Wind Turbine Power output forecasting based on weather conditions

Team members : Gokul Soni

FINAL PROJECT REPORT

1. INTRODUCTION

1.1 OVERVIEW

Wind energy forecasting plays an important role in wind energy utilization,especially wind speed forecasting, which is a vital component of wind energy management.Over the last decade there has been rapid growth in wind generation of electricity, with the installed wind power capacity worldwide has increased almost fourfold from circa 24.3 GW (Giga Watts) to an expected 203.5 GW(Giga Watts) .In power systems, balance is maintained by continuously adjusting generation capacity and by controlling demand.

1.2 PURPOSE

As wind is inherently variable, wind power is a fluctuating source of electrical energy. Short-term forecasts (ranging from 1 h up to 72 h) are useful in power system planning for unit commitment and dispatch, and for electricity trading in certain electricity markets where wind power and storage can be traded or hedged. Hence , the main purpose behind building this product is to forecast wind turbine energy output , in a short term (ranging from 1 hour up to 72 hour).

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM:

There exist a number of technological, environmental and political challenges linked to supplementing existing electricity generation capacities with wind energy. Here, mathematicians and statisticians could make a substantial contribution at the interface of meteorology and decision-making, in connection with the generation of forecasts tailored to the various operational decision problems involved. Indeed, while wind energy may be seen as an environmentally friendly source of energy, full benefits from its usage can only be obtained if one is able to accommodate its variability and limited predictability. Based on a short presentation of its physical basics, the importance of considering wind power generation as a stochastic process is motivated. The conventional moving-average statistical models were proven to be less efficient in forecasting the wind energy, as the wind speed is inherently variable quantity.

2.2 PROPOSED SOLUTION:

To overcome the disadvantages of conventional models, advanced deep learning models such as LSTM (Long Short Term Memory) , can be used to map the inherently variable attribute to a complex function. LSTM is well-suited to predict time series given time lags of unknown duration. Relative insensitivity to gap length gives an advantage to LSTM over alternative RNNs, hidden Markov models and other sequence learning methods. Since the wind speed and wind degree vary from place to place ,it is not possible to develop a generalized model which can predict the energy output of the wind farms located all around the world. As a part of our problem statement, Muppandal wind farm situated in TamilNadu at coordinates 8.2600 N 77.5475 E, is taken as a use case to train the LSTM model.The proposed LSTM model consists of four layers and computationally less expensive than other deep learning LSTM architectures.

The architecture of deployed LSTM model :

```
Model: "sequential"
_________________________________________________________________
Layer (type)            Output Shape         Param #
=================================================================
lstm (LSTM)            (None, 4)            96
_________________________________________________________________
dense (Dense)           (None, 1)            5
=================================================================
Total params: 101
Trainable params: 101
Non-trainable params: 0
_________________________________________________________________
```

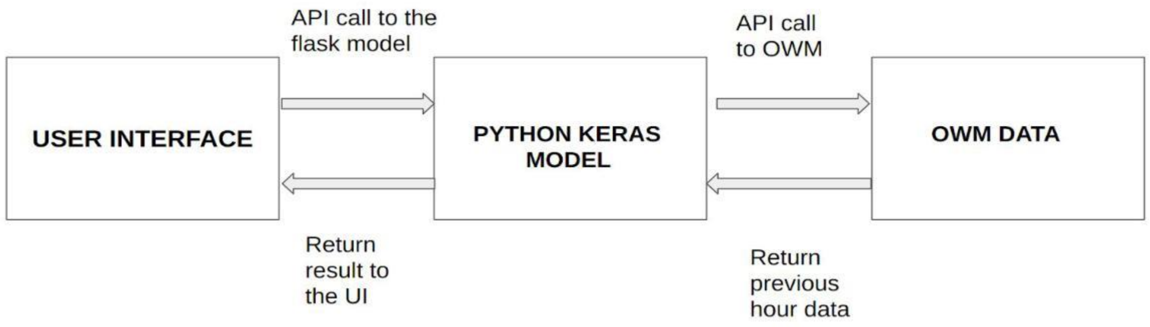
3. THEORETICAL ANALYSIS

3.1 BLOCK DIAGRAM:

The three main components of our project are :

1. The User Interface - A flutter mobile application
2. The REST API - Deep Learning model
3. OpenWeatherMap API - To get the previous hour's weather data.

