ANALYZING WIND POWER DATA USING MACHINE LEARNING

ANALYZING WIND POWER DATA USING MACHINE LEARNING

A project report submitted in partial
fulfillment of the requirements for the degree of
Master of Science in Computer Science

By

Guru Nishok Radhakrishnan
Master of Science, University of Colorado, Denver, 2019
Bachelor of Technology, Anna University, 2017

12/2019
University of Colorado Denver

ABSTRACT

Wind power is one of the major sources of renewable energy. In order to level up the various other alternative sources of energy, numerous numbers of wind turbines are being deployed at both on-shore and off-shore. Thus, the power generated by wind turbines is increasing rapidly, thereby taking the generated power for business purposes. A dependable forecast for the speed of wind is highly needed which plays a vital role in conversion mechanisms pertaining to the Wind Energy systems. The generated power is mainly obtained from the speed of the wind. Even though, the speed of the wind is subjected to change regularly, but with the help of the fact that it follows a specific pattern over a specific time period. A precise and reliable wind speed forecast is needed to solve the problems faced by the variable wind energy conversion systems. This variation in the wind speeds makes it hard for a machine learning model to identify a pattern in the speed of the wind. So as a result, we consider this problem as a forecasting problem where this pattern of the time series generated is taken into consideration which helps us to gain some helpful information which we will be using it for the prediction of the power. We use the machine learning model, Long Short-Term Memory, which is known to be accurate in learning the patterns of the wind data and predicting the generated power.

This Project Report is approved for recommendation to the Graduate Committee.

Project Advisor:

Dr. Liang He

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Problem

There is a high inclination in renewable energy which leads to some problems occurring due to the variable production of the energy[10]. Therefore, a prediction mechanism which is reliable and accurate is much needed for the deployment of the wind turbines which in turn helps in constructing the wind farms. Generally, we use the machine learning algorithms to derive a prediction method which is applied to the data set to predict the result. These models which are built follows a statistical observation of the data sets, hence the term data-oriented models. These developed algorithms using machine learning are helpful in predicting which mainly focus on the time which varies from minutes to hours and is deployed in controlling the wind turbine to produce power with high efficiency. The model we build in this project is purely based on the measurements of the time series of the wind power that is being generated. We use the machine learning model, Long Short-Term Memory which helps in identifying the patterns in the data and predicts the results which is the generation of the power.

1.2 Project Statement

In this proposal, we build a machine learning model which would overcome the issues faced by considering the problem as a time series pattern [6]. Long short-term memory (LSTM) is a constructive unit which will build various levels of recurrent neural network. An LSTM network is generally developed in order to solve the problem of the vanishing gradient and the exploding problem when the recurrent neural networks are being trained. The LSTM network consists of many numbers of hidden layers, as a result, it has more memory units and it has a network of deep stacked LSTM's which helps in remembering the pattern of the previous outcomes and is best suited for studying the observations of the past. Hence, we use the LSTM network as a model in our project to predict the power generated by the system.

1.3 Approach

A recurrent neural network (RNN) [14] comprises of various units of LSTM which is called as the network of the LSTM. Generally, a unit of LSTM will consist of a cell and three gates namely, input, output, forget. An LSTM model helps in computing the time lags of the various sizes and the duration of the events by means of classifying, processing, and predicting the pattern of the time series. These networks have an upper hand over other predictive models such as hidden Markov models and various other learning models which are sequentially used in applications in terms of the relative remembrance of the length of the gap.

1.4 Organization of this Project Report

Chapter 2 discusses the description of the Neural Network, Recurrent Neural Network, Long Short-Term Memory network. Chapter 3 gives information pertaining to the architecture model and high-level design. Chapter 4 describes the various methodologies deployed in the model, its results and the analysis. Chapter 5 comprises of the conclusion, the main contributions and the future works that can be done to enhance the model.

2. BACKGROUND

2.1 Key Concepts

Section 2.1 in this project includes a brief description of the Neural network, Long Short-term Memory and the Recurrent neural networks. Section 2.2 includes various areas of the research that have been performed related to this project.

2.1.1 Neural Network

The group of algorithms which are mainly designed in order to detect patterns are called the neural networks [4]. They use the perception of the machine by clustering and labeling the data as raw data and interpret the problem. The recognized patterns generally comprise of the numbers which are stored in the vectors. These vectors help in storing and translating all the real-time data i.e., textual data, pictorial data, sound data, and the time series data.

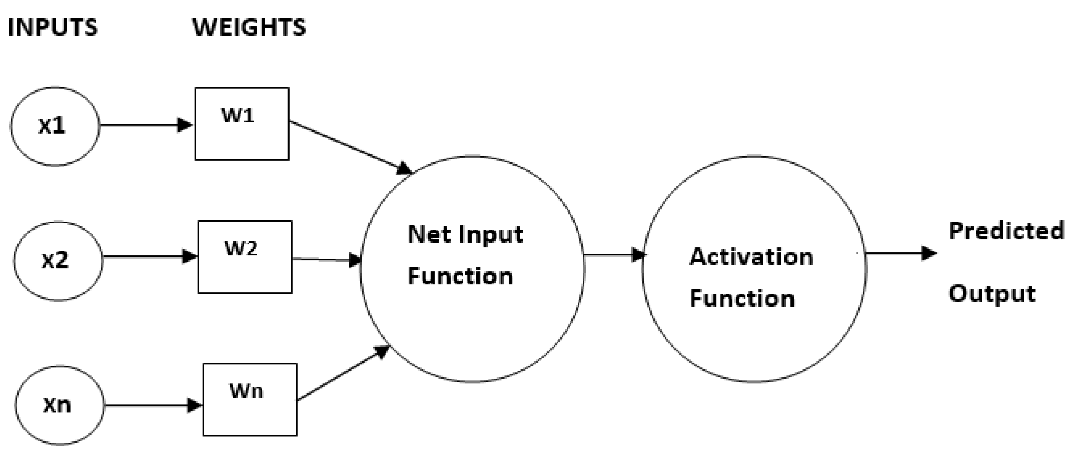


Figure 1.1: Representation of a Neural Network

There are multiple layers in the neural networks. Each of those layers contains its nodes. The main part of the computation will happen on these nodes of various layers. The input data along with the set of weights are combined by the node. This combination of both input-weight is added, and this result will be provided to the node's activation function. A flow representation of the neural network can be found in figure 1.1.

The layer of the node will look like a row of switches which will be active and inactive as the input values are being fed on through the neural network. One node's output will the successive node's input which starts from the initial layer of the neural network as shown in figure 1.2. The goal would be in designing a model which would select the weights which are adjustable and pair it up along with the various features of the input.

