PREDICTING THE ENERGY OUTPUT OF WIND TURBINES BASED ON WEATHER CONDITIONS

A MINI PROJECT-I REPORT

Submitted by

ROHITH VISHWA V.S	**(1901204)**
SHARMILI B.J	**(1901217)**

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

SRI RAMAKRISHNA ENGINEERING COLLEGE
[Educational Service: SNR Sons Charitable Trust]
[Autonomous Institution, Accredited by NAAC with 'A' Grade]
[Approved by AICTE and Permanently Affiliated to Anna University, Chennai]
[ISO 9001:2015 Certified and All Eligible Programmes Accredited by NBA]
Vattamalaipalayam, N.G.G.O. Colony Post,

COIMBATORE – 641 022
ANNA UNIVERSITY: CHENNAI 600 025

MAY 2021

SRI RAMAKRISHNA ENGINEERING COLLEGE

COIMBATORE - 641022

BONAFIDE CERTIFICATE

Certified that this Project Report **“Predicting The Energy Output Of Wind Turbines Based On Weather Conditions”** is the bonafide work of

Rohith Vishwa V.S	**(1901204)**
Sharmili B.J	**(1901217)**

who carried out the project work under my supervision.

SIGNATURE

Mrs.S.Birundha

SUPERVISOR

Assistant Professor

Department of Computer Science and Engineering,

Sri Ramakrishna Engineering College,

Vattamalaipalayam,

N.G.G.O Colony Post,

Coimbatore – 641 022.

SIGNATURE

Dr. A. Grace Selvarani

HEAD OF THE DEPARTMENT

Department of Computer Science and Engineering,

Sri Ramakrishna Engineering College,

Vattamalaipalayam,

N.G.G.O Colony Post,

Coimbatore – 641 022.

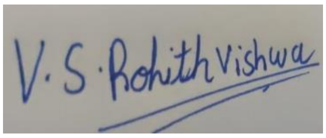
Submitted for the Autonomous project VIVA - VOICE held on____________.

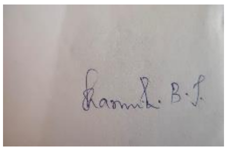
INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We affirm that the project work titled " **PREDICTING THE ENERGY OUTPUT OF WIND TURBINES BASED ON WEATHER CONDITIONS** " being submitted in partial fulfillment for the award of Bachelor of Engineering is the original work carried out by us. It has not formed the part of any other mini project work submitted for award of any degree or diploma, either in this or any other University.



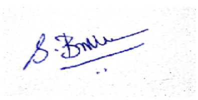


(Signature of the Candidates)

ROHITH VISHWA V.S (1901204)

SHARMILI BJ (1901217)

I certify that the declaration made above by the candidates is true.



(Signature of the guide)

Mrs.S.Birundha,

Assistant Professor,

Department of CSE.

ACKNOWLEDGEMENT

We express our gratitude to **Sri. D. LAKSHMINARAYANASWAMY,** Managing Trustee, **Sri. R. SUNDAR,** Joint Managing Trustee, SNR Sons Charitable Trust, Coimbatore for providing excellent facilities to carry out our project.

We express our deepest gratitude to **Dr. N. R. ALAMELU,** Principal, for her valuable guidance and blessings.

We thank **Dr. A. GRACE SELVARANI,** Professor and Head, Department of Computer Science and Engineering who modeled us both technically and morally for achieving great success in life.

We sincerely thank our Project Coordinator, **Dr. T. GUHAN,** Assistant Professor (Sl.Gr), Department of Computer Science and Engineering for his great inspiration.

Words are inadequate to offer thanks to our respected guide. We wish to express our sincere thanks to **Mrs. S. BIRUNDHA,** Assistant Professor, Department of Computer Science and Engineering, who gave constant encouragement and support throughout this project work and who made this project a successful one.

We also thank all the staff members and technicians of our Department for their help in making this project a successful one.

TABLE OF CONTENTS

ABSTRACT

Tropical countries like India have great potential as a major source of green energy. However, in order to utilize the energy completely, proper provisioning is required. So there exists a need to forecast and estimate the energy output of windmills. Forecasting wind energy at station level is a big challenge as it gives only macro level information. With the recent phenomenal growth of applications in artificial intelligence (AI), it is also possible to use data-driven models based on AI, especially deep learning for short-term forecasting of wind energy. In this document we have proposed a forecasting methodology using the Long Short -Term Memory (LSTM) model, which is a deep learning approach for time series data analysis. The simulations using these models have been validated against the true observation at station scale.

The capability of the proposed method is demonstrated through various error matrices and found to have better performance. The output of the proposed method has the potential to improve the short-term wind speed prediction capability at station level.