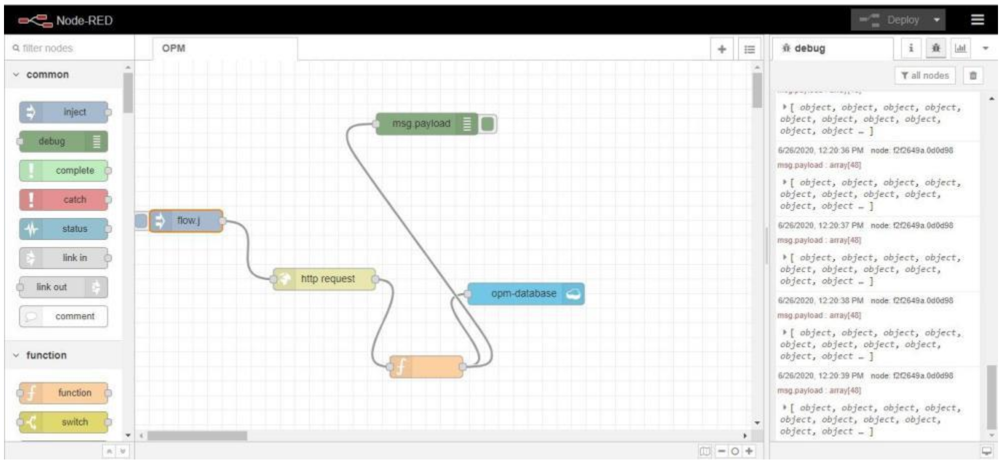
The block diagram below explains the working of the components.



3.2 HARDWARE/SOFTWARE DESIGNING:

The product designing involves training a model and deploying it using flask .The flask app is then deployed in PaaS infrastructure by Cloud Service providers. The flask app is designed as a RESTful API ,where the User Interface curls the result from server and displays it to the user. User Interface is built using flutter . The model is keras LSTM model with backend as Tensorflow. Instead of converting the model to tflite and directly deploy it in mobile ,we deployed a REST api using flask and made the UI to curl the required information. By this way the end UI application becomes light weight and uses less computational time and space.

Node RED Template to collect dataset from pen weather map API :



```
▸[ object, object, object, object,
object, object, object, object,
object, object … ]
6/26/2020, 12:20:36 PM   node: f2f2649a.0d0d98
msg.payload : array[48]
▸[ object, object, object, object,
object, object, object, object,
object, object … ]
6/26/2020, 12:20:37 PM   node: f2f2649a.0d0d98
msg.payload : array[48]
▸[ object, object, object, object,
object, object, object, object,
object, object … ]
6/26/2020, 12:20:38 PM   node: f2f2649a.0d0d98
msg.payload : array[48]
▸[ object, object, object, object,
object, object, object, object,
object, object … ]
6/26/2020, 12:20:39 PM   node: f2f2649a.0d0d98
msg.payload : array[48]
▸[ object, object, object, object,
object, object, object, object,
object, object … ]
```

4. EXPERIMENTAL INVESTIGATIONS

1. The following observations were madewhile testing the API call to the OWM

A. The API call is made as :
https://api.openweathermap.org/data/2.5/weather?APPID=91ffd96ea9b199fc8b27691304cbc549&lat=11.5450&lon=79.5212

