INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
ALLAHABAD

6^{TH} SEMESTER PROJECT

Forecasting Energy Trading Prices Using Time Series Models

Submitted To:
Dr. Pavan Chakraborty
IIIT Allahabad

Submitted By:
Ishan Arora (IIT2017501)
Vikrant Singh (IIT2017502)
Srikar Anand (IIT2017504)
Akshay Gupta (IIT2017505)
Naman Deept (IIT2017507)

CERTIFICATE FROM SUPERVISOR

We hereby declare that the work presented in this mid semester project report of B.Tech (IT) 6th Semester entitled ”Forecasting Energy Trading Prices Using Time Series Models” , submitted by us at Indian Institute of Information Technology, Allahabad, is an authenticated record of our original work carried out from January 2020 to February 2020 under the guidance of Dr. Pavan Chakraborty.

Due acknowledgements have been made in the text to all other material used. The project was done in full compliance with the requirements and constraints of the prescribed curriculum.

Date : 20-02-2020
Place : IIIT Allahabad

Supervisor:
Dr Pavan Chakraborty

Acknowledgments

The timely and successful completion of the book could hardly be possible without the helps and supports from a lot of individuals. We will take this opportunity to thank all of them who helped us either directly or indirectly during this important work.

First of all we wish to express our sincere gratitude and due respect to our supervisor Dr Pavan Chakraborty, Faculty,IIIT Allahabad. We are immensely grateful to him for his valuable guidance, continuous encouragements and positive supports which helped me a lot during the period of my work. We would like to appreciate him for always showing keen interest in my queries and providing important suggestions.

We also express whole hearted thanks to our friends and classmates for their care and moral support.

Contents

1 Abstract

This report describes the 6th semester project our group is working on, titled "Forecasting Energy Trading Prices Using Time Series Models". Time series analysis have been the subject of a great interest in the recent times. Time series analysis is used for prediction and forecasting. It has many real life applications in field of business, sales, etc such as economic and sales forecasting. In this paper, the energy trading price has been predicted using the time series analysis of various features such as weather, stock. This has been done through the use of ARIMA, HMM and the neural network such as LSTM.

2 Introduction

Time series analysis is of utmost importance in the modern world where every industry wants to enhance its revenue by predicting the future of their product and adjusting their production , consumption and maintain their market . Time series analysis of data not only helps in forecasting but also helps in understanding the past of subject in consideration statistically . It also helps in learning the trends which affects various business domain , also it can be used for monitoring physical systems , software systems , financial trading systems , weather forecasting etc .
Time series analysis and forecasting of next-day energy trading prices has been chosen because it has great effect on an economy . Here , multivariate features affecting energy trading prices will be taken into consideration that affects the energy trading prices and time series analysis will be done . For example ,

- Coal based electricity prices are dependent upon coal production , coal import and oil prices .
- Solar energy based electricity are dependent upon weather conditions on daily basis .

Taking these into considerations for a particular energy trading , this project intends to find and propose the trend between these features and energy trading prices .

Under this project, basically we had performed time series analysis for forecasting which essentially has two parts, firstly forecasting of weather variables like solar irradiation, wind speed, temperature etc. and secondly predicting the final power output (or efficiency of a photo-voltaic panel). Time Series Analysis helps us to understand what are the underlying forces leading to a particular trend and helps us in forecasting and monitoring the data points by fitting appropriate models to it. Traditionally, time series forecasting was under the scope of statistics and models like Auto-regressive Integrated Moving Average (ARIMA), but due to possible

non-linearity in the data, we had also used Long Short Term Memory (LSTM) models which are more reliable and accurate over this kind of data. We had also implemented it through Hidden Markov Models (HMM) and the results have been compared.
Dataset collection and combining is a specifically hectic task in performing Multivariate Time Series Analysis. The datasets are not easily available in a single file. One has to manually collect various datasets which contain various types of variables and combine them together.

3 Motivation

We aim to explore new possibilities and incorporate them in our everyday lives to make it more comfortable and easy going. Time series analysis has been in use around us for the longer time. It has been used in statistics, geophysics, data mining, etc for the forecasting purpose. It also has a great impact in the field of business to predict sales and stock prices. The main motive for using the time series is that it can predict or forecast the future.

