Energy Analytics - Assignment 2

Lukas Raad - S204701
Tais Abrahamsson - S193900
Filipe Branco - S226424

April 2023

Workload	**Task 1**	**Task 2**	**Task 3**	**Task 4**
Coding	All	All	All	All
Writing	All	All	All	All
Editing	All	All	All	All

Task 1: ARIMA Model

We begin by splitting the data into a training data set and test data set. We produce the following plot.

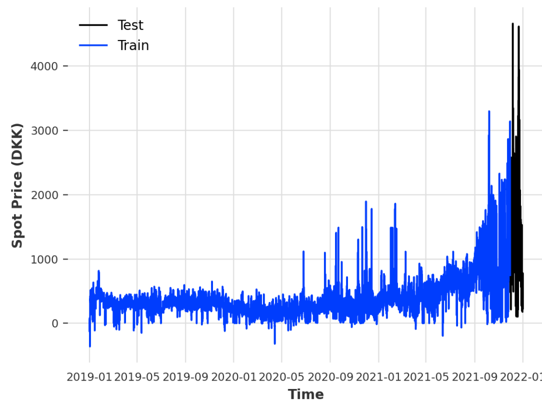


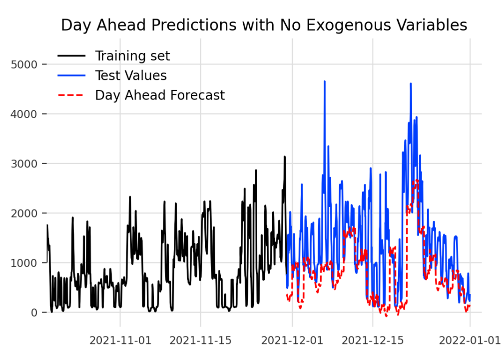
Figure 1: Price data vs. Hour in UTC split between train and test data

1.1 ARIMA model with no exogenous variables

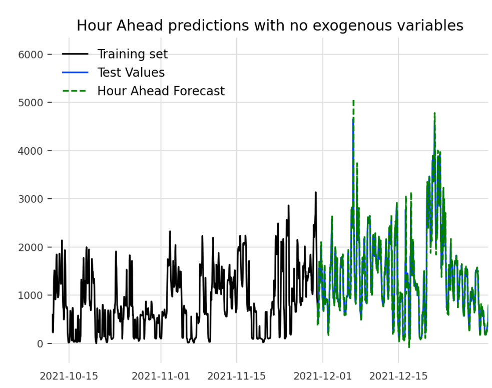
For predicting the electricity prices we can begin with an ARIMA model, with no exogenous variables. In order for the training of the model to be more optimized we also used a fourier feauturizer while assuming a periodicity of 24 hours for the price data with $k = 3$ cosine and sine terms. We tried several values starting with $k = 12$ and reduced the number to find the one that allowed for the fastest run time but also the best best prediction.

In order to save time, we only run the fit on first 1000 values of the training data. This does not affect the resulting model, and cuts down on run time.

First off, we perform predictions day ahead predictions, and get the resulting plot



(a) Day Ahead Predictions with Seasonal ARIMA Model



(b) Hour ahead predictions with Seasonl ARIMA

The day ahead predictions result in a mean squared error of 986.36 and the hour ahead predictions 255.68.

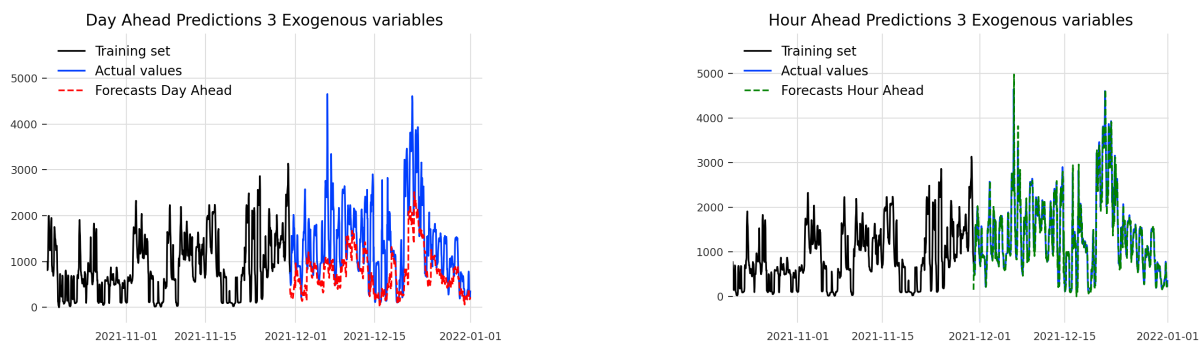
1.2 Adding exogenous variables (SARIMAX)

We can further improve this model by adding exogenous variables in order to train a SARIMAX model. We can add the Exogenous variables

- CentralPowerMWh
- LocalPowerMWh
- OffshoreWindGe100MW_MWh

We first tried SolarPowerLt10kWMWh, OffshoreWindGe100MWMWh, and GrossConsumptionMWh. We found that while these do improve the model slightly, we figured that maybe Central Power and LocalPower would have a stronger effect on electricity prices and we did find that these exogenous resulted in lower RMSE's.