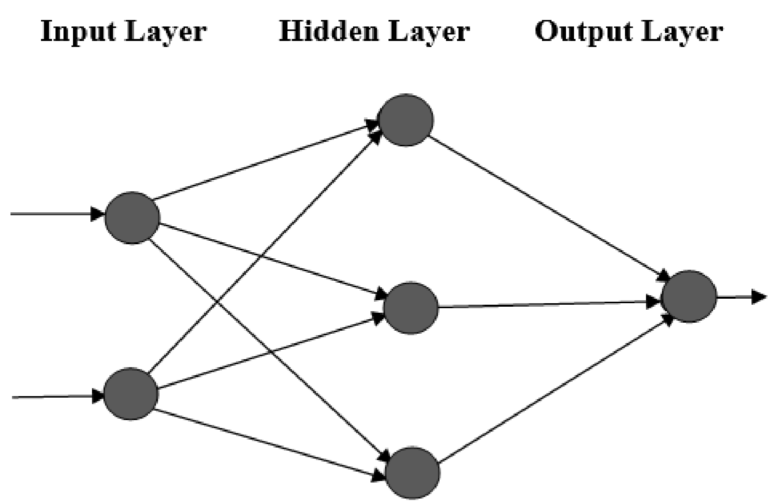


Figure 1.2: Layers of a Neural Network

2.1.2 Recurrent Neural Network

Recurrent Neural Networks [14] are considered to be the strongest type of neural networks because of the fact that the capacity of these networks to have internal memory. Since the RNN's have their internal memory, they will be having the capacity to remember important values of the input fed to them, which helps them in computing the result of the successive output. The diagrammatic representation of the Recurrent Neural Network is represented in figure 1.3. However, the Recurrent Neural Network has its own limitations. Gradient descent is a function which measures the deviation of the output with respect to the change in the input values. The limitation of the RNNs is the exploding gradient problem where the designed algorithm will start assigning different weights without any specific reason. The other limitation includes the vanishing gradient problem where the obtained values of the gradient descent are very small, and the developed model will halt learning, or it will ,in turn, take a long time to compute.

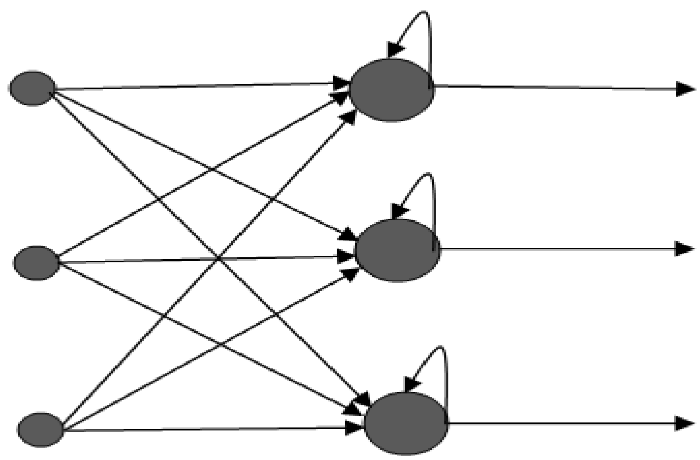


Figure 1.3: Representation of a Recurrent Neural Network

2.1.3 Long Short-Term Memory

Long Short-Term Memory (LSTM) is basically an extended form of Recurrent Neural Networks [8]. LSTM will have a better memory which enables them to remember the time lags which are longer in duration. There are also several layers in LSTM which comprise of various layers of an RNN. Together as a collection, they are called as a LSTM network. The LSTM network will help the RNN in remembering the patterns over a longer period of time. The main reason being that the memory unit of the LSTM will behave as a real-time computer which has the capacity of manipulating the information from its location in the memory. The memory unit in the LSTM is called as the gated cell which will have the authority to either store or delete the information. The weight assignment is also learned by the algorithm. The usage of LSTMs overcomes the limitations of the exploding gradient and the vanishing gradient problems that were discussed in the Recurrent Neural Networks. Figure 1.4 represents the various connections which are traveling in and out of the LSTM network and it was taken from [8]. The weighted connections are learned when the data set is being trained, by which the operation of the various gates change.

