# Electricity Price and Intraday Spread Forecasting

Rui Ding

May 1, 2019

## 1 Introduction

We are interested in the forecasting of prices and price volatility, i.e. intraday price spread, of electricity in the Day Ahead Market. Our primary data sources are from CAISO oasis platform where hourly Local Marginal Prices (including subsidiaries such as congestion, loss, and energy) and hub level load data are available. We are also interested in other fundamental variables such as fuel prices and temperature. We primarily focus on statistical and computational methods for point forecast purposes.

## 2 Literature Review

Both the original price time series, on a nodal level, and the computed intraday price spread time series are sequence of data that follows a normal trend plus occasional spikes. Hence it is reasonable to model the normal price part with a time series model such as ARIMAX and the spike part with other forecasting methods. Most of the existing approaches to forecast electricity prices are reasonably effective for normal prices but they cannot deal with the price spikes accurately. GARCH process has been tested to simulate price spikes in original price series but it has not been able to incorporate spikes with a height usually observed. To this end their are several methods proposed to deal with the price spike components.

At the very basics we reviewed the various models proposed in Electricity price forecasting: A review of the state-of-the-art with a look into the future, Rafal Weron 2014. While our goal certainly include understanding the market driving factors, we care for point forecasts more since that would optimize our revenue. Hence we skip the discussion of multi-agent models, fundamental models, and reduce-form models which deal with the underlying nature of price movement more than the predicting power of the model. Common statistical models reviewed by this paper include AR models, ARMA models, ARIMA models, ARMAX models, GARCH models, etc. We focus on ARMAX and ARIMAX models which include exogenous variables that we believe are driving factors such as temperature and system demand (load). To deal with nonlinearity and complexity we rely on computational intelligence models such as ANN and RNN, particularly LSTM models.

Price Forecasting in the Day-Ahead Energy Market by an Iterative Method with Separate Normal Price and Price Spike Frameworks, Voronin and Partanen 2013 proposed an iterative hybrid method to separately predict the normal prices and price spikes in the Finnish day-ahead market, which can be extended to the CAISO market. In their framework, a wavelet transformation is first applied to decompose the price series into less volatile components. Then an ARIMA model is used to capture the cyclicality of the series, which is combined with a neural network to produce normal price forecasts by capturing both the linear and nonlinear patterns between target and exogenous variables. For price spike value prediction a k-NN is applied. While the wavelet transform can be substituted with simple differencing techniques to reach a stationary series. The key part in normal price forecasting is a seasonal ARIMA model. To capture a linear trend and seasonality in the time series, one-hour and 24-hour differencing is used. A compound classifier with relevance vector machine, decision trees, and probabilistic neural nets are used to make a single classifier for price spike occurrence predictions. A probability threshold is implemented to decide if a spike will occur at the current data point which is trick used to compensate for the scarcity of spike events. A kNN is used to predict the actual spike value once a spike is predicted to occur. A very important definition made in this paper is the electricity price spike, which is defined as the price surpassing a specified threshold calculated as the mean plus three times the standard deviation. The proposed method used adapted mean average percentage error to evaluate the forecast results and it also cared for the spike prediction accuracy separately. This method generally outperforms competing forecasting models due to its ability to separately predict price spikes. Exogenous variables can be incorporated into the feature selection process easily. A potential issue could be the running time of the computations.

Our data has a embedded network structure such that the node being predicted has a neighborhood of nodes close to it. We would like to have a model that forecasts the prices for all these nodes together.In Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms, Lago, Ridder, and Schutter 2018, the authors proposed a hubrid DNN-LSTM network to simulatneously forecast day-ahead prices in several countries. A CNN model is also proposed using max pooling. It is interesting to note that based on their results DNN actually outperforms LSTM and CNN models, who have higher computational powers in some sense. The reason might be that DNN has the fewest number of parameters so that it requires less data to train properly. Depending on the amount of data we have, DNN might be better in terms of the testing performance. DNN also makes no assumption about the data input hence allowing for any kind of correlation between lagged values.