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

Over the last decade, increasing environmental pollution and depletion of the fossil fuel reserves has been the most heard terms. This has encouraged the search of clean and pollution free sources of energy. Wind energy is one among them and also a significant source of energy. Meteorological parameters such as wind speed, relative humidity and air temperature are taken into account.

Due to the presence of parameter estimation and prediction, it is well-known that the usage of neural networks comes into play. Neural networks are nonlinear processing systems composed of interconnected units which are called neurons. This helps us to perform ill-defined problems and complex tasks as well to solve complex problems.

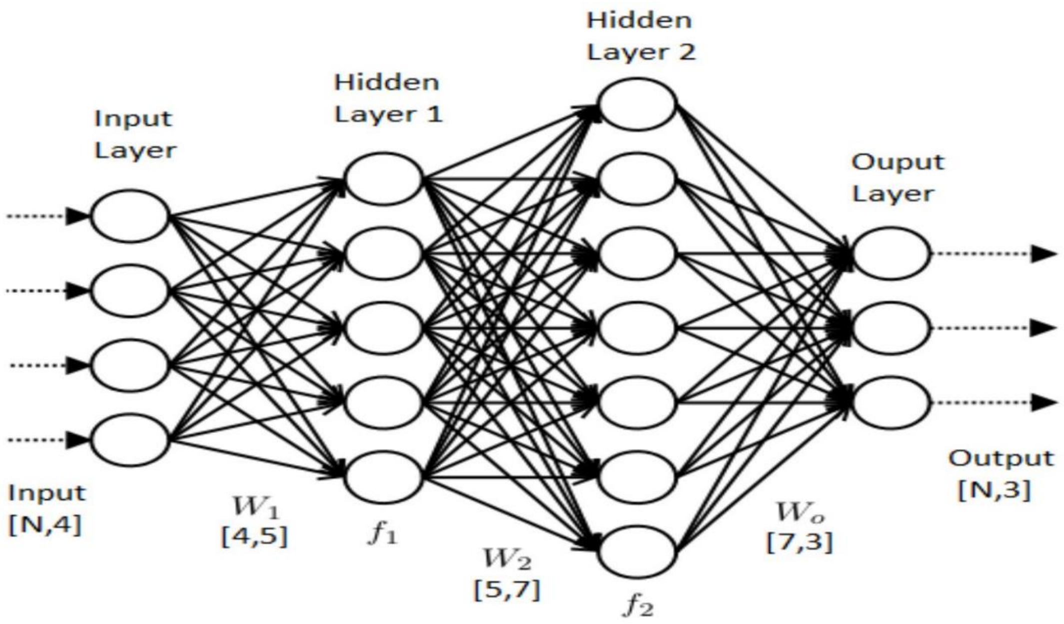


Figure 1.1: Layers Of Neural Networks.

Specifically, neural networks are useful when it is necessary to build a model from existing data, when it is necessary to simulate the behavior of systems characterized by noisy and incomplete data. With the usage of proper experimental data in combination with a lstm algorithm. It is known that wind speed has a major influence on the power output; moreover it also depends on air density, which is in turn affected by the relative humidity.

CHAPTER 2

LITERATURE SURVEY

The university of oldenburg developed an approach called previento. They use the Deutschland model or nowadays the Lokalmodell (LM) of the German Weather Service (DWD) as the NWP model. A good overview over the parameters and models influencing the result of a meteorological short-term forecasting system has been given by Mönnich. He found that the most important of the various submodels being used is the model for the atmospheric stability.