B. Data format was :

{"coord":{"lon":79.52,"lat":11.55},"weather":[{"id":804,"main":"Clouds","description":"overcast clouds","icon":"04d"}],"base":"stations","main":{"temp":307.96,"feels_like":309.56,"temp_min":307.96,"temp_max":307.96,"pressure":1004,"humidity":44,"sea_level":1004,"grnd_level":1001},"wind":{"speed":3.5,"deg":246},"clouds":{"all":99},"dt":1594797171,"sys":{"country":"IN","sunrise":1594772734,"sunset":1594818588},"timezone":19800,"id":1265504,"name":"Kurinjippādi","cod":200}

2. The Deep Learning model was wrapped into a REST API using python flask.

The following observations were made:

A. The API call through command line :

```
curl -X POST http://127.0.0.1:9000/classifier/run -F hours=2
```

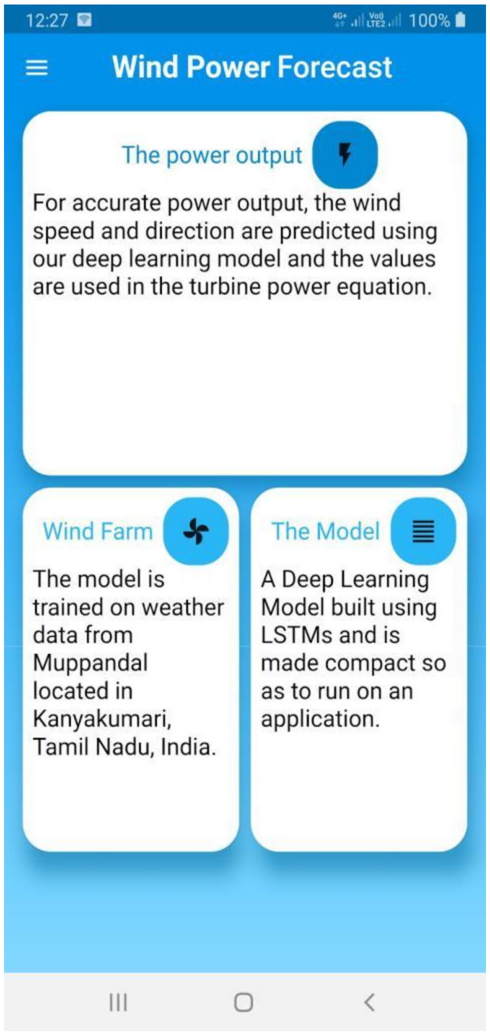
B. The output was a JSON file containing 48 point predictions for the 2nd hour from the current time. The values are in "Watts".

{"0": 768.4502600042123, "1": 814.0369279865768, "2": 832.3292903087915, "3": 944.5268658793321, "4": 1039.0213742995713, "5": 1058.224997083237, "6": 1091.4272547338721, "7": 1091.7142301871086, "8": 1095.3782208277971, "9": 1030.3317998409636, "10": 810.3161207405408, "11": 904.5119927805358, "12": 846.532385601517, "13": 802.8210420295138, "14": 756.8727587807564, "15": 661.045999996351, "16": 661.045999996351, "17": 587.7967236092414, "18": 545.617009705704, "19": 521.4073792682636, "20": 523.5256246325578, "21": 526.6416485536314, "22": 594.7362959862292, "23": 703.5707449777448, "24": 688.9395977818278, "25": 692.5602284043447, "26": 825.0787992579982, "27": 768.4502600042123, "28": 934.6029109254856, "29": 937.1340668495069, "30": 1068.8131276422555, "31": 1089.6279626547278, "32": 1060.398416867273, "33": 977.1844294438039, "34": 718.546177563003, "35": 536.0834077233976, "36": 519.4632591657241, "37": 512.7894153446409, "38": 512.3798130002434, "39": 511.98343461023427, "40": 507.01895849493957, "41": 512.3798130002434, "42": 509.5562122981227, "43": 525.8299686856212, "44": 527.4758411067927, "45": 535.0229336329543, "46": 565.6652169527182, "47": 629.4395511038049}

3. The following are the screenshots of the working flutter application :

Icon for the application

This is the home screen of the application.



