PREDICTING THE ENERGY OUTPUT OF WIND TURBINE BASED ON WEATHER CONDITION

IBM NALAIYA THIRAN REPORT

Submitted by

S.THAMIZHARASI - 814619106004

A.KAVIYA -814619106001

K.KAVIYA -814619106002

A.VANAJA-814619106302

Department Of

ELECTRONICS AND COMMUNICATION ENGINEERING

Date	18 November 2022
Team ID	**PNT2022TMID045948**
Project Name	**Predicting the output energy of wind turbine based on weather condition**
Maximum Marks	8 Marks

INDUSTRY MENTOR: NIDHI **FACULTY MENTOR: RAJA J**

TRICHY ENGINEERING COLLEGE

TRICHY – 621 132

LIST OF CONTENTS:

1.INTRODUCTION :

Wind speed/power has received increasing attention around the earth due to its renewable nature as well as environmental friendliness.
With the global installed wind power capacity rapidly increasing, the wind industry is growing into a large-scale business. Reliable short-term wind speed forecasts play a practical and crucial role in wind energy conversion systems, such as the dynamic control of wind turbines and power system scheduling.
A precise forecast needs to overcome problems of variable energy production caused by fluctuating weather conditions.
Power generated by wind is highly dependent on the wind speed. Though it is highly non-linear, wind speed follows a certain pattern over a certain period of time. We exploit this time series pattern to gain useful information and use it for power prediction
.

1.1 PROJECT OVERVIEW :

Category:

Machine Learning

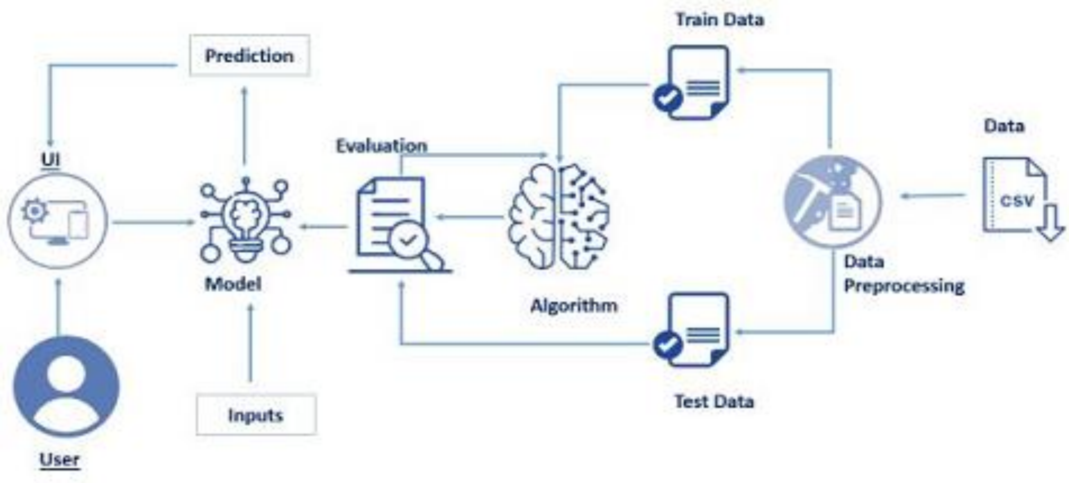
Skills Required:

Python, Python Web Frame Works, Python For Data Visualization, Data Preprocessing Techniques, Machine Learning, IBM Cloud, IBM Watson Studio, Python-Flask.

Project Description:

Wind power generation differs from conventional thermal generation due to the stochastic nature of wind. Thus wind power forecasting plays a key role in dealing with the challenges of balancing supply and demand in any electricity system, given the uncertainty associated with the wind farm power output. Accurate wind power forecasting reduces the need for additional balancing energy and reserve power to integrate wind power. For a wind farm that converts wind energy into electricity power, a real-time prediction system of the output power is significant. In this guided project , a prediction system is developed with a method of combining statistical models and physical models. In this system, the inlet condition of the wind farm is forecasted by the auto regressive model.

Technical Architecture:



1.2 PURPOSE:

- Accurate wind power forecasting **reduces the need for additional balancing energy and reserve power to integrate wind power**.
- For a wind farm that converts wind energy into electricity power, a real-time prediction system of the output power is significant.
- Wind energy **plays an increasing role in the supply of energy worldwide**.

2.LITERATURE SURVEY :

2.1 EXISTING PROBLEM:

- **Turbines produce noise and alter visual aesthetics:**
  Wind farms have different impacts on the environment compared to conventional power plants, but similar concerns exist over both the noise produced by the turbine blades and the visual impacts on the landscape .
- **Sound and visual impact** are the two main public health and community concerns associated with operating wind turbines. Most of the sound generated by wind turbines is aerodynamic, caused by the movement of turbine blades through the air.

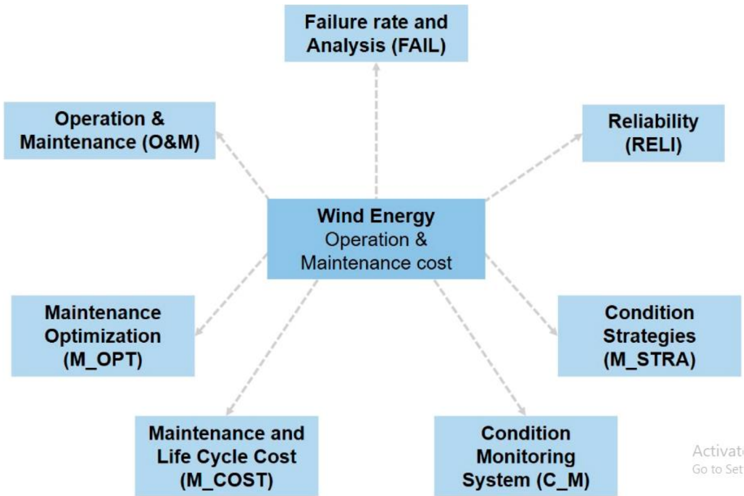
2.2 REFERENCE:

S.NO	AUTHOR/YEAR	TITTLE	TECHNIQUE USED	MERITS	DEMERITS
1.	A .Clifton /2012	Using machine learning to predict wind turbine power	Machine learning.	Strong function wind speed.	Affected by turbulence and shear.

		output.			
2.	Aman Bahugun /2013	Predicting the energy output of wind turbine based on weather conditions watson auto AI	IBM WATSON AUTO AI machine learning.	Predicted more accurately.	However ,in another study it was found that the prediction errors do not satisfy the Kolmogorove Smirnov test for normal distribution.
3.	Haroon Rashid/2020	Forecasting of wind turbine output power using machine Learning.	Machine learning	Accurate predicting of output power.	Absolute errors for the proposed model.
4.	Katya Vladislavleva/ 2019	Predicting the Energy Output of Wind Farms Based on Weather Data: Important Variables and their Correlation	wind energy, prediction, genetic programming, DataModeler	A good prediction of the energy output.	However, levels of production of wind energy are hard to predict as they rely on potentially unstable weather conditions present at the wind farm.
5.	J K Lundquist and P.Fleming1/2012	Using machine learning to predict wind turbine power output	Machine learning, classification and regression trees, wind energy, wind turbine	Reduce bias in power predictions that arise because of the different turbulence and shear at the new site, compared to the test site.	Changes of wind direction with height, non-uniform shear, and the state of the turbine were not considered here but may impact turbine deployment sites.
6.	Aoife M. Foley/2020	Review Current methods and advances in . forecasting of wind power	Meteorology Numerical weather prediction Probabilistic forecasting	Thus wind power forecasting plays a key role in dealing with	Overall accurate wind power prediction reduces the financial and

		generation	Wind integration wind power forecasting	the challenges of balancing supply and demand in any electricity system, given the uncertainty associated with the wind farm power output.	technical risk of uncertainty of wind power production for all electricity market participants.