Plotting the predictions, we get



(a) Day ahead predictions

(b) Hour ahead predictions

Figure 3: Predictions using 3 exogenous variables with SARIMAX model

The Day ahead model has a root mean squared error of 903.41 and the hour ahead has 253.97. As we can see this is an improvement on the model with no exogenous variables.

One way of verifying how good the model is would to compare with the daily and weekly persistences. We compute the weekly and daily persistence for the training data.

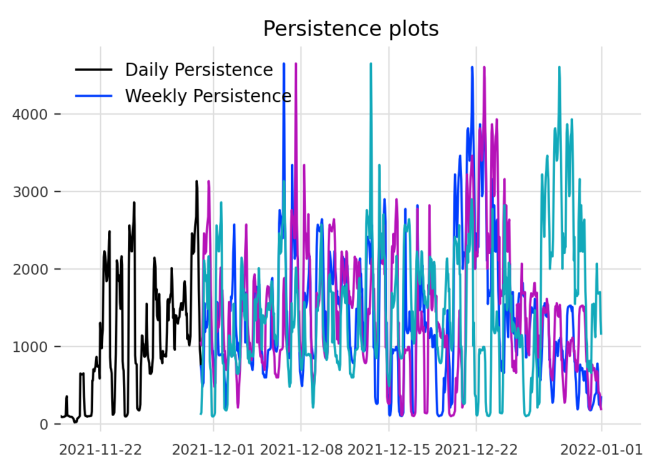


Figure 4: Weekly and Daily Persistence of the training data

The persistence simply assumes that the data would continue like its last step. Therefore, in order to check if our model is good the mean squared error of the persistence with the test data should be bigger than the MSE of the model. Calculating the RMSE for the persistence. We compare the daily persistence with the test data and get a value of 777.56. Comparing with the weekly persistence we get 1084.74. Both these values are bigger than the RMSE between the model and the test data. Therefore, with this information and observing the graphs and the RMSE values, we can assume that our model is good.

Task 2: Temporal Convolutional Neural Network

The goal for this task was to develop a temporal Convolutional Neural Network in order to predict electricity spot prices as accurately as possible, and compare its performance with the ARIMA model from Task 1. We chose to predict the prices for the DK2 area only.

2.1 Hour Ahead Forecast

In order to properly compare both models, we used the same exogenous variables that we thought would yield the best forecasts throughout for the tCNN: CentralPowerMWh, LocalPowerMWh, OffshoreWindGe100MWMWh. For the hyperparameters, it was a matter of trial and error. We started with a small kernel size and slowly incremented it by 1, changing all the other hyperparameters around it(keeping the diation base below the kernel size). Furthermore, in order to avoid overfitting and still allowing the network to learn useful patterns from the training data, we decided to set the dropout at 10%. We ended up using the following hyperparameters, as they yielded the lowest RMSE score from the trials we ran:

Input Chunk Length = 85
Output Chunk Length = 24
Number of Epochs = 15
Dropout = 10%
Dilation Base = 2
Weight Norm = $True$
Kernel Size = 6
Number of Filters = 4
Number of Epochs for the Validation Period = 1
Random State = 0

