Author(s): Rantonen, Mika; Korpihalkola, Joni

Title: Prediction of spot prices in Nord Pool's Day-ahead market using machine and deep learning

Year: 2020

Version: Accepted version (final draft)

Please cite the original version:

Rantonen, Mika; Korpihalkola, Joni (2020). Prediction of spot prices in Nord Pool's Day-ahead market using machine and deep learning. In G. Nicosia, V. Ojha, E. La Malfa, G. Jansen, V. Sciacca, P. Pardalos, G. Giuffrida, R. Umeton (Eds.) Machine Learning, Optimization, and Data Science. LOD 2020. Lecture Notes in Computer Science, vol. 12565. Springer, Cham.

DOI: https://doi.org/10.1007/978-3-030-64583-0_59

Prediction of Spot prices in Nord Pool's Day-ahead market using Machine learning and Deep learning

Rantonen Mika[1 [0000−0002−5320−0853]] and Korpihalkola Joni[1 [0000−0001−6434−1240]]

Institute of Information Technology, JAMK University of Applied Sciences (JAMK), Piippukatu 2, 40100 Jyväskylä, Finland

Abstract. Aim of this paper is to describe and compare the machine learning and deep learning based forecasting models that predict Spot prices in Nord Pool's Day-ahead market in Finland with open-source software. The liberalization of electricity markets has launched an interest in forecasting future prices and developing models on how the prices will develop. Due to the improvements in computing capabilities, more and more complex machine learning models and neural networks can be trained faster as well as the growing amount of open data enables to collect of the large and relevant dataset. The dataset consist of multiple different features ranging from weather data to production plans was constructed. Different statistical models generated forecasts from Spot price history and machine learning models were trained on the constructed dataset. The forecasts were compared to a baseline model using three different error metrics. The result was an ensemble of statistical and machine learning models, where the models' forecasts were combined and given weights by a neural network acting as a metalearner. The results also prove that the model is able to forecast the trend and seasonality of Spot prices but unable to predict sudden price spikes.

Keywords: Machine Learning · Deep Learning · SPOT price prediction.

1 Introduction

Nowadays, the liberalization of power markets enables to trade electricity, it can be bought and sold as any other commodity. The participants of electricity market have to optimize profits and risks for example to accurately forecast of short-term future electricity prices. [5]

Finnish area spot prices are defined on Nord Pool's Elspot market. Nord Pool is a public company that operates on two different markets, Elspot market and Elbas market. Elspot market is the day-ahead market, which is the focus of this publication. The participants in the Nord Pool Elspot market are mostly electricity retailers and large production plants on the buyer side and owners of large power plants on the seller side. Elspot market is divided into bidding areas, each with a transmission system operator responsible for monitoring congestion in their grid. Physical power contracts are traded in Elspot market for the next day, which is why it is also referred to as a day-ahead market. Participants in the Elspot market have until 12:00 CET to submit their purchase and sell orders for the 24-hour period on the next day. The Spot market members can submit hourly orders, block orders, exclusive group orders and flexi orders. [14].

The electricity prices are affected by many variables, such as electricity consumption, outside temperature and electricity production plans. The relevant variables are defined using data analysis methods to build a relevant dataset. The dataset will be used to train machine learning models and neural networks to predict future spot prices. Statistical models will also be used to forecast spot prices based on price history data [14].

