

# IBM Recommendation Model To Utilize Energy From Wind Farm

-BY HOROSCOPER"S

## SCOPE:-

- Our **Ultimate Aim** Is To **Competitive Real Time** , **Reduce Energy Wasteage** And **Losses In Power Grid**.
- By Using Our Model,we Can Predict The **Valuable Time** Of Power Production.
- **Rest Of Plant** Can Be Provided **Frequently** By **Predicting** The **Unworthy Time** By **Forecatsing** Energy,so We Can **Save Machine** Break Down's.
- We Can Solve The **Future Energy Needs**.
- As Well **Availbiality Of Energy** Can Be Determined.

## LITERATURE REVIEW:-

➡️ We get into this great paper to get prerequisite in this domain.

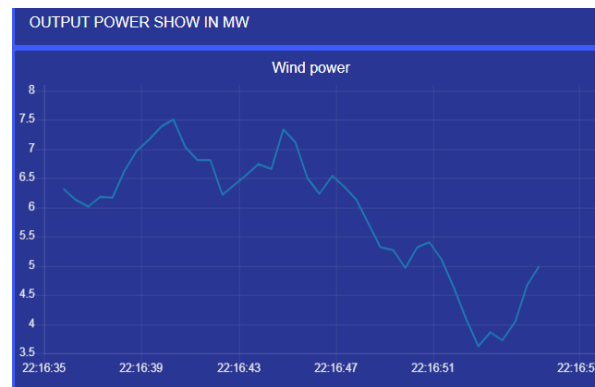
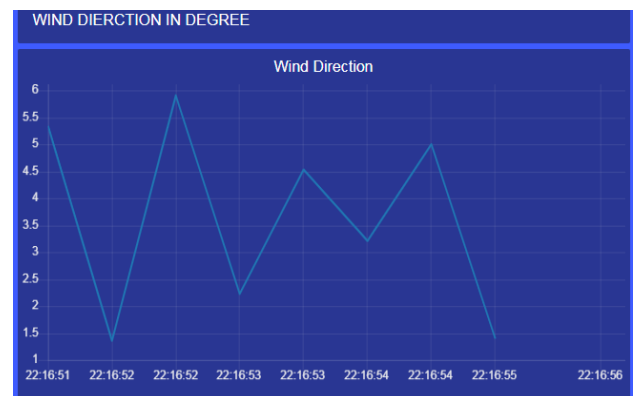
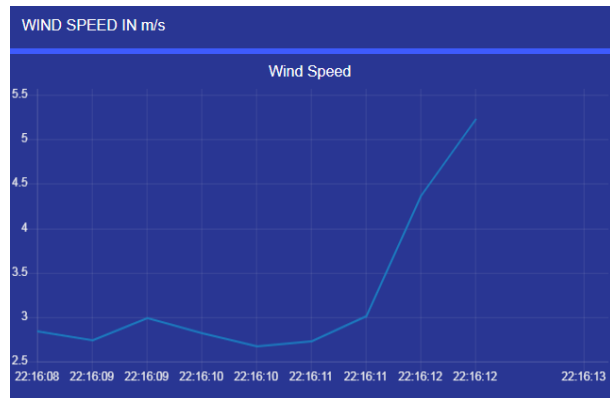
➡️ below mention's are our aspects from this paper

### **Article A New Hybrid Approach to Forecast Wind Power for Large Scale Wind Turbine Data Using Deep Learning with TensorFlow Framework and Principal Component Analysis**

-Mansoor Khan , Tianqi Liu ,\* and FarhanUllah

- The principal components are then used as input data to **deep learning** based on the TensorFlow framework. **PCA provides feature extraction and selection.**
- Here, **acc, val\_loss, val\_acc** represents accuracy, validation loss and validation accuracy, respectively. The loss, accuracy and loss, and **validation loss** are **calculated for wind power forecasting.**
- The proposed deep learning algorithm is applied to PCs **to forecast wind power.** The Keras API is used with **TensorFlow to configure** a more reliable **neural network.**

## Forecasted Output Of Our Model:-



The Above Forecasting Done For 24 Hrs On Date 28-10-2021

## Work Process:-

- IBM Account Creation.
  - Smart Internz assigning template.
  - Assigning team.
- Fetching Input.
- Selection of algorithm.
- Data preparation.
- Deploying Model.
- GUI creation.

## Fetching Input:-

➔ Collecting of data from various website.