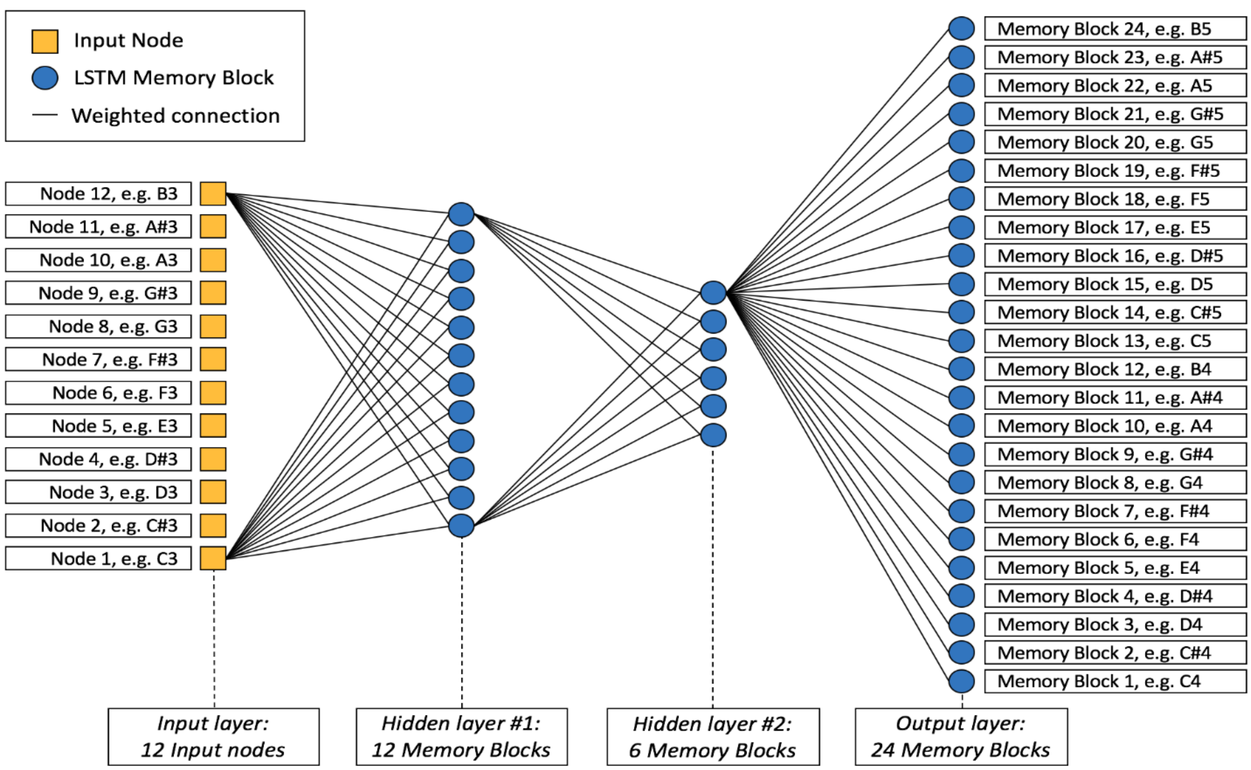


Figure 1.4: Representation of Long Short-Term Memory Network

2.2 Related Work

There has been a broad view of an experiment in the field of generation of power. Costa et al. [2] paper discuss the several methods of models designed for the short-term prediction using mathematical context, statistical analysis and the physical models over the last three decades. Soman et al. [16] paper give us a broad perspective about the techniques pertaining to the forecasting problems. From the past research and the studies, it is seen that learning the model statistically provides a positive outcome when it comes to the prediction regarding the short-term energy. Foresti et al. [5] paper provide a view of a linear regression model with the help of the support vector machines enable a way to detect the patterns of the wind speed predictions.

The paper by Mohandes et al. [13] uses the neural networks in order to predict the power of the wind which is being generated. The paper also discusses the comparisons of the regressive models with the network of backpropagation. The other similar work includes the paper by Catalao et al. [1] which works on training the feedforward network using the L-M algorithm in order to obtain the forecasting of the wind power which proved to be a more efficient model than other prediction mechanisms. The paper by Ziqiao et al. [7] introduces a method to predict the power generated by the wind turbines in a short-term by means of using the neural networks with the help of speed of the wind and the direction of the wind from the data set. The proposed model with respect to the data which is being used provided accuracies that are high. Nils et al. [3] paper provide a model of prediction in order to overcome the problems faced by the renewable energy supplies which are highly needed in order to manage the conventional wind energy power plants.

3. ARCHITECTURE

3.1 High-Level Design

The problem of the prediction is divided into two categories namely the estimation phase and prediction phase. The estimation phase helps in determining the total energy produced that is the production of the power provided the conditions of the weather like pressure, speed of the wind and the temperature. The steps include loading the pre-processed data set and splitting them into a train-test set[15]. Identify a suitable look backs for the time frame. This is followed by reshaping the data and fitting the data set into the LSTM model and the power generated by the system will be predicted with a Root Mean Square Error. We have designed the model [9] such that the problem of computing the precise generation of the power is considered as the time series problem. This is because we know that the pattern of the wind follows a specific pattern for a specific time of the year, month or day.

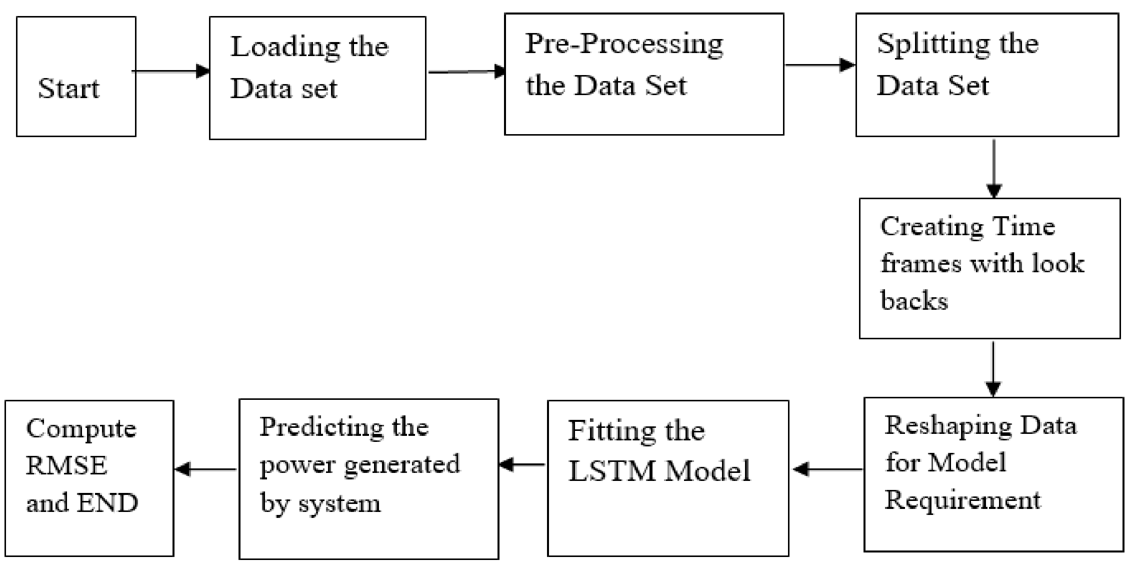


Figure 2.1: Flow Diagram of the Working Model

The model we are using, Long Short-Term Memory, helps in easy learning in order to find the wind patterns from the data set and make the prediction of the power. The capability of memory remembrance in the LSTM makes it a standalone over other models of neural networks such as Markovian models and the recurrent neural networks.

There are three gates in the LSTM network [8]. They are the input, output and the forget gates. The functions of the gates are as follows, the input gate helps in deciding to feed the new input into the network or not. The forget gate will decide which unrelated information can be deleted. The output gate decides whether to cause an impact in the particular time step using the output value. The cell above is solely authorized to keep track of the various sequences of the outputs.

In [7]:
```
''' Loading data '''
dataframe = pd.read_csv('C:/Users/gurun/Desktop/CPS/DATASET/WIND/dataset1.csv')
dataframe.head()
```

Out[7]:

	Year	Month	Day	Hour	Minute	Wind Direction (deg)	Wind Speed (m/s)	Air Temperature ('K)	Air Pressure (atm)	Air density (kg/m^3)	Power generated by system (MW)
0	2007	1	1	0	0	244.189	11.673	275.580	97808.504	1.223	15.701
1	2007	1	1	0	5	244.755	11.395	275.537	97815.832	1.224	15.498
2	2007	1	1	0	10	245.644	11.468	275.486	97823.160	1.224	15.568
3	2007	1	1	0	15	246.863	11.636	275.428	97825.896	1.224	15.691
4	2007	1	1	0	20	247.287	11.727	275.342	97832.304	1.225	15.719

Figure 2.2: Representation of the original data set

3.2 Data Set

The data set used for this project was taken from the department of wind energy on the website of the national renewable energy laboratory [11]. The data set has a total of 105,120 entries of a wind turbine which records the data of wind power generation for every 5 minutes throughout the year. The data set was subjected to pre-processing in order to find any outliers. The data set contained various metrics which was recorded during the generation of power. It includes the temperature of the air, atmospheric pressure, the direction of the wind, speed of the wind, and finally the wind energy power generated by the system. The data set also included the timestamp as an attribute for the whole year with the interval of every five minutes. The original data set which was downloaded from the website of the National Renewable Energy Laboratory (NREL) is represented in figure 2.2.