4 Problem Definition

In this paper titled as "Forecasting Energy Trading Prices Using Time Series Models", we had implemented an efficient system for predicting the prices of energy trading. For this purpose different data-sets related with various features such as weather and stock prices can be used as these prices depend on these features. To study and predict the results, time series analysis need to be done on these data-sets. Time series analysis can be done through the use various tools and technologies such as ANN, SVM, correlation, etc. In this paper we had tried to implement time series analysis using the ARIMA (Auto Regressive Integrated Moving Average), Hidden Markov Model(HMM) and the LSTM (Long Short Term Memory) network . We also compared the results found from these three models.

5 Literature Review

5.1 Forecasting Next-Day Electricity Prices by Time Series Models

This paper titled as "Forecasting Next-Day Electricity Prices by Time Series Models" was published in IEEE 2002 by Francisco J. Nogales, Javier Contreras, Member,

IEEE, Antonio J. Conejo, Senior Member, IEEE, and Rosario Espínola. In the electricity markets, both the electricity producers and the electricity consumers require a tool which could predict the prices of the electricity with higher accuracy. This paper has given two efficient and accurate model to predict the electricity prices which is based on the time series analysis using dynamic regression and transfer function models.
The first method called dynamic regression approach relates the price at particular hour t with the prices at the past time i.e. at hours t,t-1, ... and the values of demands at previous times t,t-1, and so on. The second model called transfer function approach assumes that the price and the demand series are stationary i.e. their mean and covariance is constant. This approach uses ARMA model.
To study the results of these models, datasets from two different markets was collected, one from Spanish and other from the Califorian market. Average error found in Spanish market was around 5% while in Californian market was 3%. Spanish market was more volatile compared to Californian because of less competition, also the results were less accurate and more uncertain during the peak hours in Spanish markets because of high demands. The results obtained from these models were sufficient to predict the prices so that these could help the producers and the consumers.

5.2 Time Series Analysis of Household Electric Consumption with ARIMA and ARMA Models

The current various researches have used the method of forecasting with time series data such as the electric power consumption.The time series analysis of the given production deals with a regression problem for obtaining the deterministic output, the classical regression model for obtaining the desired is not sufficient.Instead, the introduction of correlation as a phenomenon that may be generated through lagged linear relations leads to proposing the autoregressive (AR) and autoregressive moving average (ARMA) models. Adding non-stationary models to the mix leads to the autoregressive integrated moving average (ARIMA) model popularized in the landmark work by Box and Jenkins (1970). Autoregressive models are based on the idea that the current value of the series, x_t, can be explained as a function of p past values, $x_{t-1}, x_{t-2}, ..., x_{t-p}$.This also contains the gaussian noise which needs to be estimated using gaussian distribution equations.We will be using ARIMA model to generate the future use of the energy consumption from the given present datasets, along with this model Bayesian Vector Auto-regression Model (BVAR) can also be used to get the desired prediction in the energy utilisation and consumption trends.

6 Proposed Methodology

For studying the forecasting trend and it's analysis, there comes a very important to select the proper model, the desired selection is a must as it should reflect the underlying structure and trends involved in the forecast. Based on the type of problem the proposed model for the given can be the one which fits under the criterion of non-stationary time-series and we initially preferred the linearity in studying the analysis (rather than assuming the non linear time-series) since the model can be observed to be linear from practical aspect.

6.1 Dataset Aggregation and Preparation

Dataset has been aggregated from various sources and combined to form single dataset with features that is impact-ful and affect greatly the everyday trading prices. Dataset has been collected from :

- Wholesale electricity Data and SEDS data provided by US EIA [2000-2013].[5]

- SP index daily price data, Yahoo finance [2000-2013].[6]

Various features are then extracted to apply ARIMA , Hidden Markov Model and LSTM model and then the results are compared.