12:27 100%

Wind Power Forecast

The power output

For accurate power output, the wind speed and direction are predicted using our deep learning model and the values are used in the turbine power equation.

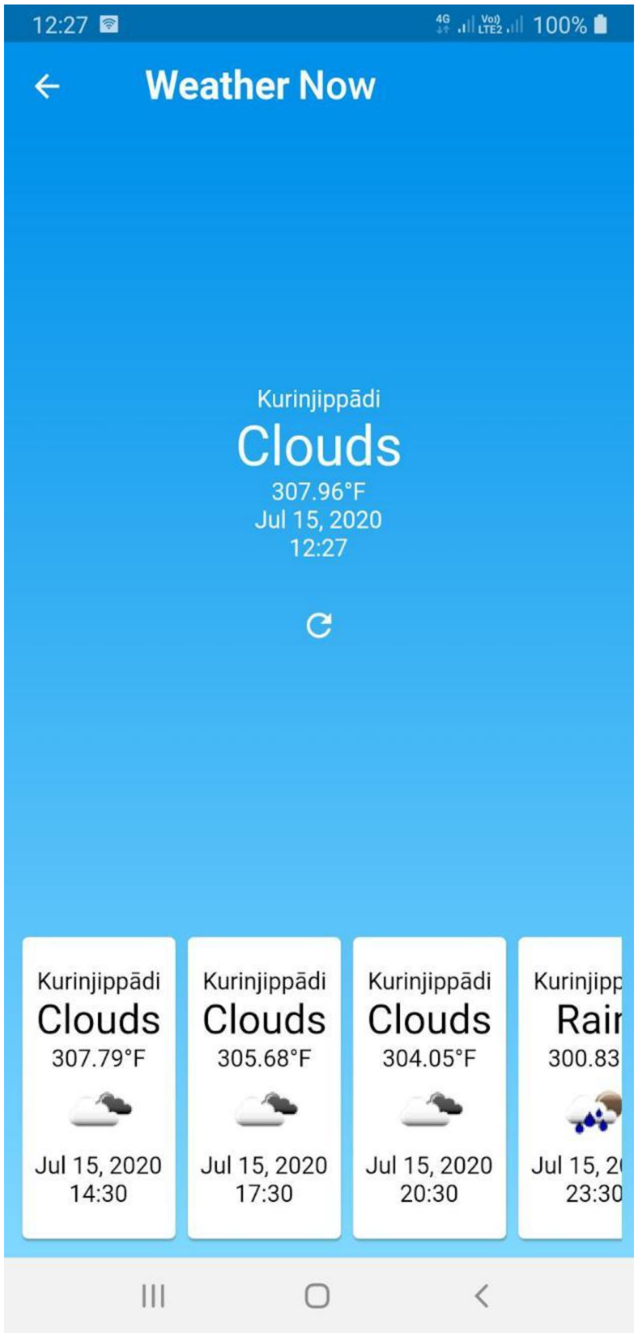
Wind Farm

The model is trained on weather data from Muppandal located in Kanyakumari, Tamil Nadu, India.

The Model

A Deep Learning Model built using LSTMs and is made compact so as to run on an application.