We assume that similar models can be applied directly to the price spread time series with slight modifications since the spike behavior is more significant in that data.

# 3 Data Description

Our primary data source is CAISO, which provides different frequencies of data on various variables of interest. The main data set is the LMP hourly data set which records hourly local marginal prices for about 6000 different nodes in California region in the recent years. We have loaded the data from March 2017 to March 2019. The LMP comes with detailed components which are congestion, loss, and energy. On the other side, we have system demand forecast for different regions as well as load for different regions. We will need to correlate the nodes with these regions based on documentations or longitude latitude information. These data come in directly from the CAISO API and there are no missing values for our purposes. We could also do the analysis using the 15 minute level data of the same categories but the data scraping process would take too long so we stick with hourly data for now. We are also looking for fuel price data such as gas prices and local temperature data.

# 4 Proposed Method

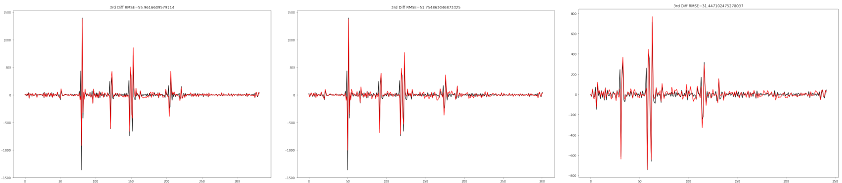
Our preliminary model is a naive statistical model called the ARIMAX model. Based on our observation, the price time series mostly correspond to one with a unit root so we perform a first difference before fitting an ARMA model. Specific order parameters are to be tuned. We add exogenous variables and their lagged values to the feature set as well. We would like to have a single model for all the nodes so the training data would be a stack of vectors that are taken from all the nodal time series slices.

Moving forward, we would like to implement a model that separately predicts price spike magnitude and occurrences. This would be adding another model on top of the ARIMA model which is used to predict normal prices. A potentially different feature set would be used for a kNN classification and prediction for spikes.

Our ultimate goal is to employ a deep learning framework that incorporates nonlinear relationships as well.

# 5 Preliminary Results

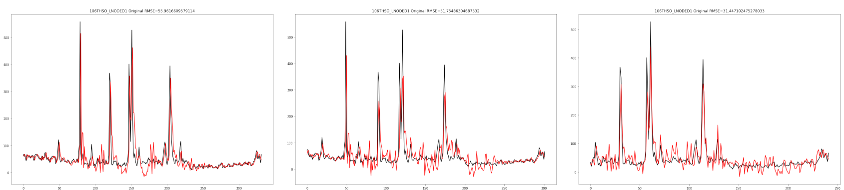
We start with models that use the price spread series only due to uncertainty about the exogenous variables. Hence, we are building a purely statistical model on the stacked data set consisting of about 6000 nodes. The data tends to exhibit spike behaviors and the data distribution doesn't vary much across nodes so we can process them as a since data source. Upon trying different models we found that a third order differencing on the original series would yield the best results. Based on our goal to train a unified model for all nodes we selected an order p for the AR model and preprocessed all our data into training and testing sets. We tested p equals 30, 60, and 120 yielding R square values of 0.85, 0.88, and 0.91. The third difference predictions are shown below for the three models:



(a) p=30 (b) p=60 (c) p=120

Figure 1: AR models on third differenced series with different order parameters

A single node is used to demonstrate the prediction performance which is shown below.



(a) p=30 (b) p=60 (c) p=120

Figure 2: ARIMA(p,3,0) models with different order parameters

We believe that identifying spike behaviors is crucial to our analysis so we define a threshold $\mu + 3 * \sigma$ where $\mu$ is the time series mean and $\sigma$ is the standard deviation. For example, in the figure below we demonstrate how spikes are identified for a particular node.