6.2 Auto-regressive Integrated Moving Averages(ARIMA)

The Auto-Regressive moving average model (ARMA) model is useful for studying the uni-variate time series. ARMA model as the name suggests is the combination of Auto-Regression (AR) and Moving-Averages(MA). The mathematical equation for regression consists of the present output as the linear function of the past output with the error term included. Mathematically the AR(p) model can be expressed as :

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t$$

where ϕ_i are the estimates and ϵ_i is the error value corresponding to the time 't'. The estimates are found using Regression Algorithms either using Least Squares or Gradient Descent ,the value of error is a probabilistic function (can be assumed Gaussian).The term 'c' is also included in the estimate as this parameter also needs to be calculated.
The moving average MA(q) assumes the past error values as explanatory variables.so

,instead of dependence upon the previous instances of variables , these are dependent upon the error values,Hence new prediction MU(q) will be given as:

$$y_t = \mu + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t$$

Here θ are the model parameters, and μ is the mean values. So combining these will give the ARMA model.The problem with this model is that it can be used only in the case of univariate ,time-stationary models.But in practice the data shows the non stationary behaviour and so we need to use the concept of generalised ARMA model ,also referred to as ARIMA model (Auto-Regressive Integrated Moving Averages),this covers the case of non stationary data as well. ARIMA is a triple (p,d,q) meaning that the values p,d,and q needs to be defined. The mathematical formulation of the Arima(p,d,q) is given as:

$$(1 - \sum_{i=1}^{p} \phi_i L^i)(1 - L)^d y_t = (1 + \sum_{j=1}^{q} \theta_j L^j)\epsilon_t$$

Here the parameters in the triple p,d, and q are non negative integers, Generally the value of d is considered to be 1 as this covers most of the cases, d=0 gives back our natural ARMA model. In our experiment we considered the value of d as 1. This is a nice generalisation to our ARMA model since ARMA(p,0,0) gives our AR(p) model and ARMA(0,0,q) gives our MA(q) model. So this covers both of them.The model is most widely used in many practical applications for prediction of stock prices, that are non stationary (i.e mean,variance, and standard deviation of the model are not same over time).

6.3 Long Short-Term Memory (LSTM)

Long Short Term Memory (LSTM) are the special kind of Recurrent Neural Networks (RNN). LSTM have the capability to learn the long term dependencies. LSTM initially given by Hochreiter and Schmidhuber in 1997. LSTM can be used on a large variety of problems and hence it is now widely used in the field of deep learning. LSTM has feedback connections. LSTM can used to process both the single data points like images as well as sequences of data like videos. LSTM can also handle the problem of the Long Term dependency.

The common architecture of LSTM unit consists of the cell and the three gates which are input gate, output gate, and forget gate. The job of cell is to remember the values over the arbitrary intervals of the time while the job of the gates is to regulate and control the flow of data and information from in and out of the cell. The cell keeps the track of the dependencies between the elements of the imput sequence. The input gate's work is to regulate the flow of new values in the cell. Forget gate controls that for how much unit of time value should be present in the cell. Output's gate job is to check that for how much extent the value present in the cell will contribute to the computation of the output activation of the LSTM unit. The activation functions used for the gates of LSTM are usually sigmoid activation function and the hyperbolic tangent function.