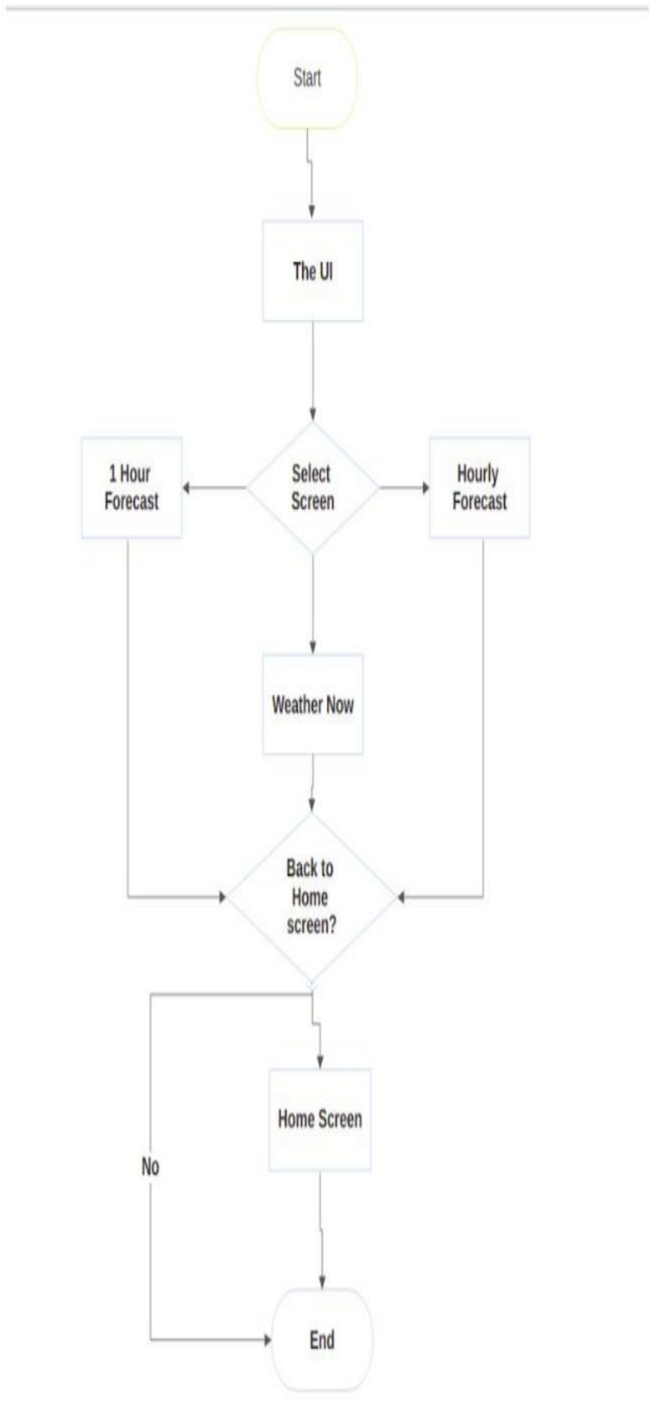
This is the Weather Now tab, for displaying the current weather at a particular location.



Kurinjippādi
Clouds
307.96°F
Jul 15, 2020
12:27

Kurinjippādi	Kurinjippādi	Kurinjippādi	Kurinjipp
Clouds	Clouds	Clouds	Rai
307.79°F	305.68°F	304.05°F	300.83
Jul 15, 2020	Jul 15, 2020	Jul 15, 2020	Jul 15, 2
14:30	17:30	20:30	23:30

5. FLOW CHART


Start
The UI
1 Hour Forecast
Select Screen
Hourly Forecast
Weather Now
Back to Home screen?
Home Screen
No
End