•Kaggle-

<https://www.kaggle.com/berkerisen/wind-turbine-scada-dataset>

•Github-[https://github.com/ShashwatArghode/Wind-Energy-Prediction-using-LSTM/blob/master/AL\\_WIND\\_07\\_12.xlsx](https://github.com/ShashwatArghode/Wind-Energy-Prediction-using-LSTM/blob/master/AL_WIND_07_12.xlsx)

➔ Importing data in **IBM watson studio** in a click.

```

In [40]: import sys
import pandas as pd
from botocore.client import Config
import boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
client_64c8547f35d43b5b41b68c871da81d6 = boto3.client(service_name='s3',
    aws_access_key_id='HcWLN0vsl13y1209RQ1H4v35TyJy80d0cua4f1',
    aws_secret_access_key='ibm_auth_endpoint=https://iam.cloud.ibm.com/oidc/token',
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.eu-gov-objects.storage.service.networklayer.com')

body = client_64c8547f35d43b5b41b68c871da81d6.get_object(Bucket='hanif-donotdelete-pr-nxileyclobvpg',Key='winddirtime.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, '__iter__'): body.__iter__ = types.MethodType(__iter__, body)

series = pd.read_csv(body,index_col='Date/Time')
series.head()

```

Out[10]:

Date/Time	Wind Direction (°)
01 01 2018 00:00	259.994904
01 01 2018 00:10	266.641113
01 01 2018 00:20	272.564789
01 01 2018 00:30	271.258087
01 01 2018 00:40	265.674286

## Selection Of Algorithm:-

- Choosing **Best algo** . is the **main** process ever in **forecasting**.
- Normally algo. Like **SVM, Arima, Autoarima** are **used** for **time series** prediction .
- But in this recommendation model **we tried LSTM** algo ,which is one of library of **keras**.
- We have done this **forecasting** in **three segment**,  
Wind Speed , Direction, Power.
- **Wind Speed** and **Wind Direction** using **Istm** model.
- **Power Forescating** using **AutoAi**,The facility provied by our **IBM Watson studio**.

## Future Engineering:-

### ▶ FOR WIND SPEED AND DIRECTION:-

- To fit our model we done some **preprocessing** ,
- Split up data into **train** and **test** .
- Converting the data into **scalar form** , then take **difference of past data**.
- Setting the **epoch** ,finding **study rate**, fix the **batch size**, selection of **neurons** and **validate**.

### ▶ FOR POWER:-

- We done it in **auto AI**.
- For **power** prediction we give the **input** from the **ouput** of **former prediction**.

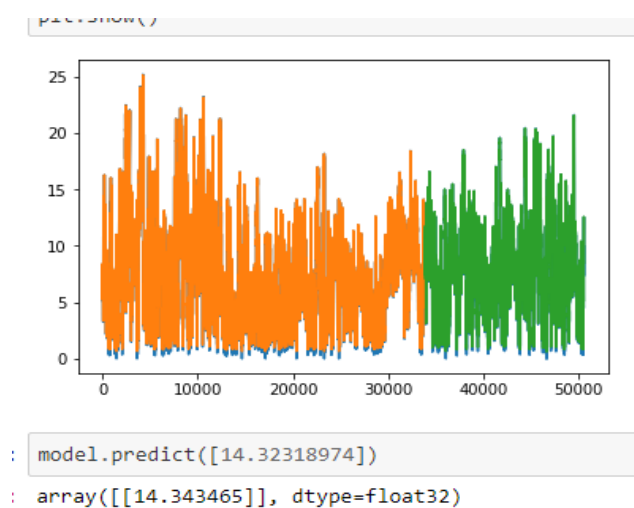
#### SPLITTING OF DATA

```
[8]: X = df[['WindSpeed', 'WindDirection']].values
      print(X)
      y = df['ActivePower'].values
      print(y)
      from sklearn.model_selection import train_test_split
      x_train,x_test,y_train,y_test = train_test_split(X,y, test_size = 0.2)

      [[ 5.31133604 259.99490356]
       [ 5.67216682 268.64111328]
       [ 5.2160368  272.56478882]
       ...
       [ 8.43535805  84.74250031]
       [ 9.42136574  84.2979126 ]
       [ 9.97933197  82.27462006]]
      [ 380.04779053 453.76919556 306.37658691 ... 2201.10693359 2515.6940918
       2820.46606445]
```

## Validation of our Prediction:-

- ➡ The plot shows the **test and train data** fitted with our model.
- ➡ The RMSE the value showing below are **model fitted by power** data which is especially done by **Auto Ai Service** By Our **IBM Watson Studio**.