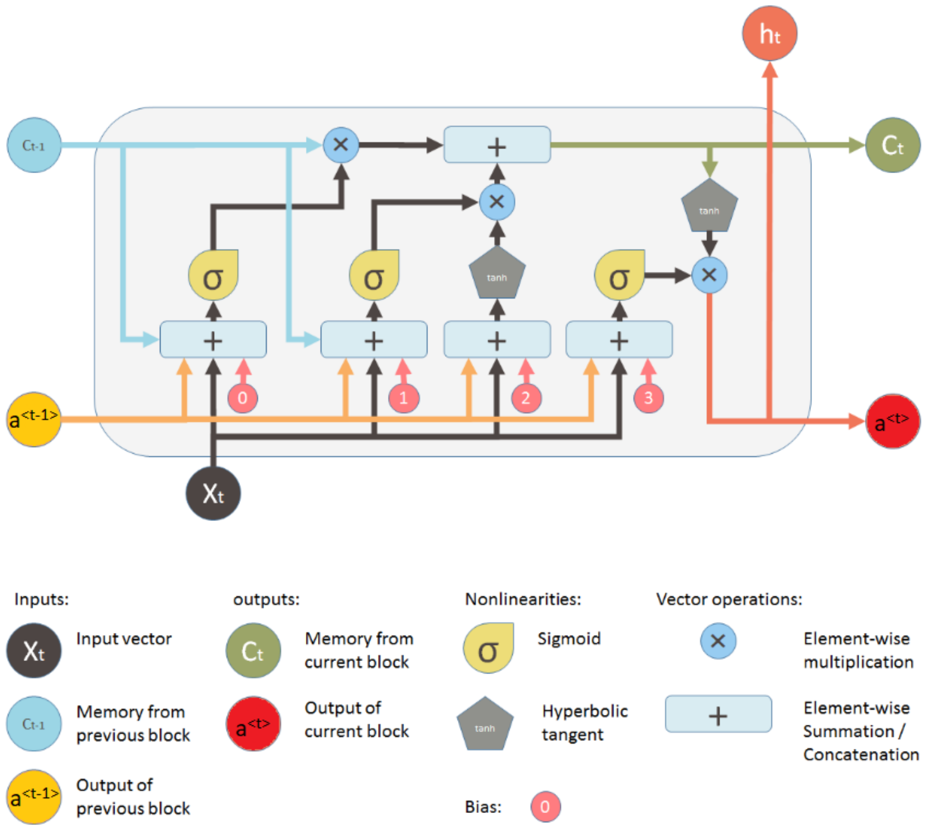

Figure 6.2.1 : Common LSTM unit architecture
Source: Adapted from [3]

Architecture for a common LSTM unit is shown in Figure 6.2.1. THe role of forget layer is to make a decision that what should be thrown away from the cell state which make use of the sigmoid layer. It takes $a^{<t-1>}$ and x^t as a input and produces a output for every number in the cell $c^{<t-1>}$ between 0 and 1. The meaning of the 1 is that model should completely keep this while 0 means it should completely forget it. Now the input gate by the use of the sigmoid layer decides that which values should be updated. After this a tanh activation layer creates a whole new vector of the values denoted by $\tilde{c}^{<t>}$ which we can add to model. The two layers i.e. i^t and $\tilde{c}^{<t>}$ on combining decides that what new values should be stored in the cell. To update new cell c^t, old cell c^{t-1} is multiplied with the forget gate f^t and then bias i^t * $\tilde{c}^{<t>}$ is added to it. To decide the output of the LSTM unit, first output gate layer is computed using the sigmoid activation layer which decides what values should be kept in output. Now cell $c^{<t>}$ is passed through hyperbolic tangent activation function tanh, which is then multiplied with the output gate layer $o^{<t>}$ to get the desired output. The job of the tanh activation function is to keep the values between -1 and 1. All the required equations of LSTM unit and how they are connected is shown in Figure 6.2.2.

$$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$$
$$i^{<t>} = \sigma(W_u[\, a^{<t-1>}, x^{<t>}] + b_u)$$
$$f^{<t>} = \sigma(W_f[\, a^{<t-1>}, x^{<t>}] + b_f)$$
$$o^{<t>} = \sigma(W_o[\, a^{<t-1>}, x^{<t>}] + b_o)$$
$$c^{<t>} = i^{<t>} * \tilde{c}^{<t>} + f^{<t>} * c^{<t-1>}$$
$$a^{<t>} = o^{<t>} * \tanh c^{<t>}$$

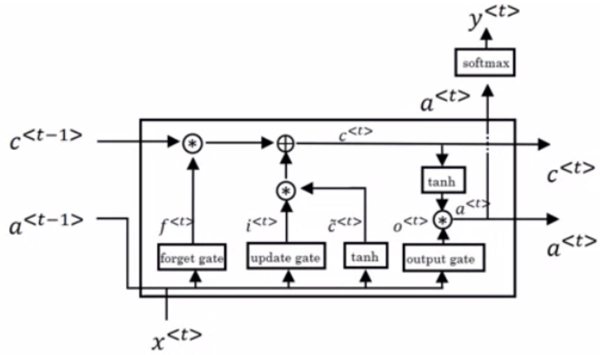

Figure 6.2.2 : Equations for the LSTM
Source: Adapted from [4]

