallel. The linear bypass can not be applied in this variation. The multi-variate model is tested with additional features like historical generation levels for brown and hard coal, nuclear and renewable power plants, as well as historical electric load. The new data is resampled to one-hour resolution by taking the arithmetic mean over all quarter-hours of one hour. It is scaled to 0-1 range by applying Equation 5 prior to model training. None of the above features or any combination among them is able to improve the model performance. The resulting MAE values mostly lie well above 6.5€/MWh and are therefore substantially worse than the price-only approach. The multi-variate time series approach will not be considered any further.

Improvements and Future Steps

Although the final time series model yields satisfactory results, there is a lot of potential for improvement. Due to the limited scope of this work, only parameters of a relatively fixed architecture are varied in the testing procedure. It might be beneficial to experiment with additional layers and layer sizes. Apart from that, the model should be tested extensively on variations of the input vector size (more than 168 historical hours), scaling and outlier removal procedures (e.g. robust, quartile-based scaling) or convolutional filter size and stride.

4.2 Module II: The Renewable Generation Model

Renewable generation levels have a major impact on the Day-Ahead electricity price. The renewable generation model aims at giving precise estimations of the generation levels, based on geo-spatial weather data. The main idea behind this model is to correctly reproduce the *heterogenous distribution* of renewable generation power plants over Germany. If the geographic area of Germany is split into equally sized squares, each square contains a certain 'density' of renewable generation capacity, say solar power plants. A high solar radiation in this square results in a higher generation output, compared to the same radiation level in a square with less photovoltaic power plants. The model has to be able to infer that generation capacity distribution based on weather data alone: It is only shown certain weather variables, say direct and diffuse radiation during one hour, for each square in a grid that spans all of Germany, and the solar generation level for all of Germany in that particular hour. Ideally, the model now applies weights to every square, which represent the generation capacity in that area. By applying higher weights for a high capacity area and vice versa, the model should be able to correctly infer the total generation level from input weather data.

The weather input data includes 145 latitudinal and 110 longitudinal steps, making up for 15950 squares in total. Using these as a direct input vector for a model is highly unlikely to yield good results, due to the enormous size of the input space. Instead, a convolutional neural network is applied. This network does not only reduce the size of the two-dimensional input matrix by applying filters and strides, it also keeps the spatial proximity relation between the squares intact. Furthermore, if multiple filters and multiple channels are used, the model is able to evaluate a combination of multiple, two-dimensional input matrices containing different weather variables. This concept can be compared to the popular approach for image recognition based on convolutional neural networks. In this application, each channel represents a color channel, each entry in the input matrix a pixel value from the input picture. In case of weather data, each weather variable is one channel, each square in the grid an entry in the input matrix.

The solar generation model consists of an input layer with three channels, which are direct and diffuse solar radiation and total precipitation, a convolutional layer with 64 filters of size 2 and stride 2 (in both dimensions), a second convolutional layer with 12 filters of size 5 and stride 5 and a final fully connected output layer with one node. The first layer reduces the input space from 5km by 5km squares in the original input to 10km by 10km squares. The 64 filters allow for different combinations of input channel magnitudes. The second layer reduces the feature space more drastically, due to its bigger filter and stride, and reduces the output feature maps to 12. The output is flattened to a vector of length 1848 (every node in all feature maps of the second layer) and fed to the dense layer, which produces a single output value, the generation level. In this model, all layers use the ReLU activation function. The model is trained with the adam optimizer using a mean squared error as the loss function and ten training epochs. Input and output data is scaled to a range of 0-1 by dividing every data point by the maximum observed value for that variable in the whole input data set. The available training data (every hour of the year 2015) is split into a training and a test set of size 0.2. Only full days containing all 24 hourly time steps of that day are picked by the splitting procedure in order to keep the daily structure intact for visualization purposes.

A randomized validation split containing 20% of the samples from the leftover training set is used to choose the best model from all training epochs. Again, the Keras library is used for implementation, choosing a mean squared error loss function, the adam optimizer and 10 epochs for training.

The results for the solar power generation model are given in Figure 8 for the whole test set and in Figure 9 for a smaller subset. The overall model performance is surprisingly precise for the relatively simple modeling approach. General trends in generation levels and the typical daily cycle are mostly matched quite accurately. The total magnitude is often not met on point; there are still some over- or under-prediction errors present in the forecast. However, the overall tendency is met for most days in the test set.

