|  |
| --- |
| Vellore Institute of Technology, Chennai    DATABASE SYSTEMS: DESIGN AND IMPLEMENTATION  CSE5003  Project on Wind Data Cleaning and Prediction  Submitted by  Kalpalathika N 20MAI1006  Pattan Afrid Ahmed 20MAI1016  Poorvaja P 20MAI1009  MTECH CSE (AI and ML)  Under the guidance of  Dr. [Sajidha SA](https://scholar.google.com/citations?user=gfiUz5IAAAAJ&hl=en&oi=ao) |

TABLE OF CONTENTS

[ABSTRACT 3](#_Toc57040205)

[1 INTRODUCTION 4](#_Toc57040206)

[2 REQUIREMENTS 5](#_Toc57040207)

[3 DESCRIPTION OF DATA SET 6](#_Toc57040208)

[4 LITERATURE REVIEW 7](#_Toc57040209)

[5 PROPOSED WORK 8](#_Toc57040210)

[5.1 Wind Data Cleaning Module 8](#_Toc57040211)

[5.2 Database module 10](#_Toc57040212)

[5.3 Prediction Module 12](#_Toc57040213)

[5.3.1 Long Short-Term Memory (LSTM) 12](#_Toc57040214)

[5.3.2 Random Forest Regressor (RFR) 12](#_Toc57040215)

[5.3.3 Fbprophet (FP) 13](#_Toc57040216)

[5.3.4 Auto Regressive Integrated Moving Average (ARIMA) 13](#_Toc57040217)

[6 PERFORMANCE EVALUATION 15](#_Toc57040218)

[7 ALGORITHMS 16](#_Toc57040219)

[8 SCREENSHOTS 18](#_Toc57040220)

[9 CONCLUSION 23](#_Toc57040221)

[10 REFERENCES 24](#_Toc57040222)

# ABSTRACT

Renewable energy resources are environmentally clean and found abundant in nature. To increase green energy penetration and reduce global warming, wind energy is one of the promising technologies that is developing and promising to satisfy our future energy needs. In this project, the data collected from the mast are analyzed for missing or erroneous values. Data cleaning is done to improve the quality of the data and to improve the accuracy and reliability in the wind speed prediction result. The data is stored in the PostgreSQL database for future reference. The cleaned data is passed as input for four models namely Autoregressive Integrated Moving Average (ARIMA) [1], fbprophet, Long-Short Time Memory (LSTM) [8] and Random Forest Regressor (RFR) [3] to predict the wind speed and their results are compared.

# INTRODUCTION

Renewable energy sources are available in abundance naturally and can be utilized for power generation to satisfy industrial and commercial needs. The renewable sector is witnessing a phenomenal growth, and accurate prediction is essential. Wind energy is proposed as an important source of alternative energy in recent years. It has more advantages with respect to other sources in terms of installation and generation cost. The advantages of wind energy are combined with a few difficulties like high uncertainty, limited predictability and wind power energy is not entirely deliverable.

A measurement tower or met mast is a free standing [tower](https://en.wikipedia.org/wiki/Tower), which carries measuring instruments with [meteorological](https://en.wikipedia.org/wiki/Meteorological) instruments such as [thermometers](https://en.wikipedia.org/wiki/Thermometers) and [instruments to measure wind speed](https://en.wikipedia.org/wiki/Anemometer) such as anemometers, wind wane. Measurement tower is crucial in the development of [wind farms](https://en.wikipedia.org/wiki/Wind_farms). Before [developers construct a wind farm](https://en.wikipedia.org/wiki/Wind_farm#Location_planning), they first measure the wind resource on a prospective site by erecting measurement towers. Typically these mount [anemometers](https://en.wikipedia.org/wiki/Anemometer)  and other measuring instruments at a range of heights up to the hub height of the proposed wind turbines, and log the wind speed data at frequent intervals (every ten minutes) for at least one year and preferably two. The data allow the developer to determine if the site is economically viable for a wind farm, and to choose wind turbines optimized for the local wind speed distribution.

