Day-Ahead Electricity Price Forecasting via Machine Learning

May 2021

**Table of Contents**

[Abstract 3](#_Toc71566177)

[Introduction 3](#_Toc71566178)

[Related work 3](#_Toc71566179)

[Methods 3](#_Toc71566180)

[VMD 4](#_Toc71566181)

[GSA 4](#_Toc71566182)

[LSTM 4](#_Toc71566183)

[Dataset and Features 5](#_Toc71566184)

[Data Pre-processing, Model Creation and Data Flow 5](#_Toc71566185)

[Experiments / Results/ Discussions 6](#_Toc71566186)

[Conclusion / Future work 7](#_Toc71566187)

[Contributions 8](#_Toc71566188)

[References 9](#_Toc71566189)

# Abstract

Electricity price forecasting is the curial step to optimize energy efficiency in the electricity market. However, due to the sophistication in electricity price prediction, the accurately 24-hour electricity price forecasting is the one of the key factors to increase efficiency and economics of the power plants. In this project, the model includes variational mode decomposition (VMD), long short-term memory (LSTM). The dataset that we used in this study is one-year hourly 2020 California Independent System Operator (CAISO) dataset. The model performance illustrated that mean absolute error is 0.0766. The results illustrated that VMD-LSTM model electricity price forecasting is significantly accurate and stable.

# Introduction

We are tackling the day-ahead electricity price forecasting to optimize the power plants efficiency, financial gain and prevent wasting unnecessary electric energy. The accurate prediction of day-ahead price can help economic operation of power plants and effectively predict future electricity load in the short period of time to prevent unexpected power outage. This is an application of real-world data via machine learning algorithm. Forecasting application in this regard is the act of prediction based on previous historical time series data.

Time series data may be characterized with complex non-linear interrelations [1] and as such, functions that are applied to explore data which have specific objectives to analyze, model, extract knowledge and understand dense dynamic relationships between label and independent features. The most useful type is Short-term Forecasting – prediction for few seconds, hours, days, weeks, or months [4]. In this project we designed a machine learning model suitable for time series datasets and specifically applicable for short-term energy price prediction applications.

# Related work

A. Heydari et al., 2020 [1] predicted short-term electricity price and load for isolated power plants and the model obtained accurate and stable results. Their proposed model was VMD-GSA-GRNN. Their work is the key inspirational factor to our project.

# Methods

Our original plan was the mixed machine learning approach (VMD – GSA – LSTM). Unfortunately, we had some limitations to use Gravitational Search Algorithm (GSA) in this project. However, we discuss it in the methods. Our designed model includes VMD for filtering, denoising and generating the features of the original electricity price data, and Long Short-Term Memory (LSTM) for training, validating, and testing the time series data and thus generate price forecasting. This project framework does not include the GSA for feature selection. The project framework shown as below:

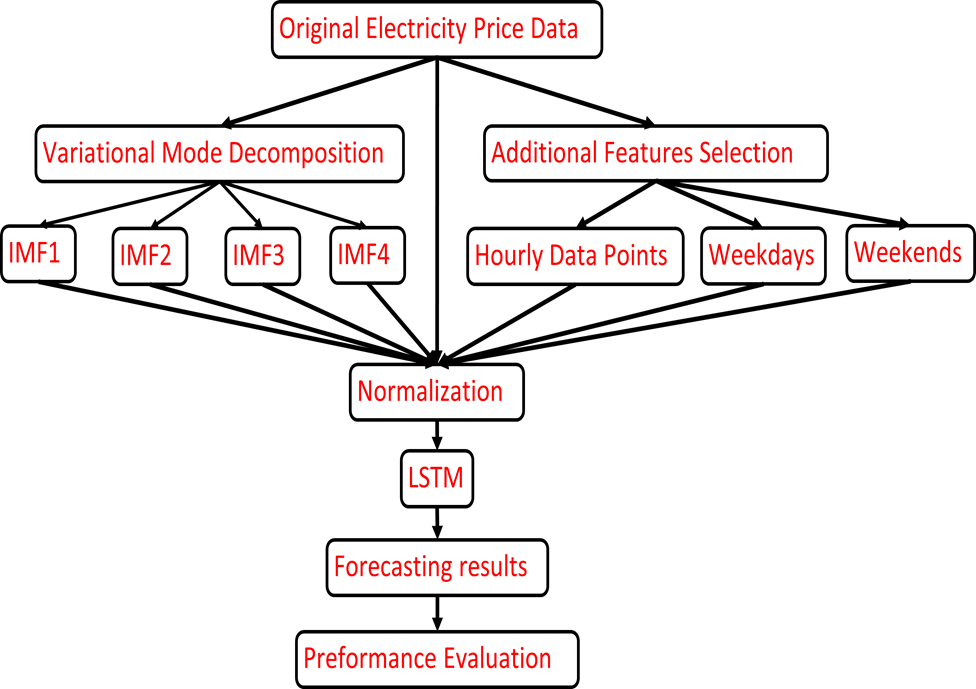


Figure 1. The model Framework

## VMD

The VMD method proposed by Dragomiretskiy and Zosso in 2014 [1][3]. It is a novel method of non-recursive signal processing designed to decomposing a dimensional signal to independent modes. The goal of VMD is to decompose a real valued (electricity price) input signal f, into a discrete number of sub-signals (modes),uk. The mode has specific sparsity properties while reproducing the input. The original electricity price signal decomposed into 4 sub-signals as an independent subseries and denoted by Intrinsic Mode Function (IMF1…IMF4th), it represents decomposed signals from high to low frequency (Fig 2.). Each mode is compacting around a center pulsation wk, it is to be determined with decomposition. There are three steps to make it work:

• Obtaining the unilateral frequency spectrum of every subseries uk, through Hilbert transform computing analytic signal,

• Gaining the corresponding estimated center frequency through modifying the mode frequency spectrum,

• Assessing each mode bandwidth through the H1 Gaussian smoothness of the decomposed signal.

Each IMF series will be prepared as the new input values for LSTM as training, validation, and testing data.

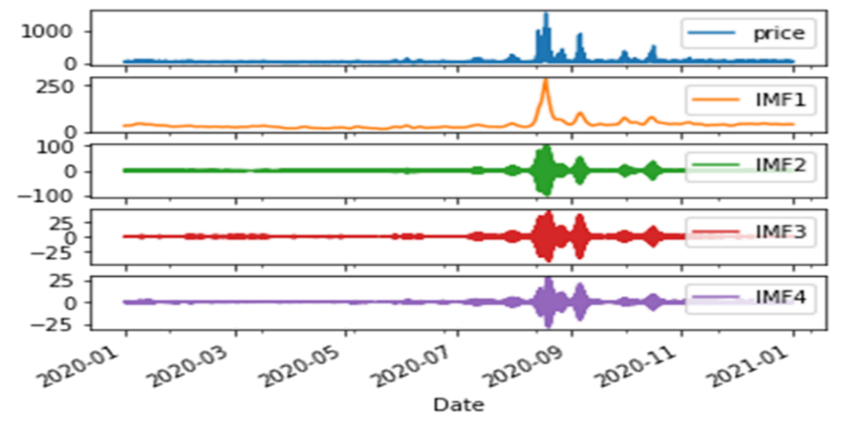


Figure 2. One-year decomposition results

## GSA

GSA is a nature inspired metaheuristic optimization technique inspired by Newton’s Law of gravitation and motion. It is used to generate optimal features from diverse array of features to assure model stability, minimize prediction error and reduce computational time and energy [1]. It is initiated by generating a random number of agents with initial masses. It achieves the objective function by the movement of objects with lower mass to objects of higher mass till all agents converge. At the point of convergence, optimal solution is assumed to have been found and the search terminates. GSA generates the optimal solution when it reaches termination criteria. [2]

## LSTM

LSTM is a deep learning neural network with backpropagation support. This is a special kind of Recurrent Neural Network (RNN) which works as a composition of long-term and short-term memory. LSTM overcomes the vanishing gradient issue of RNN during the training of a neural network [5][6][7][8]. LSTM efficiently identifies hidden patterns and potential of the data through a continuous self-learning process with the help of gates, and activation functions. One of the distinguishable factors in the LSTM network is the memory cell (LSTM cell). The usually hidden layers of a deep neural network are replaced by memory cells in LSTM architecture [8]. LSTM cell includes an input gate, a forget gate, and an output gate in the memory block [6][8]. This network works through a sigmoid layer, a tanh layer, and pointwise multiplication, pointwise addition operation. LSTM knows how to maintain cell state and can control input flow.