Now with the necessary data and the above knowledge of the LSTM, a model of LSTM was built which has total 18 number of the input features which are High Price, Low Price, Change , Daily Volume, Number of Trades , Number of Companies , DNI , Temperature , DHI , Pressure , Relative Humidity , Wind Speed , Open , High , Low , Close , Adj Close , Volume . The 85 percent of dataset has been used for the testing purpose and the remaining 15 percent has been used for the testing purposes. For the pupose of training the model of LSTM, the Keras API and Tensorflow has been used. LSTM model was trained using the number of epochs equal to 20 with the use of the single dense layer. The dropout value for the model was taken as 0.2. The objective of the dropout values was to avoid the situation of over-fitting. For the purpose of the optimisation and faster convergence mini batch gradient descent was used and the batch size was taken as 64. These details of LSTM model is also shown in Figure 6.2.3.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_5 (LSTM)                (None, 100)               47600
_________________________________________________________________
dropout_5 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 101
=================================================================
Total params: 47,701
Trainable params: 47,701
Non-trainable params: 0
_________________________________________________________________
```

Figure 6.2.2 : Details of LSTM model

6.4 Hidden Markov Model (HMM)

The Forward–backward algorithm used in HMM was first described by Ruslan L. Stratonovich in 1960 and in the late 1950s in his papers in Russian. The Hidden Markov Models were later described in a series of statistical papers by Leonard E. Baum and other authors in the second half of the 1960s. One of the first applications of HMMs was speech recognition, starting in the mid-1970s. Markov models are becoming increasingly famous now-a-days as a means of modelling various phenomena in different disciplines. Hidden Markov Model (HMM) is one such model.