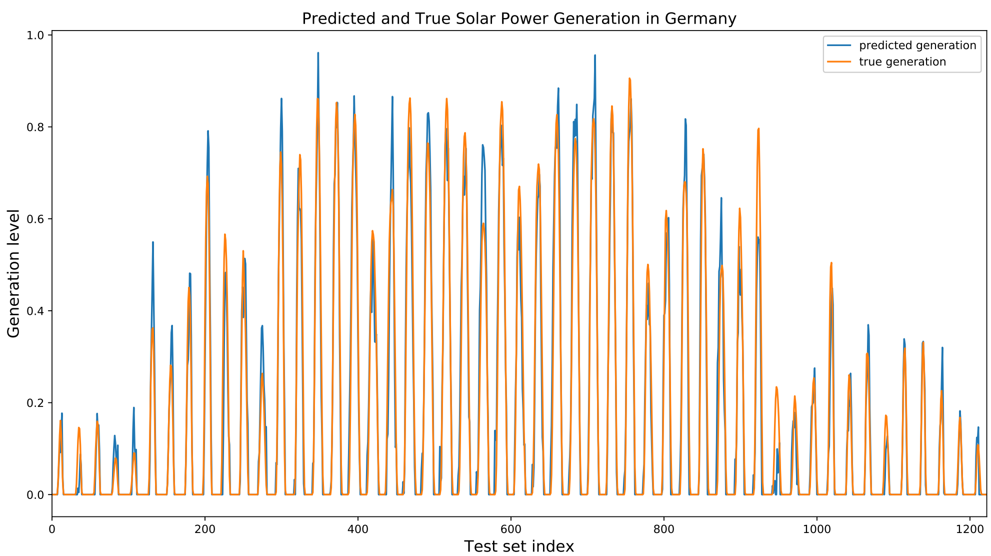

Figure 8: Predictive performance of the solar power generation model on the full test set.

The same model architecture and training procedure is applied to zonal and meridonal wind velocities as input data to predict wind power generation (aggregated onshore and offshore). The results for the wind power prediction on the test set are depicted in Figure 10 and Figure 11. Again, the model achieves very pleasing results. There are no substantial prediction errors visible, just minor precision errors. The general training and validation loss of the wind power model are actually even smaller than those of the solar power model.

Improvements and Future Steps

Although the renewable generation models yield very pleasing results, there is still room for improvement. In addition to more experiments with different model architectures, a grid search for hyperparameter tuning is likely to improve the results even further. Apart from that, additional input variables might be tested for their effect on the overall predictive power. Solar power plants for example are also sensitive to ambient temperature. Wind power generation varies with air density, which might be influenced by pressure levels, air humidity and temperature. All these variables could be tested within the model framework. This is out of the scope of this work, but most likely a promising approach for further improvement in the future.

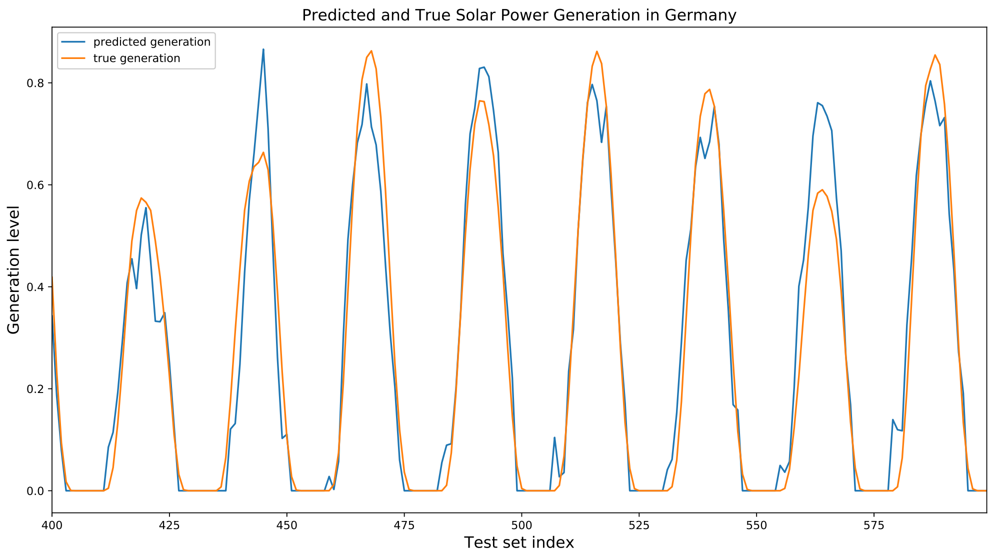

Figure 9: Predictive performance of the solar power generation model on a subset of the test set.