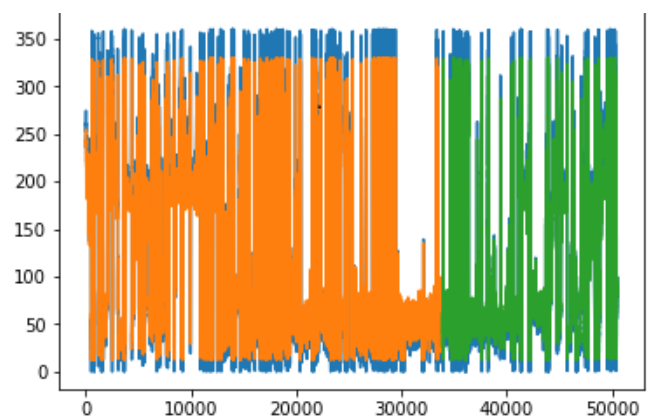
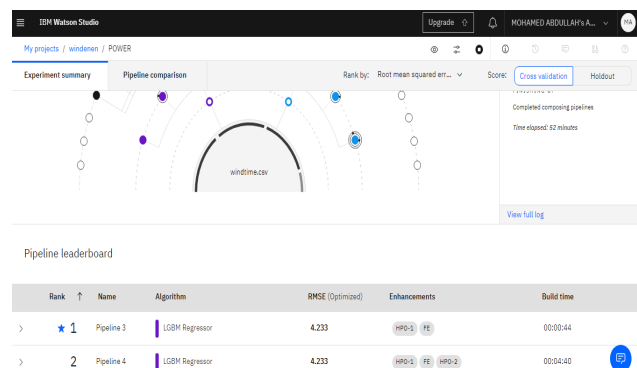


```

: import math
  # Estimate model performance
  trainScore = model.evaluate(trainX, trainY, verbose=0)
  print('Train Score: %.2f MSE (%.2f RMSE)' % (trainScore, math.sqrt(trainScore)))
  testScore = model.evaluate(testX, testY, verbose=0)
  print('Test Score: %.2f MSE (%.2f RMSE)' % (testScore, math.sqrt(testScore)))

```

Train Score: 0.58 MSE (0.76 RMSE)  
Test Score: 0.54 MSE (0.74 RMSE)





## Deployment:-

- ➔ **IBM Watson Studio** made **Deployment** easy in few steps.
- ➔ For deployment purpose ,we created a service credentials in **Watson mechaine learning service**.
- ➔ After creating credentials **load the model in service provider**,We **get the guid**
- ➔ that's all its done

```

from watson_machine_learning_client import WatsonMachineLearningAPIClient

uml_credentials = {
    "apikey": "wetr0xhgm2d4o82ou7_7uag1009#amdj55cde-vstg",
    "iam_apikey_description": "Auto-generated for key 22942c3d-89db-432c-aaab-378896575988",
    "iam_apikey_name": "Service credentials-2",
    "iam_role_crn": "crn:vi:bluemix:public:iam::::serviceRole:Writer",
    "iam_serviceid_crn": "crn:vi:bluemix:public:iam-identity::a/1af47091ccdc47489f9955bf348a1583::serviceid:ServiceId-ata649d-7e6c-4f1b-9e25-799208ee5c1a",
    "instance_id": "fdcbcfed-e3a5-4918-af08-d1453d727e98",
    "url": "https://eu-gb.ml.cloud.ibm.com"
}

client = WatsonMachineLearningAPIClient(uml_credentials)

metadata = {
    client.repository.ModelMetaNames.AUTHOR_NAME: 'Mohamed Abdullah',
    client.repository.ModelMetaNames.AUTHOR_EMAIL: 'abdullahmohamed2540@gmail.com',
    client.repository.ModelMetaNames.NAME: 'Prediction of Wind Direction',
    client.repository.ModelMetaNames.FRAMEWORK_NAME: 'tensorflow',
    client.repository.ModelMetaNames.FRAMEWORK_VERSION: '1.15',
    client.repository.ModelMetaNames.FRAMEWORK_LIBRARIES: [{'name': 'keras', 'version': '2.2.4'}]
}

model_result_path = 'direction_prediction.h5'
lstm_model.save(model_result_path)

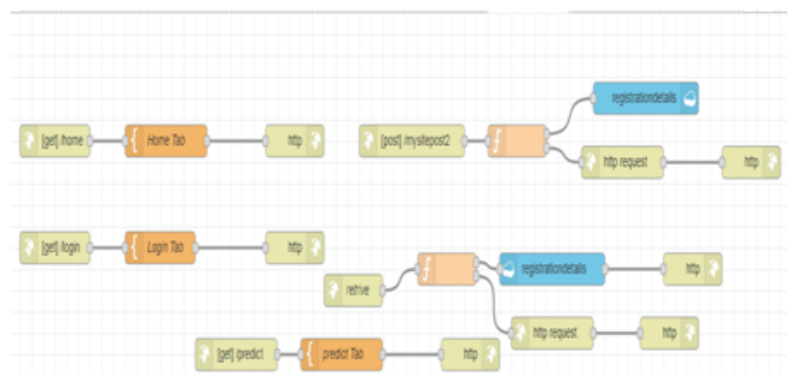
!tar -czvf direction_prediction.tgz direction_prediction.h5
direction_prediction.h5

stored_data = client.repository.store_model(model='direction_prediction.tgz', meta_props=metadata, training_data=-X, training_target=y)