2 Related works

Based on the market needs and the evolution of neural networks, several methods already been proposed in the literature to model the electricity price. Li et al.[9] presents that the methods can be categorized into equilibrium analysis, simulation methods, time series, econometric methods, intelligent system methods, and volatility analysis. The publication compares these methods and techniques based on the model classification, time horizon and prediction accuracy. The summarized information is helpful to the verification, comparison and improvement of a specific method or hybrid method for electricity price forecasting in the competitive environment. Tan et al. [12] proposes a price forecasting method based on wavelet transform combined with Autoregressive Integrated Moving Average (ARIMA) and generalized autoregressive conditional heteroskedasticity (GARCH) models. The results show that the proposed method is far more accurate than the other compared forecast methods. Knapik [8] uses the autoregressive ordered probit, a Markov model, and an autoregressive conditional multinomial model to analyze the drivers of the process and to forecast extreme price events. The best forecasts of the extreme price events are obtained based on the ACM(1; 1) model. Amor e. al. [3] proposes the predictive performance of the proposed hybrid k -factor GARMA-LLWNN model which provides evidence of the power compared to the hybrid ARFIAM-LLWNN model, the individual LLWNN model and the k -factor GARMAFIGARCH model resulting a robust forecasting method. Karabiber et al. [7] presents the three individual models for forecasting in the Danish Day-Ahead Market. The used models were Trend and Seasonal Components (TBATS), ARIMA and Artificial Neural Networks (ANN) in which ARIMA and ANN methods are used with external regressors. All three models have surpassed the benchmark seasonal naïve model and ANN have provided the best results among the three models. Aggarwal et al. [2] presents an overview of different price-forecasting methodologies and key issues have been analyzed. They concluded there is no systematic evidence of out-performance of one model over the other models on a consistent basis. Beigait et al. [4] presents, there are many approaches which can be used for electricity price forecasting, features such as multiply seasonality, high volatility and spikes make it difficult to achieve high accuracy of prediction in Lithuania's electricity price zone. The highest average accuracy during forecasting experiments was achieved using Elman neural network, however the most accurate prediction (MAPE error equal = 2.94 %) was made by using Jordan network. Voronin [14] presents a model which aims not only to predict dayahead electricity prices and high degree of accuracy but also price spikes in Nord Pool. The proposed models are based on an iterative forecasting strategy implemented as a combination of two modules separately applied to normal price and price spike prediction in which the normal price module employed the previously applied forecasting technique that was a mixture of Wavelet transform (WT), linear SARIMA and nonlinear Neural Network (NN). The price spike module was a combination of the spike probability and the spike value forecasting models. Wang et al.[15] propose an electricity price forecasting framework which consists of two-stages feature processing and an improved Support Vector Machine (SVM) classifier has been proposed to solve this problem. A new hybrid feature selector based on Grey Correlation-Analysis (GCA) is used to process the n-dimensional time sequence as an input. Furthermore, Kernel Principle Component Analysis (KPCA) is applied to extract new features with less redundancy to boost SVM classifier in accuracy and speed. Furthermore, the differential evolution (DE) algorithm obtains the appropriate super parameters for differential evolution DE based Support Vector Machine (DESVM) automatically and efficiently. Conejo et al.[6] presents the recommended set of variables are demand, differenced demand, and electricity prices lagged by 24 and 168 hours. Furthermore, for week-ahead forecasts, the regression model with an hourly approach is recommended, while for day-ahead forecasts, the Seasonal Autoregressive Integrated Moving Average (SARIMAX)-model is recommended, which also includes electricity prices lagged by 1 and 2 hours as input variables and includes electricity prices lagged by 1 and 2 hours as input variables. Su et al. [11] proposed data-driven predictive models for natural gas price forecasting based on common machine learning tools such a as ANN, SVM, gradient boosting machines (GBM), and Gaussian process regression (GPR). Results show that the ANN gets better prediction performance compared with SVM, GBM, and GPR. Brusaferri et al. [1] proposed a probabilistic energy price forecast based on Bayesian deep learning techniques. The results show the capability of the proposed method to achieve robust performances in out-of-sample conditions while providing forecast uncertainty indications.

3 Dataset

3.1 Data sources and Feature Engineering and Importance

The electricity prices are affected by many variables therefore the dataset has be to collected from many different resources. Therefore the data can be various format and the granularity of data can vary drastically. Electricity prices are highly volatile due to the fact that there is no economically viable way to store large amounts of electricity. Unexpected demands in electricity, shortages, transmission failures, generator failures are some of the usual causes for price spikes [10]. Since the spot price is dictated by the market, some price spikes may occur due to market gaming or false speculations [8].

