# DNN model of Electricity Price Forecasting


### Capstone Project Proposal - Margaret Toohey

*Udacity Machine Learning Nanodegree*


## Domain Background

Electricity price forecasting (EPF) is concerned with the prediction of the price of wholesale electricity, for a region or energy market. Like any commodity, accurate price prediction is crucial for energy companies and suppliers to be able to bid into and participate in the energy market. Unlike other commodities, the stability of power systems is bound by a constant balance between production and consumption, and excess electricity on the system cannot be stored. Another unique feature of the electricity market is the high variability in the demand, based on weather (temperature, wind-speed, sunlight), time of the day (on-peak vs. off-peak) and more long term seasonal effects (winter, summer, weekends, holidays). [1] The variability caused by these factors leads to a market unlike other commodities markets, exhibiting both seasonal trends and unanticipated price spikes or drops.

Another major factor in the EPF is the amount of Renewable Energy Sources (RES) that are available to the market. RES (e.g. Wind, solar and hydro etc. electricity sources) have an impact on the price of electricity, and generally the generation of RES leads to a decrease in the wholesale price of electricity, since the RES marginal costs are close to zero. The drawback, however, is that the amount of RES available to the power system is not constant, and is variable over a number of factors, such as weather. [2]

This is where EPF becomes a societal problem, and where my personal interest comes from – accurate electricity price prediction may lead to more RES projects participating in the market, leading to cleaner and more sustainable energy production, and ultimately lower the cost of energy for the consumer.

Traditionally, statistical methods have been used for EPF, however in recent years, Machine Learning (ML) methods have become more popular. Time series and regression models have been used with good effect, but due to the complexity of the markets, predictions from these models does not give accurate long term EPF. Due to the complexity and the large feature space of the markets, Deep Neural Networks may be the key to accurate medium to long term EPF. This is the basis of my project proposal.

## Problem Statement

The goal of this project is to create a DNN model, based on historical price data and other factors, which can accurately predict electricity prices for a given market, which could be used by energy companies for day-ahead price forecasting and hence successful bidding into an energy market.

# Datasets and Inputs

The dataset I propose to use to train, tune and test my forecasting model is historical price data from the Scandinavian electricity market, NordPool. As mentioned above, electricity prices are effected by exogenous variables*, such as the RES generation and the demand on the power system. As such, two additional time series will be used in the dataset; the day-ahead load forecast and the day-ahead wind generation. In order to correctly model the various time variability such as daily (on-peak vs off-peak), weekly (weekday vs weekend) and seasonal (winter vs summer, holidays, etc) changes, 6 years of data, from 01.01.2013 to 24.12.2018, in hourly increments will be used. The model will be split into 3 sections; the first section (containing two years of data) will be used for training. The next section (containing 2 years of data) will be used to tune the hyperparameters. The final two year section will be used to test the accuracy of the model. This data is freely available from NordPool. [3]

The input features from the NordPool data includes the following:

- Historical day-ahead prices of the previous three days and one week ago (i.e. $p_{d-1}$ $p_{d-2}$ $p_{d-3}$ $p_{d-7}$)
- The day ahead forecasts of the load and the wind generation
- Historical day-ahead forecasts of the load and the wind generation from the previous day and one week ago
- A dummy variable to represent the day of the week.

This results in a total of 241 input features for the model.

*Exogenous variables are those whose value is determined outside of the model, and are imposed on the model. For example, in a simple supply and demand model, a change in consumer taste will lead to an exogenous change in demand, which will impose a change in price and quantity transacted. [4]

# Solution Statement

I propose to solve the EPF problem described above by:

1. Download and pre-process the historical price data from an energy market (I will use data from the Scandinavian market, NordPool)
2. Train a DNN to predict the wholesale electricity price using the historical price data.
3. Tune the hyperparameters and optimize the features of the model using a validation dataset from the same market but a different time period.
4. Test the accuracy of the model using a test dataset, ensuring data that has never been seen by the model before is used for testing. The Mean Absolute Percentage Error (MAPE) and the Mean Average Error (MAE) metrics will be used to determine the accuracy of the model.
5. The final step of the project will be to compare the accuracy of my model with a State of the Art industry standard, from this EPF toolbox.