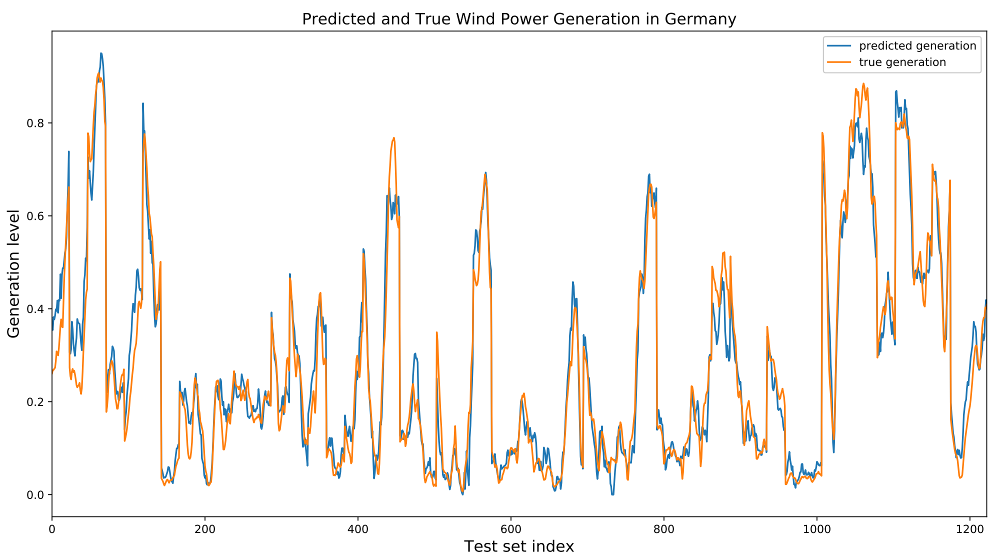

Figure 10: Predictive performance of the wind power generation model on the full test set.

4.3 Module III: Aggregating all Inputs

The preliminary time series based estimations of Module I can now be refined with the renewable generation estimations and the external load forecasts for the day of interest. These inputs represent all features that are available before Day-Ahead market closing in a real world application ('final model'). The model performance of this model is compared to two idealized benchmarks. The first benchmark ('perfect information') includes real gen-

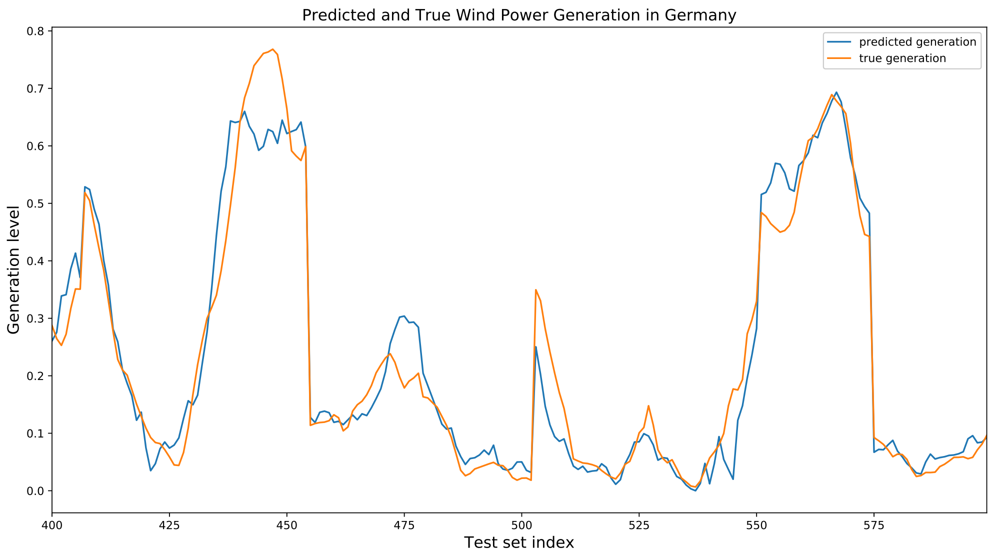

Figure 11: Predictive performance of the wind power generation model on a subset of the test set.