3.3 Data Pre-Processing

The real-world data which is collected into a data set is not perfect. The collected data set will contain some outliers, inconsistent data, and the data which tends to be noisy. These kinds of data must be removed. In order for the better prediction of the model, we have to pre-process the data set in order to obtain the desired results. This pre-processing includes the cleaning of the data where we try to identify any missing values and fill out with the average of the nearest neighbor, followed by the normalization of the data set. This normalization technique will alter the numerical values in the columns by means of the scaling without the deformation of the information of values provided in the data set. This, as a result, will help the LSTM model to predict the future wind power generation as close as possible. The data pre-processing also included the step to convert certain attributes measures in the original data set for better understanding purposes. For instance, the measure of the unit of Air temperature was converted from Kelvin to Celsius and the Power generated by the system was converted from the unit of Megawatt to Kilowatt. The time stamp attribute was combined into a single column i.e., the year, month, day, hour, a minute into a separate column named Date-Time index. This is done since we consider the problem as a time series problem in order to predict the amount of the power generated by the wind turbine in the future. The screenshot of the pre-processed data set can be found in figure 2.3. This pre-processed data set will be used for training Long Short-Term Memory to predict the power generated by the system.

In [10]:
```
dataframe = pd.read_csv("C:/Users/gurun/Desktop/CPS/DATASET/WIND/DPPdataset.csv",index_col="DATETIME")
dataframe
```

Out[10]:

DATETIME	Air Temperaure ('C)	Air Density (kg/m^3)	Wind Speed (m/s)	Wind Direction (deg)	Power (KW)
1/1/2007 0:00	2.430	1.223	11.673	244.189	15701
1/1/2007 0:05	2.387	1.224	11.395	244.755	15498
1/1/2007 0:10	2.336	1.224	11.468	245.644	15568
1/1/2007 0:15	2.278	1.224	11.636	246.863	15691
1/1/2007 0:20	2.192	1.225	11.727	247.287	15719
1/1/2007 0:25	2.128	1.225	11.501	247.035	15602
1/1/2007 0:30	2.073	1.225	10.830	247.833	14972
1/1/2007 0:35	2.046	1.226	9.824	252.287	13272
1/1/2007 0:40	2.018	1.226	8.955	259.919	11073
1/1/2007 0:45	1.923	1.227	8.942	265.988	11043
1/1/2007 0:50	1.719	1.228	9.802	268.478	13241
1/1/2007 0:55	1.466	1.229	10.812	268.183	14966
1/1/2007 1:00	1.258	1.230	11.374	266.350	15497
1/1/2007 1:05	1.093	1.231	11.490	264.592	15609
1/1/2007 1:10	0.965	1.232	11.222	262.625	15358
1/1/2007 1:15	0.880	1.232	10.746	260.878	14912
1/1/2007 1:20	0.825	1.232	10.361	258.753	14396
1/1/2007 1:25	0.797	1.232	10.153	255.858	13976

Figure 2.3: Representation of the pre-processed data set

3.4 Implementation

Every attribute in the pre-processed data set will be used for the estimation phase of the project while only the attributes Date Time and amount of power generated by the system will be used by the prediction phase of the project. The phase of the estimation will predict the amount of wind power generated provided the other features like the direction of the wind, and the speed of the wind is given. Since the temperature and the wind directions are being provided to the learning model, it makes the problem solvable for the LSTM network which with the help of current environmental conditions and the previous pattern of the weather computes the wind energy power generated by the system.

The LSTM is run on a Keras library with the forecasting problem as a multivariate time series[6]. Once the basic model of the LSTM network was formed, initial experiments were carried out in determining the look backs for the model. Lookbacks or the time delay is the lag, a term used in the Long Short-Term memory which provides us with the information of a number of steps used by the designed LSTM model in order to predict the next result. It was found that when the number of look backs was 8, there was a good prediction of the power generated by the system was obtained. The phase of estimation models is useful when the trends of the patterns of the environmental conditions are given, then we can compute the power generated by the system.

The prediction experiments are done purely on the time series analysis. The power generated by the system is predicted without any knowledge of the weather conditions in the future. This is also a serious problem since we are predicting an outcome without any given data like the direction of the wind, speed of the wind but with just making an analysis in the previous patterns in the data by means using the LSTM model.

Data set to this type of model will only contain the Date Time index and the power generated by the system as a supervised form of learning which will be fed into the LSTM network. The LSTM model will learn the previous patterns in the data and with the help of the gained knowledge, it will predict the power generated by the system. For predicting the future values, we use the validation of the forward walk and evaluate the results.

4. METHODOLOGY, RESULTS, AND ANALYSIS

4.1 Methodology

The entire project is worked on a Windows 10 operating system and the programming language used here is the Python. This is implemented in Jupyter notebook which is an open source web application. There are various packages of the machine learning have been used here which includes, Keras [9], this library is a neural networking open source library which runs above the TensorFlow. Pandas, this machine learning library helps in the analysis and the manipulation of the data. NumPy, a library for scientific computation for the python. Seaborn, this library helps in representing graphs which are statistical, generally a data visualization library. Once the necessary libraries have been imported in the Jupyter notebook, the pre-processed data set have to be fed into the LSTM model and then the two parts of the project namely the estimation and the prediction phases are evaluated to obtain the results.

4.2 Results

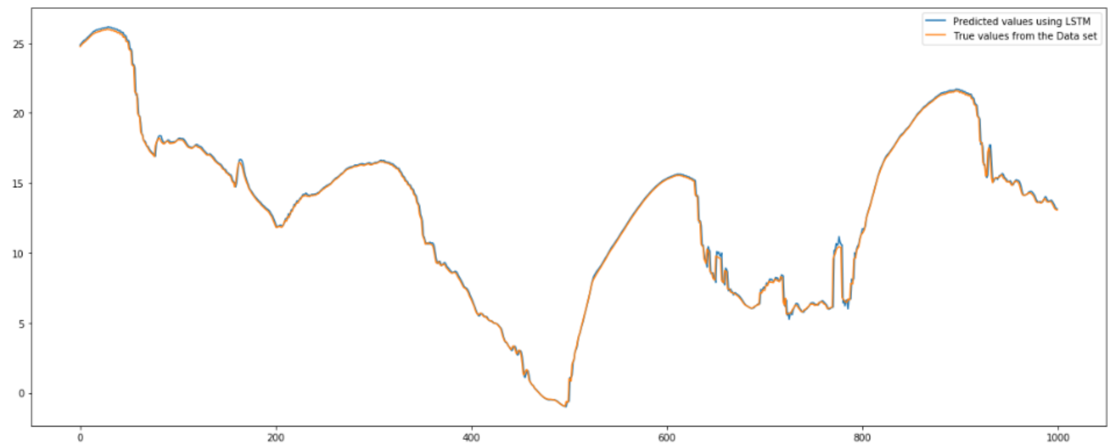
The section 4.2.1 consists of the results, mean squared error and the graph obtained for the Estimation part of the project. The section 4.2.2 consists of the results, mean percent error and graph obtained for the Prediction part of the project.