2.3 PROBLEM STATEMENT DEFINITION :



3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:

3.2 IDEATION & BRAINSTORMING:

BRAINSTROM & IDEA PRIORITIZATION:

Use this template in your own brainstorming sessions so your team can unleash their .Once all sticky notes have been grouped, give each cluster a sentence-like label.

If a cluster is bigger than six stickynotes, tryandseeifyouandbreakitupintosmallersub-groups.

Prioritize

Yourteamshouldallbeonthesamepageaboutwhat'simportantmoving forward. Place your ideas on this grid to determine whichideasareimportantandwhicharefeasible.

Afteryoucollaborate

Youcanexportthemuralasanimageorpdftosharewithmembersofyourcompanywho mightfind it helpful..

A ***Teamgathering***

Definewhoshouldparticipateinthesessionandsendaninvite.Sharerelevantin for mationorpre-workahead.

B ***Setthegoal***

Think aboutthe problemyou'llbe focusingonsolving inthebrainstormingsession.

C ***Learnhowtousethefacilitationtools***

UsetheFacilitationSuperpowerstorunahappy.

3.3 PROPOSED SOLUTION:

ProposedSolutionTemplate:

Projectteamshallfillthefollowinginformationinproposedsolutiontemplate.

S .No.	Parameter	Description
1.	ProblemStatement(Problemtobe solved)	Our time –tested kombi Box produce gives a higher priority to those forecasts having the lowest prediction error in respective weather situation.
2.	Idea/Solutiondescription	Our aim is to map weather data to energy production
3.	Novelty/ Uniqueness	A good overview on the different methods that were recently applied in forecasting of wind power generation can be found in .
4.	SocialImpact/ CustomerSatisfaction	If there's a lot of wind, you get more energy output than if there's less wind, which means you will likely want to do maintenance when the winds are low to minimize downtime.
5.	BusinessModel(RevenueModel)	Real time projections for solar power, including behind-the-meter generation Grid-oriented forecasts
6.	Scalabilityofthe Solution	The model prediction is then showcased on user interface to predict the energy output of wind turbine

3.4 PROBLEM SOLUTION FIT:

1. CUSTOMERSEGMENT(S):

Windflowsoverthebladescreatinglift (similartotheeffect onairplanewings),whichcauses the blades to turn.

The blades areconnected to a drive shaft that turns an electricgenerator,whichproduces (generates).

2JOBS-TO-BE-DONE/PROBLEMS:

Turbinesproducenoiseandaltervisualaesthetics

Wind farms have different impactson the environment compared toconventionalpowerplants,butsimilar concerns exist over both thenoiseproducedbytheturbinebladesandthevisualimpactsonthe landscape.

3. TRIGGERS:

The wind speed is always fluctuating andthus the energy content of the wind is alsoalways changing. Exactly how large thevariationisdependsboth ontheweather and onlocalsurfaceconditionsandobstacles.

4.EMOTIONS:

BEFORE / AFTER

Before:who live in close proximity to windturbines say they experience sleepdisturbances, headaches and concentrationproblems.

After: Between working long hours, climbingturbinesmultipletimes aday,anddealingwithextreme heat in the summer and cold in thewinter.

5.AVAILABLESOLUTIONS:

These residential windmills cangenerate electricity by churning thewind through its blades, which in turnrotates the turbine and generatespower, which can meet the needs of asmall family. The energy that is yieldedfrom these wind turbinesisclean, renewable, andarealsocost-effective.

6.CUSTOMERCONSTRAINTS:

- Intermittent
- Low operatingcosts**Noise** and visual pollutionEfficient useoflandspace
- Some adverse environmentalimpact
- Windenergyisajobcreator

7.BEHAVIOUR:

The wind speed is always fluctuatingandthus the energy content of the wind isalso always changing.

Exactly how largethevariationis depends bothontheweather and on local surface conditionsandobstacles.

8.CHANNELSofBEHAVIOUR:

Online:

Anefficientautomatedapproach towindfarmoperationmonitoringispresented.

Offline:

Product isavailableforofflineusage.

9.PROBLEMROOTCAUSE:

The degradation, weakening and debondingof the adhesive layers (in the trailing orleading edges or on the spar/shell joint) isoneofthemainprocessesleadingtowind turbinebladefailure.

10.YOURSOLUTION:

Larger rotor diameters allow wind turbines to sweepmore area, capture more wind, and produce moreelectricity. A turbine with longer blades will be able tocapture more of the available wind than shorterblades—eveninareaswithrelativelyless wind.

4. REQUIREMENT ANALYSIS:

4.1. FUNCTIONAL REQUIREMENTS:

Followingarethefunctionalrequirementsoftheproposedsolution.

FRNo.	FunctionalRequirement(Epic)	SubRequirement(Story/Sub-Task)
FR-1	UserRegistrationandl ogginginbyentering theirusernameand password.	RegistrationthroughForm.
FR-2	User Confirmation byvalidatingtheusernam ewithrespectto thepassword	Confirmationviapop-upMessage.
FR-3	Displayingthefurtheri nformationabout Theapplication.	Byselectingtheaboutbuttonthedetailsoftheapplicati onwillbedisplayed.
FR-4	Validatingthecityname.	System checks whether the city entered by theuser is present or not. If present it will collectthefurtherdetailselseitwilldisplaythepop-upmessageas error inthecity.
FR-5	Checkingthedatatypeoftheva lue.	Systemchecksforthedatatypeofthevalueenteredb y theuser.
FR-6	Validatingallrequiredfields.	Before predicting the output the systemcheckswhetherallthevaluesareentere dbytheuserand checkswhetherallvaluesarecorrect.
FR-7	Displaying weatherConditionsforagiv encity.	ItdisplaystheweatherofthecitywhichhaveBeensel ected.
FR-8	Displaying predictedEnergyoutp utpower.	The predicted output will be displayed asamountofwindenergypowergenerated.