Diagram, schematic

Description automatically generated

Graphical user interface, application

Description automatically generated

Figure 3. A LSTM Cell [4]

# Dataset and Features

This section describes the dataset and its features. This also includes data preprocessing steps, data flow, data preparation, and model creation.

**Data Description**

We considered CAISO historical timeseries data to evaluate our neural network model in training, validation, and test phases. We collected this dataset from the energyonline.com [9]. This time series contained hourly electricity price from January to December 2020. This dataset represented San Diego Gas & Electric (SDGE) electricity market.

# Data Pre-processing, Model Creation and Data Flow

After getting the decomposed signal inputs from the VMD, we had our primary dataset which contained timeseries, price, and four IMFs. Then, we went through the following data pre-processing tasks to finally forecast hourly electricity price for 24 hours.

1. **Stationarity Check:** Stationarity verifies that the statistical properties in a time series dataset do not change over time (e.g., mean, standard deviation, variance etc.). We followed three methods given in the following for stationarity check. Our datasets passed this test.
2. **Visualize Plots**: Drew a histogram plot of the time series data and visually checked if there are any non-stable trends or seasonality (Nonstationary). Seasonality is the presence of variations that occur at specific regular intervals.
3. **Summary Statistics**: Checked the summary statistics on the data for seasons or random partitions. Then, compared the mean or standard deviation of those partitions.
4. **Statistical Tests**: Performed statistical tests, like Augmented Dickey-Fuller (ADF) test to check stationarity in data.
5. **Add Extra Features:** Added more features like hour data, weekdays/weekends indexes so that our model can learn more about our data.
6. **Normalization/ Standardization:** Used Z- score normalization to handle outliers in our train dataset. It is measured by subtracting the mean from the original data points and dividing by the standard deviation.
7. **Data Preparation:** Machine learning model usually required three types of datasets to perform any experimental analysis, (i) ‘train dataset’ to train the model, (ii) ‘validation dataset’ for evaluation the quality of the model, (iii) ‘test dataset’ to test the model after the model has gone through the validation process. In this project, we considered a 70%, 20%, and 10% split for creating training, validation, and test datasets, respectively. We did not randomly shuffle the data during splitting to reserve the sequence in the data.
8. **Sliding Window Method:** Designed a sliding window method which considered prior time steps (168 hours) to predict the next time steps (24 hours). This sliding window technique is used in this project to train the model and forecast electricity price on test data.
9. **Create Dataset:** Created a python method to convert the time series DataFrame into tensor data by using preprocessing.timeseries\_dataset\_from\_array() function. This function converted the train, validation, and test dataset into tensor data to feed into the Neural Network.
10. **Model Creation (LSTM):** Each LSTM cell contains 5 layers. Three of them are sigmoid and two are tanh layers. Besides, to design our model we used (i) input layer for inserting data by following the setup of sliding window method; (ii) hidden layer which has 32 neurons to work repeatedly until get best result; (iii) Dense layer to get output from previous layer which piped through 192 hidden neurons (1 dimensional tensor); and (iv) output (reshape) layer to generate the output. We set 150 epochs and batch size was 168 in this model. Adam optimizer was used for the optimization of this model.
11. **Train the Model:** Trained our model with 70% of the data. During the training, the LSTM algorithms detect the pattern in the data that gradually design a map in between input data attributes and the desired output.
12. **Validate the Model:** Calculated loss in training and validation dataset to compare and improve the model performance. Mean Square Error (MSE) method was used to measure the loss in this model.
13. **Predict using Test Dataset:** Based on sliding window method electricity price forecasting was accomplished on the test dataset. Mean Absolute Error (MAE) was used to measure the performance of this model on the test dataset.

# Experiments / Results/ Discussions

Electricity price forecasting is one of the most critical issue in the economical operation of power system. High accuracy in the day-ahead price prediction can increase the profitability of electricity market. Our model on CAISO data gave us ignorable error and impressive performance on electricity price prediction.