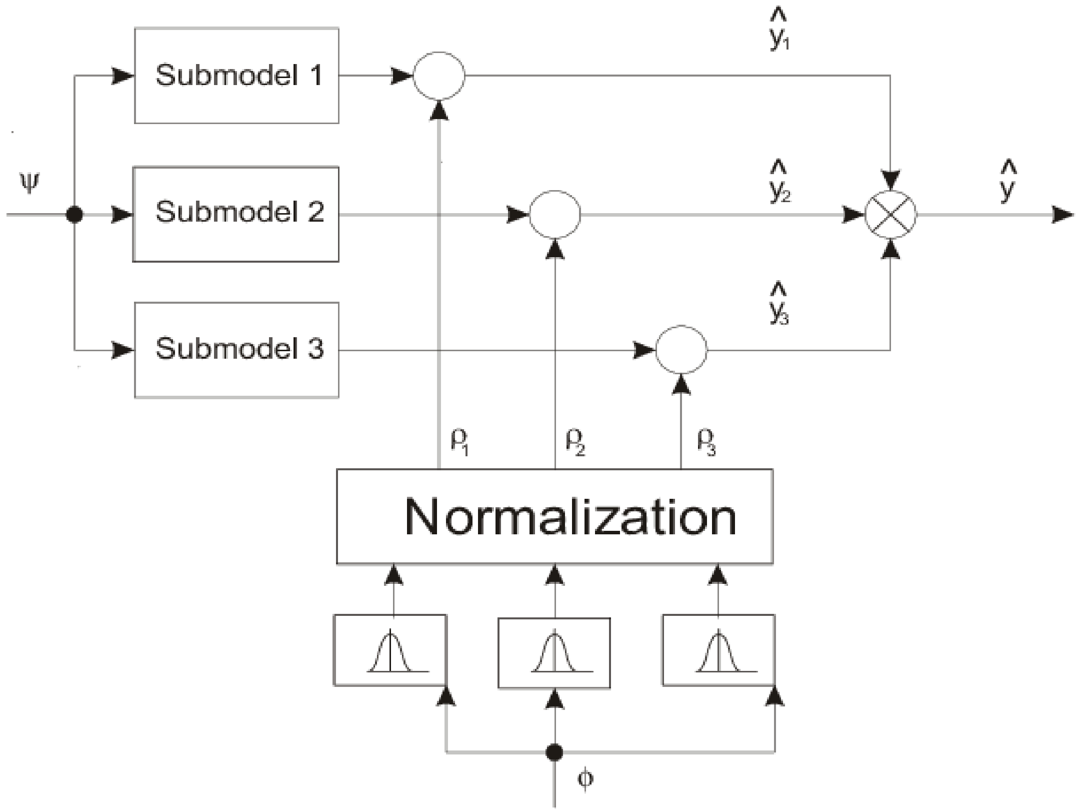


Figure 2.1: Lokal model or Deutschlandmodell

The NWP model changed frequently; the use of a recursive technique was recommended. A large influence was found regarding the power curve. The theoretical power curve given by the manufacturer and the power curve found from data could be rather different. Actually, even the power curve estimated from data from different years could show strong differences. The latter might be due to a complete overhaul of the turbine. The largest influence on the error was deemed to come from the NWP model itself. LocalPred and RegioPred are a family of tools developed by Martí Perez (formerly CIEMAT, now CENER). It involves adaptive optimization of the NWP input, time series modelling, mesoscale modelling with MM5, and power curve modelling. He could show for a case of rather complex terrain near Zaragoza (Spain), that the resolution of HIRLAM was not good enough to resolve the local wind patterns.

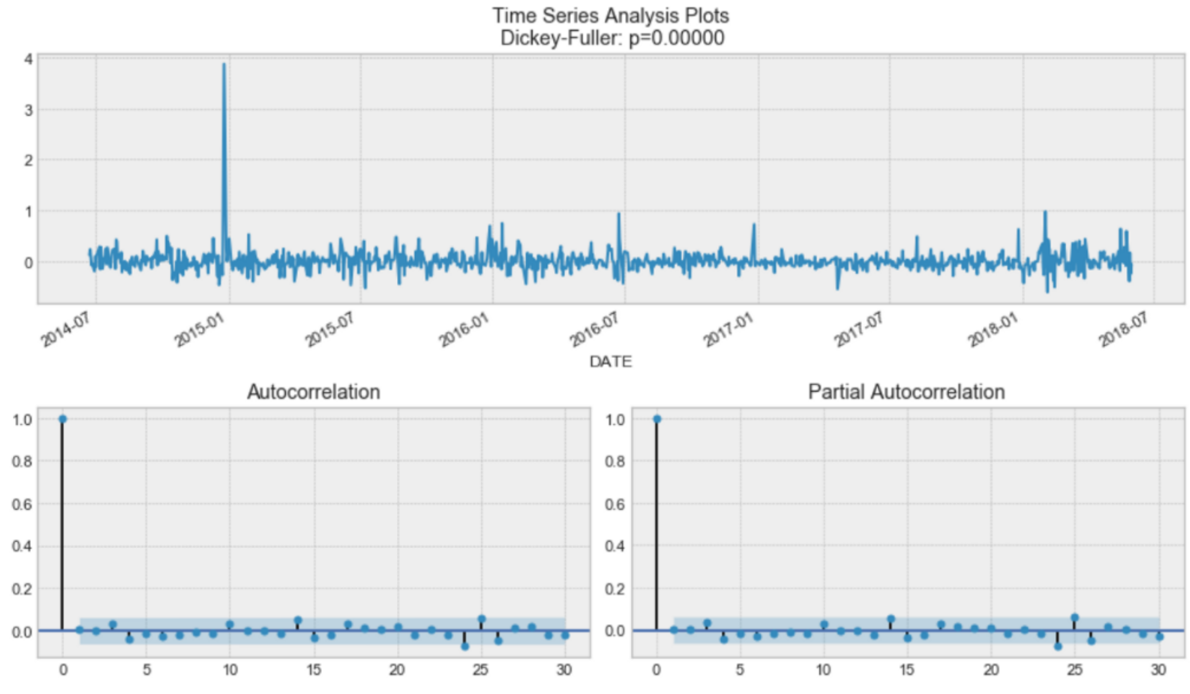


Figure 2.2: Time Series Aanalysis Plots

A new approach was described by Jorgensen, they integrate the power prediction module within the NWP itself. They call it HIRPOM (HIRlam Power prediction Model). Moehrlen has looked at the resolution needed for successful application of NWP forecasting. In different runs with horizontal model resolutions of 30 km, 15 km, 5 km and 1.4 km for two months in January 2001, the most common statistical accuracy measures did improve only slightly with higher resolution. However, peak wind speeds were closer to the measured values for the high-resolution forecasts. For the higher resolution forecasts, the best model layers were ones closer to the ground than in the coarser models.