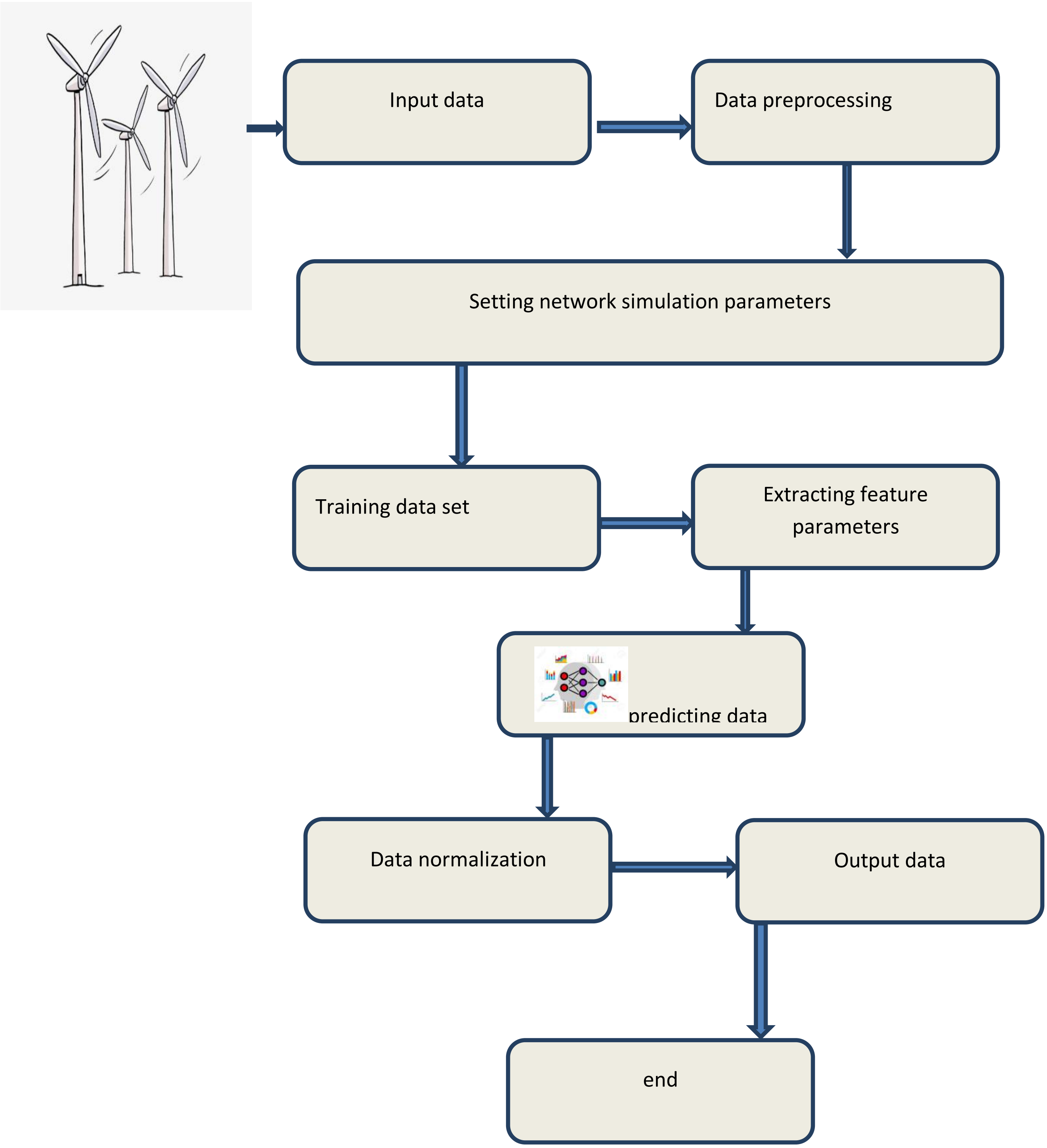
4.2. NON-FUNCTIONAL REQUIREMENTS:

Followingarethenon-functionalrequirementsoftheproposedsolution.

FRNo.	Non-FunctionalRequirement	Description
NFR-1	Usability	Thesystemsatisfiestheusergoalsandtheapplicationi seasytouse.
NFR-2	Security	Thedataprovidedtosystemwillbeprotectedfrom Attacksandunauthorizedaccess
NFR-3	Reliability	Thesystemwillprovidetheconsistencyinoutputwith outproducinganerror.
NFR-4	Performance	Theperformancewillneverdegradeeventhewor kloadis increased.
NFR-5	Availability	Theapplicationisavailablefor24*7
NFR-6	Scalability	Thesystemcanbeusedaswebapplicationaswella s mobile application with a sufficientinternet availability.

5. PROJECT DESIGN:

5.1 DATA FLOW DIAGRAMS:



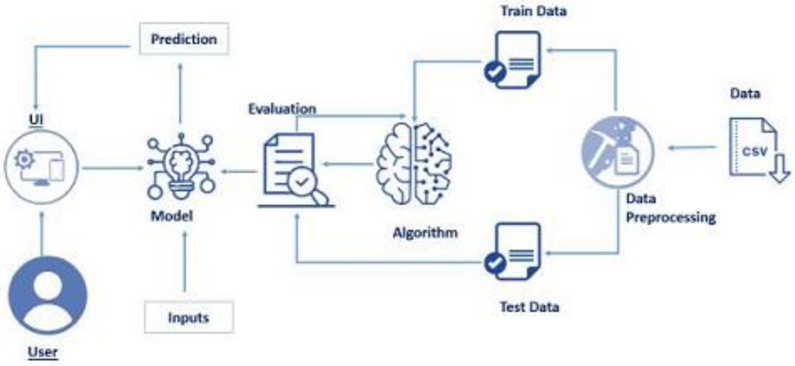
5.2 SOLUTION & TECHNICAL ARCHITECTURE:

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Wind power generation differs from conventional thermal generation due to the stochastic nature of wind.
- Thus wind power forecasting plays a key role in dealing with the challenges of balancing supply and demand in any electricity system, given the uncertainty associated with the wind farm power output.
- The inlet condition of the wind farm is forecasted by the auto regressive model.
- We report on the correlation of the different variables for the energy output.