An Accurate and reliable wind speed prediction is vital for wind farm planning and operational planning for electrical networks. To improve the accuracy of wind speed prediction, many forecasting approaches have been proposed however these models typically do not account for the importance of data pre-processing and are limited by the use of individual models [2] [4] [5] [6] [7] [9] [10]. Our proposed work handles both data pre-processing and prediction using real-time data. Dataset contains aesthetics like Date/time, pressure, and temperature, wind speed at different heights of 50m, 80m, and 100 m and mean wind speed are considered. For data cleaning, interpolation method is used and for prediction, wind speed is labeled. In between the cleaning and prediction process the data storing and retrieving is done using PostgreSQL database.

# REQUIREMENTS

|  |  |  |
| --- | --- | --- |
| Items | Description | Remarks |
| Software | Python, PostgreSQL |  |
| Data | Renewable energy data (Wind data sources are getting identified) | 10 min of data for around 2 years is used. |

# DESCRIPTION OF DATA SET

The data set collected from measurement tower (as shown in figure 1) consists of the wind data at the interval of 10 mins interval and contains parameters such as date and time, mean and standard deviation of the wind speed at different heights, pressure and temperature for the period of 2 years from 2013 October to 2015 September (as shown in table 1).



**Figure 1 Photo of the Measurement Tower (Source: Wikipedia)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date/Time** | **WS\_50\_mean** | **WS\_50\_stdev** | **WS\_80\_mean** | **WS\_80\_stdev** | **WS\_100\_mean** | **WS\_100\_stdev** | **Ta\_10\_mean** | **Pbaro\_5\_mean** |
| 1/10/2013 10:00 | 5.62 | 0.43 | 5.71 | 0.43 | 5.61 | 0.43 | 27.81 | 1004.2 |
| 1/10/2013 10:10 | 5.72 | 0.48 | 5.85 | 0.5 | 5.85 | 0.43 | 27.82 | 1004.1 |
| 1/10/2013 10:20 | 6 | 0.36 | 6.05 | 0.46 | 6.05 | 0.37 | 27.76 | 1003.9 |
| 1/10/2013 10:30 | 6.49 | 0.41 | 6.58 | 0.53 | 6.46 | 0.43 | 27.77 | 1003.9 |
| 1/10/2013 10:40 | 6.74 | 0.45 | 6.82 | 0.56 | 6.86 | 0.39 | 27.8 | 1003.8 |
| 1/10/2013 10:50 | 7.15 | 0.38 | 6.45 | 0.66 | 7.17 | 0.38 | 27.87 | 1003.6 |
| 1/10/2013 11:00 | 7.13 | 0.44 | 6.23 | 0.59 | 7.25 | 0.41 | 27.9 | 1003.5 |
| 1/10/2013  11:10 | 7.16 | 0.58 | 6.31 | 0.56 | 7.28 | 0.49 | 27.95 | 1003.4 |

**Table 1 Sample Mast Data**

# LITERATURE REVIEW

In recent years, clean energies, such as wind power have been developed rapidly. Especially, wind power generation becomes a significant source of energy in some power grids. On the other hand, based on the uncertain and non-convex behavior of wind speed, wind power generation forecasting and scheduling may be very difficult. In this paper, to improve the accuracy of forecasting the short-term wind speed, deep learning time series prediction based on Long Short Term Memory neural networks (LSTM) [8]. The proposed wind speed forecasting strategy is applied to real-life data from Sotavento that is located in the south-west of Europe, in Galicia, Spain, and Kerman that is located in the Middle East, in the southeast of Iran. The presented numerical results demonstrate the efficiency of the proposed method, compared to some other existing wind speed forecasting methods.

Wind speed forecasting models and their application to wind farm operations are attaining remarkable attention in the literature because of its benefits as a clean energy source. In this paper, we suggested the time series machine learning approach called random forest regression for predicting wind speed variations. The computed values of mutual information and auto-correlation shows that wind speed values depend on the past data up to 12 hours [3]. The random forest model was trained using ensemble from two weeks data with previous 12 hours values as input for every value. The computed root mean square error shows that model trained with two weeks data can be employed to make reliable short-term predictions up to three years ahead.