Since the spot price is calculated for every hour of the day, and the price data is available in an hourly frequency, it is necessary for all others variables to be in an hourly frequency. The fluctuating spot price is clearly cyclical. There is a clear 24-hour cycle of how the price usually behaves, a weekly cycle with the price being lower on the weekends and a yearly cycle with price being higher in winter times. Simply using a number to describe the time does not convey its cyclical nature to a machine learning model. Feature engineering principles are used to create new features. With time series data, rolling mean and standard deviation with varying window sizes was used as features. A rolling mean with a window size of 36 for example would calculate the mean of the latest 36 spot prices. A lagged version of spot price can also be used as a feature. Spot prices lagged by one day and one week were added as features, because the adequate forecast results can be achieved by just using the prices from the day before or week before. Having too many features can negatively affect the accuracy of the models. A list of features to be deleted was created from the result of random forest algorithm's permutative importances function.

Feature importances were listed from the trained random forest model. This provides a percentage for each feature meaning how much the feature affects the final prediction. To compare the random forest's Gini importances and permutative importances, both were retrieved from the same random forest model. According to Gini importance, the precipitation and cyclical month features had zero impact on the final prediction. The most significant features were the spot price one week ago, spot price 24 hours ago, Prophet's daily seasonality and Prophet's additive terms. All these features had an importance score of 0.05 or higher. Permutation importance, calculated Prophet's daily seasonality to be the most important feature with a 0.15 importance score. The second-best feature was Prophet's additive terms, all other features were below 0.07 score. Interestingly, energy consumption in Sweden, temperature in Sweden and Finland and yearly seasonality of Prophet gained a negative importance score. The results of Gini and permutation importances are vastly different. Both agree that precipitation data at its current form is not useful. However, no further assumptions regarding what features to cut can be made. Testing of one-hot-encoded data, the importance scores claimed that the one-hot encoded features had no importance on the prediction. However, when the random forest was trained without one-hot encoded features, the losses were higher. This most likely means that the feature importance score is not compatible with one-hot encoded features and may not always provide informative results.

4 Predicting the Day-ahead hourly prices

4.1 Autoregression

An autoregression (AR) model was trained that forecasts the next 24 hours. The model is evaluated on the whole test dataset using the walk-forward validation method. As would be expected, the model reacts a day late to the sudden upward or downward spikes in spot prices as seen in Figure 1. However, if there are no sudden spikes, the model is able to produce an acceptable forecast on normal workdays. For a simple model, it achieves quite low mean absolute error (MAE) and root-mean-square errors (RMSE).