LANDBERG MODEL

Landberg developed a short-term prediction model based on physical reasoning similar to the methodology developed for the European Wind Atlas. It is the perfect example for the model chain in the introduction. Landberg used the Danish or Riso version for all the parts in the model: the HIRLAM model of the DMI as NWP input, the Wasp model from Riso to convert the wind to the local conditions and the Riso PARK model to account for the lower output in a wind park due to wake effects. He found that for the MOS to converge, about 4 months' worth of data were needed (which might not be available when setting up the model for a new customer). If the wind from one of the upper NWP levels is used, the procedure is as follows: from the geostrophic wind and the local roughness, the friction velocity u* is calculated using the geostrophic drag law. This is then used in the logarithmic height profile, again together with the local roughness. If the wind is already the 10m-wind, then the logarithmic profile can be used directly.

For the model power prediction for the total region is calculated as a sum of the predictions for the sub areas. The final prediction of the wind power production for the total region is calculated as a weighted average of the predictions.

LONG SHORT -TERM MEMORY

Long Short-Term Memory (LSTM) is a Recurrent Neural Network (RNN) architecture used in the field of deep learning. unlike standard feed forward neural networks, LSTM has feedback connections. It can not only process single data points(such as images),but also entire sequence of data.

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three *gates* regulate the flow of information into and out of the cell.

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.