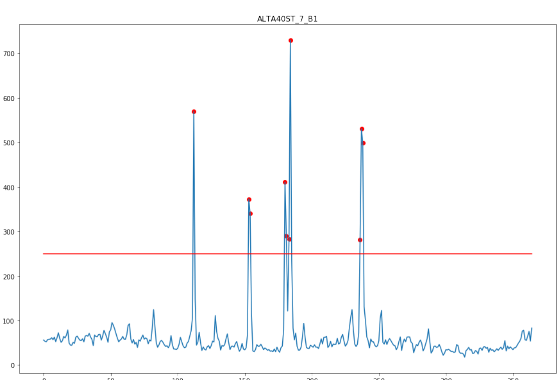
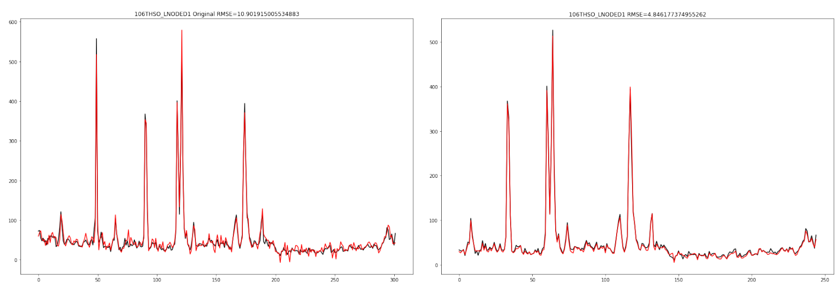



Figure 3: Spikes

## 6 Improved Results on Price Spread Forecasting

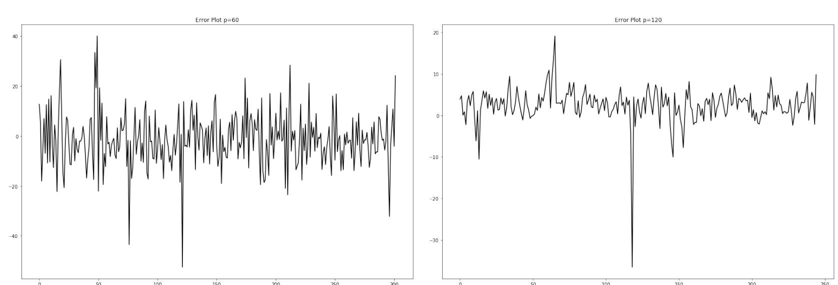
Our primary focus is on the price spread forecast so we employ a more powerful model, the neural network architectures, to predict the intraday price spread time series values. More specifically, we process the original price spread data with a past window size taken as the input features, but now we are predicting the next day point forecast using a fully connect network with hidden layers that we can modify.

The input feature length are selected to be 60 or 120 as we compare their performances. The first hidden layer has 12 cells and the second has 8 cells, both with ReLu activation functions. The output is a single cell which predicts the next day price spread. We train the network with 15 epochs minimizing the mean squared error loss using the Adam optimizer. As the result, the model shows significant improvement over the ARIMA models we used previously in terms of the RMSE metric and the percentage error metric.




(a) p=60 (b) p=120

Figure 4: ANN spread forecasts with different feature length parameters

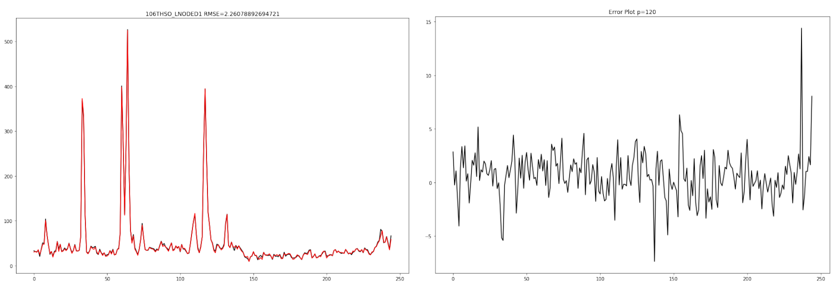



(a) p=60 (b) p=120

Figure 5: ANN models error plot

We can see from the test RMSE results that the higher order model gives a better result with a RMSE of only 4.8 which corresponds to about 10 percent in terms of percentage error of the price spread series. This is much better than the RMSE from ARIMA models which is usually around 50 to 60. The error plots are also shown above.