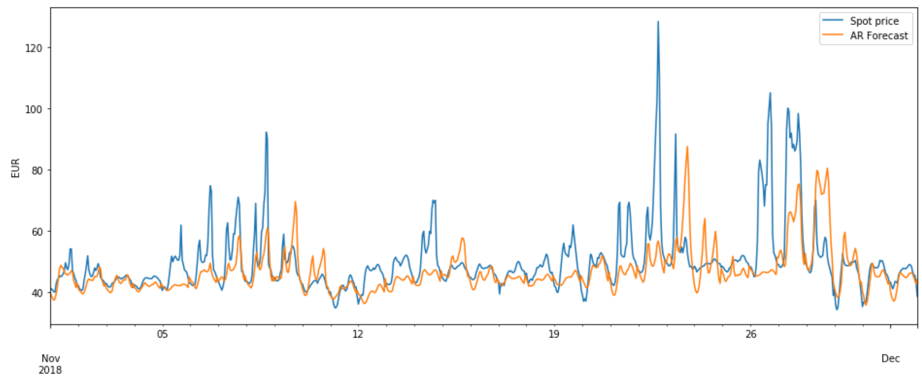

Fig. 1. Autoregression model predictions compared to real spot prices

4.2 Prophet

The prophet model can be tuned by changing the following hyperparameters: *changepoints*,*n_changepoints*, *changepoint_range* and *changepoint_prior_scale*. Changepoint is when the time series trend changes suddenly and *changepoint_ prior_scale* determines how much the changepoints fit into the data. In case of the spot price, the spot price weekly trend lowers during the weekends and yearly trend lowers during the summers, at least usually. The effects of holidays can also be changed. It was seen earlier that the spot price trend lowers during winter holidays. Prophet model forecasts from spot price history data alone, since the use of exogenous variables is not supported. The prophet model is fast to train; therefore, a grid search was implemented to find the best hyperparameters. According to [13] the model can perform worse with more data, since a longer history can mean that the model is overfit to past data that is maybe not as relevant in the future. Thus, the length of the training data was also included in the grid search. After grid search, the best parameters for the model were found. The changepoint prior scale was at 0.05, n_changepoints at 48, holidays_prior_scale at 10, seasonali-ty_prior_scale at 10, seasonality mode in additive and the model was only given data from 2018. The model was made to predict on the test dataset, and the forecasts were compared to actual spot prices. The model predicts spot prices on average to be 7.2 euros lower or higher than the real value. When plotting the forecast values and actual values in a line graph in Figure 2, it can be seen that the model has learned the daily and weekly seasonality well. The weakness of this model is that it is unable to predict sudden spikes in the spot price. The Prophet model provides a data frame of yearly, monthly, weekly and daily seasonality along with additive terms.

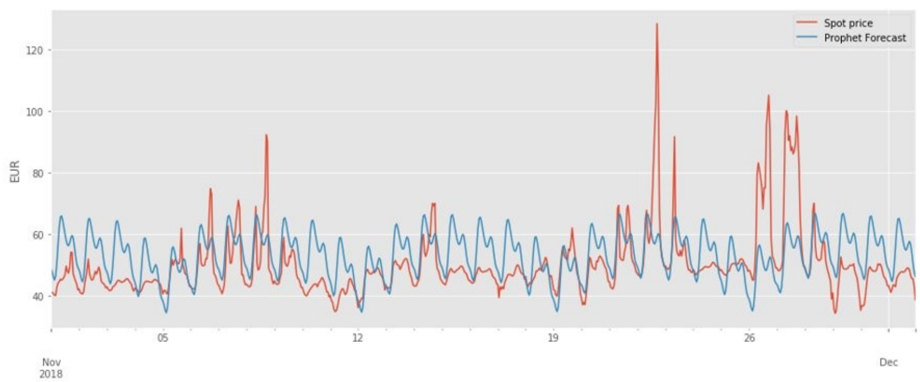

Fig. 2. Prophet's forecasts and spot prices

4.3 Onestep LSTM

A multivariate Long short-term memory (LSTM) network uses multiple features to predict the spot price. In total there are 26 features in the training dataset. The created network uses the spot price and other features at timestep *t-1* to predict the spot price at timestep *t*. The network is a simple network with a single hidden layer, which is an LSTM layer that is fed the input directly and that connects to a fully-connected layer. The fully-connected layer then gives the spot price value as output. The model uses the last 48 timesteps, which means that the input shape will be 48 rows long with 26 columns. The model provides good results, as can be seen in Figure 3, however it is not suited well for spot price forecasting, since the buy and sell orders are sent the day before, which means that predicting one hour ahead has no real-world application. This model should be modified to make predictions more than *t+1* timesteps ahead. Hyperparameter optimization was conducted in a brute force way of simply trying different combinations of hyperparameters and calculating RMSE and MAE values to determine the best hyperparameters. The tested hyperparameters are listed in Table 1.

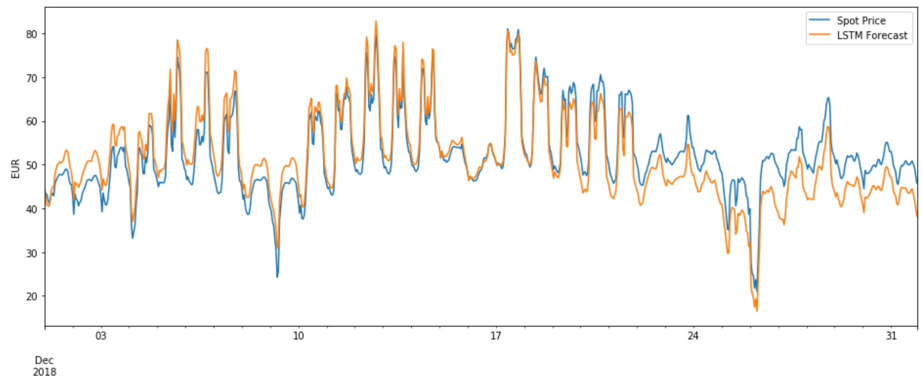

Fig. 3. Onestep LSTM forecasts (yellow) and spot prices (blue)