eration levels for all major power generation technologies (brown coal, hard coal, nuclear, pumped hydro, river hydro, biomass, other), the external forecasts for wind and solar power, the external load forecast and the time series model prediction output as features. This model represents near-perfect information and thereby a hard upper performance boundary that is achievable with the chosen learner. A second benchmark includes external forecasts for wind and solar power, as well as load forecasts and time series model prediction output ('renewable benchmark'). This benchmark serves as a basis to show potential improvements of the renewable generation model. It is based on perfect renewable generation forecast information, the quantity that the renewable generation model tries to mimic, while all other features are held equal. Again, all data is sampled to one-hour intervals by taking the mean over sub-hour samples. In contrast to the previous modules, the data is not scaled since only scale invariant regressors are used in this part of the project.

These different feature sets are used to train a regression model in order to output a final prediction of hourly Day-Ahead prices, one hour at a time. A Gradient Boosting Regressor is chosen as the main learning algorithm. Gradient Boosting is an ensemble method that combines multiple weak learners, like regression trees, in an iterative fashion to form one strong learner. Gradient Boosting iterates over sequential *stages*, each in which a weak learner is fit on a subset of the data. On an intuitive level, this subset is chosen based on 'hard' data samples that previous stages performed badly on. By splitting the data in each stage, the algorithm constructs an ensemble of learners, each fit to a particular subset of the data. The final prediction is derived by weighting and averaging the predictions of all weak learners. Gradient Boosting is quite popular among the available machine learning algorithms, since it delivers good results regularly and is quite robust to overfitting. Moreover, it is scale invariant, the input features can remain in their original scale. This is quite helpful for this particular application, since price and power generation have different units and numerical magnitudes.

The model is implemented using the off-the-shelf Gradient Boosting Regressor of the scikit-learn python library [17]. In a first test, library default parameters are applied. Only data for the year 2015 can be used here, since overlapping weather data and renewable generation/forecast data is only available for this particular year. The data is split into a training and test set (size 0.2 of the whole data) by random splitting. The performance of the different parameter combinations on the full test set are shown in Table 6. There are some major implication to be

	MAE	MPE	MSE	RMSE
'perfect information'	3.28	24.39	19.26	4.39
'renewable benchmark'	3.64	21.96	23.35	4.83
'final model'	3.74	23.38	24.67	4.97

Table 6: Performance metrics for the Gradient Boosting Regressor on different feature subsets.

drawn from these results. Firstly, the final model performs quite well in direct comparison with two benchmarks. The perfect information model is ahead in terms of error metrics by quite a significant margin. Compared to the renewable benchmark, the final model is able to achieve a comparable performance, missing the benchmark by only 0.1€/MWh mean absolute error. The overall performance of the final model constitutes a significant performance improvement, compared to the former best MAE of 5.47€/MWh achieved by optimized time series model. Besides these main findings, a very interesting implication can be drawn from fitting the renewable benchmark on actual generation levels instead of the forecasted ones (all else equal): The model performs slightly *worse* with an MAE of 3.66€/MWh, indicating that expected renewable generation has indeed more explanatory power than actual generation.

In order to maximize the performance, a grid search 3-fold cross validation is applied. The following parameters are used to construct the different parameter permutations: The loss function is chosen from *least squares*, *least absolute deviation* or *huber*, the learning rate from $\{0.05, 0.1, 0.3\}$, the total number of weak learners (regression trees) applied are $\{100, 1000, 5000\}$, the minimum number of samples in a single data split chosen from $\{2, 10, 20\}$ and the maximum depth of the weak learner tree from $\{3, 5, 10\}$, which makes up for a total number of 243 parameter permutations. The best result is achieved with a combination of the huber loss function, a learning rate of 0.05, 100 base estimators, a minimum number of 20 samples per split and a maximum depth of 5 for the weak learner regression trees. A similar parametrization is applied to the benchmark feature sets, choosing 1000 weak learners for the renewable benchmark and 5000 weak learners for the perfect information benchmark instead of 100. Performance metrics for the optimized models are depicted in Table 7. The parameter opti-