One step ahead wind speed forecasting models were compared. A univariate model was developed using a linear autoregressive integrated moving average (ARIMA) [1]. This method’s performance is well studied for a large number of prediction problems. The main objective was to compare the impact of the various meteorological variables on the performance of the multivariate model (NARX) of wind speed prediction with respect to the high performance univariate linear model.

# PROPOSED WORK

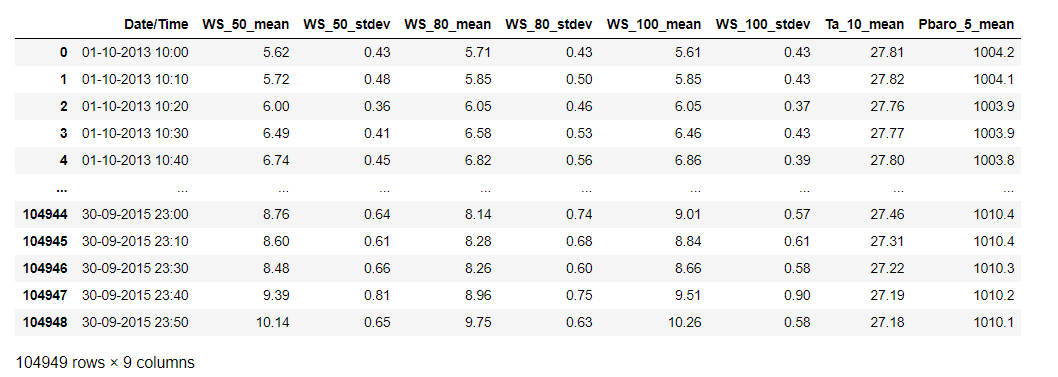
In the proposed work, we have three modules:

1. Wind Data Cleaning Module
2. Database Module
3. Prediction Module

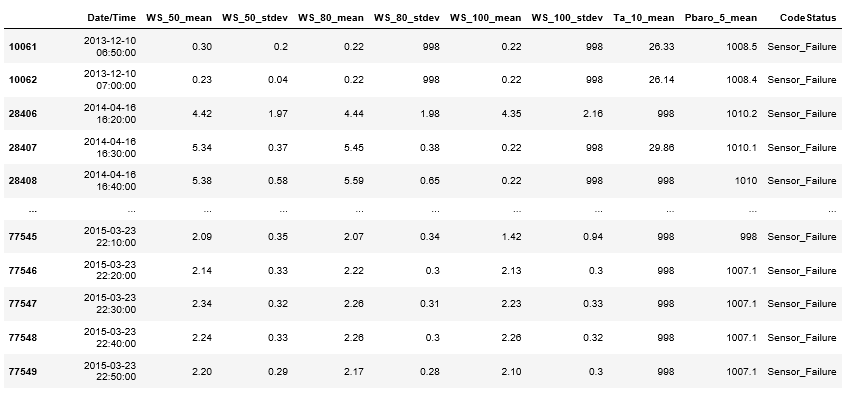
## Wind Data Cleaning Module

In this module, missing values are identified and cleaned using pandas library in python. Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data. Pandas has fast and efficient DataFrame object with default and customized indexing, Data alignment, integrated handling of missing data and time series functionalities. The missing values are handled using fillna(), interpolate() and imputer() methods. We are using interpolate() with padding method. Interpolation through padding means copying the value just before a missing/error entry and filling the missing/error entry in a series.

First step is to load the excel file as dataset (as shown in figure 2). Then identify the columns with missing/error values and replaces it with the error code based on the error value.Identified error rows with error status are stored as error log dataset (as shown in figure 3). Then interpolation is performed and the cleaned dataset is stored as cleaned dataset (as shown in figure 4).