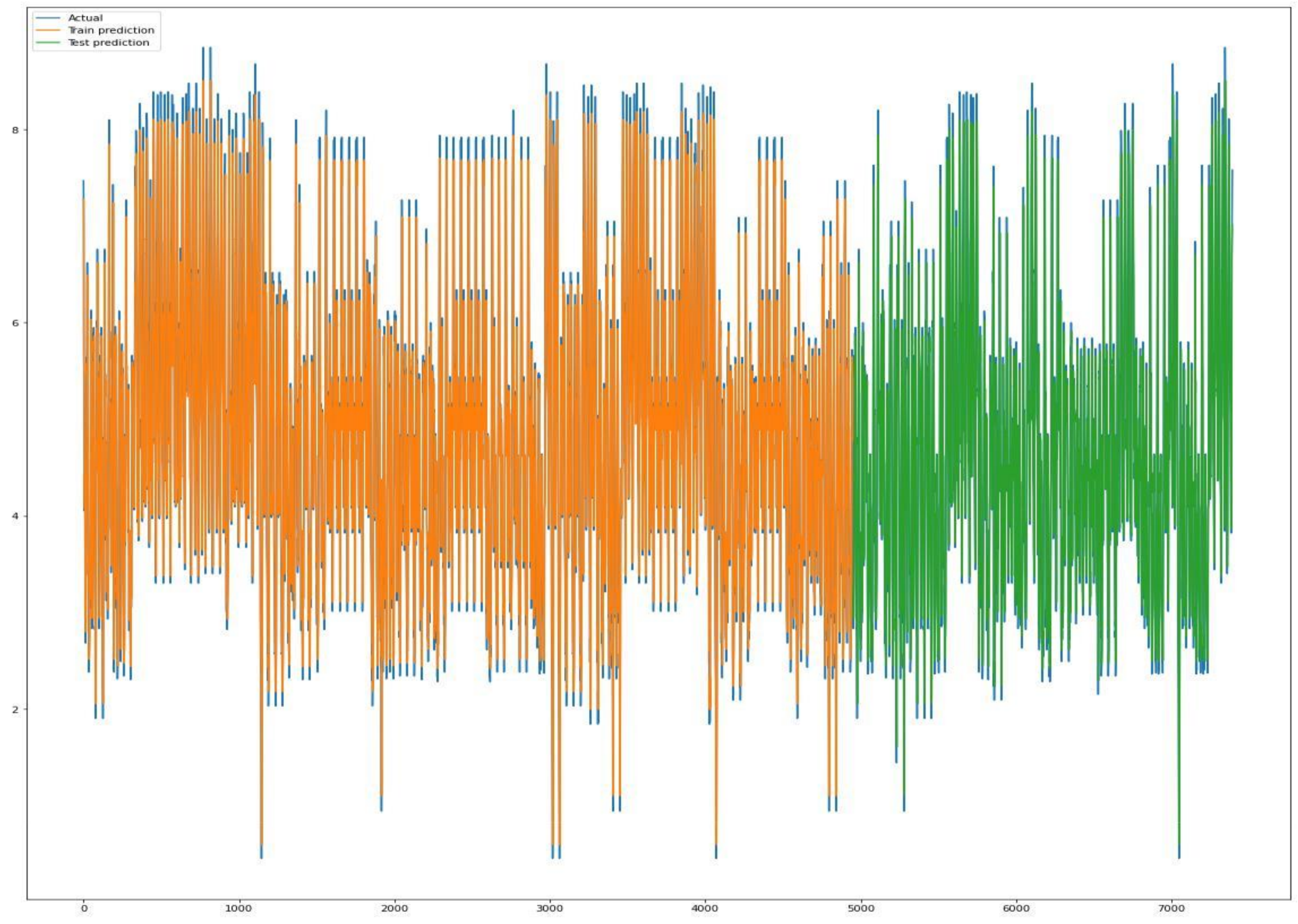
6. <u>RESULT</u>

The input data set is extracted from open weather map API using API calls . For every call , hourly forecast of weather is retrieved .The data received contains 48 entries ,i.e, for every 1.25 min the weather conditions are recorded and retrieved . Total of 150 hours of data is collected from 26-06-2020 to 03-07-2020. The model is trained and validated using this dataset.

Train Score: 0.57 RMSE

Test Score: 0.56 RMSE

The resultant plot of train and test:



The weights of the trained model was saved in '.h5' files ,which is then loaded into the model architecture . Flask app is created and hosted in cloud service .The important function of flask app is to call Open weather Map(OWM) API and collect the hourly weather data.The OWM API returns the current hour weather forecast in json format.