Upon modifying the network structure we can further reduce this error. A change in the number of hidden cells to 30 and 10 gives an RMSE of only 2.26 which corresponds to only 4.6 percent error of the test series.




(a) p=120 Forecasts

(b) p=120 Errors

Figure 6: ANN models refined

We tested the same model on pure price predictions as well and found it to suit that purpose as well. We trained a NN with p=60 on a selected portion of the entire dataset due to huge data size.The testing result shows an RMSE of 5.95 which correponds to about 17 percent error of the original electricity price series in the test node.

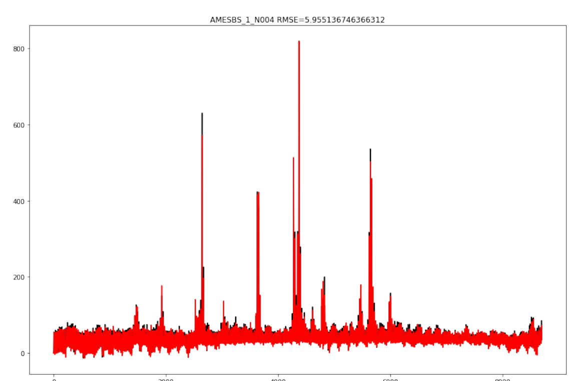
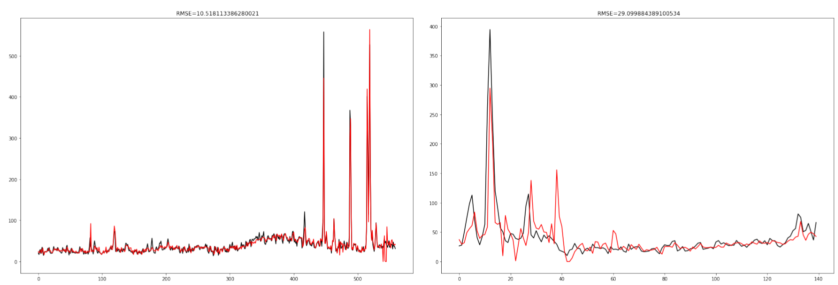



Figure 7: Price Forecasting with ANN

## 6.1 Problems with Out-of-sample Testing and Improved Results

As we load more data from past years we began to realize that the previously trained neural network doesn't work well on truly out-of-sample datasets from different time horizons. One problem is that the training and testing data we had from 2017-2018 year corresponds to a split between the thousands of nodes but not the times. In fact, although different nodes have different prices and hence spreads, they generally conform to a specific shape and thus they are highly correlated. Using some nodes' data to train the network and testing on other nodes resolve to an overfitting issue and the network, when trained properly and well enough, already absorbs all the needed information and will predict very accurately on the testing set. To avoid such issue, we used a longer dataset from 2016 to 2018. We separated the training set and testing set on the time horizon and performed another neural network learning model.



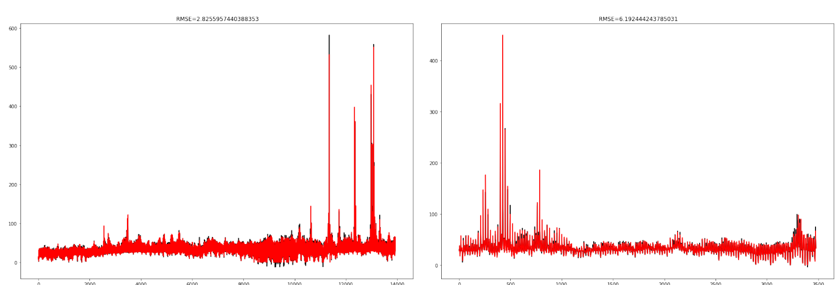

(a) Training Data

(b) Testing Data

Figure 8: ANN models on 2 years data

We observe that the first year's pattern looks extremely different from the second year's pattern, indicating that any model based on purely past lagged values will not be able to capture the real driver behind the spread spike occurrences. The current training error is 25 percent and testing error is 76 percent which show much room for improvement, but the forecast is able to capture some essential patterns of the future spreads.