Table 1. LSTM hyperparameters

Hyperparameters	Values
Epochs	20,50
Neurons	10,20,40,60,80
Optimizers	Adam, RMSProp, SGD
Learning Rates	0.001, 0.0005, 0.0001
Batch Sized	1,4,12,24,48,168

4.4 Encoder-decoder LSTM

In order to forecast more than one hour ahead of the current time, the model needs to have a multistep output. In other words, the previous model was made to predict, the output was a single float number, the output needs to be modified to be a sequence of numbers that would be the forecast horizon. The encoder-decoder LSTM contains two sub-models, the encoder and decoder. The encoder creates an internal representation of the input sequence, which the decoder will then use to predict a sequence. This is called a sequence-to-sequence model which has been used to automate translations between languages. The input and output timesteps that the model is trained on can be modified *i.e.* the model can take as input a week's worth of data or only a day's worth of data and forecast either the next day or the next week spot prices. This was used to change the forecast horizon of the model. The network has two hidden layers and utilizes a repeat vector to repeat the input between two LSTM layers. The CuDNNLSTM is an optimized version of the regular LSTM layer, which can be only trained with Nvidia's graphics processing units (GPU). In the final version, two more hidden LSTM layers were added. The network training time increased slightly, but test results are better. Input weight regularization and low learning rate of 0.00001 were also found to reduce loss and improve forecast accuracy. Learning rates of 0.01 or higher made the model ignore daily seasonality and draw flat lines across the forecast horizon as a result. This problem also appeared when the forecast horizon was a week or longer. The hyperparameters that achieved the best results are listed below in Table 2.

Table 2. Best LSTM Encoder-decoder training hyperparameters

Hyperparameters	**Values**
Optimizer	Adam
Learning rate	0.00001
cu_dnnlstm_2 layer	100
cu_dnnlstm_3 layer	100
dense_2 layer	100
dense_3 layer	100
Epochs	50
Batch size	12

4.5 CNN-LSTM

The previous LSTM Encoder-decoder was changed to include convolution layers. This was achieved by adding two one-dimensional convolution layers and using one-dimensional max pooling, as visualized in Figure 4. The output from the convolutional layers is flattened and sent to the LSTM layer. This architecture is similar to the encoder-decoder, in this case the encoder is the convolutional neural network (CNN) and the decoder is the LSTM.

4.6 Ensemble learning

Ensemble learning was implemented by creating a stacking model, where predictions of multiple models were combined. The best combination method was found to be the metalearner, where the neural network was a fully-connected neural network with two hidden layers. The models to combine where selected from previous test results. For a 24-hour forecast, the best result was achieved by combining the forecasts of the LSTM network, ARe model and Prophet. A 36-hour forecast was also created to test how the performance compared to a 24-hour forecast. In the 36-hour ensemble model, three LSTM models were combined with AR and prophet model forecasts. In the 48-hour forecast, a combination of four LSTM models achieved the best results.

4.7 Forecast accuracy evaluation

The accuracy of forecasting methods and trained forecasting models will be measured on a test dataset. The test dataset will be separate from the training