Example - Solution Architecture Diagram:



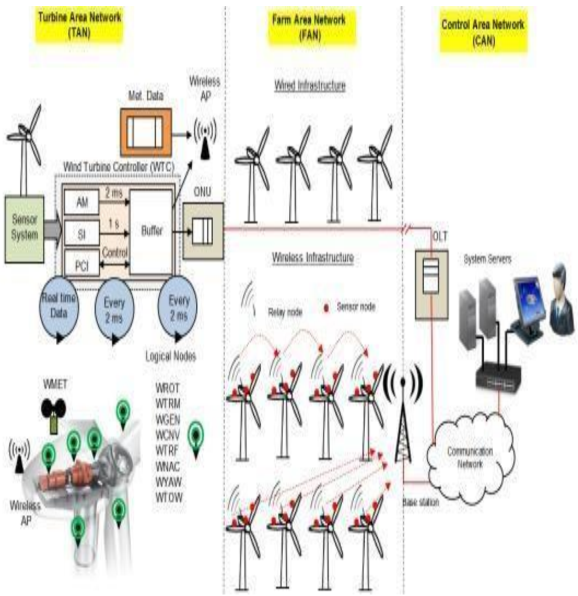
Reference: https://github.com/SmartPracticeschool/llSPS-INT-3437-Predicting-the-Energy-Output-of-Wind-Turbine-Based-on-Weather-Conditions-Watson-Auto-

TechnicalArchitecture:

TheDeliverableshall includethearchitecturaldiagramasbelowandtheinformationasperthetable1&table2

Example:Predictingtheenergyoutputofwindfarmbasedonweatherconditions.

Reference: https://www.mdpi.com/1996-1073/7/6/3900



Guidelines:

The proposed communication network architecture for the Smart-WPFconsists of three networks: the turbine area network (TAN), the farm area network(FAN), and the control area network (CAN).

It consists of hierarchical architectureswhereLevel1isasensornetworkinasinglewindturbine,Level2isthewindturbine-to-wind turbine interaction in the WPF, Level 3 is the local control center to windturbine interaction, and Level 4 is the farm-to-farm interaction to optimize gridoperation.

Inordertoimplementhierarchicalnetworkarchitectures,ahybridcommunication solution is considered. EPON-based architecture represents a wiredsolution,whileZigBee-Proisconsideredforthewirelesssolution.Inthiswork,Levels1 and 2 are explained in more detail, while Levels 3 and 4 are out the scope of thiswork

6. PROJECT PLANNING & SCHEDULING:

6.1 SPRINT PLANNING & ESTIMATION:

Sprint	SprintStartDate	SprintEndDate(Planned)	Story PointsCompleted(asonPlannedEndDate)
Sprint-1	01 Nov2022	14 Nov2022	20
Sprint-2	07 Nov2022	14Nov2022	20
Sprint-3	07Nov2022	15Nov2022	20
Sprint-4	09 Nov2022	15Nov2022	20

6.2 SPRINT DELIVERY SCHEDULE:

Sprint	SprintStartDate	SprintEndDate(Planned)	Story PointsCompleted (ason PlannedEndDate)	SprintReleaseDate(Actual)
Sprint-1	01 Nov2022	14 Nov2022	20	14 Nov 2022
Sprint-2	07 Nov2022	14Nov2022	20	14 Nov2022
Sprint-3	07Nov2022	15Nov2022	20	15 Nov2022
Sprint-4	09 Nov2022	15Nov2022	20	15 Nov2022

7. CODING & SOLUTIONING:

7.1 FEATURE 1:

```
Saving Wind Dataset.csv to Wind Dataset.csv
```

In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import joblib
```

Data Preprocessing

In [3]:

```
path = "Wind Dataset.csv"
df = pd.read_csv(path)
df.rename(columns={"Date/Time":"Time", "LV ActivePower (kW)":"ActivePower(KW
                   "Wind Direction(°)":"Wind_Direction"},
                   inplace=True)
```

In [4]:

```
df.head()
```

0	01 01 2018 00:00	380.047791	5.311336	416.328908259.994904	
1	01 01 2018	453.769196	5.672167	519.917511	268.641113

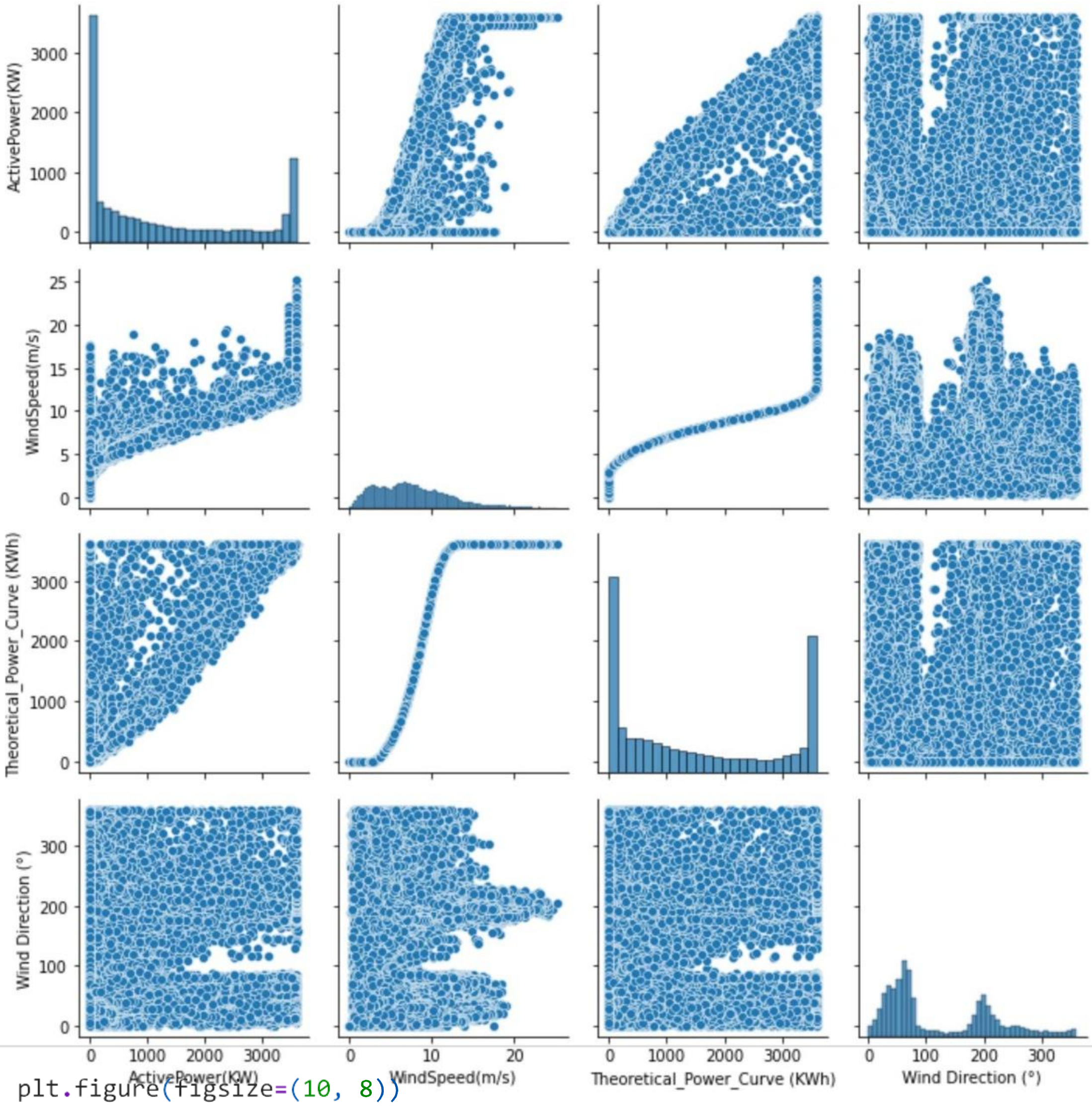
	00:10				
2	01 01 2018 00:20	306.376587	5.216037	390.900016	272.564789
3	01 01 2018 00:30	419.645904	5.659674	516.127569	271.258087
4	01 01 2018	380.650696	5.577941	491.702972	265.674286