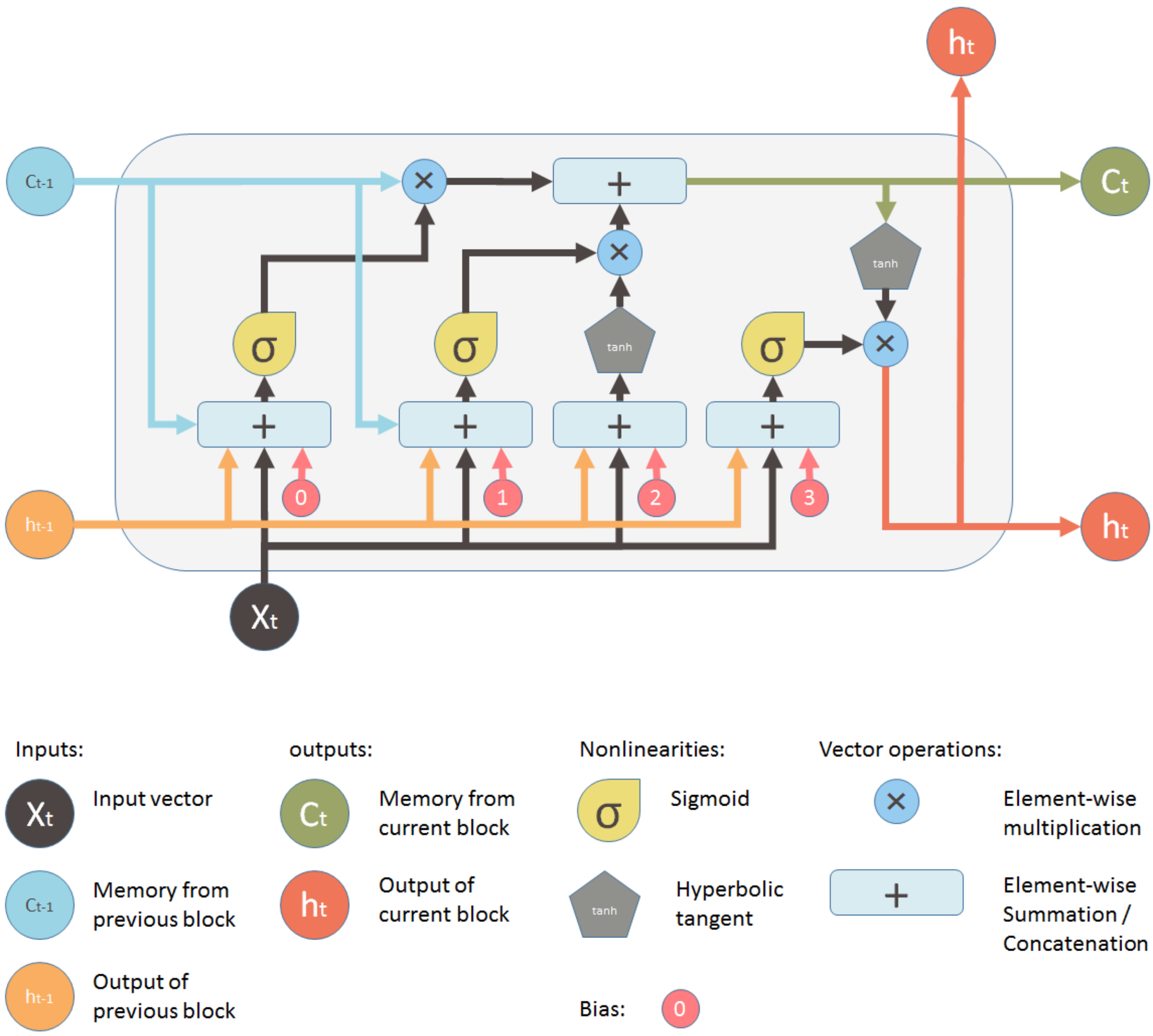


Figure 2.3:LSTM Model

2.1 EXISTING SYSTEM AND ITS DIS ADVANTAGES

A precise and accurate forecast needs to overcome problems of variable energy production caused by fluctuating weather conditions.

Accurate and reliable wind speed forecasts are significant challenge due to its high rates of change, highly nonlinear behavior with no typical patterns, and dependency on atmospheric pressure, temperature which results in large uncertainties of wind speeds.

2.2 PROPOSED SYSTEM AND ITS ADVANTAGES

The plan is to interpret this problem as a time series forecasting problem. Although the wind speed is highly unpredictable and changes a lot, it follows a particular pattern for a certain period of time.

To carry out time series data prediction LSTM is used.

Long Short -Term Memory (LSTM) machine learning model which is best known for time series data prediction, is used to learn these patterns in wind and make a prediction to learn these patterns in wind and make prediction about the energy output.

This model has been used to learn these patterns in wind speed and make a prediction about the energy output. LSTM creates both long and short term memory components. It is always a great tool for sequences. The constant error back propagation within memory cells results in LSTM's ability to bridge very long-time lags.

In finite state automata or hidden markov models LSTM does not require an prior choice of a finite number of states. In principle it can deal with unlimited state numbers. Relative insensitivity to gap length gives an **advantage** to **LSTM** over alternative RNNs

CHAPTER 3

PROBLEM STATEMENT

3.1 OBJECTIVE OF THE WORK

To get a suitable dataset and build a machine learning model using an suitable algorithm to predict the energy output of wind turbines.

3.2 PROBLEM DESCRIPTION

The problem is to build a model for prediction of energy in wind turbines. It is a known fact that wind energy is highly depended on wind speed. As the wind speed does not follow a certain pattern for long time, the prediction becomes even tougher. The built model and the chosen algorithm should be able to solve these problems in order to get a forecasting output.

3.3 MODULE DESCRIPTIONS

1. Data collection :

   An apt dataset should be chosen in order to meet the requirements. The features in the dataset is studied and checked for the relevancy to build the model.

   A dataset featuring the Date and Time, Pressure, Wind Direction, Wind Speed and Pressure generated by the system with 26281 epochs.

DateTime	Pressure | (atm)	Wind direction | (deg)	Wind speed | (m/s)	Power generated by system | (kW)
2007-01-01 00:00:00	0.890467	104	3.429	0
2007-01-01 01:00:00	0.890583	106	3.579	0
2007-01-01 02:00:00	0.890366	102	4.307	1986.68
2007-01-01 03:00:00	0.890326	99	4.562	2597.61
2007-01-01 04:00:00	0.890549	98	4.553	2555.46
2007-01-01 05:00:00	0.890821	98	4.695	2922.34
2007-01-01 06:00:00	0.890946	98	4.713	2972.42
2007-01-01 07:00:00	0.890966	98	4.491	2396.37
2007-01-01 08:00:00	0.890917	96	4.549	2498.91
2007-01-01 09:00:00	0.890871	99	4.315	1955.02
2007-01-01 10:00:00	0.890754	103	3.675	665.675
2007-01-01 11:00:00	0.890708	111	3.343	0
2007-01-01 12:00:00	0.890653	121	3.377	0
2007-01-01 13:00:00	0.89048	118	3.39	0
2007-01-01 14:00:00	0.890236	123	3.443	0
2007-01-01 15:00:00	0.889675	146	5.877	7836.49
2007-01-01 16:00:00	0.889413	144	5.925	8120.03
2007-01-01 17:00:00	0.889285	141	5.745	7399.98
2007-01-01 18:00:00	0.889011	137	5.413	5628.63
2007-01-01 19:00:00	0.888572	137	4.728	3121.51
2007-01-01 20:00:00	0.888269	141	4.627	2802.61
2007-01-01 21:00:00	0.888004	149	4.7	2897.21
2007-01-01 22:00:00	0.887654	168	4.349	2099.38
2007-01-01 23:00:00	0.887518	196	3.816	926.474
2007-01-02 00:00:00	0.887388	224	4.615	2762.95
2007-01-02 01:00:00	0.887207	240	5.688	7032.75
2007-01-02 02:00:00	0.886902	234	9.961	39739.9
2007-01-02 03:00:00	0.886431	236	13.915	52866.1
2007-01-02 04:00:00	0.886067	235	14.656	52780.7
2007-01-02 05:00:00	0.885695	235	16.009	52682
2007-01-02 06:00:00	0.88514	234	17.109	52669.9