4.2.1 Estimation

The estimation makes use of the multivariate time series forecasting model along with the LSTM Model. Two experiments are performed on this part of the project. One year of data was divided into two batches of the train-test set for the experiment carried here. It is seen that for the first batch the train-test was split into a 70-30 batch which means 70 percent of the total 105,120 values of the data are used for the purpose of the training and was used to predict the remaining 30 percent of the data of the wind power generated. The results obtained in this experiment was considered good with the Variance 1.000 and Root Mean Square Error (RMSE) 0.175 is observed for the conducted experiment. The graph obtained is represented in figure 3.1.

```
In [11]:  #Plotting the first 1000 entries to see prediction
          from matplotlib import pyplot
          pyplot.figure(figsize=(20,8))
          pyplot.plot(Y_pred[:1000], label= 'Predicted values using LSTM')
          pyplot.plot(YTest[:1000], label= 'True values from the Data set')
          pyplot.legend()

Out[11]: <matplotlib.legend.Legend at 0x264b7576c88>
```




```
In [16]:  from sklearn.metrics import r2_score
          print('Variance : %.3f' % r2_score(YTest, Y_pred))

          Variance : 1.000

In [13]:  from sklearn.metrics import mean_squared_error
          print("Mean squared error: %.3f" % mean_squared_error(YTest, Y_pred))
          print("Root mean squared error: %.3f" % sqrt(mean_squared_error(YTest, Y_pred)))

          Mean squared error: 0.031
          Root mean squared error: 0.175
```

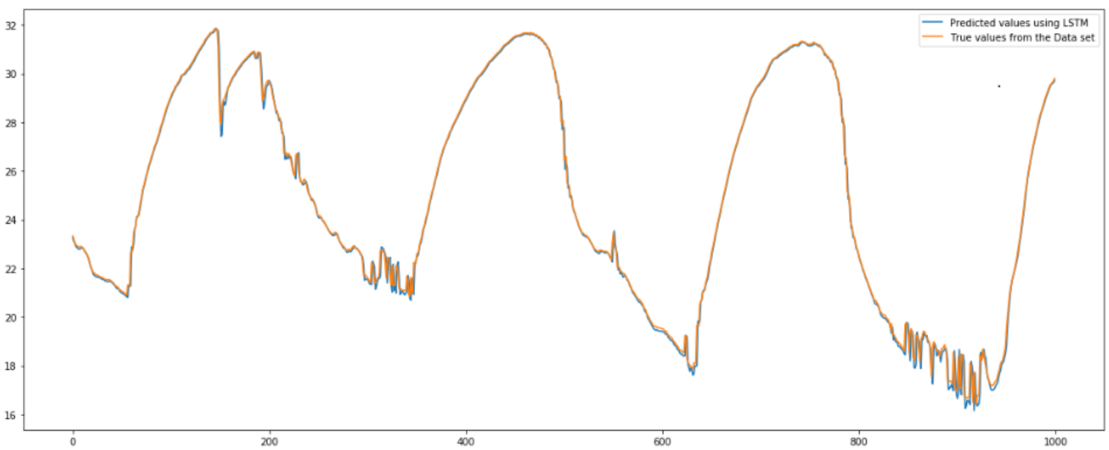
Figure 3.1: Estimation graph for 70-30 train-test split

The second experiment was performed as a 60-40 train-test split were 60 percent of the pre-processed data are used for the purpose of the training which was used to predict the remaining 40 percent of the data of the wind power being generated. The results obtained were outstanding

Variance of 1.000 and an RMSE 0.236 of is observed for the conducted experiment. The graph obtained is represented in figure 3.2.

```
In [11]:  #Plotting the first 1000 entries to see prediction
          from matplotlib import pyplot
          pyplot.figure(figsize=(20,8))
          pyplot.plot(Y_pred[:1000], label= 'Predicted values using LSTM')
          pyplot.plot(YTest[:1000], label= 'True values from the Data set')
          pyplot.legend()

Out[11]: <matplotlib.legend.Legend at 0x203c8bb6cc0>
```




```
In [12]:  from sklearn.metrics import r2_score
          print('Variance : %.3f' % r2_score(YTest, Y_pred))

          Variance : 1.000

In [13]:  from sklearn.metrics import mean_squared_error
          print("Mean squared error: %.3f" % mean_squared_error(YTest, Y_pred))
          print("Root mean squared error: %.3f" % sqrt(mean_squared_error(YTest, Y_pred)))

          Mean squared error: 0.056
          Root mean squared error: 0.236
```

Figure 3.2: Estimation graph for 60-40 train-test split

4.2.2 Prediction

In this part of the project, we predict the power generated by the system by means of analyzing the pattern of the previous years of the data. Data used by this prediction model will be just the Date Time index and the power generated by the system. Once the designing of the model is over, a series of experiments are carried out in order to determine the perfect look back size that the model should use, in order to get a good prediction of the power generated by the system. This is followed by the setting the number of epochs and certain parameters which fits right for the model and then experimenting to obtain the final predictions on the power generated. The results of the conducted experiments hold the hyperparameters as follows, the size of the batch as 1, number of epoch iterations as 6, the neuron number as 10 and a constant lag of 24. Five experiments were conducted on the prediction part and their result is as follows,