```
ActivePower(KW)                  0
WindSpeed(m/s)                   0
Theoretical_Power_Curve (KWh)    0
Wind Direction (°)               0
dtype: int64
```

In [6]:

```
sns.pairplot(df)
```

Out[6]:



In [7]:

```
plt.figure(figsize=(10, 8))
corr = df.corr()
ax = sns.heatmap(corr, vmin = -1,vmax = 1, annot = True)
bottom, top = ax.get_ylim()
ax.set_ylim(bottom + 0.5, top - 0.5)
print(corr)
```

```
                               Wind Direction (°)
ActivePower(KW)                         -0.062702
WindSpeed(m/s)                          -0.077188
Theoretical_Power_Curve (KWh)           -0.099076
Wind Direction (°)                       1.000000
```

In [5]:
```
df.isnull().sum()
```

Out[5]: Time 0

In [8]:

```
df["Time"] = pd.to_datetime(df["Time"], format = "%d %m %Y %H %M", errors =
```

Splitting the data to Train and Test

In [9]:

```
y = df["ActivePower(KW)"]
X = df[["Theoretical_Power_Curve (KWh)", "WindSpeed(m/s)"]]

from sklearn.model_selection import train_test_split
train_X, val_X, train_y, val_y = train_test_split(X, y, random_state=0)
```

Model Building

In [10]:

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error,r2_score

forest_model = RandomForestRegressor(n_estimators = 750, max_depth = 4, max_

forest_model.fit(train_X, train_y)
```

Out[10]:

```
RandomForestRegressor(max_depth=4, max_leaf_nodes=500, n_estimators=750,
                      random_state=1)
```

In [11]:

```
power_preds = forest_model.predict(val_X)

print(mean_absolute_error(val_y, power_preds))
print(r2_score(val_y, power_preds))
```

```
0.911349642890175
```

In [12]:
```
joblib.dump(forest_model, "power_prediction.sav")
```

Out[12]:
```
['power_prediction.sav']
```

In [13]:
```
df
```

Out[13]:

	Time	ActivePower(KW)	WindSpeed(m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)
0	NaT	380.047791	5.311336	416.328908	259.994904
1	NaT	453.769196	5.672167	519.917511	268.641113
2	NaT	306.376587	5.216037	390.900016	272.564789
3	NaT	419.645904	5.659674	516.127569	271.258087
4	NaT	380.650696	5.577941	491.702972	265.674286
...	...	...	...	...	...
50525	NaT	2963.980957	11.404030	3397.190793	80.502724
50526	NaT	1684.353027	7.332648	1173.055771	84.062599
50527	NaT	2201.106934	8.435358	1788.284755	84.742500
50528	NaT	2515.694092	9.421366	2418.382503	84.297913
50529	NaT	2820.466064	9.979332	2779.184096	82.274620

50530 rows × 5 columns

:

8.TESTING:

8.1 TEST CASES:

	TestingType	FunctionResult
TestingNo:1	Functionalitytesting	Yes
	Usabilitytesting	Yes
	Interfacetesting	Yes
	Performancetesting	Yes(medium)
	Securitytesting	Yes

	TestingType	FunctionResult
TestingNo:2	Functionalitytesting	Yes
	Usabilitytesting	Yes
	Interfacetesting	Yes
	Performancetesting	Yes(medium)
	Securitytesting	Yes

	Testing Type	Function Result
TestingNo:3	Functionality testing	Yes
	Usability testing	Yes
	Interface testing	Yes
	Performance testing	Yes
	Security testing	Yes

	Testing Type	Function Result
TestingNo:4	Functionality testing	Yes
	Usability testing	Yes
	Interface testing	Yes
	Performance testing	Yes
	Security testing	Yes

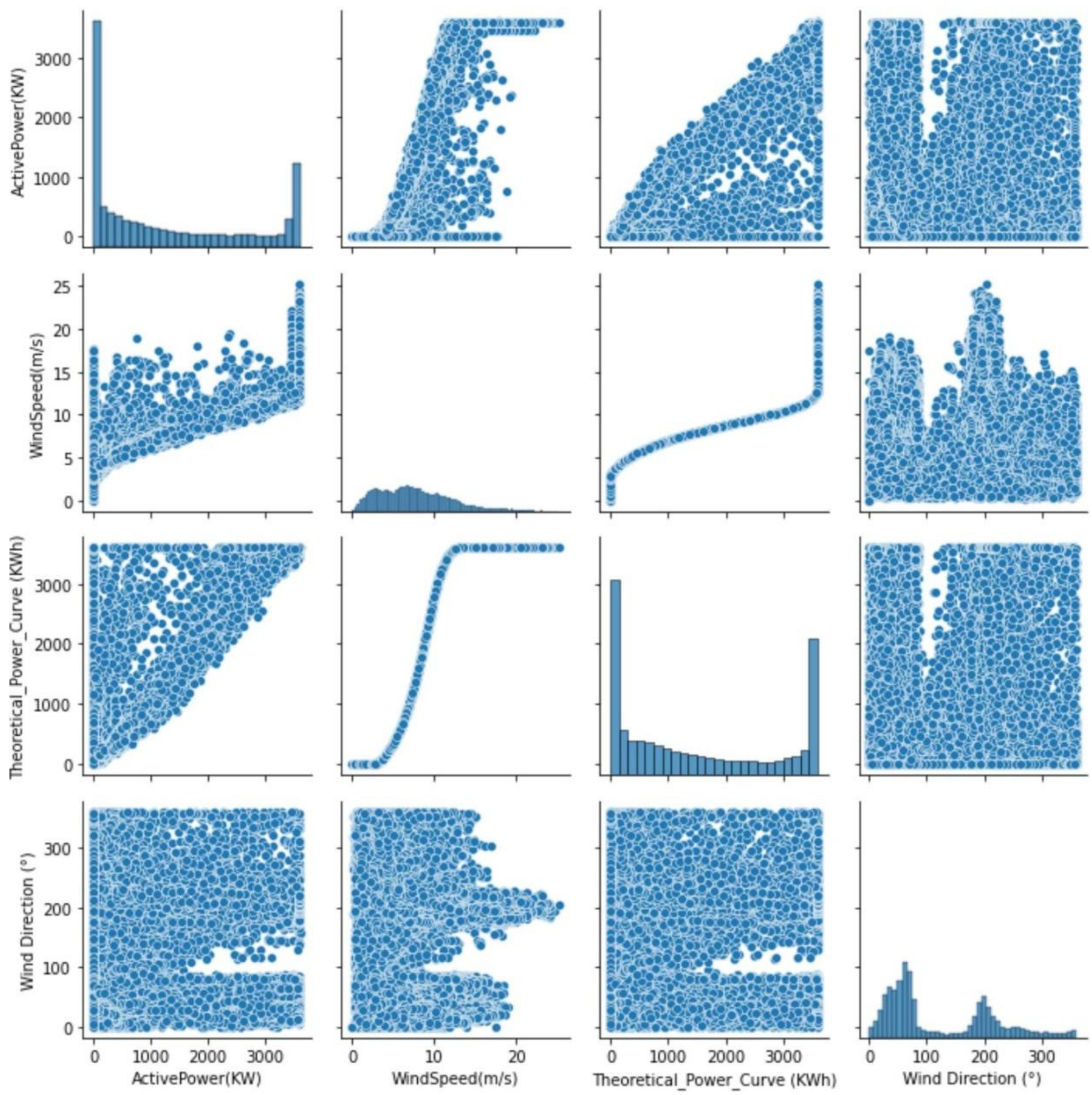
9.RESULT:

9.1 PERFORMANCE METRIES:

DATA PREPROCESSING:

Out[4]:

0	01 01 2018 00:00	380.047791	5.311336	416.328908	259.994904
1	01 01 2018 00:10	453.769196	5.672167	519.917511	268.641113
2	01 01 2018 00:20	306.376587	5.216037	390.900016	272.564789
3	01 01 2018 00:30	419.645904	5.659674	516.127569	271.258087
4	01 01 2018	380.650696	5.577941	491.702972	265.674286

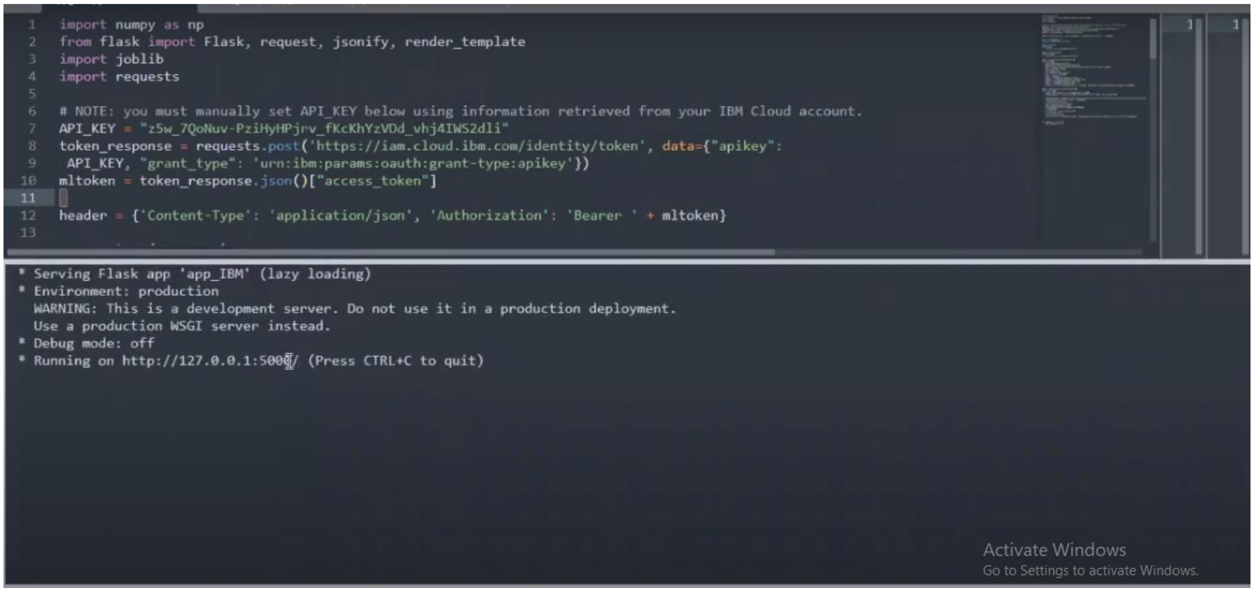


ActivePower(KW)
WindSpeed(m/s)
Theoretical_Power_Curve (KWh)
Wind Direction (°)

MODEL BUILDING:

```
Out[10]:  RandomForestRegressor(max_depth=4, max_leaf_nodes=500, n_estimators=750,
                                random_state=1)
```

FINAL RESULT:



10. ADVANTAGES& DISADVANTAGES:

ADVANTAGES:

- Weather Underground Services provide very accurate Historical Weather Data which increased the accuracy of model.
- Website is more convenient to use due to zero storage.
- With Choosing city, Website can accurately predict power output using weather condition.

DISADVANTAGES:

- Weather API is paid and the free version provide limited API requests per day.
- Android Website can be deployed on IBM Cloud.
- No free server available on IBM Cloud for deploying Backend.

11. CONCLUSION:

lusions In this study we showed that wind energy output can be predicted from publicly available weather data with accuracy at best 80% $R2$ on the training range and at best 85, 5% on the unseen test data.

We identified the smallest space of input variables (windGust2 and dewPoint), where reported accuracy can be achieved, and provided clear trade-offs of prediction accuracy for decreasing the input space to the windGust2 variable.

We demonstrated that an off-the-shelf data modeling and variable selection tool can be used with mostly default settings to run the symbolic regression experiments as well as variable importance, variable contribution analysis, ensemble selection and validation.

12. FUTURE SCOPE:

Most wind power forecasting models study ‘regular’ wind conditions.

The EU funded project called ‘Safewind’ aims to improve wind power prediction overchallenging and extreme weather periods and at different temporal and spatial scales.

Development activities are on-going to reduce error in the wind power prediction, to improve regionalized wind power forecasting for on - shore windfarms and to derive methods for wind power prediction for offshore wind farms.

It is possible that use of ensemble and combined weather prediction methods methods together may enhance forecasting.

If the error in wind power forecasting and prediction is reduced then electricity markets can trade with more certainty.

Contract errors as a function of time in electricity markets can be as high as 39% for a forecasting lead time of 4 h Gubina et al.

present a new tool called the WILMAR and ANEMOS scheduling Methodology (WALT) to reduce the number of thermal generators on stand-by or in reserve using the probability of generation outages and load shedding are system reliability criteria instead of generation adequacy based solely on generation outage.

The wind and load forecast errors are modelled using a Gaussian stochastic variable approach.

However, in another study it was found that the prediction errorsdo not satisfy the KolmogoroveSmirnov test for normal distribution.

In Ramìrez and Carta , it was shown that, the use of autocorrelated (and thus not independent) successive hourly mean wind speeds, though invalidating all of the usual statistical tests, has no appreciable effect on the shape of the pdf estimated from the data.