Figure 3.3.1: Sample Dataset

2. Exploratory Data Analysis:

    The obtained data is visualized using appropriate graphs and studied.

3. Preprocessing and Feature scaling :

    Feature scaling is a method used to normalize the range of independent variables. The outlier is found by plotting the graph. The found outliers are then treated.

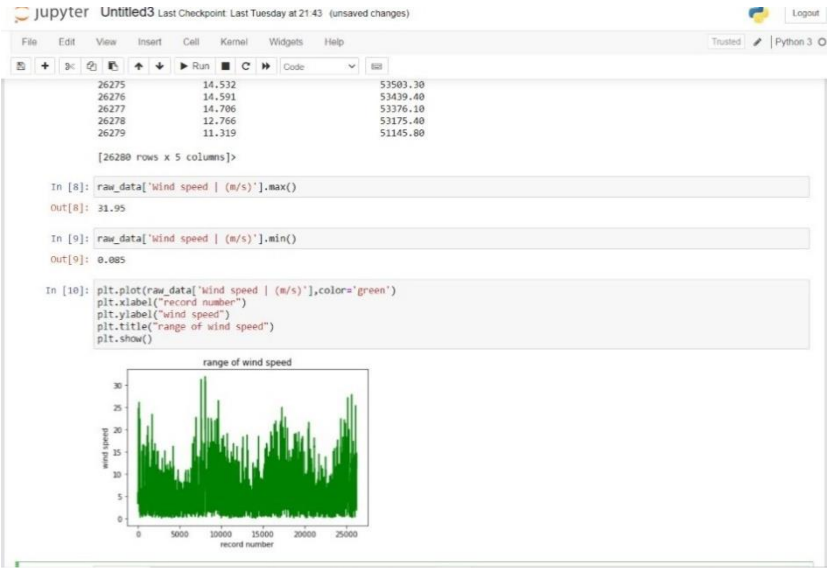


Figure 3.3.3: Preprocessing and Future Scaling.

4. Machine learning algorithm:

    LSTM algorithm has been chosen and the dataset is sent in the input gate of it for the algorithm to analyze it.

    The model uses 26281 epochs and a batch size of 120 for the algorithm to analyse the energy output.

5.Deployment in Google Colab:

    The model is built in Google Colab using LSTM algorithm.

CHAPTER 4

SYSTEM SPECIFICATIONS

4.1 HARDWARE COMPONENTS

As the project is completely based on model machine learning there is no significant hardware involved in this.

4.2 SOFTWARE COMPONENTS

Tools used: Google Colab

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

Language used: python3

Libraries used:

1. Pandas - It is mainly used for data analysis. Pandas allows importing data from various file formats such as comma-separated values, JSON, SQL, Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features.

2. NumPy - NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

3. matplotlib.pyplot - matplotlib. pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

4. Seaborn - Seaborn is a library in Python predominantly used for making statistical graphics. Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

5. Windrose - Windrose is a Python library to manage wind data, draw windroses (also known as polar rose plots), and fit Weibull probability density functions. The initial use case of this library was for a technical report concerning pollution exposure and wind distributions analyzes.

6. Datetime - Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

CHAPTER 5

RESULTS

We came up with the idea of predicting the energy output of wind turbines based on its weather conditions which serves the purpose of predicting the energy output in advance and so the natural energy of the wind power is saved for future use. The energy output predicted in this technique proves the best state running of the wind turbines based on its average loss.