**Model Loss**

MSE is utilized to investigate the model loss on training and validation datasets. MSE is the measure of the average of the squared of the errors that means average squared difference between the predicted values and actual values [10]. The MSE is a measure of the quality of the model, always non-negative and the values closer to zero is always better. Figure 4 shows the model loss during training and validation process. The x-axis represents 150 epochs for our model and the y-axis represents the loss on each epoch. The right - down corner of Fig 4. shows that the training loss and validation loss of this model embrace each other and closer to zero. This ensured that this model was neither underfitting nor overfitting, rather a good fitting model on this data.

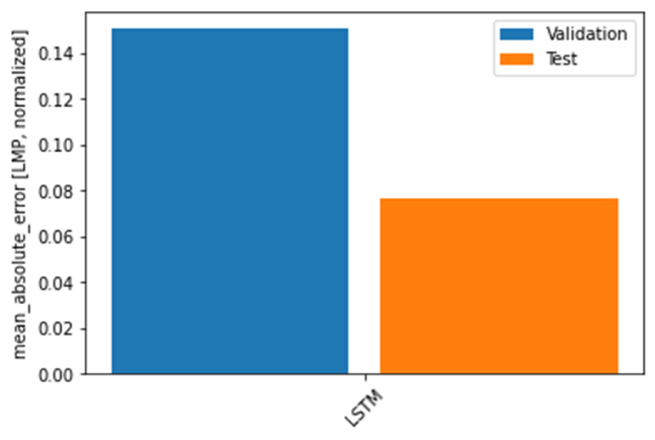
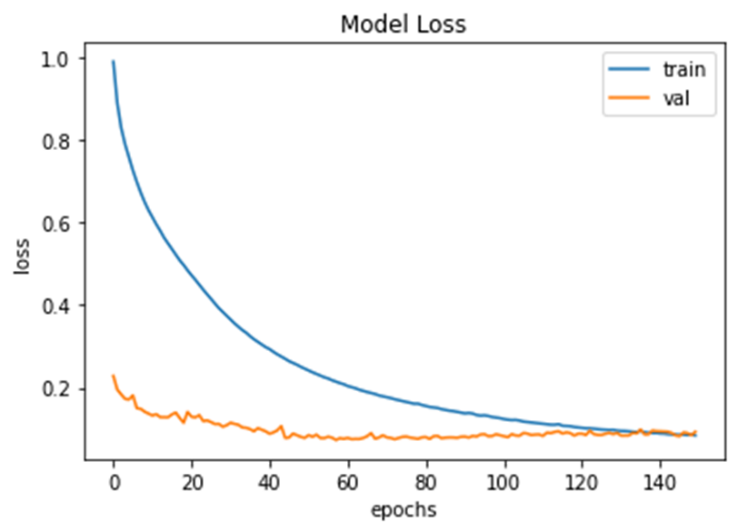
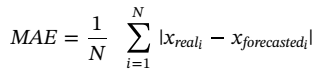


Figure 4. Model Loss Figure 5. Model Performance

**Model Performance**

To evaluate the performance of this model, MAE was chosen. MAE measures the average of the difference between the actual price label and the predicted price. The error calculation was measured by the following equation:



Here, N = 24 hours, Xreal = Original price, Xforecasted = Predicted price by our model. The lower the error means high accuracy in the price prediction. Figure 5 shows the MAE by this model on the validation and test datasets. The model MAE on the test dataset showed only 0.0766, which was ignorable error and indicative of impressive performance of this deep learning model.

We divided the entire year data into four seasons like the following, (i) March to May is Spring, (ii) June to August is Summer, (iii) September to November is Fall, and (iv) December to February is Winter. The following table 1 shows MAE for each season. All these seasonal errors are less than one which is acceptable and good for a model. Unfortunately, the summer season data was overfitting the model. We are working on it to make summer data a good fit for our model.

Table 1: MAE on Seasonal Data

|  |  |
| --- | --- |
| **Season** | **MAE** |
| Spring (March - May) | 0.6783 |
| **Summer (June - August)** | **0.5811** |
| Fall (September - November) | 0.1685 |
| Winter (December - February) | 0.3003 |
| **Year (January - December)** | **0.0766** |

**Price Prediction**

Figure 6 shows the predicted results on hourly electricity price. Each of these three subplots represent a window of 192 hours. These are randomly picked window, but the data are sequential. In the graph the blue portion is the previous 168 hours (1 week) of price, the green circles are 24 hours (1 day) original data labels, and the orange cross are 24 hours predicted price. These plots illustrate that the price prediction impressively follow the trend which is incredibly good forecasting.