	MAE	MPE	MSE	RMSE
'perfect information'	2.27	27.98	10.35	3.22
'renewable benchmark'	3.18	24.63	18.59	4.31
'final model'	3.46	25.99	22.24	4.72

Table 7: Performance metrics for the Gradient Boosting Regressor on different feature subsets after cross validated parameter optimization.

mization improves the model performance on all feature sets and metrics, except for the maximum prediction error (MPE). It seems like average model performance is traded for an increased maximum errors. Given the lower mean squared error on all subsets however, these larger errors can only be present in very rare occasions, otherwise this metric would not have improved by such a significant margin. The main performance metric, MAE, improved substantially to 3.46€/MWh for the final model. A direct comparison to the optimized benchmark results show that the margin of feature improvement for the final model is bigger than indicated by the previous unoptimized parametrization. The renewable benchmark beats the final model by almost 0.4€/MWh MAE, which is a direct measure of potential improvements that might be achieved by further improving the renewable generation model from Module II. The advantage of perfect information in the form of additional features becomes even more obvious in case of model optimization: The perfect information benchmark is able to achieve a MAE of 2.27€/MWh. Hence, additional features, here generation data for the forecasting day, harbor a huge potential for model improvement. If one could find a reliable way to model these features too, this would most likely improve the model precision by a significant margin.

Nevertheless, the final model performance is quite compelling. Overall, the mean absolute error has improved from 8.52€/MWh of the naive benchmark to 5.47€/MWh for the optimized time series model down to 3.46€/MWh for the final model including renewable generation and load forecasts.

There are some examples from current literature that deal with the problem of Day-Ahead electricity price forecasting and could serve as an additional benchmark for the final model: [2] use a neural network approach to predict Day-Ahead prices for the New England and Italian market. [15] use an artificial neural network approach and time series prices as well as load forecasts to predict prices in the Italian market. [4] apply different preprocessing procedures and a multi-layer perceptron to predict Day-Ahead prices for the German bidding zone. [3] test different time series modeling approaches to predict prices in the Nord Pool electricity market. [16] conduct an extensive study of time series prediction models for estimations of electricity spot price distributions. [11] use ARMA based time series approaches to predict future prices in the Nord Pool market area.

It is quite difficult to compare the results of these studies to what has been achieved in this project, mainly due to differences in data sets, performance metrics, general model design and performance evaluation methods. However, it is safe to state that none of the sources above is able to deliver a better average result over a comparable test set. Many sources are achieving comparable results to the time series model presented in this work. The most compelling and comparable performance is given in [16]. The authors too use all available Day-Ahead prices for the German bidding zone and enrich them with long-term coal prices, CO_2 prices, gas prices as well as wind and solar power forecasts from external vendors. Their best model achieves a mean absolute error of 4.05 €/MWh and a root mean squared error of 5.47€/MWh for a test set in the year 2015. The approach presented in this work achieves a MAE of 3.46€/MWh and a RMSE of 4.72€/MWh for a very similar feature base and evaluation framework. Please note that these results have to be taken with a good grain of salt. As explained earlier in this work, actual weather measurements are used for the renewable generation level prediction, due to the lack of real world forecasting data. A qualitative argument is given for why using actual weather forecasts as inputs instead might yield a comparable or even better result. As long as this hypothesis is not proven empirically, the results of a model including the renewable generation model results cannot be called 'superior' to any other approach. However, there is a strong indication that the model has a potential to outperform the approaches presented in the literature above.

5 Conclusion and Future Steps

In this project a combination of three different supervised learning approaches is used to predict Day-Ahead electricity prices for the German bidding zone. It is shown that a neural network time series model based on a combination of convolutional and recurrent layers is able to deliver a good basic estimation for all hourly prices of the forecasting day. A renewable generation model based on a convolutional neural network is then used to predict hourly solar and wind power generation for the German bidding zone during the forecasting day. Together with external load forecasts provided by the German Transmission System Operators, the renewable generation forecast and the time series based price estimation are used in a Gradient Boosting Regressor to generate a final prediction for the Day-Ahead prices. This prediction is able to beat a naive benchmark model by a significant margin. Under some fair assumptions the whole approach outperforms comparable results for Day-Ahead price prediction found in current literature.