4.2.2.1 Predicting 12 hours of data

The model successfully predicted the future power generated for 12 hours with the same hyperparameters as mentioned above. The Root Mean Square Error was 16.036. Figure 4.1.1 depicts the computation of Root Mean Square Error and 4.1.2 represents the prediction for 12 hours of the power generated by the system.

```
Every 5th Minute Entry=119, Predicted_Value=15241.600541, Expected_Value=15669.000000
Every 5th Minute Entry=120, Predicted_Value=15641.847111, Expected_Value=15807.000000
Every 5th Minute Entry=121, Predicted_Value=15779.236350, Expected_Value=15878.000000
Every 5th Minute Entry=122, Predicted_Value=15849.791006, Expected_Value=15940.000000
Every 5th Minute Entry=123, Predicted_Value=15911.525213, Expected_Value=16000.000000
Every 5th Minute Entry=124, Predicted_Value=15971.444050, Expected_Value=16000.000000
Every 5th Minute Entry=125, Predicted_Value=15971.544758, Expected_Value=16000.000000
Every 5th Minute Entry=126, Predicted_Value=15971.817839, Expected_Value=16000.000000
Every 5th Minute Entry=127, Predicted_Value=15972.248718, Expected_Value=15834.000000
Every 5th Minute Entry=128, Predicted_Value=15806.819725, Expected_Value=15724.000000
Every 5th Minute Entry=129, Predicted_Value=15697.512091, Expected_Value=15645.000000
Every 5th Minute Entry=130, Predicted_Value=15619.307593, Expected_Value=15404.000000
Every 5th Minute Entry=131, Predicted_Value=15379.187349, Expected_Value=15277.000000
Every 5th Minute Entry=132, Predicted_Value=15253.133361, Expected_Value=15535.000000
Every 5th Minute Entry=133, Predicted_Value=15512.125971, Expected_Value=15767.000000
Every 5th Minute Entry=134, Predicted_Value=15745.144201, Expected_Value=15848.000000
Every 5th Minute Entry=135, Predicted_Value=15827.164197, Expected_Value=15905.000000
Every 5th Minute Entry=136, Predicted_Value=15885.160121, Expected_Value=16000.000000
Every 5th Minute Entry=137, Predicted_Value=15981.103371, Expected_Value=16000.000000
```

In [16]:
```
from sklearn.metrics import mean_squared_error
print("Root mean squared error: %.3f" % sqrt(mean_squared_error(Expected_Val, Y_pred)))
```

```
Root mean squared error: 16.036
```

Figure 4.1.1: Calculation of Root Mean Square Error

```
In [19]:  # Plot between the predicted power by the system and the number of hours
           from matplotlib import pyplot
           pyplot.figure(figsize=(20,8))
           pyplot.plot(Expectation_plot, label="True values")
           pyplot.plot(Pred_plot, label="Predicted values")
           pyplot.xlabel("Number of hours of future prediction/12 units=1 hour")
           pyplot.ylabel("Power generated by system in (kW)")
           pyplot.legend()
           pyplot.show()
```

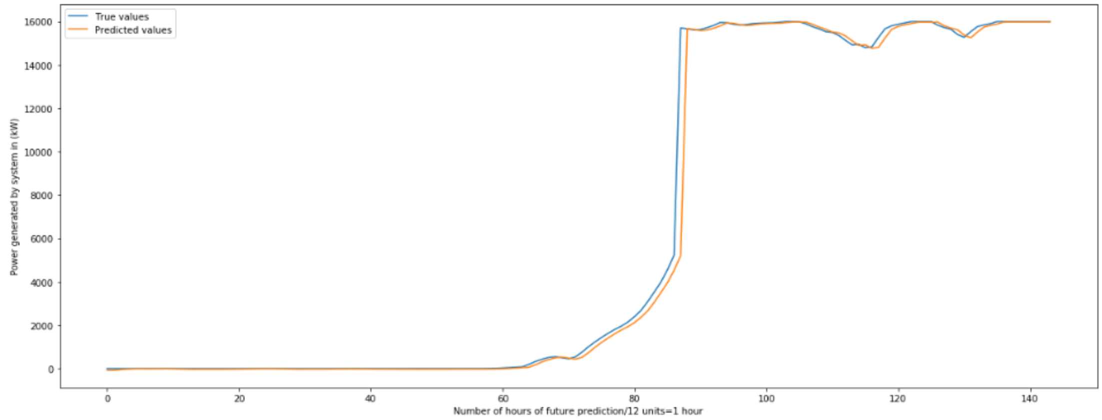


Figure 4.1.2: Prediction graph of power generated for 12 hours

4.2.2.2 Predicting 24 hours of data

The designed model predicted 24 hours of the power generated by the system and the Root Mean Square Error was found to be 31.495. Figure 4.2 represents the prediction graph of power generated for 24 hours.

```
Root mean squared error: 31.495

In [19]:  # Plot between the predicted power by the system and the number of hours
           from matplotlib import pyplot
           pyplot.figure(figsize=(20,8))
           pyplot.plot(Expectation_plot, label="True values")
           pyplot.plot(Pred_plot, label="Predicted values")
           pyplot.xlabel("Number of hours of future prediction/12 units=1 hour")
           pyplot.ylabel("Power generated by system in (kW)")
           pyplot.legend()
           pyplot.show()
```

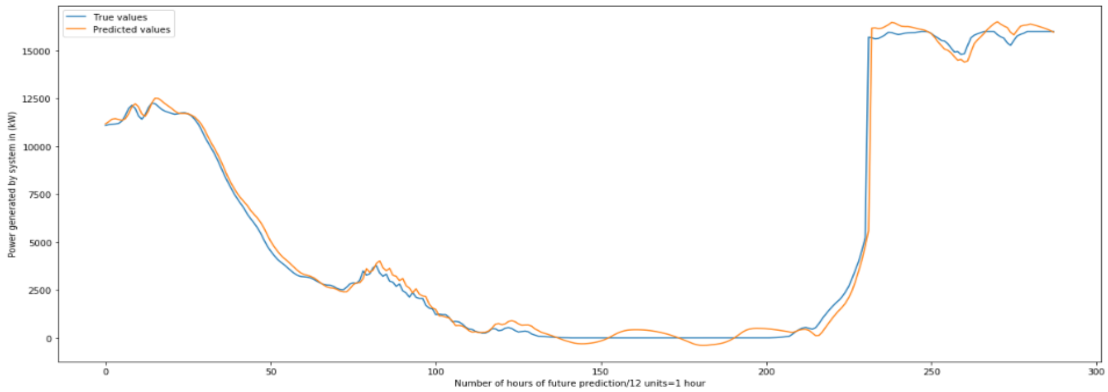


Figure 4.2: Prediction graph of power generated for 24 hours

4.2.2.3 Predicting 48 hours of data

The power generated by the system for 48 hours was calculated using the LSTM model and the Root Mean Square Error obtained was 239.609. Figure 4.3 represents the prediction graph of power generated for 48 hours.

```
Root mean squared error: 239.609
```

```
In [19]: # Plot between the predicted power by the system and the number of hours
from matplotlib import pyplot
pyplot.figure(figsize=(20,8))
pyplot.plot(Expectation_plot, label="True values")
pyplot.plot(Pred_plot, label="Predicted values")
pyplot.xlabel("Number of hours of future prediction/12 units=1 hour")
pyplot.ylabel("Power generated by system in (kW)")
pyplot.legend()
pyplot.show()
```