**Figure 2 Uncleaned Dataset**



**Figure 3 Error Log Dataset**

# 

**Figure 4 Cleaned Dataset**

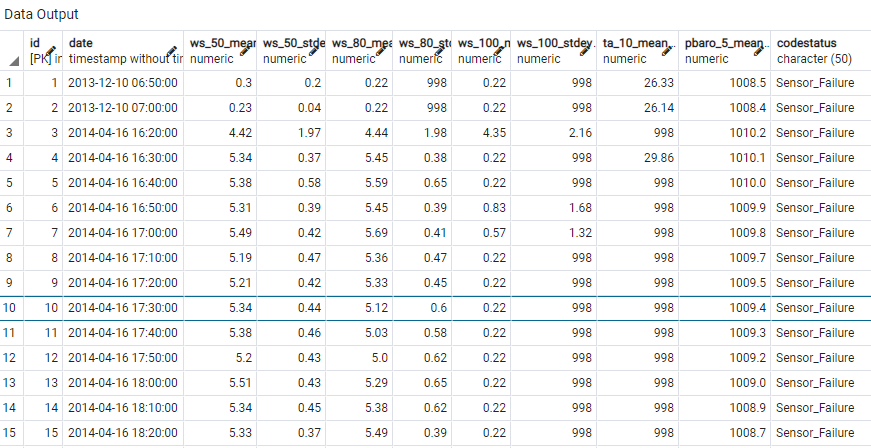
## Database module

PostgreSQL is an open source relational database management system (DBMS) developed by a worldwide team of volunteers. PostgreSQL is not controlled by any corporation or other private entity and the source code is available free of charge. Three tables are created and the values are inserted by connecting python and PostgreSQL. Python has various database drivers for PostgreSQL. Currently, the [psycopg](http://initd.org/psycopg/)2 is the most popular PostgreSQL database adapter for the Python language. The psycopg2 fully implements the Python DB-API 2.0 specification.

The uncleaned data table consists of original data dataset which contains missing or error values (as shown in figure 5). The error log table contains the error rows with error code status which can be used for future reference (as shown in figure 6). The cleaned data table consists of the data with interpolated values (as shown in figure 7).

## 

**Figure 5 Original Data Table**



**Figure 6 Error Log Data Table**

# 

**Figure 7 Cleaned Data Table**

## Prediction Module

Mainly focusing on Autoregressive Integrated Moving Average (ARIMA), fbprophet and Long-short Time Memory (LSTM), Random Forest Regressor (RFR) prediction models.

* Fbprophet is improvement to the ARMA/ARIMA model.
* It has inbuilt functions for seasonality and trends.
* By training the model with proper tuning, we predict the wind speed.
* When it comes to LSTM, it’s special kind of RNN which overcome the vanishing gradient problem.

### Long Short-Term Memory (LSTM)

The long short-term memory network (LSTM), which is a variant of RNN. An LSTM unit shown in Figure 1 is composed of a cell, an input gate, an output gate, and a forget gate. The unique structure of LSTM can effectively solve the problems of gradient disappearance and gradient explosion problems in the training process of RNN.

### Random Forest Regressor (RFR)

Random forest is a non-parametric ensemble based learning technique used for both classification and regression problem. It is an extended version of decision tree algorithm which works on a set of rules and the possible outcomes to form a tree-like structure. For an incorrect rule adds the impurity to the subsequent nodes, a high risk of error propagation is always associated with decision trees.

Random forest algorithm eliminates error diffusion property inherent in decision trees by constructing multiple decision trees. Random samples of given data set are generated and fed to several tree-based learners to form a random forest. Splitting condition for each node in a tree is based on only the randomly selected predictor attributes which lower the error rate by avoiding the correlation among the trees.

Random forest regression is a non-parametric regression technique in which the functional relation between dependent and independent variables are captured from the features of the data. In simple words random forest algorithm can be explained by a three-step procedure as follows:

1. From the given data select random samples with replacement

2. At each level, split node properly to get the best split until a maximum level of tree is obtained

3. Repeat the second step until a satisfied number of trees are generated.

### Fbprophet (FP)

To use Prophet for forecasting, first, a Prophet() object is defined and configured, then it is fit on the dataset by calling the fit() function and passing the data.