# Benchmark Model

The model I will use to benchmark mine will be an open-source DNN model created specifically for the purpose of benchmarking EPF models, with the aim of standardising the way ML algorithm performance is measured in the context of EPF. [5] The benchmark model is provided by the authors as an open source python library, [6] specifically designed to provide a common research platform for EPF researchers.

# Evaluation Metrics

I will use the Mean Average Percentage Error (MAPE) and the relative Mean Absolute Error (rMAE) and the Root Mean Squared Error (RMSE) to calculate the accuracy of my model. MAPE, MAE and RMSE are commonly used to quantify the error of forecasting models [7].

$$\mathrm{MAPE} = \frac{1}{24\,N_\mathrm{d}} \sum_{d=1}^{N_\mathrm{d}} \sum_{h=1}^{24} \frac{|p_{d,h} - \hat{p}_{d,h}|}{|p_{d,h}|},$$

$$\mathrm{rMAE} = \frac{\frac{1}{24N_\mathrm{d}} \sum_{d=1}^{N_\mathrm{d}} \sum_{h=1}^{24} |p_{d,h} - \hat{p}_{d,h}|}{\frac{1}{24N_\mathrm{d}} \sum_{d=1}^{N_\mathrm{d}} \sum_{h=1}^{24} |p_{d,h} - \hat{p}_{d,h}^{\mathrm{naive}}|},$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{24\,N_\mathrm{d}} \sum_{d=1}^{N_\mathrm{d}} \sum_{h=1}^{24} (p_{d,h} - \hat{p}_{d,h})^2},$$

Where $p_{d,h}$and $\hat{p}_{d,h}$are the real price and the forecasted price, respectively, on day *d* and hour h, and $N_d$ is the number of days in the dataset.

# Project Design

The feature space of these data sets is quite large, as mentioned above, so I propose using NN as they can deal with feature correlation themselves. DNNs have shown good results for EPF in the literature, so I propose using this method for my prediction model.

The main components needed for this solution are data processing, a training function, a model that can be trained and then deployed via a predict method, that can be used to pass in test data to the trained model.

The first steps will be to make sure the data is consistent, delete any empty rows or columns from the sets, and the combine each of the different forecasts (dah ahead price, load and wind generation for all time frames) into one pandas Dataframe. The training, validation and test datasets must be split here (the last 2 years of data will be used for testing) and the target values (the actual price) will be separated from the train and validation test sets. Once this is complete, the data will be loaded to S3.

Next the model must be defined. The basic steps for an NN model are:

- Define the network, based on the chosen parameters, e.g. number of input features, hidden layers and the output dimension, etc.
- Apply the linear transformations, and apply dropout if necessary.
- Define the feed forward behaviour of the network, output of the layers, activation function etc.

This model will be called during training, and the training data will be passed in, along with an error or loss function (i.e. the method by which we define how our model is performing, compared to the target value). The next step during training is to perform backpropagation of the error.

Once we have a training method, an estimator object can be created. The estimator will be trained and then a validation dataset will be used for hyperparameter tuning.

Then the model will be deployed to an endpoint in order to create a predictor. Finally the model will be evaluated using the test data and the accuracy metrics described above.

# References


[1] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future,," *International Journal of Forecasting,* vol. 30, no. 4, pp. 1030-1081, 2014.

[2] F. N. Luigi Grossi, "Robust forecasting of electricity prices: Simulations, models and the impact of renewable sources,," *Technological Forecasting and Social Change,* pp. 305-318, 2019.

[3] "Historical Market Data," 2020. [Online]. Available: https://www.nordpoolgroup.com/historical-market-data/. [Accessed 18 11 2020].

[4] N. G. Mankiw, Macroeconomics, third edition., Worth, 1997.

[5] G. M. D. S. R. W. Jesus Lagoa, "Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark," *Renewable and Sustainable Energy Reviews,* 2020.

[6] "EPF Toolbox," Github, 20 08 2020. [Online]. Available: https://github.com/jeslago/epftoolbox. [Accessed 18 11 2020].

[7] Y. W. Shuman Luo, "A two-stage supervised learning approach for electricity price forecasting by leveraging different data sources,," *Applied Energy,* vol. 42, pp. 1497-1512, 2019.