HMM is a statistical generative model which considers the system being transitioned in certain number of states. The model assumes that the state is can not directly observed, but the dynamics or the output of the state can be observed. In the model, time may be assumed as a discrete quantity. Also, current value of the state is assumed to be dependent only on the previous values, but not the future values. For each unique pair of states, there is a certain probability that the state may be changed from the first to the second. A state may not even transit, it may just stay in the same state. All the transition probabilities can be represented in a matrix, and it is called transition matrix.

Mathematically,

- X_n, Y_n and X_n is not directly observable
- X_t depends only on X_{t-1}
- Y_t depends only on X_t

Let us define the terminologies in HMM.
Let the number of states be n. The history may be written as

$$x_1, x_2, ..., x_T, \text{ for each i, } 1 \leq x_i \leq n.$$

The transition probability from state i to state j may be written as

$$\mathrm{P}(X_{t+1} = j | X_t = i) = \gamma_{ij}$$
$$\gamma = [\gamma_{ij}], \text{ for } 1 \leq i, j \leq n$$

γ is the one-step transition matrix. k-step transition matrix can be calculated as $\Gamma(k) = \gamma^k$
Let μ_i and σ_i be the mean and the variance of the guassian distribution of state i
One of the key interests of HMM is the marginal distribution, $\mathrm{P}(Y_t)$. The output of the model will give us conditional distribution, i.e., $\mathrm{P}(Y_t|X_t = i)$. But we need to marginalise over all possible solutions

$$\mathrm{P}(Y_t) = \Sigma P(X_t = i)P(Y_t|X_t = i)$$

Dealing with joint likelihood of observed data and unobserved states

$$\mathrm{P(X,Y)} = \Pi P(X_t|X_{t-1})P(Y_t|X_t)$$

Forward-Backward Algorithm to calculate the probability of data sequence.

$$\mathrm{P}(X_t|Y_{1:T}) = \mathrm{P}(X_t|Y_{1:t}, Y_{t+1:T}) = \mathrm{P}(X_t|Y_{t+1:T})\mathrm{P}(Y_{t+1:T}|X_t)$$

7 Requirements

- Intel i7 or above processor.
- RAM - 8GB minimum.
- Operating System - Linux (Ubuntu), Windows
- Python 3.6
- Tensorflow v1.14 with Keras API
- Pandas , Numpy , SKlearn statsmodel python libraries
- matplotlib and seaborn libraries for plotting purposes
- jupyter notebook or spyder IDE
- Google Colab resources
- Google Chrome web browser for Google Collaboratory

8 Experimental Data and Results

8.1 Results obtained using ARIMA

The ARIMA model has mainly three parameters :

p : The number of Auto-Regression

q : The moving average parameter

d : The non-seasonal difference

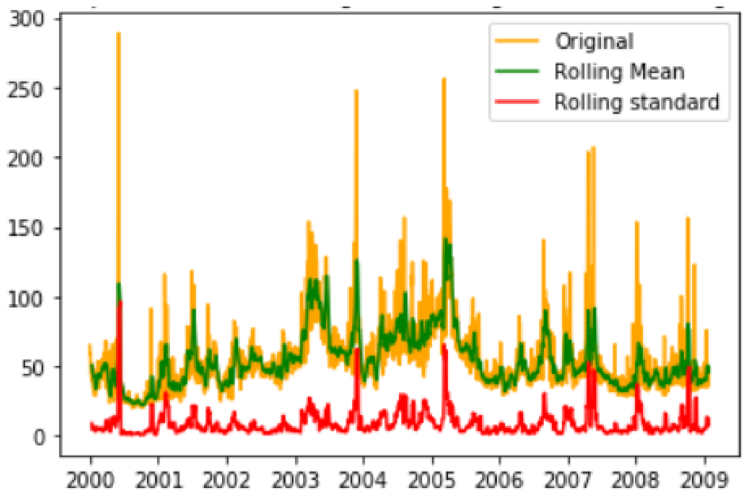