Especially the renewable generation model applied in this project yields surprisingly good results. The model is able to predict the major characteristics of hourly solar and wind power generation in Germany, based only on a few weather variables with only a few small scale errors on unseen data. Precise renewable generation forecasts might not only be helpful for price estimation but also for a variety of energy system related models or simulations. The time series model training proved to be particularly different in terms of performance stability. Performance varied quite significantly among training iterations. In the end, elastic net regularization was found to be the key for stable performance and reliable overall results.

There are some major aspects for further improvement that might be applied to the modeling approach used in this project:

- Include more generation data. Apart from renewable generation, other technologies such as coal or nuclear generation have been shown to have a significant influence on the price estimation. Including historical values of these features in a multi-variate time series model has not been successful. A more promising approach would be to create distinct models for generation prediction for each feature, similar to the renewable model in this work. These predictions could be used to further improve price forecasts.
- Procure or engineer more features. For example, one could introduce a feature that accounts for holidays in the time series data (they should resemble weekends in terms of electricity prices). Additional power systems related information like power plant downtimes, international energy trade or underlying raw-material prices (coal, gas etc.) are other potential candidates to further improve the model performance.
- Gather real world weather forecast data over a substantial time horizon from the DWD weather forecasting service platform to prove the validity of the renewable model for generation forecast prediction.
- Enhance the renewable generation model with more weather variables like air pressure, ambient temperature or humidity levels.
- Run large scale computational tests on the time series and renewable generation model to further optimize parameters, model design and feature preprocessing. Only a few tests have been conducted in this work due

to time and resource limitations.

- Try out different model architectures and inputs for the time series model, e.g. more historical input data per sample.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] N. Amjady, A. Daraeepour, and F. Keynia. Day-ahead electricity price forecasting by modified relief algorithm and hybrid neural network. *IET Generation, Transmission & Distribution*, 4(3):432, 2010.

[3] Rita Beigaite. Electricity price forecasting for Nord Pool data. pages 37–42, 2017.

[4] Marin Cerjan, Marin Matija, and Marko Delimar. Dynamic hybrid model for short-term electricity price forecasting. *Energies*, 7(5):3304–3318, 2014.

[5] François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[6] Johanna Cludius, Hauke Hermann, Felix Chr. Matthes, and Verena Graichen. The merit order effect of wind and photovoltaic electricity generation in germany 2008-2016: Estimation and distributional implications. *Energy Economics*, 44(Supplement C):302 – 313, 2014.

[7] DWD. Regional reanalysis COSMO-REA6 FTP Server. `ftp://ftp-cdc.dwd.de/pub/REA/`.

[8] ENTSO-E. ENTSO-E Transparency Platform. `https://transparency.entsoe.eu/dashboard/show`.

[9] EPEXSPOT. EPEX SPOT SE: Day-Ahead Auction. `http://www.epexspot.com/en/market-data/dayaheadauction`.

[10] European Commission. Commission Regulation (EU) No 543/2013 of 14 June 2013 on submission and publication of data in electricity markets and amending Annex I to Regulation (EC) No 714/2009 of the European Parliament and of the Council Text with EEA relevance. `http://eur-lex.europa.eu/eli/reg/2013/543/oj`, 2013.

[11] Tryggvi Jónsson, Pierre Pinson, Henrik Aalborg Nielsen, Henrik Madsen, and Torben Skov Nielsen. Forecasting electricity spot prices accounting for wind power predictions. *IEEE Transactions on Sustainable Energy*, 4(1):210–218, 2013.

[12] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. `http://arxiv.org/abs/1703.07015`, 2017.

[13] Christopher Olah. Conv Nets: A Modular Perspective. `http://colah.github.io/posts/2014-07-Conv-Nets-Modular/`, 2014.

[14] Christopher Olah. Understanding LSTM Networks. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`, 2015.

[15] Ioannis P. Panapakidis and Athanasios S. Dagoumas. Day-ahead electricity price forecasting via the application of artificial neural network based models. *Applied Energy*, 172:132–151, 2016.

[16] Christian Pape, Arne Vogler, Christoph Weber, By Christian Pape, Arne Vogler, Oliver Woll, and Christoph Weber. Forecasting the distributions of hourly electricity spot prices - Accounting for serial correlation patterns and non-normality of price. 2017.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[18] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[19] Rafał Weron. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, 30(4):1030–1081, 2014.