We apply the same change of method on pure price data from the two years' dataset.




(a) Training Data

(b) Testing Data

Figure 9: ANN on Two Year Price Data

The training error is 9 percent and testing error is 19 percent. This result is better than applied to the spread forecasts due to the abnormal data of the spread sequence.

## 7 Spike Classification

Based on our prior discussion, the ideal way to forecast spikes is to build a machine learning classification model to predict if a spike is occurring at the given point or not. Then we build a regression solely on the spike data points to predict the magnitude of the spike. To this end we need several exogenous variables to be the predictors of the spike behavior.

## 8 Year Ahead Forecasts

Another question of interest is how shall we forecast the entire year ahead price spread sequence given the past years' data. To this end we have three approaches. The naive method is to simulate every day's point forecast using the same type of models as before and use the simulated value as the true value to add to the data set. As we add in new data we move forward with our window slice to predict on day ahead. Although simple to implement, this simulation method tends to not show any effect of the price volatility towards further into the future as the initial price spread signal will be diminished as we progress if a time series model is implemented. The errors also accumulate very quickly as the forecasts move forward.

However, if we switch to a neural network regression which we discussed before, then the initial local errors are very low and hence we get accurate predictions for a longer period of time into the future. As we build our future predictions on these simulated values we don't get too large a global error since local errors are significantly reduced. Using 120 days as the past window slice to train the neural network and testing this simulation based forecasting on a stand out complete year data starting from only having an initial input vector for the first point forecast, we obtain an RMSE of 5.473 which corresponds to about 11 percent in terms of the mean percentage error over the time series. The resulting simultaneous forecasts for a whole year time frame is shown below:

Another approach is to build a regression model to forecast year ahead value with the one-year lagged value plus categorical variables. Since we can train the model using the 2017-2018 and 2018-2019 data, we can predict 2019-2020 price spread using the 2018-2019 data set. The data sample is large enough because we can stack the 6000 nodes' data since they are similar in nature. The categorical variables will consist of 12 month indicators and 24 hour indicators. Hence our input feature length is 37 and output length is 1. We can check the regression behavior and modify the model later.

A third possible approach is to use a sequence to sequence learning model such as a RNN framework. The entire year's data for one node will be taken

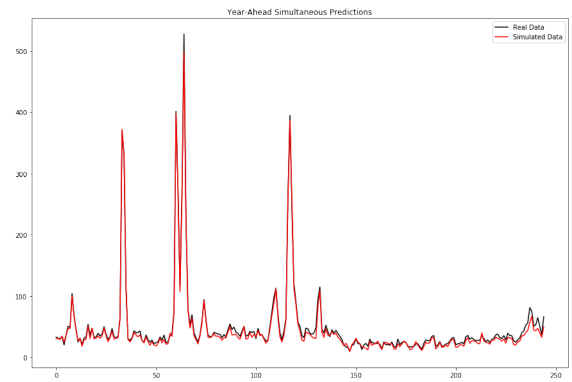
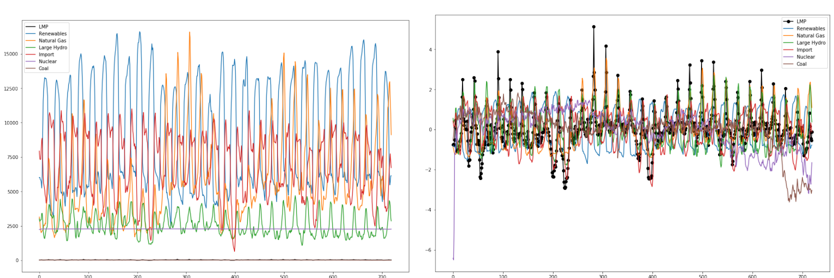



Figure 10: Year Ahead Price Forecasting with ANN Simulations

as one training sample and the output value will be the next year's entire data vector. So the input length and output length both equal to 365 or some other prediction length.