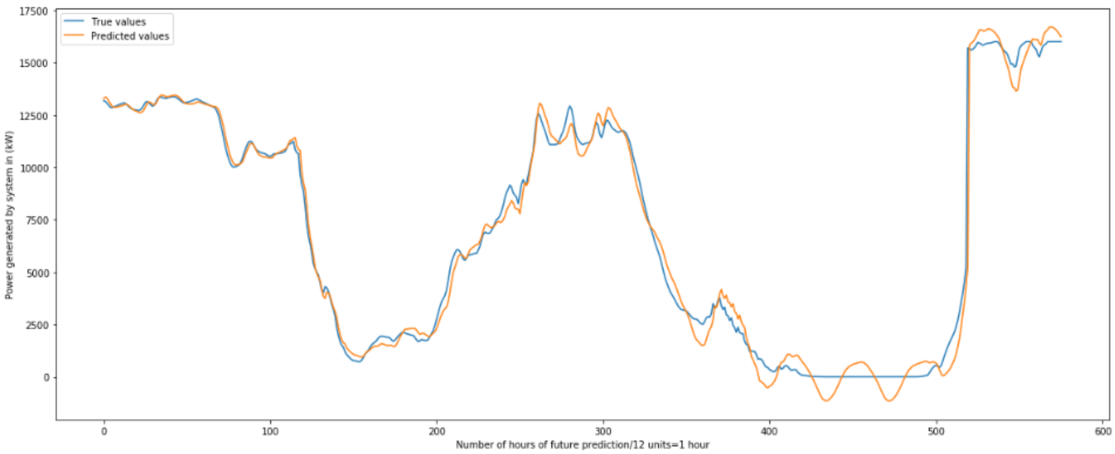


Figure 4.3: Prediction graph of power generated for 48 hours

4.2.2.4 Predicting 1 week of data

The power generated by the system for 7 days was obtained from the designed model. There were some brilliant results which were obtained with the Root Mean Square Error of 58.774. This was really great since the LSTM model was able to find the pattern for a week's data. Figure 4.4 represents the prediction graph of power generated for a week.

```
Root mean squared error: 58.774
```

```
In [19]:
# Plot between the predicted power by the system and the number of hours
from matplotlib import pyplot
pyplot.figure(figsize=(20,8))
pyplot.plot(Expectation_plot, label="True values")
pyplot.plot(Pred_plot, label="Predicted values")
pyplot.xlabel("Number of hours of future prediction/12 units=1 hour")
pyplot.ylabel("Power generated by system in (kW)")
pyplot.legend()
pyplot.show()
```

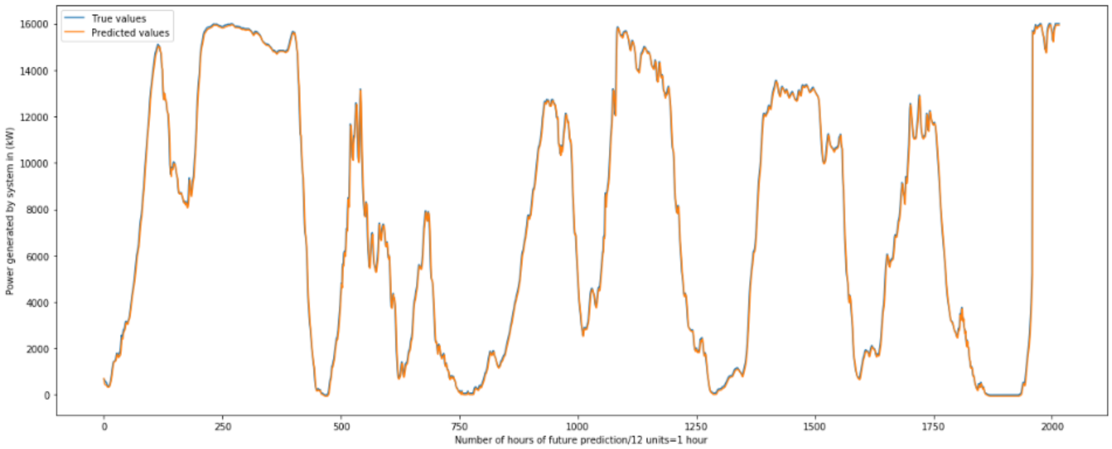


Figure 4.4: Prediction graph of power generated for a week

4.2.2.5 Predicting 1 month of data

One complete month of power generated by the system is predicted by the designed model and Root Mean Square Error of 56.040 was obtained. This is really good for a generation of wind power which solely depends on the speed of the wind which follows periodic patterns. Figure 4.5 represents the prediction graph of power generated for one month.

```
Root mean squared error: 56.040
```

```
In [19]:  # Plot between the predicted power by the system and the number of hours
          from matplotlib import pyplot
          pyplot.figure(figsize=(20,8))
          pyplot.plot(Expectation_plot, label="True values")
          pyplot.plot(Pred_plot, label="Predicted values")
          pyplot.xlabel("Number of hours of future prediction/12 units=1 hour")
          pyplot.ylabel("Power generated by system in (kW)")
          pyplot.legend()
          pyplot.show()
```

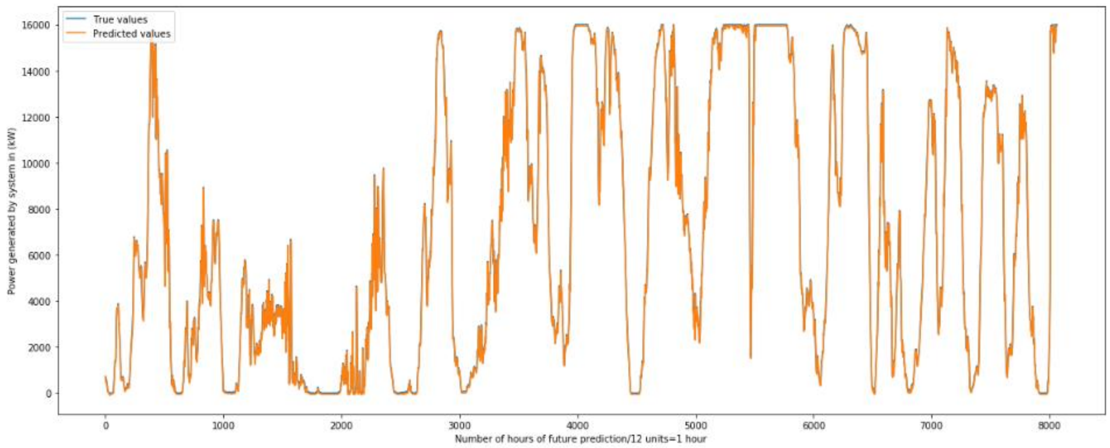


Figure 4.5: Prediction graph of power generated for one month

4.3 Analysis

In the experiments conducted above it is evident from the line graph that the Long Short-Term Memory model which is designed in the project has the ability to study the pattern of the data set and as a result it provides a good prediction of the power generated by the system for various time intervals in the future up to one month. Figure 4.6 represents a graph that is drawn with hours of the data predicted and the Root Mean Square Error obtained from the experiment on the x-axis and y-axis respectively. It is seen that the prediction was outstanding for our observations being considered for 12 hours, 24 hours, 1 week, 1 month.