Figure 8.1.1 : Graph of actual vs predicted training prices using different methods (X axis: date in year and Y axis: price in $ per megawatt)

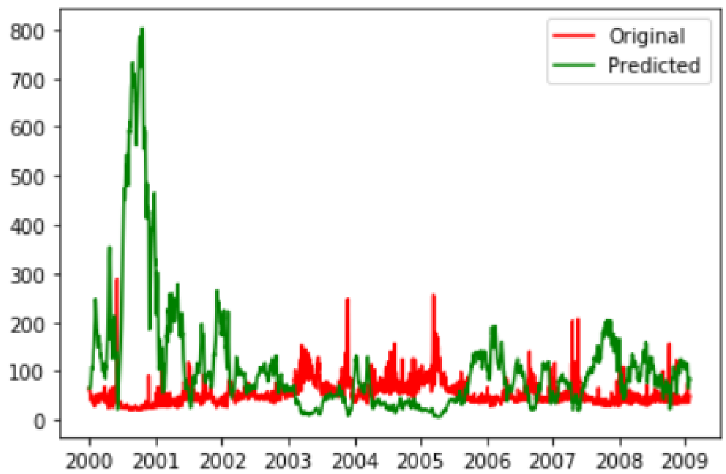

Figure 8.1.2 : Graph of actual training prices vs predicted training

prices (X axis: date in year and Y axis: price in $ per megawatt)

Root mean squared error for prediction with ARIMA model is **147.5509182239792**.

8.2 Results obtained using HMM

The hidden markov model makes prediction of the future based on solely the present and past values . This model used only past and present values of stock prices to predict the future prices.

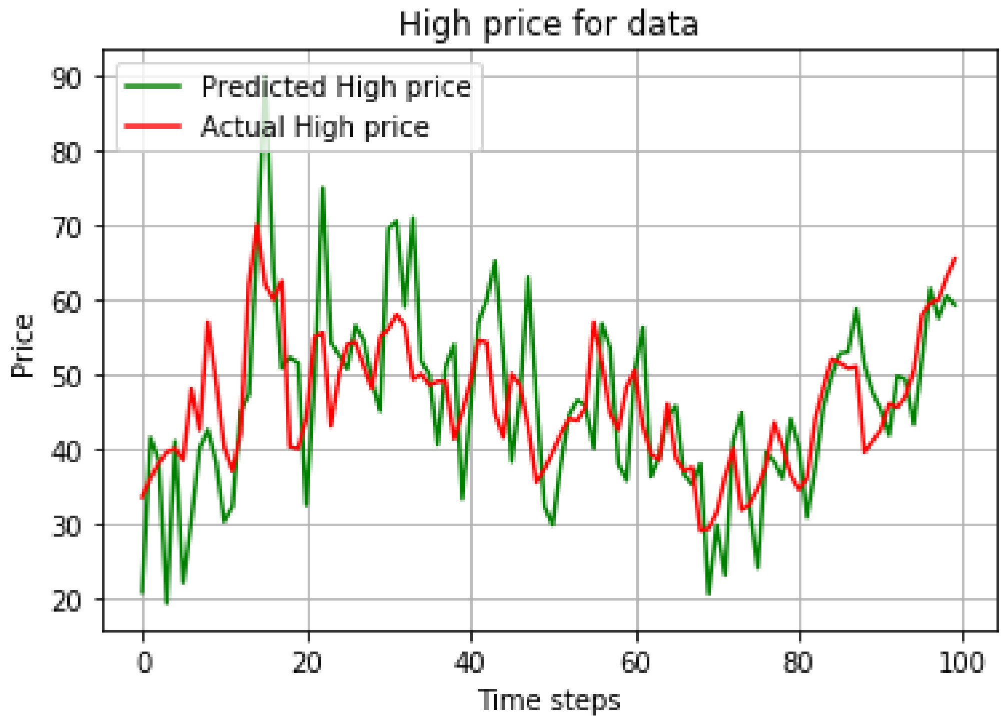

Figure 8.2.1 : Graph of actual vs predicted trading prices

Root mean squared error on high prices using Hidden Markov model is **9.273716892379236**.

8.3 Results obtained using LSTM

The LSTM model used 85% of the data to train the model and the 15% of the data to test the model and root mean square error of the model is calculated .

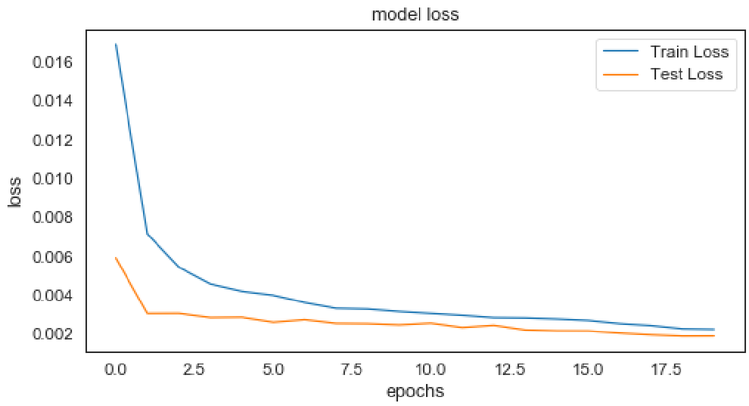

Figure 8.3.1 : Graph of training loss and testing loss vs epoch

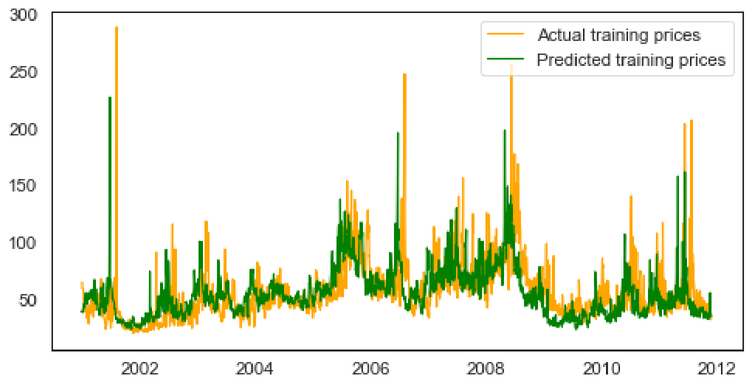

Figure 8.3.2 : Graph of actual training prices vs predicted training prices (X axis: date in year and Y axis: price in $ per megawatt)

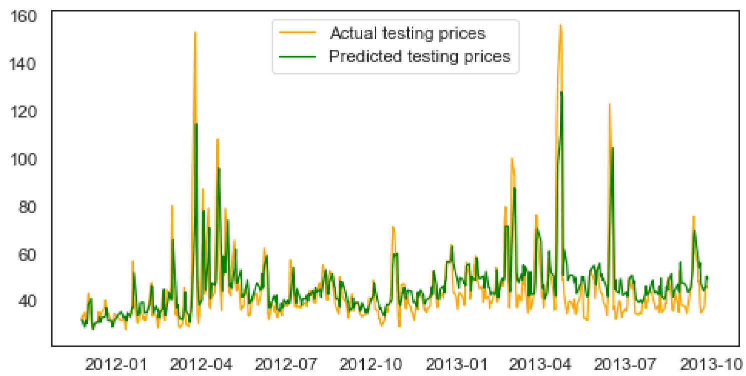

Figure 8.3.3 : Graph of actual testing prices vs predicted testing prices (X axis: date and Y axis: price in $ per megawatt)

Training set root mean square error of the LSTM model : **12.518183038575403**
Training set mean absolute error of the LSTM model : **8.148153889445059**

Testing set root mean square error of the LSTM model: **11.747577291919768**
Testing set mean absolute error of the LSTM model: **7.403125770781229**

9 Conclusion

Three different time series models has been used in time series analysis and future price prediction of stock prices . ARIMA model showed a large rmse value of **147.5509182** for this multi-variate time series model. Hidden Markov Model was a rather simple uni-variate time series model which showed rmse value of **9.27371689237**. This model may fail to perform well under multi-variate conditions. Lastly, LSTM model is presented which performed the best among the three models with training set rmse value of **12.518183038575403** and the testing set rmse value of **11.747577291919768** when given the multi-variate time series data. On comparison ,it can be concluded that the LSTM model was better performing when given a multi-variate time series data which can deal with multiple features efficiently.