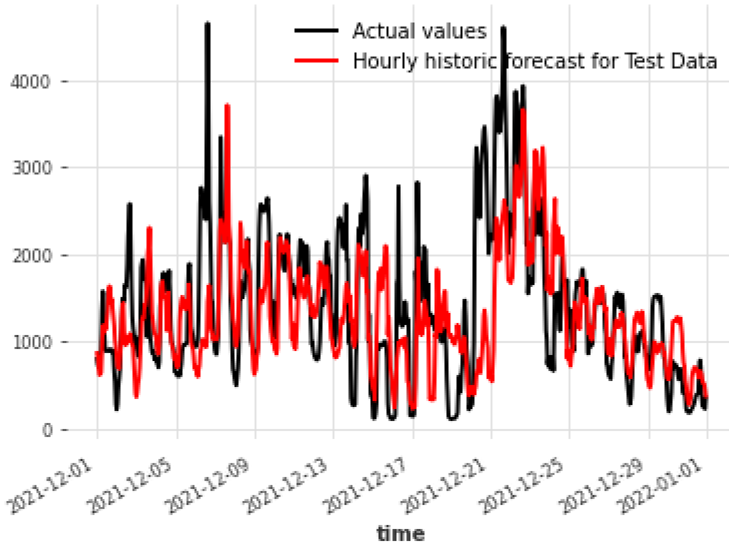


Figure 5: Hour-Ahead Prediction for tCNN

The Hour-Ahead prediction model resulted in an RMSE of 693.32, compared to the daily persistence RMSE of 777.56, a 12.16% decrease from the daily persistence error calculated in Task 1. This shouldn't be surprising, as tCNNs are much better at capturing complex interactions between different time series than persistence models, especially when three exogenous variables are introduced. In addition, whereas in Task 1 we ended up not using all of the training data available due to runtime efficiency, this was not a problem for the tCNN model which was easily able to handle the large-scale dataset it was given.

2.2 Day Ahead Forecast

Using the same exogeneous variables and parameters as was used in the hourly prediction, we ran the tCNN with a Forecast Horizon of 24 hours instead of 1 which yielded the following prediction:

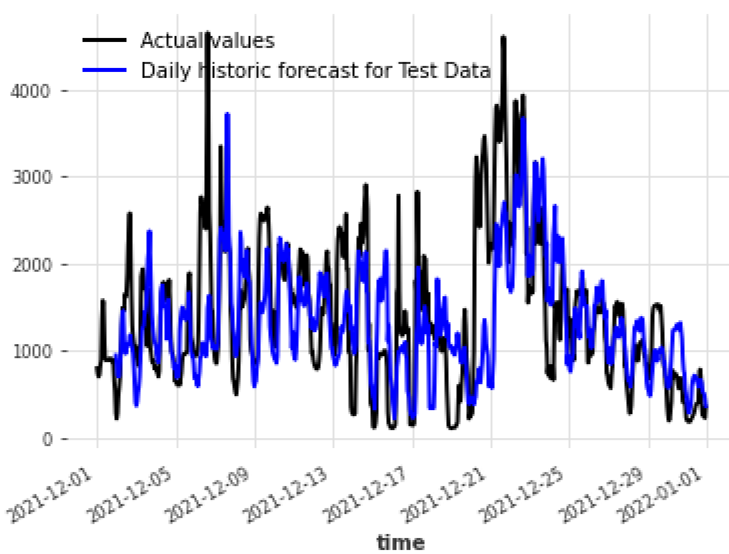


Figure 6: Day-Ahead Prediction for tCNN

The RMSE value that was generated for the Day-Ahead tCNN prediction model was 693.01, as compared to the weekly persistence RMSE of 1084.74, this time a 56.54% decrease from the persistence RMSE calculated in Task 1. This larger RMSE difference between the two models is a testament to how the persistence model performs considerably worse when it comes to forecasting larger intervals of data, which are more susceiptable to fluctuations and noise. On the other hand, both the hourly and daily tCNN forecasts maintained a constant RMSE of 693 throughout, proving it to be a more reliable and consistent forecasting method.

Task 3: Performance of ARIMA and tCNN models

3.3 Comparing the two models

In terms of accuracy, for the hour and day ahead predictions. We found that in the case of the ARIMAX model, the day ahead predictions were worse than the hour ahead predictions. With an RMSE of 986 for the day ahead predictions and 256 for the hour ahead. However in the case of the tCNN, the hour and day ahead yielded similar results with an identical RMSE of 630. This makes the model

For the ARIMA model, it took around 3 minutes to train the model with the full training set using a k-value of 3. For the tCNN, the run time of training the model varied based on the number of epochs, but it took 4 minutes and 23 seconds to train the full data set with an epoch number of 15. Therefore, we do notice a difference between the two models' computational times, with the tCNN taking longer due to its larger number of parameters, and to its use of more extensive training.

The tCNN uses ten hyperparameters: Input Chunk Length, Output Chunk Length, Number of Epochs, Dropout, Dilation Base, Weight Norm, Kernel Size, Number of Filters, Number of Epochs for the Validation Period, Random State. The ARIMA, however, only uses two: M and K, seasonality and number of sines and cosines for the Fourier Featurizer. Out of the two models we found the ARIMA process easier to train simply because there were less parameters and considerations to worry about.

3.4 Recommendations

Given the previous considerations, I would recommend the ARIMAX model. This one has faster run times, is simpler to use and provides better results with hour ahead predictions. It only provides worse predictions in the case of day-ahead predictions. However, I don't think this out weighs the far more accurate hour ahead predictions, 630 for tCNN and 253 for the ARIMAX.

Task 4: Using forecasting model for bidding