The fit() function takes a DataFrame of time series data. The DataFrame must have a specific format. The first column must have the name ‘ds‘ and contain the date-times. The second column must have the name ‘y‘and contain the observations.

|  |  |  |
| --- | --- | --- |
| **S.No** | **Parameter** | **Description** |
| 1 | yearly seasonality | Fit yearly seasonality |
| 2 | weekly seasonality | Fit weekly seasonality |
| 3 | daily seasonality | Fit daily seasonality |
| 4 | change points | List of dates at which to include potential changes. |

The Prophet package provides intuitive parameters which are easy to tune. Even someone who lacks expertise in forecasting models can use this to make meaningful predictions for a variety of problems. Using time as a regressor, Prophet is trying to fit several linear and non linear functions of time as components. Usually, some popular error terms such Root Mean Square Error (RMSE) is used during the modeling evaluation.

### Auto Regressive Integrated Moving Average (ARIMA)

An ARIMA model is a class of statistical models for analyzing and predicting time series data. It explicitly caters to a suite of standard structures in time series data, and as such provides a simple yet powerful method for making skillful time series forecasts. ARIMA is an acronym that stands for Auto Regressive Integrated Moving Average. It is a generalization of the simpler Auto Regressive Moving Average and adds the notion of integration. Each of these components are explicitly specified in the model as a parameter. A standard notation is used of ARIMA (p,d,q) where the parameters are substituted with integer values to quickly indicate the specific ARIMA model being used. The parameters of the ARIMA model are defined as follows:

**p**: The number of lag observations included in the model, also called the lag order.

**d**: The number of times that the raw observations are differenced, also called the degree of differencing.

**q**: The size of the moving average window, also called the order of moving average.

**Decompose:** The decomposition of time series is a statistical task that deconstructs a time series into several components, each representing one of the underlying categories of patterns. Statsmodels python library and seasonal\_decompose, autocorrelation\_plot methods are used.

# PERFORMANCE EVALUATION

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Based on a rule of thumb, it can be said that RMSE values between 0.2 and 0.5 shows that the model can relatively predict the data accurately.

---- (1)

Where represent an actual value represents the forecasted value

# ALGORITHMS

**7.1 LSTM Algorithm:**

**Step-1:** normalize the dataset

**Step-2:** split into train and test sets

**Step-3:** reshape into X=t and Y=t+1

**Step-4:** reshape input to be [samples, time steps, features]

**Step-5:** create and fit the LSTM network

**Step-6:** make predictions

**Step-7:** calculate root mean squared error

**7.2 RandomForestRegressor Algorithm:**

**Step-1:** input data for regressor.

**Step-2:** Create five columns (T1, T2, T3, T4, T5) and fill them with consecutive wind speeds.

**Step-3:** Split the train, test, and validation data.

**Step-4:** 100 estimators are divided, 50 parallel & fit for the RFR algorithm which classifies columns (T1, T2, T3, T4, T5)