# 9 Adding Exogenous Variables for Explanatory Power

As mentioned earlier, the patterns in price spread data or pure price data cannot be explained well enough by its own past history. There are many fundamental driving factors of the price sequence of electricity (LMP). Some of the economic factors include supply and demand of many types of fuels and energy. Other exogenous variables include temperature and precipitation. We test if adding in supply data from six different energy sources will improve the neural network learning performance.

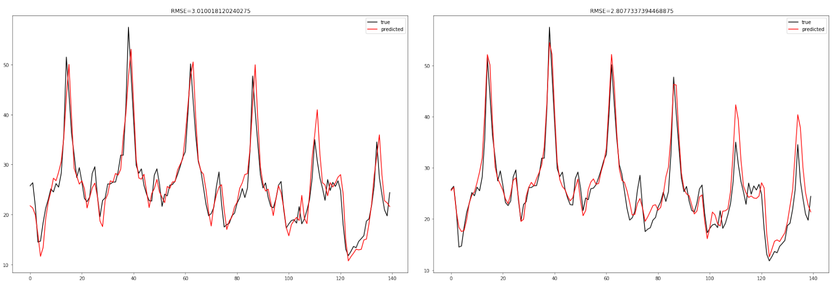



(a) Six Exogenous Series

(b) After Applying Scaling

Figure 11: Exogenous Variables

As an example we compare the hourly price data of the LMP from 2018 June to that of the supply data of the same month, preprocessed into hourly frames, for Renewables, Natural Gas, Large Hydro, Import, Coal, and Nuclear. The figure below shows the raw data we get. A standard scaler was applied to the different datasets for uniformity and we stacked a past historic window of values into one feature vector for each point forecast of the LMP price.




(a) Without Exogenous Variables (b) With Exogenous Variables

Figure 12: ANN Testing Performance on 2018.06 Price

The results show that with the addition of exogneous variables and after sufficient training and tuning, the error reduces from 12 percent to 10 percent and the important features of the dataset such as the occurrence of spikes are restored in the predicted series in a timely fashion.

We continue on to perform predictions of the spread sequence using exogenous variables as our data gets larger. As a preliminary result we obtained the correlation matrix between spread and the six exogenous variables.

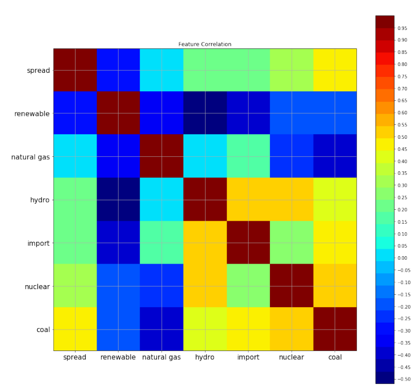



Figure 13: Exogenous Variable Correlation Matrix

# 10 Neural Network and ARIMA-X Models for Price Predictions with Supply and Demand Data

At this point we were able to retrieve sufficient exogenous variable data to conduct tests on the predictive power of supply and demand variables from energy sources such as natural gas, coal, nuclear, renewables, and large hydro. The data ranges from June 2018 to February 2019.