13. APPENDIX:

13.1 SOURCE CODE:

- **app.py**

```
import numpy as
npimport streamlit
as stimport pandas
as
pdimportdatetime

importplotly.graph_objectsasgoi
mportbase64

import
timeimporttensor
flow


st.set_page_config(page_ti
tle="DEEPWIND",page_icon
=""

)

old_models=tensorflow.keras.models.load_model('model.h5')



#setbackground,usebase64toreadlocalfiledef
get_base64_of_bin_file(bin_file):

    withopen(bin_file,'rb')asf:d
        ata=f.read()

    returnbase64.b64encode(data).decode()


defset_png_as_page_bg(png_file):

bin_str=get_base64_of_bin_file(png_file)
page_bg_img='''

<style>

    body {

    background-
    image:url("data:image/png;base64,%s");background
    -size:cover;
```

```
    }
</style>
    '''%bin_str

st.markdown(page_bg_img,unsafe_allow_html=True)re
    turn
set_png_as_page_bg('gr.gif')

defhome():
    return"welcome"

def
    predict(temperature,pressure,wind_speed,wind_direction):values=np.array([[tempe
    rature,pressure,wind_speed,wind_direction]]
```

```
prediction=old_models.predict(values.reshape(-
1,1,4),batch_size=1)print(prediction)

    returnprediction

defmain():

st.sidebar.markdown("<h1style='text-                    Navigation Bar</h1>",
align:center;color:black;'>unsafe_allow_html=True)

  nav =st.sidebar.radio("",["Home     ","User defined Prediction","Forecasting"])

  ifnav =="Home     ":

st.markdown("<h1 style ='color:black; text_align:center;font-family:times new roman;font-
size:20pt; font-weight: bold;'>DEEPWINDS</h1>", unsafe_allow_html=True)
st.markdown("<h1style='color:brown;text_align:center;font-weight:bold;font-
size:19pt;'>MadebyQuadTechieswith</h1>",unsafe_allow_html=True)

st.markdown("<h1 style ='color:black; text_align:center;font-family:times new roman;font-
weight:bold;font-size:16pt;'>◆WINDPOWERPREDICTIONDLWEB-
APP◆/h1>",unsafe_allow_html=True)


  if nav =="User definedPrediction":

set_png_as_page_bg('gra(1).jpg')

st.markdown("<h1 style='text-align: center; color: green;'>UserInput Parameters</h1>",
unsafe_allow_html=True)

withst.beta_expander("Preferences"):

st.markdown("<h1style='text-align:left;font-weight:bold;color:black;background-

color:white;font-                          ½(°C) </h1>",unsafe_allow_html=True)
    size:11pt;'>Temperaturecol1,col2=st
    .beta_columns(2)

    withcol1:

min_temp=st.number_input('Minimum Temperature(°C)',min_value=-
89,max_value=55,value=-15,step=1)

    withcol2:
```

```
max_temp=st.number_input(' Maximum Temperature(°C)',min_value=-
88,max_value=56,value=50,step=1)

st.markdown("<h1 style='text-align: left; font-weight:bold;color:black;background-
color:white;font-size:11pt;'> Wind Speed (m/s) </h1>",unsafe_allow_html=True)

    col1,col2 =
    st.beta_columns(2)with col1:

min_speed=st.number_input('     MinimumWindSpeed

(m/s)',min_value=0,max_value=99,value=1,step=1)wit
    h col2:

max_speed=st.number_input('     MaximumWindSpeed

(m/s)',min_value=2,max_value=100,value=27,step=1)st.
write("")

  temperature=st.slider('Temperature  ½[°C]', min_value=min_temp, step=1,
max_value=max_temp,value=max_temp)

  pressure =st.slider('Pressure  [atm]', 0.9, 1.0, 1.
```

```
wind_speed =st.slider('Wind Speed [m/s]',min_value=min_speed, step=1,

        max_value=max_speed,value=max_speed)

        wind_direction =st.slider('Wind Direction [deg]', 0, 1, 360)
           result=""

           ifst.button("Predict"):

              result =
        predict(temperature,pressure,wind_speed,wind_direction)st.balloons(
        )

        st.success('PredictedPoweris{}kW'.format(result))


            if nav =="Forecasting":

       set_png_as_page_bg('04.gif')

        st.markdown("<h1 style='text-align: center; color:black ;'> FORECASTING </h1>",
        unsafe_allow_html=True)

          #Setup fileupload

        st.markdown("<h1 style='text-align:center; color:white;background-color:black;font-
        size:14pt'>   Upload yourCSV orExcel file. (200MB max) </h1>", unsafe_allow_html=True)

        uploaded_file = st.file_uploader(label="",type=['csv',
             'xlsx'])global df

             ifuploaded_fileis not None:

               print(uploaded_file)

        st.markdown("<h1 style='text-align:center; color:black;background-
        color:lightgreen;font-size:14pt'> Fileupload
        successful </h1>",unsafe_allow_html=True)

               try:

                 df=pd.read_csv(uploaded_file)s
        t.write(df)


               exceptExceptionase:
```

```
df =
pd.read_excel(uploaded_file)st.write(df)

st.markdown("<h1style='text-align:center;color:black;background-color:powderblue;font-size:14pt'>INPUTDATA</h1>", unsafe_allow_html=True)

trace
=go.Scatter(x=df['DateTime'],y=df['Powergeneratedbysystem|(kW)'],mode='lines',name='Data
')

layout=go.Layout(title="",xaxis={'title':"Date"},yaxis={'title' :"Powergeneratedbysystem
|(kW)"})

fig = go.Figure(data=[trace],
layout=layout)st.write(fig)

df1=df.reset_index()['Powergeneratedbysystem|(kW)']i
mportmatplotlib.pyplot asplt

st.write("\n")
```

```
st.markdown("<h1style='text-align:center;color:black;background-color:powderblue;font-size:14pt'>##INPUTDATAINTERMSOFNO. OFHOURS##</h1>",

unsafe_allow_html=True)

        trace = go.Scatter(x = df1.index,y = df['Power generated by system | (kW)'],mode =
'lines',name ='Data' )

        layout = go.Layout(title = "",xaxis = {'title' : "No. of hours"},yaxis = {'title' :
"Powergeneratedby system (kW)"})


        fig=go.Figure(data=[trace],layout=layout)#f
         ig.show()

st.write(fig)

        from sklearn.preprocessing import
        MinMaxScalerscaler=MinMaxScaler(feature_range=
        (0,1))df1=scaler.fit_transform(np.array(df1).reshape
        (-1,1))

    ##splitting dataset into train and test
splittraining_size=int(len(df1)*0.65)test_siz
e=len(df1)-training_size

        train_data,test_data=df1[0:training_size,:],df1[training_size:len(df1),:1]


        import numpy

      # convert an array of values into a dataset
                     matrix#convertan
   arrayofvaluesintoadatasetmatrixdefcreate_data
                 set(dataset,time_step=1):

           dataX,dataY=[],[]

               foriinrange(len(dataset)-time_step-1):

                        a = dataset[i:(i+time_step), 0]###i=0, 0,1,2,3-----
                 99100dataX.append(a)
```

```
dataY.append(dataset[i + time_step,
0])returnnumpy.array(dataX),numpy.array(data
Y)

#reshapeintoX=t,t+1,t+2,t+3andY=t+4tim
e_step=30

X_train, y_train = create_dataset(train_data,
time_step)X_test,ytest=create_dataset(test_data,
time_step)

#reshapeinputtobe[samples,timesteps,features]whichisrequiredforLSTMX_trai
n=X_train.reshape(X_train.shape[0],X_train.shape[1], 1)

X_test=X_test.reshape(X_test.shape[0],X_test.shape[1],1)#
Create theBILSTMmodel

fromtensorflow.keras.modelsimportSequential
fromtensorflow.keras.layersimportDense

fromtensorflow.keras.layersimportLSTM

fromtensorflow.keras.layersimportBidirectional
model=Sequential()

model.add(Bidirectional(LSTM(250,input_shape=(1,30))))m
odel.add(Dense(1))

model.compile(loss='mae',optimizer='adam')
```

```
model.fit(X_train,y_train,validation_data=(X_test,ytest),epochs=10,batch_size=64,verbose=1)
        importtensorflowas tf

train_predict=model.predict(X_train)test_predict=model.predict(X_te
st)

   #Transformback to original
formtrain_predict=scaler.inverse_transform(train_predict)test_predict=
scaler.inverse_transform(test_predict)

   ###CalculateRMSEperformancemetricsim
        portmath

        fromsklearn.metricsimportmean_squared_error
math.sqrt(mean_squared_error(y_train,train_predict))

   ###TestDataRMSEmath.sqrt(mean_squared_error(ytest,test_predict))
   ### Plotting

    # shift train predictions for
plottinglook_back=30

trainPredictPlot=numpy.empty_like(df1)t
rainPredictPlot[:,:] =np.nan

trainPredictPlot[look_back:len(train_predict)+look_back,:]=train_predict#
   shift test predictions forplotting

testPredictPlot =
numpy.empty_like(df1)testPredictPlot[:,
:] =numpy.nan

testPredictPlot[len(train_predict)+(look_back*2)+1:len(df1)-1, :] =
   test_predict#plot baselineand predictions

st.markdown("<h1style='text-align:center;color:black;background-color:powderblue;font-

size:14pt'>TRAINANDTEST DATA          </h1>",unsafe_allow_html=True)


        #plt.plot(scaler.inverse_transform(df1))plt.plot(scaler.inverse_transfor
m(df1), color="blue", linewidth=1, linestyle="-")plt.xlabel('No.ofhours')
```

```
   # Set the y axis label of the current
axis.plt.ylabel('Powergeneratedbysystem|(k
W)')



plt.plot(trainPredictPlot,label='Train Data',color="black",linewidth=2, linestyle="--
")plt.plot(testPredictPlot,label='Test Data',color="orange",linewidth=2, linestyle="--
")plt.legend(loc="upperleft")

     #plt.show()
st.pyplot(plt)



x_input=test_data[len(test_data)-30:].reshape(1,-
1)temp_input=list(x_input)temp_input=temp_inp
ut[0].tolist()



   #demonstratepredictionfornext24hoursfr
        omnumpy import array

lst_output=[
]n_steps=30i
=0
```

```
        while(i<24):if(len(temp_i
          nput)>30):#print(temp
          _input)

x_input=np.array(temp_input[1:]
)x_input=x_input.reshape(1,-1)

x_input = x_input.reshape((1, n_steps,
1))yhat = model.predict(x_input,
verbose=0)temp_input.extend(yhat[0].to
list())temp_input=temp_input[1:]lst_out
put.extend(yhat.tolist())

i=i+1

          else:

x_input = x_input.reshape((1,
n_steps,1))yhat=model.predict(x_input,v
erbose=0)

           print(yhat[0])temp_inpu
t.extend(yhat[0].tolist())

           print(len(temp_input)
)lst_output.extend(yhat.tolist())
i=i+1



        print(lst_output)day_new=np.arange(
1,31)day_pred=np.arange(len(df1),len(df1)+2
4)

        import matplotlib.pyplot as
        pltprint(len(df1))progress=st.
        progress(0)

        for i in
range(100):time.sleep(0.
1)progress.progress(i+1)s
t.balloons()
```

```
st.markdown("<h1style='text-align:center;color:black;background-color:powderblue;font-
size:14pt'>PREDICTEDRESULTSFORNEXT24 HOURS</h1>",

unsafe_allow_html=True)plt.plot(day_pred,scaler.inverse_transform(lst_output),color="green",
linewidth=1.5, linestyle="--",marker='*',markerfacecolor='yellow',markersize=7)

plt.legend('GTTP',loc="upperleft")


plt.xlabel('No.ofhours')

    # Set the y axis label of the current
axis.plt.ylabel('Powergeneratedbysystem|(k
W)')


st.pyplot(plt)

st.markdown("<h1 style='text-align: center; color:black ;background-
color:yellow;font-size:14pt'>G-GivenData, \nT-Train Data,\nT-
TestData,\nP-Predicted Results</h1>",unsafe_allow_html=True)
```

```
st.write(scaler.inverse_transform(lst_output))

ifname_____=="main":

main()
```