The model was experimented to run with various epoch size in order obtain a good result because, when the speed of the wind is less than the specific range, then the machine learning model developed was not able to predict well.

In [10]:
```
Hours_of_data_predicted = [12,24,48,168,672]
Root_Mean_Squared_Error = [16.036,31.495,239.609,58.774,56.040]
```

In [11]:
```
import matplotlib.pyplot as plt
plt.xlabel('Hours of predicted data')
plt.ylabel('Root Mean Square Error')
plt.plot(Hours_of_data_predicted,Root_Mean_Squared_Error)
plt.show()
```

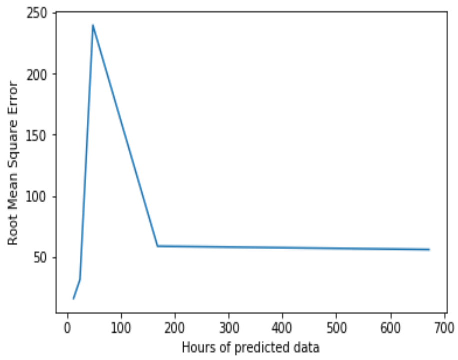


Figure 4.6: Hours of predicted data vs Root Mean Square Error

5. CONCLUSIONS

5.1 Summary

The observations of the annual and seasonal changes of the generated power were analyzed using the designed Long Short-Term Memory model. The designed model was able to study the specific patterns in the data and predict the results of the power generated by the system accordingly. The root mean square error for the estimation of the 70-30 set was found to be 0.175 and for the 60-40 train-test batch was found to be 0.236 for the estimation, which is very good for the problem like a weather forecasting using time series prediction. The prediction results for the future power generated was interesting for the given data set where the average RMSE obtained was 80.39. Thus, with the designed model, we can predict the future power generated. This will help in lowering the operating levels of the wind turbines in the farm which in turn will cut costs for the periodic replacements of the wind turbine blades and improves the capability of the turbine design which will yield higher productivity.

5.2 Contributions

Devising the algorithm plan for constructing the Long Short-Term memory and to make it run on an environment was complex. There were problems in designing the LSTM model. The training of the datasets takes longer than usual to train since it goes deep into the LSTM's stack of memory. The LSTM model provides results which overfit the data and hence many experiments were run on detecting the perfect look backs for actuating the next step of the model. There was also a difficulty in constructing the model for the data set used because the model assigned some random initial weights which affected the final predictions, that is, the power generated by the system.

The designed model will be very much effective in overcoming the problems of exploding gradient descent and the vanishing gradient descent. Thus, LSTM model overcomes the above two limitations and provides good prediction results for the power generated by the system. This future prediction will result in the efficient construction of the project, planning ahead maintenance. This in turn will lead to the reduced penalties and charges which is being incurred on the wind farms. The day ahead prediction of the power generated by the system will improve the trading of the wind power in the market and provide some excellent, useful knowledge in the real world.

5.3 Future Work

In this project, the predictions were made for an hour and use the obtained results to compute the next hour and repeating the same until the result is obtained. The future work will be in developing the model with a look ahead prediction. The model must be developed in such a way that it must be able to predict X hours or energy in one go. That is, the prediction has to be made for X hours without feeding the results to the memory and then rolling on to the next. The root mean square error and the obtained results must be compared with the previous experiments done in this project. Also, in order to study the patterns of every feature in the data set accurately, a new model can be designed which uses the combination of an algorithm of a Long Short-Term Memory and the decision tree which can improve the results considerably by a good margin.

REFERENCES

[1] Joao P.S Catalao, Hugo M.I Pousinho, Victor M.F Mendes, "An artificial neural network approach for short-term wind power forecasting in Portugal",15th International Conference on Intelligent System Applications to Power Systems (2009).

[2] Alexandre Costa, Antonio Crespo, Jorge Navarro, Gil Lizcano, Henrik Madsen, Everaldo Feitosa, "A review on the young history of the wind power short-term prediction", Renewable Sustainable Energy Revolution 12(6), 1725-1744(2008).

[3] Nils Andre Treiber, Justin Heinermann and Oliver Kramer, "Wind Power Prediction with Machine Learning", Computational sustainability chapter, January (2015).

[4] "A beginner's guide to neural networks and deep learning", https://skymind.ai/wiki/neural-network.

[5] Loris Foresti, Devis Tuia, Mikhail Kanevski, Alexei Pozdnoukhov, "Learning wind fields with multiple kernels", Stock. Env. Res. Risk Assess. 25(1), 51-66(2011)

[6] "Multi-step Time Series Forecasting with Long Short-Term Memory networks in Python",https://machinelearningmastery.com/multi-step-time-series-forecasting-long-short-term-memory-networks-python/.

[7] Ziqiao Liu, Wenzhong Gao, Yih-Huei Wan, Eduard Muljadi, "Wind Power Plant Prediction by Using Neural Networks", IEEE Energy Conversion Congress and Exposition (ECCE) (2012).

[8] "A free Encyclopedia article from Wikipedia about the Long Short-Term memory explaining its architecture", https://en.wikipedia.org/wiki/Long_short-term_memory .

[9] "Multi-variate Time Series Forecasting with Long Short-Term Memory in Keras", https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/ .

[10] Shuhui Li, Donald C. Wunsch, Edgar A. O'Hair and Michael G. Giesselmann, "Using Neural networks to estimate wind turbine power generation", IEEE Trans. On Energy Conversion, vol.16, pp.267-282, Sep.2001.

[11] "Wind turbine data set acquired from the website of National Renewable Energy Laboratory (NREL)", https://www.nrel.gov/grid/western-wind-data.html .

[12] Andrew Kusiak, Haiyang Zheng and Zhe Song, "Short-term prediction of wind farm power-A data mining approach", IEEE Trans on Energy Conversion, vol.24, no.1, pp.125-135, Mar. 2009.

[13] Mohamed Ahmed Mohandes, Shafiqur Rehmann, Talal O. Halawani, "A neural networks approach for wind speed prediction", Renewable Energy 13(3), 345-354 (1998).

[14] A introduction to the concept of Recurrent Neural Network towards Machine Learning "https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5".

[15] A guide for the time series prediction with the Long Short Term Memory in kaggle, "https://www.kaggle.com/jphoon/bitcoin-time-series-prediction-with-lstm".

[16] Saurabh S. Soman, Hamidreza Zareipour, O.P Malik , Paras Mandal, "A review of wind power and wind speed forecasting methods with different time horizons", North American Power Symposium(NAPS), pp. 1-8 (2010).