Preprocessing:

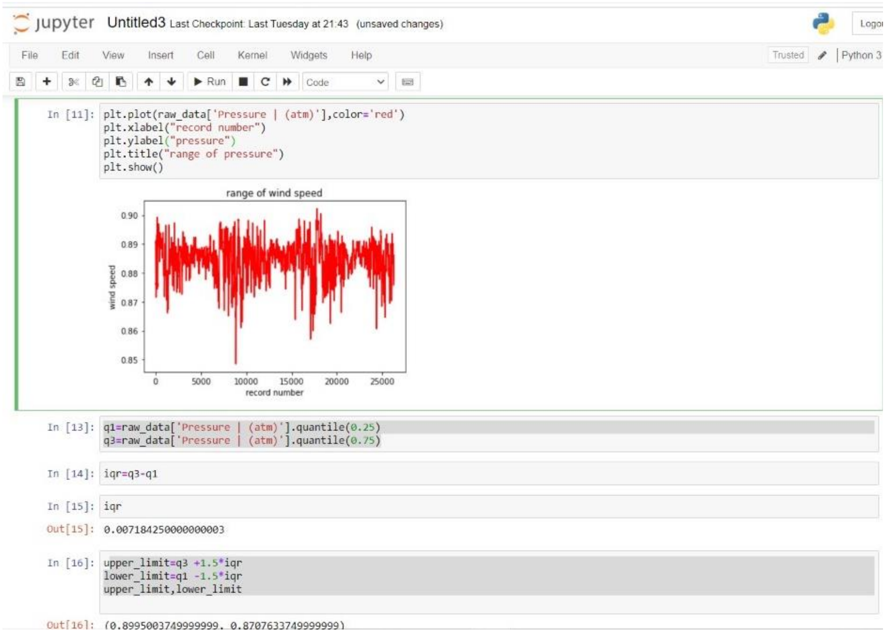


Figure 5.1: outliers

Outliers were found in the dataset as showed in the previous graph and were removed. The following image shows the graph after the removal of outliers.

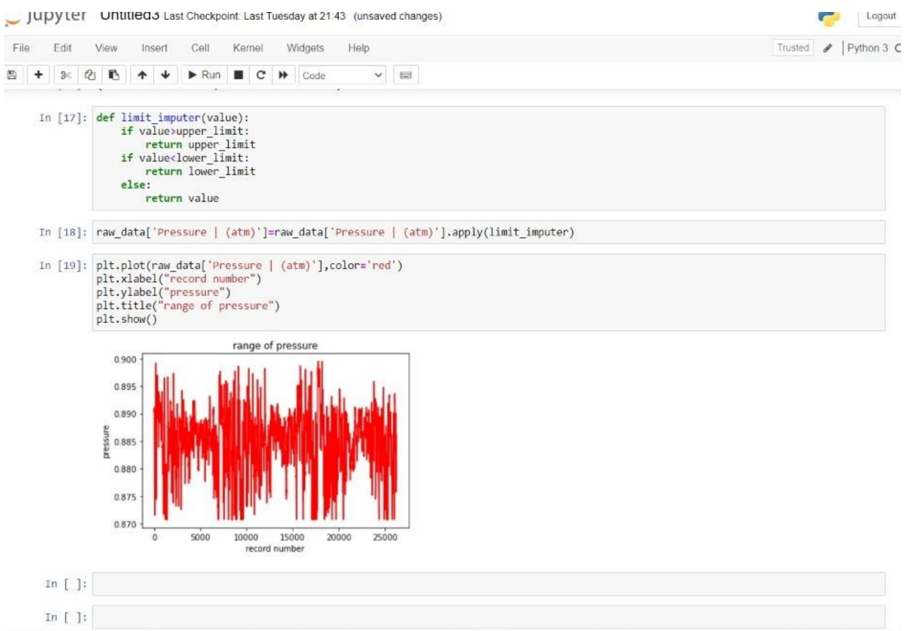
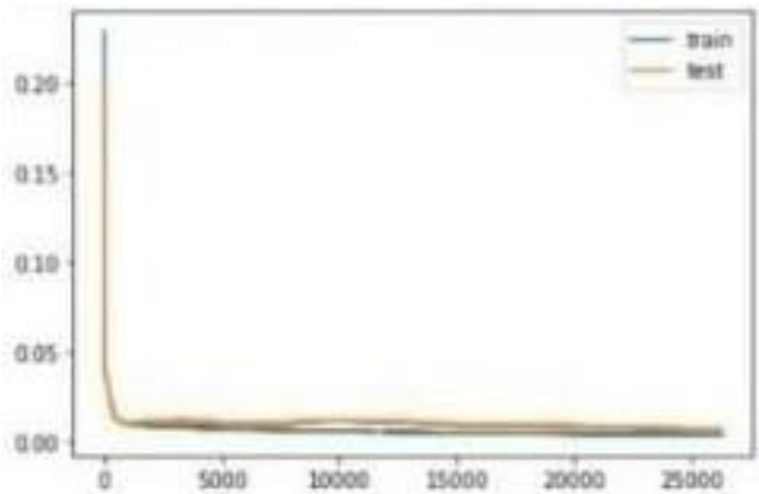


Figure 5.2: data after removing Outliers.

After implementing the the model we obtained an average loss of 0.0175. By which we were able to figure out the accuracy as 98.25%.