**Step-5:** calculate root mean squared error

**7.3 fbprophet Algorithm:**

**Step-1:** splitting the data and renaming ‘df’ and ‘y’

**Step-2:** split into train and test sets

**Step-3:** fitting the data and defining seasonalities

**Step-4:** predicting the values of ‘y’ column

**Step-5:** calculate root mean square error

**Step-6:** changepoints of the wind speed

**7.4 Auto Regressive Integrated Moving Average Algorithm:**

**Step-1:** train the standardization.

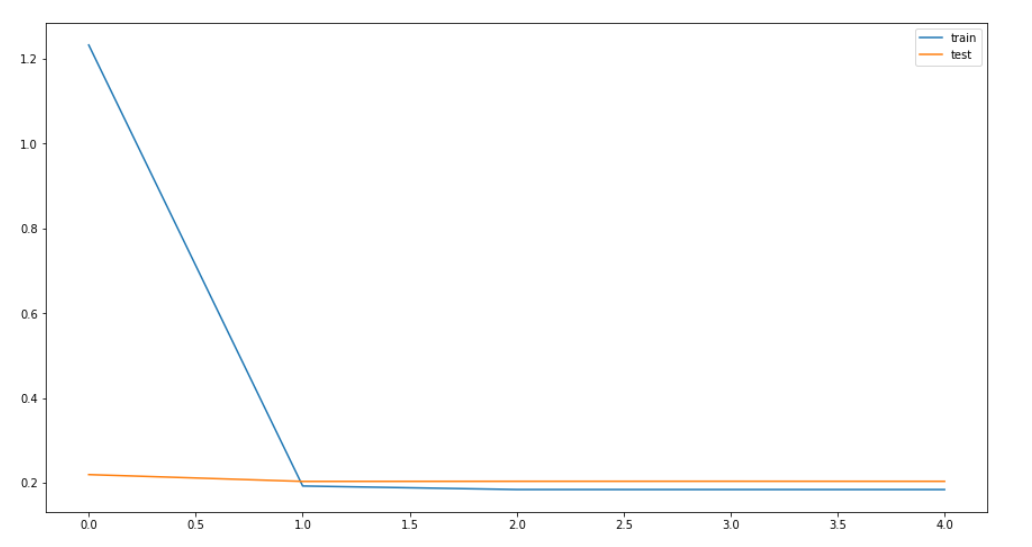
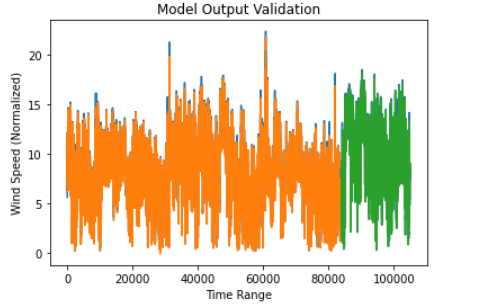
**Step-2:** standardization the dataset and print the first 5 rows.

**Step-3:** preprocessing data.

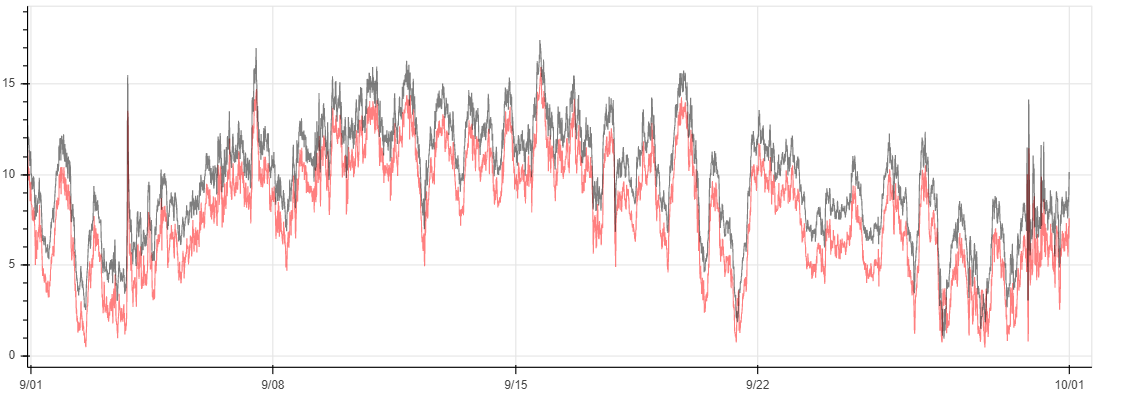
**Step-4:** fitting the data for ARIMA model.