model.py

```
import pandas as
pdimport
datetimeimportnum
pyasnp

fromkeras.model
simportSequenti
alfromkeras.layer
simportDense

fromkeras.layersimportLSTM

fromkeras.layersi
mportBidirectional
importpandas as
pd

importkeras

'''Loadingdata'''

df =
pd.read_excel('
Dataset.csv')df
=df.drop(colum
ns=['DateTime']
)''' Cleaning
Data
'''#dataframe.d
rop['Date'].valu
es
```

```
df['Power generated by system | (kW)'].replace(0,
np.nan,
inplace=True)df['Powergeneratedbysystem|(kW)'].filln
a(method='ffill',inplace=True)


X=df.drop(columns=['Powergeneratedb
y system|(kW)'])Y = df[['Power
generated by system |
(kW)']]X=np.array(X).reshape(-1,1,4)

Y=np.array(Y).reshape(-1,1,1)


model=Sequential()

model.add(Bidirectional(LSTM(100,
activation='relu',input_shape=(-
1,1,4))))model.add(Dense(1))

model.compile(loss='mae',optimizer='adam',metrics=['accuracy'])

model.fit(X,Y,epochs=100,callbacks=[keras.callbacks.EarlyStopping(patience=3)])


test_data = np.array([[-
4.858,0.989741,6.651,273]])o=model
.predict(test_data.reshape(-
1,1,4),batch_size=1)print(o)


# Saving
model to
diskmodels=
model.save('
model.h5').
```