The flask app preprocess the data and predicts the nth hour wind speed values .
The power is calculated using the given formula:

$$\text{Power} = k\, C_p\, 1/2\, \rho\, A V^3$$

Where:

P = Power output, kilowatts

Cp = Maximum power coefficient, ranging from 0.25 to 0.45, dimension less (theoretical maximum = 0.59)

ρ = Air density, lb/ft3

A = Rotor swept area, ft2 or π D2/4 (D is the rotor diameter in ft, π = 3.1416)

V = Wind speed, mph

k = 0.000133 A constant to yield power in kilowatts

air density is taken as 1.23 kg/m^3 , radius of the rotor blades is taken as 45 m as standard size varies from 24 -45m .The User can retrieve the predicted power for nth hour from current time. Since the python-flask app is a RESTful api ,the predicted power can be retrieved using a curl command. The UI makes an api call to python-flask app to retrieve predicted output.Since Open Weather Map (OWM) returns 48 data points for an hour , the predicted power output contains 48 datapoints ,i.e. every 1.25 min the app makes a power prediction.The final power prediction is displayed in the UI. Flutter app serves as a user interface(UI) .The resultant power is in kW .

7. ADVANTAGES AND DISADVANTAGES

ADVANTAGES :

1. Intuitive User Interface

2. Faster Predictions on just a click of a button

3. Accurate predictions upto 1 hour to 72 hours

4. 48 predictions per hour for more accurate Power Output visualization

5. The Power grids can make use of this app to efficiently monitor the wind turbine power output.

6. An additional feature : Weather Now, to display the current weather at the location is added to the application.

7. The model is also trained to predict direction of the wind and can be used for the alignment of the turbines.

DISADVANTAGES:

1. On device Deep Learning model cannot be used. The predictions must be made through

2. The time taken for predictions is thus comparatively little less.

3. The model is trained only on Muppandal, Kanyakumari, Tamil Nadu,India weather data i.e.Region specific. There is scope for generalizing the model to suit for various other regions.

8. APPLICATIONS

1. The same model mentioned above can also be applied to other data such as Solar energy, tidal energy etc.
2. The model can be transfer-learned for making predictions for other locations, instead of training from scratch. This reduces the time taken for training.
3. The application can also be used as a standalone REST API calls

9. CONCLUSION

A Deep Learning model, based on Long Short Term Memory(LSTMs) has been developed to forecast the Power output of the wind turbine. A simple, efficient and a versatile model is built keeping in mind the diversity of data, computational complexity and overhead involved in making API calls to the model for prediction.
The product can increase the accuracy of the forecasting the output from the wind turbine .Overall accurate wind power prediction reduces the financial and technical risk of uncertainty of wind power production for all electricity market participants

10. FUTURE SCOPE

1. A more generalized model can be developed to suit forecasting for different locations.
2. The model can be developed to make predictions on other sources of data such as solar power, tidal power etc.
3. On-device model can be developed to make much more faster predictions.

11. BIBILIOGRAPHY

1. LSTMs https://keras.io/api/layers/recurrent_layers/lstm/
2. Flask https://pypi.org/project/Flask/

3. Deploying flask app to IBM cloud

4. Deploying flask app to Heroku
https://towardsdatascience.com/deploying-a-deep-learning-model-on-heroku-using-flask-and-python-769431335f66

5. Virtual environments in python Pipenv & Virtual Environments

6.HTTP client and servers for Dart https://dart.dev/tutorials/server/httpserver

7. Deploying ML models as REST API
https://medium.com/analytics-vidhya/deploy-ml-models-using-flask-as-rest-api-and-access-via-flutter-app-7ce63d5c1f3b

8. Flutter UI
https://medium.com/@lets4r/flutorial-create-a-staggered-gridview-9c881a9b0b98

9. Time Series Forecasting
https://courses.analyticsvidhya.com/courses/creating-time-series-forecast-using-python

https://pub.dev/packages/tflite#-installing-tab-

https://github.com/danielegrattarola/spektral

https://arxiv.org/abs/1703.07015