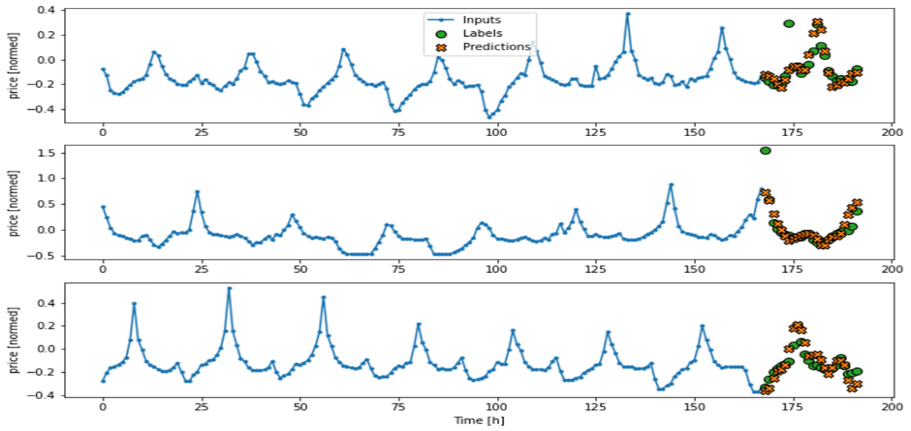


Figure 6. Price Prediction using our Model

# Conclusion / Future work

The project results showed that MAE was significantly small. But still there is room for improvement by exploring optimization algorithms for better the model performance. The results do not include the GSA for feature optimization and selections. We are still working on the VMD-GSA-LSTM framework to improve the model performance even further. At the same time, we plan to solve summer data overfitting problems. We will also focus on incorporating weather data to investigate its effects on overall model performance and electricity price forecasting accuracy.

# References

[1] Azim Heydari, Meysam Majidi Nezhad, Elmira Pirshayan, Davide Astiaso Garcia, Farshid Keynia, Livio De Santoli, Short-term electricity price and load forecasting in isolated power grids based on composite neural network and gravitational search optimization algorithm, Applied Energy, Volume 277,2020. https://doi.org/10.1016/j.apenergy.2020.115503.

[2] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, “GSA: A Gravitational Search Algorithm,” Inf. Sci. (Ny)., vol. 179, no. 13, pp. 2232–2248, 2009, doi: 10.1016/j.ins.2009.03.004.

[3] Gholamreza Memarzadeh, Farshid Keynia,Short-term electricity load and price forecasting by a new optimal LSTM-NN based prediction algorithm,Electric Power Systems Research,Volume 192, 2021. https://doi.org/10.1016/j.epsr.2020.106995.

[4] Dragomiretskiy, K. and Zosso, D., 2013. Variational mode decomposition. IEEE transactions on signal processing, 62(3), pp.531-544.

[5] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, “Stock price prediction using LSTM, RNN and CNN-sliding window model,” 2017 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2017, vol. 2017-January, pp. 1643–1647, 2017, doi: 10.1109/ICACCI.2017.8126078.

[6] G. Memarzadeh and F. Keynia, “Short-term electricity load and price forecasting by a new optimal LSTM-NN based prediction algorithm,” Electr. Power Syst. Res., vol. 192, no. November 2020, p. 106995, 2021, doi: 10.1016/j.epsr.2020.106995.

[7] W. Contributors, “Long short-term memory,” en.wikipedia.org, 2021. https://en.wikipedia.org/w/index.php?title=Long\_short-term\_memory&oldid=1005032489 (accessed May 03, 2021).

[8] M. Phi, “Illustrated Guide to LSTM’s and GRU’s: A step by step explanation,” towardsdatascience.com, 2018. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21 (accessed Jun. 03, 2021).

[9] <http://www.energyonline.com/Data/GenericData.aspx?DataId=22&CAISO___Day-Ahead_Price>

[10] Wikipedia contributors. (2021, April 30). Mean squared error. In *Wikipedia, The Free Encyclopedia*. Retrieved 17:58, May 10, 2021, from <https://en.wikipedia.org/w/index.php?title=Mean_squared_error&oldid=1020706752>