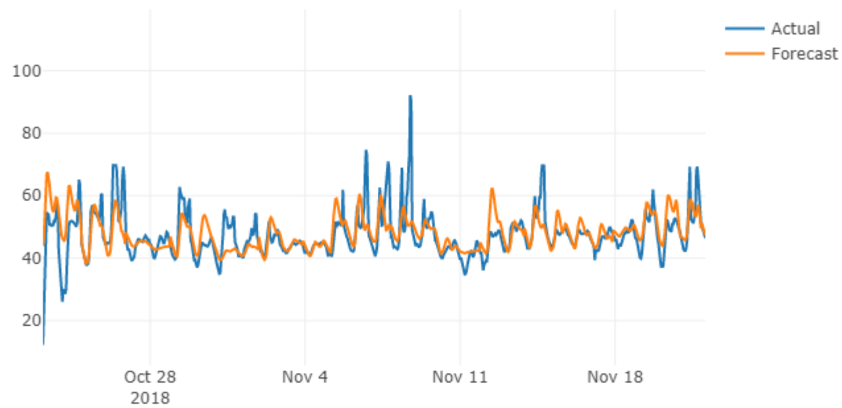

Fig. 4. CNN-LSTM results with forecast horizon of 48

data and will be a continuum of the training data timewise. The test dataset will be ten weeks of data before the end of the year 2018. This means that the test data is from October 22 to December 31 as shown in Figure 5.

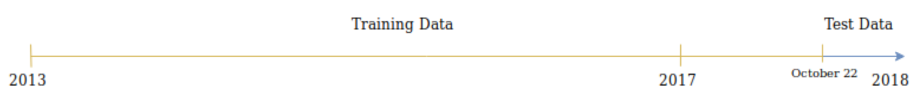

Fig. 5. Training and test data split

5 Results

The best forecasts were, depending on the forecast horizon, usually achieved with neural networks or the ensemble model, where the forecasts of models' autoregression, Prophet and multiple LSTMs were combined, and the forecasts were given weights with a separate metalearner neural network. After many rounds of feature engineering, the dataset that was constructed had the following features:

- Cyclical hour, day, weekday and month.
- Electricity consumption in Finland and Sweden
- Emission allowances price
- Hydro power production in Sweden
- Hydro reservoir levels in Norway and Sweden
- Nuclear power production in Finland and Sweden
- Precipitation near Harsprånget
- Prophet's daily, weekly and yearly seasonality
- Spot price, moving average and standard deviation from X hours ago, where X is the forecast horizon
- Temperature in Finland and Sweden
- Wind power production in Sweden

The best results from all the forecasting models were documented and written on the Tables 3 - 5. The lowest errors per error metric are bolded. The results from all the 24 hours ahead forecast models were written down on the Table 3. The best performing models were autoregressive, LSTM, CNN-LSTM and ensemble, where their error values were quite close to each other.

Table 3. 24 hours ahead forecast errors.

Model Name	MAE	MAPE	RMSE
AR	4.90	**9.04**	8.10
Prophet	6.84	9.15	13.99
LSTM	**4.80**	9.17	7.45
CNN-LSTM	5.40	9.98	7.99
Ensemble	4.96	9.41	**7.12**

Table 4 shows the results for models with forecast horizon 36. The encoder-decoder LSTM is the best model in this horizon on almost all metrics. The CNN-LSTM and ensemble were the second-best models and had almost similar error values.

Table 4. 36 hours ahead forecast errors.

Model Name	MAE	MAPE	RMSE
AR	6.39	11.88	9.53
Prophet	6.84	**9.15**	13.99
LSTM	**5.08**	9.57	**8.08**
CNN-LSTM	5.41	10.10	8.40
Ensemble	5.30	10.30	8.24

The results with forecast horizon 48 were recorded in Table 5. In this case the CNN-LSTM model was the best based on MAE and RMSE metrics. The autoregressive, LSTM and ensemble models share the second place with mostly similar error values.

To summarize, the best 24 hours ahead forecast was made by the LSTM model, where the forecast differs ±4.80 euros from the spot price on average. The best 36 hours ahead forecast was performed by LSTM again, where the forecast error is ±5.08 euros. The best 48 hours ahead was achieved by the CNN-LSTM model, where forecasts were ±5.0 euros different from spot prices on average. The best one-week ahead forecasts were achieved by the ensemble model, where the forecasts were off by ±5.08 euros on average.

Table 5. 48 hours ahead forecast errors.