**Step-5:** calculate the root mean square error.

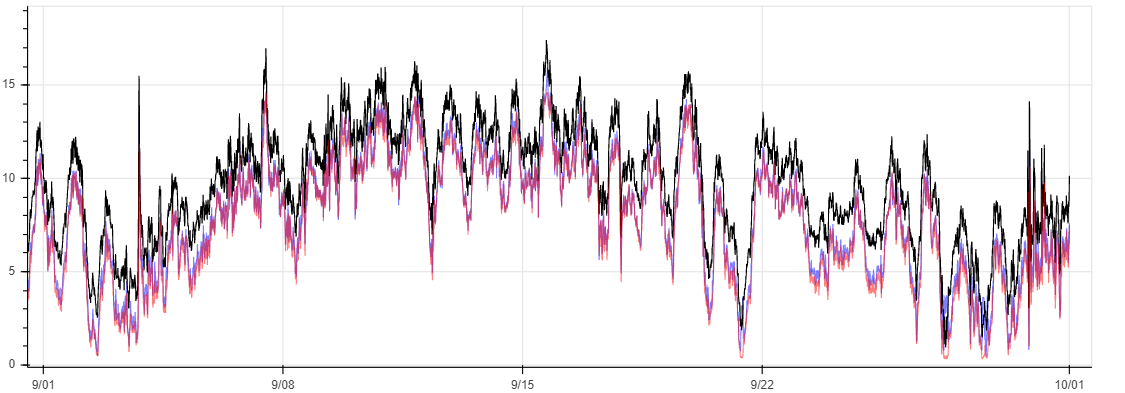
# SCREENSHOTS



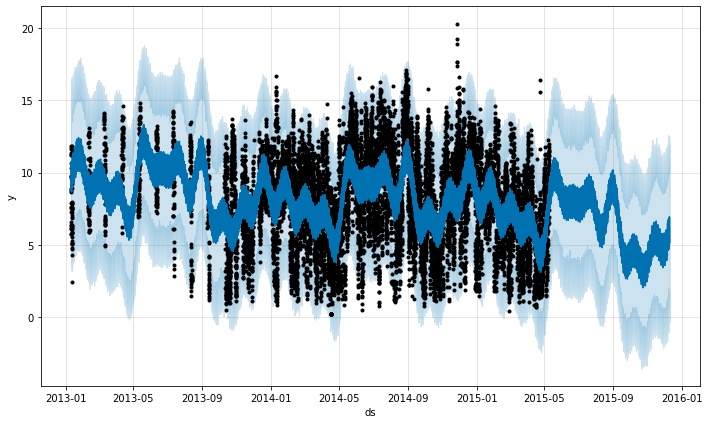
**Figure 8 LSTM Model**



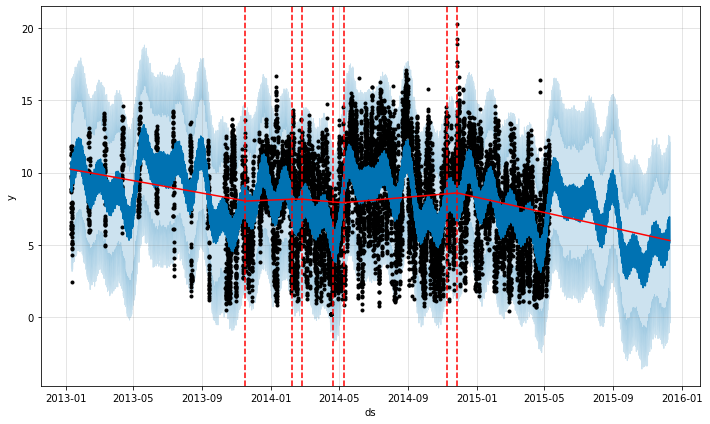
**Figure 9 Random Forest Regressor model**