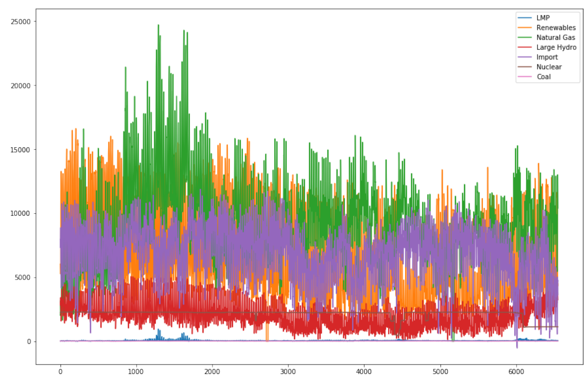
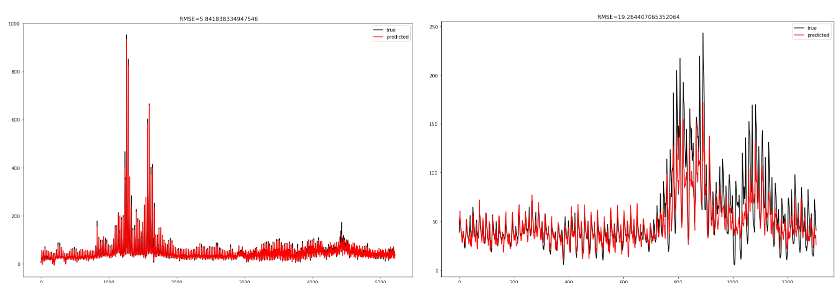



Figure 14: Exogenous Variables 2018.06-2019.02

First we add in the past supply series of these variables into the existing electricity price series while keeping the time frames uniform. Then we pass the extended feature vector into a neural network to test the prediction results on the next period (hour) price.




(a) Training (b) Testing

Figure 15: ANN Model with Exogenous Supply

The model shows an overall RMSE that is below 10. We compare its performance against an ARIMA-X model in which only the supply series of natural gas was included as the exogenous components and an ARIMA(2,1,1) model is fitted on top of it.

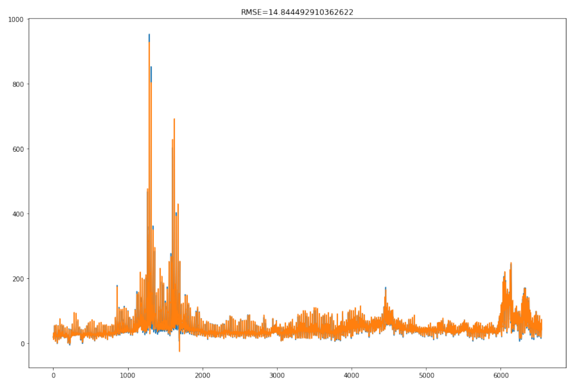



Figure 16: ARIMA(2,1,1)-X

The reported RMSE is around 14.

# 11 Out-of-sample Year-Ahead Price Spread Forecasts

We gathered three entire years' data and then performed a training on the second year's data against the first. The testing is then done on the third year's data against the second year's. This choice was deliberately made so that we can predict year ahead given the current year's data.

Specifically, we tried a neural network that takes as input the past year's lagged values and the current years lagged values together. After training a neural network we predicted the point forecast for step ahead price spreads and the results are shown below with an error nearly 60. The predictions are able to extract some crucial information about the appearance of spread spikes but the magnitude is not as high as the realized spikes are.

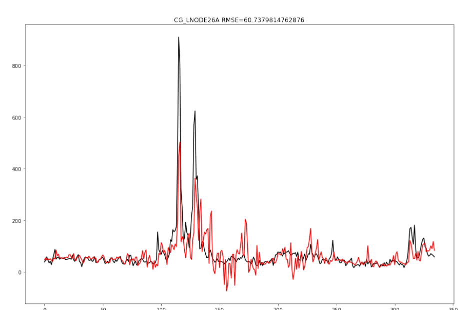



Figure 17: Year-Ahead NN Model

## 12 References

1. Electricity price forecasting: A review of the state-of-the-art with a look into the future, Rafal Weron, International Journal of Forecasting 30(2014).

2. Day-Ahead vs. Intraday-Forecasting the Price Spread to Maximize Economic Benefits, Katarzyna Maciejowska, Wweronika Nitka, and Tomasz Weron, Energies, 2019.

3.Price Forecasting in the Dat-Ahead Energy Market by an Iterative Method with Separate Normal Price and Price Spike Frameworks, Sergey Voronin and Jarmo Partanen, Energies 2013.

4. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms, Jesus Lago, Fjo De Ridder, and Bart De Schutter, Applied Energy 2018.