CHAPTER 6

CONCULSION

One of the ultimate goals of every wind power prediction model is to estimate the wind power output as early and as accurately as possible. Wind power will become more attractive for system and market operators as NWP model accuracy improves and as easier to use forecasting techniques are developed. Wind power prediction tools are invaluable because they enable better dispatch, scheduling and unit commitment of thermal generators, hydro plant and energy storage plant and more competitive market trading as wind power ramps up and down. The rapid expansion of wind generation capacity in the past 15 years has created demand for advances in wind forecasting techniques. The rapid expansion of wind generation capacity in the past 15 years has created demand for advances in wind forecasting techniques. Overall accurate wind power prediction reduces the financial and technical risk of uncertainty of wind power production for all electricity market participants. Continuing innovations in statistical and machine learning prediction techniques have also paid dividends, particularly for forecasting on very short term and short-term timescales.

Our project will be of great use to know the correct method of preventing energy loss and predicting the energy output.

CHAPTER 7

REFERENCES

[1] Alencar BDD, Affonso CDM, “Oliveira RCLDO et al (2017) Different models for forecasting wind power generation: case study”. Energies 10(12):1976.

[2] Barbounis TG, Theocharis JB, Alexiadis MC, Dokopoulos PS (2006) “Long-term wind speed and power forecasting using local recurrent neural network models”. IEEE Transactions on Energy Conversion 21(1): 273–284.

[3] Liu H, Xiwei M, Yanfei L (2018) “Smart deep learning based wind speed prediction model using wavelet packet decomposition, convolutional neural network and convolutional long short term memory network”. Energy Convers Manag 166:120–131. 2018.04.021

[4] Makridakis S, Spiliotis E, Assimakopoulos V (2018) “Statistical and machine learning forecasting methods: concerns and ways forward”. PLoS One 13(3):e0194889.

[5] Mohandes MA, Halawani TO, Hussain RS, Ahmed A (2004) “Support vector machines for wind speed prediction”. Renew Energy 29(6):939–947.

[6] Pinto T, Ramos S, Sousa MT, Zita V (2014) “Short-term wind speed forecasting using support vector machines”.Computational intelligence in dynamic and uncertain environments (CIDUE), IEEE symposium.

[7] REN21, “Renewable Energy Policy Network for the 21st Century”, 2018.

[8] GWEC, “Global Wind Report”, 2017 .

[9] GWEC, “Global Wind Report”, 2018.

CHAPTER 8

APPENDIX

8.1 SAMPLE CODING:

```
# split into train and test sets
values = reframed.values
n_train_hours = 365 * 24
train = values[:n_train_hours, :]
test = values[n_train_hours:, :]
# split into input and outputs
train_X, train_y = train[:, :-1], train[:, -1]
test_X, test_y = test[:, :-1], test[:, -1]
# reshape input to be 3D [samples, timesteps, features]
train_X = train_X.reshape((train_X.shape[0], 1, train_X.shape[1]))
test_X = test_X.reshape((test_X.shape[0], 1, test_X.shape[1]))
print(train_X.shape, train_y.shape, test_X.shape, test_y.shape)
from sklearn.metrics import mean_squared_error
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
# design network
model = Sequential()
model.add(LSTM(50, input_shape=(train_X.shape[1], train_X.shape[2])))
model.add(Dense(1))
model.compile(loss='mae', optimizer='adam')
# fit network
history = model.fit(train_X, train_y, epochs=26281, batch_size=120, validation_d
ata=(test_X, test_y), verbose=2, shuffle=False)
# plot history
plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
plt.legend()
plt.show()
# convert series to supervised learning
def series_to_supervised(data, n_in=1, n_out=1, dropnan=True):
n_vars = 1 if type(data) is list else data.shape[1]
df1 = pd.DataFrame(data)
```

```
cols, names = list(), list()
# input sequence (t-n, ... t-1)
for i in range(n_in, 0, -1):
cols.append(df1.shift(i))
names += [('var%d(t-%d)' % (j+1, i)) for j in range(n_vars)]
# forecast sequence (t, t+1, ... t+n)
for i in range(0, n_out):
cols.append(df1.shift(-i))
if i == 0:
names += [('var%d(t)' % (j+1)) for j in range(n_vars)]
else:
names += [('var%d(t+%d)' % (j+1, i)) for j in range(n_vars)]
# put it all together
agg = concat(cols, axis=1)
agg.columns = names
# drop rows with NaN values
if dropnan:
agg.dropna(inplace=True)
return agg
values = df.values
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(values)
# frame as supervised learning
reframed = series_to_supervised(scaled, 1, 1)
print(reframed.head())

# design network
model = Sequential()
model.add(LSTM(50, input_shape=(train_X.shape[1], train_X.shape[2])))
model.add(Dense(1))
model.compile(loss='mae', optimizer='adam')
# fit network
history = model.fit(train_X, train_y, epochs=26281, batch_size=120,
validation_data=(test_X, test_y), verbose=2, shuffle=False)
```