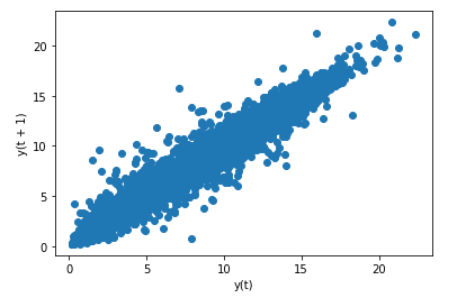
**Figure 10 RFR Model vs. LSTM Model**



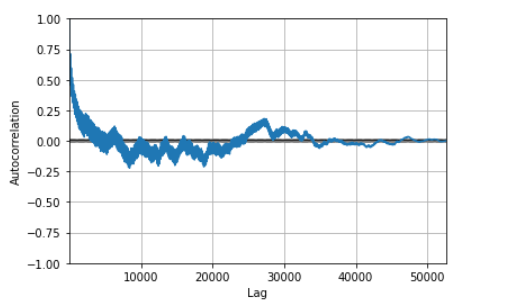
**Figure 11 Fbprophet model**



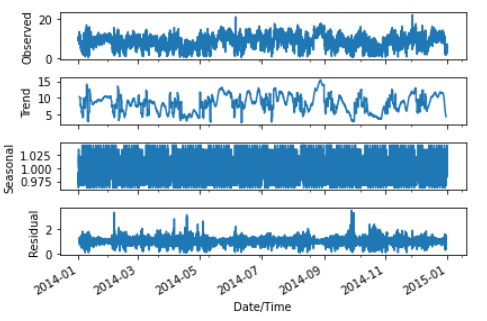
**Figure 12 Change points**



**Figure 13 a Decompose - correlation**



**Figure 13 b Decompose – correlation**



**Figure 14 ARIMA model**

# CONCLUSION

In this project, data cleaning is done using python and for storing PostgreSQL database is used. The stored data is used to perform wind speed prediction. The results of LSTM as shown in figure 8, RFR as shown in figure 9 and figure 10, ARIMA as shown in figure 14 and fbprophet models as shown in figure 11, figure 12, figure 13a and figure 13b are compared. The models are trained and tested with two years data. The root mean square error is calculated and their results are given below

1. LSTM with RMSE : 0.513
2. fbprophet with RMSE : 0.768
3. ARIMA with RMSE : 0.816
4. RFR with RMSE : 1.895

From the results presented here, it is clear that LSTM is a suitable prediction model for an effective wind speed prediction and thereby helping an efficient, cost effective wind energy management.

# REFERENCES

1. Cadenas, E., Rivera, W., Campos-Amezcua, R., & Heard, C. (2016). Wind speed prediction using a univariate ARIMA model and a multivariate NARX model. *Energies*, *9*(2), 1–15. https://doi.org/10.3390/en9020109
2. De Alencar, D. B., De Mattos Affonso, C., De Oliveira, R. C. L., Rodríguez, J. L. M., Leite, J. C., & Filho, J. C. R. (2017). Different Models for Forecasting Wind Power Generation: Case Study. *Energies*, *10*(12). https://doi.org/10.3390/en10121976
3. Drisya, G. V, Valsaraj, P., Asokan, K., & Kumar, K. S. (2017). *Wind speed forecast using random forest learning method*. *9*(06), 362–367.
4. Haddad, M., Nicod, J., Mainassara, Y. B., Masry, Z. Al, Péra, M., Haddad, M., Nicod, J., Mainassara, Y. B., Rabehasaina, L., &Masry, Z. Al. (2020). *HAL Id : hal-02867736 Wind and Solar Forecasting for Renewable Energy System using SARIMA-based Model*.
5. Kishore, S. R. N., & Vanitha, V. (2013). Wind speed forecasting for grid code compliance. *Journal of Renewable and Sustainable Energy*, *5*(6), 1–10. https://doi.org/10.1063/1.4850256
6. Liu, Y., Guan, L., Hou, C., Han, H., Liu, Z., Sun, Y., & Zheng, M. (2019). Wind power short-term prediction based on LSTM and discrete wavelet transform. *Applied Sciences (Switzerland)*, *9*(6). https://doi.org/10.3390/app9061108
7. Liu, Y., Guan, L., Hou, C., Han, H., Liu, Z., Sun, Y., & Zheng, M. (2019). Wind power short-term prediction based on LSTM and discrete wavelet transform. *Applied Sciences (Switzerland)*, *9*(6). https://doi.org/10.3390/app9061108
8. Mandal, N., &Sarode, T. (2020). Prediction of Wind Speed using Machine Learning. *International Journal of Computer Applications*, *176*(32), 34–37. https://doi.org/10.5120/ijca2020920370
9. Taylor, S. J., &Letham, B. (2017). Business Time Series Forecasting at Scale. *PeerJ Preprints 5:E3190v2*, *35*(8), 48–90. https://peerj.com/preprints/3190/%0Ahttp://ezproxy.bangor.ac.uk/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=c8h&AN=108935824&site=ehost-live%0Ahttps://peerj.com/preprints/3190/%0Ahttps://peerj.com/preprints/3190.pdf
10. Senthil Kumar, P. (2019). Improved prediction of wind speed using machine learning. *EAI Endorsed Transactions on Energy Web*, *19*(23), 1–7. https://doi.org/10.4108/eai.13-7-2018.157033