Model Name	**MAE**	**MAPE**	**RMSE**
AR	5.57	10.22	9.24
Prophet	6.84	**9.15**	13.99
LSTM	5.41	10.04	8.17
CNN-LSTM	**5.08**	9.76	**7.77**
Ensemble	5.44	10.25	8.14

6 Conclusion

The aim of the paper was to present the developed the machine learning and deep learning based forecasting models that predict Spot prices in Nord Pool's Day-ahead market in Finland with open-source software. The measurements show that the depending on the forecast horizon the neural networks or the ensemble model, where the forecasts of models' autoregression, Prophet and multiple LSTMs were combined, and the forecasts were given weights with a separate metalearner neural network. The developed machine learning model is able to forecast the trend and seasonality of Spot prices but unable to predict sudden price spikes. The price spikes may happen for a variety of reasons such as a breakdown of a transformer. The best course of action would be to design a separate model to predict price spikes and connect real time event observer. This price spike prediction model could then be combined with current models to improve performance. Further feature engineering and optimizing the used neural networks or switching to a new architecture could improve model accuracy and lower forecast error. When using sequence that were the length of a week or longer, the neural networks predictions would flatten to an almost even line. This happened because the sequence was too long and instead of fluctuating daily, the neural network thinks the best way to reduce loss is to just draw a flat line. This could be fixed by allowing length of input sequence and output sequence to be different. Future improvements will contains the data from machine failure due to the sudden peaks can be recognized.

References

1. Bayesian deep learning based method for probabilistic forecast of day-ahead electricity prices. Applied Energy **250**, 1158 – 1175 (2019)
2. Aggarwal, S., Saini, L., Kumar, A.: Electricity price forecasting in deregulated markets: A review and evaluation. International Journal of Electrical Power aand Energy Systems **31**(1), 13 – 22 (2009)
3. Amor, S., Boubaker, H., Belkacem, L.: Forecasting electricity spot price for nord pool market with a hybrid k-factor garma–llwnn model. Journal of Forecasting **37**(8), 832–851 (2018)
4. Beigaitė, R.: Electricity price forecasting for nord pool data using recurrent neural networks (2018)
5. Beigaite, R., krilavičius, T., Man, K.: Electricity price forecasting for nord pool data. pp. 1–6 (01 2018)
6. Conejo, A., Contreras, J., Espínola, R., Plazas, M.: Forecasting electricity prices for a day-ahead pool-based electric energy market. International Journal of Forecasting **21**, 435–462 (07 2005)
7. Karabiber, O.A., Xydis, G.: Electricity Price Forecasting in the Danish Day-Ahead Market Using the TBATS, ANN and ARIMA Methods. Energies **12**(5), 1–29 (March 2019)
8. Knapik, O.: Modeling and forecasting electricity price jumps in the nord poolpower market. Tech. rep., Aarhus University, Department of Economics and Business Economics,, Aarhus V, Denmark (2017)
9. Li, G., Lawarree, J., Liu, C.C.: State-of-the-art of electricity price forecasting in a grid environment (05 2010)
10. Proietti, T., Haldrup, N., Knapik, O.: Spikes and memory in (nord pool) electricity price spot prices. Tech. rep., Tor Vergata University, CEIS, Rome, Italy (2017)
11. Su, M., Zhang, Z., Zhu, Y., Donglan, Z., Wen, W.: Data driven natural gas spot price prediction models using machine learning methods. Energies **12**, 1680 (05 2019)
12. Tan, Z., Zhang, J., Wang, J., Xu, J.: Day-ahead electricity price forecasting using wavelet transform combined with arima and garch models. Applied Energy **87**(11), 3606 – 3610 (2010)
13. Taylor, S., Letham, B.: Forecasting at scale. The American Statistician **72** (09 2017)
14. Voronin, S.: Price spike forecasting in a competitive day-ahead energy market. Ph.D. thesis, LUT School of Technology , Lappeenranta University of Technology (2013)
15. Wang, K., Xu, C., Zhang, Y., Guo, S., Zomaya, A.Y.: Robust big data analytics for electricity price forecasting in the smart grid. IEEE Transactions on Big Data **5**(1), 34–45 (March 2019)