10 Future Work

A multi-variate time series hidden markov model can be implemented with the input features same as in the dataset prepared . Results of this new hidden markov model can be compared with the already implemented LSTM model and ARIMA model as in this project . Also different hyper - parameter tuning can be performed on the LSTM model to improve the efficiency of the model . New features that affect the stock prices thay may not be considered in this project can also be included to further the work in this project .We can also use the Non Linear Moving Average model (NMA) model or the Eagle Arch Model which is heteroskedastic, and hence can be used as a better implementation rather than considering the linear models.

11 References

[1] Francisco J. Nogales, Javier Contreras, Member, IEEE, Antonio J. Conejo, Senior Member, IEEE, and Rosario Espínola, ”Forecasting Next-Day Electricity Prices by Time Series Models”, IEEE TRANSACTIONS ON POWER SYSTEMS, VOL. 17, NO. 2, MAY 2002, 342-348

[2] Pasapitch Chujai, Nittaya Kerdprasop, and Kittisak Kerdprasop, ”Time Se- ries Analysis of Household Electric Consumption with ARIMA and ARMA Models”,

International MultiConference of Engineers and Computer Scientists 2013 Vol I, IMECS 2013, March 13 - 15, 2013, Hong Kong

[3] S. Yan, “Understanding LSTM and its diagrams,” Medium, 15-Nov-2017. [Online]. Available: https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714. [Accessed: 11-Mar-2020].

[4] Andrew Ng,“Deep Learning Specialization,” Coursera. [Streaming Video]. Available: https://www.coursera.org/specializations/deep-learning. [Accessed: 11-Mar-2020].

[5] Wholesale electricity Data for the PJM West Hub provided by EIA, [Online]. Available: https://www.eia.gov/electricity/wholesale/. [Accessed: 15-Feb-2020]

[6] SP index daily price data, Yahoo finance, [Online].
Available: https://finance.yahoo.com/quote/%5EGSPC/. [Accessed: 15-Feb-2020]

[7] Colah’s blog, ”Understanding LSTM Networks” Posted on August 27, 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed: 11-Mar-2020].