```

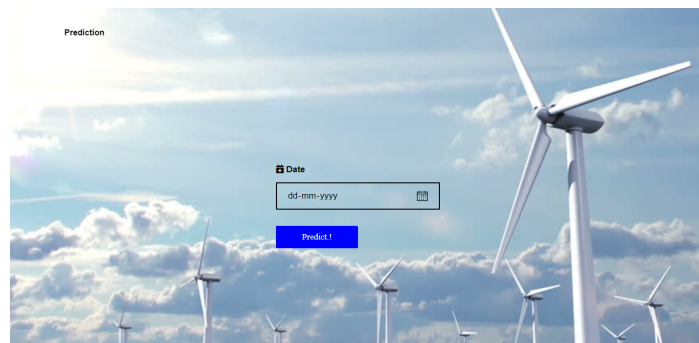
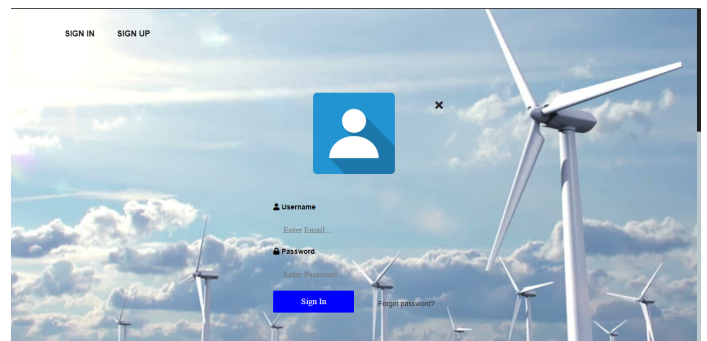
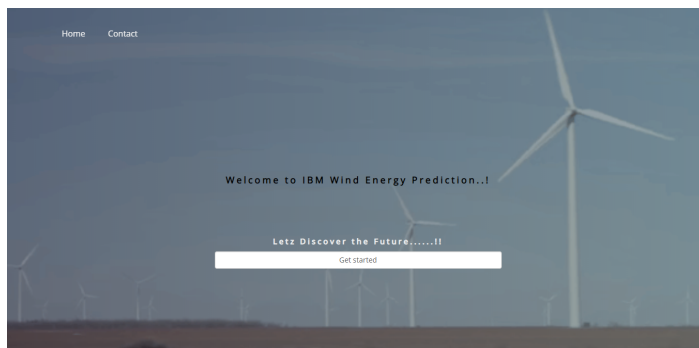
## GUI Creation:-

- ➡ We Done Our Gui With **Node-red**,it Is One Of **Service** Provied By **Our Ibm**
- ➡ Below Shown Node Are Our **Prediction Node** And Our **Gui Tab Nodes**



## Our GUI With IBM

👉 below shown image's are our different tabs in our GUI.



## HIGHLIGHT'S:-

- We Predicting **Wind Speed , Wind Direction** In **separate algo.**, and giving the **output of this prediction** to predict **power**, however real time scenario happens like this ,So our model should be best to **competitive with real time.**
- It can also **recommend** the time to **Utilize Power from grid.**
- So that we can give **rest period** to grid and **Save Energy .**

## SPECIAL THANKS TO:-

**IBM Platform And SMARTINTERNZ**

by:-

Team -Horoscoper's

**MOHAMED HANEEF I**

**MOHAMED ABDULLAH K**