EE&AE
2024 9th International Conference
on Energy Efficiency and Agricultural Engineering
27-29 June 2024, Ruse, Bulgaria
IEEE

A Computational Approach for Predicting the Short-Term Output Power of a Wind Turbine in an Urban Area

A. Fazylova, G. Mihaylov, T. Iliev, I. Stoyanov, S. Kosunalp, G. Sergazin

Institute of Telecommunications and Space Engineering, Almaty University of Power Engineering and Telecommunications named after Gumarbek Daukeyev, Almaty, Kazakhstan, e-mail: sharf_06@inbox.ru

GOAL OF THE STUDY

This paper presents an approach aimed at estimating the anticipated power output of a wind turbine by leveraging real-time and forecasted wind speed data sourced from the Internet.

METHODOLOGY OF THE INVESTIGATION

Wind speed is the main factor affecting the power output of a wind turbine. To determine the wind speed forecast formula, it was decided to use the least squares method, which makes it possible to minimize the sum of squared errors between the predicted and actual values. For the calculation, the following parameters of the wind generator were taken: the air density is 1.225 kg/m^3, the surface area of the wind generator is 10 m2, and the power factor of the wind generator is 0.35

MAIN RESULTS FROM THE STUDY

To calculate the predicted output power of the selected wind turbine for the city of Sofia (Bulgaria), software was developed that allows you to automatically receive real-time data from the site www.accuweather.com on the current and predicted wind speed. The Python code is shown in Fig. 1. The result of the output is presented in the form of a database in Fig. 2 and Table 2.

```
import requests
from bs4 import BeautifulSoup
from datetime import datetime
# Link to the weather page in Sofia
url"https://www.accuweather.com/ru/bg/sofia/51097/hour
ly-weather-forecast/51097"
# Headers for the request
headers = '{ User-Agent': 'Mozilla/5.0 (Windows NT
10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/58.0.3029.110 Safari/537.3'}
# Send a GET request to the page
response = requests.get(url, headers=headers)
# Create a BeautifulSoup object for parsing the page
soup = BeautifulSoup(response.content, 'html.parser')
# Find the table with weather data
table = soup.find_all('table')[0]
# Get the rows of the table
rows = table.find_all('tr')
# Create a list to store the data
data = []
# Skip the first row (table header)
for row in rows[1:]:
    # Get the values of the cells in the current row
    cells = row.find_all('td')
    # Get the date and time from the first cell
    date_time = datetime.strptime(cells[0].text,
'%d.%m.%Y %H:%M')
    # Get the wind speed from the second cell
    wind_speed = float(cells[1].text.split()[0])
    # Calculate the power of the wind generator
power = 0.5*1.225*10** 2*0.35*(wind_speed**3)
    # Add the data to the list
    data.append([date_time, wind_speed, power])
# Create a table as a dictionary
table = {'Date and time': [], 'Wind speed (m/s)': [],
'Power (W)': []}
# Add the data to the table
for row in data:
    table['Date and time'].append(row[0])
    table['Wind speed (m/s)'].append(row[1])
    table['Power (W)'].append(row[2])
# Add the current value and the forecast value of the
wind speed
table['Current wind speed (m/s)'] = table['Wind speed
(m/s)'][-1]
table['Forecast wind speed (m/s)'] = table['Wind speed
(m/s)'][-1]
# Print the table
print(table)
```

Fig. 1. Python source code for the output power of the wind turbine

Time	Wind Speed (m/s)	Predicted Wind Speed (m/s)	Power Output (W)
00:00	4.6	4.2	542.28
01:00	4.8	4.8	1074.24
02:00	5.3	5.4	2145.30
03:00	4.2	4.2	326.59
04:00	4.8	4.8	1074.24
05:00	5.5	5.4	1662.02
06:00	4.2	4.2	326.59
07:00	4.8	4.8	1074.24
08:00	5.4	5.4	1662.02
09:00	6.3	6.3	2569.70

Fig. 2. The result of the program execution

Table 2. Forecast and actual values

Hour	Date	Actual wind speed (m/s)	Forecast wind speed (m/s)	Power output (W)
1	23/05/01	4.6	4.2	542.28
2	23/05/01	4.8	4.8	1074.24
3	23/05/01	5.3	5.4	2145.30
4	23/05/01	4.2	4.2	326.59
5	23/05/01	4.8	4.8	1074.24
6	23/05/01	5.5	5.4	1662.02
7	23/05/01	4.2	4.2	326.59
8	23/05/01	4.8	4.8	1074.24

The forecast errors for the powers were calculated based on the wind speed data on site and according to the data after the calculation using the exponential smoothing method. The calculation results are given in Table 4. An exponential smoothing method was also applied to process the data received on site. This method allows you to get more accurate forecast data.

Table 4. Comparison of forecast errors obtained on site and data after processing

Power by actual wind speed (W)	Power by site wind speed (W)	Wind speed power after calculation (W)	MAE before calculations (%)	MAE after calculation (%)
208663.8356	158825.0781	208663.8356	9.5238	0
237081.6	237081.6	237081.6	0	0
326036.5063	337562.3781	326036.5063	1.8519	0.3704
158825.0781	158825.0781	158825.0781	0	0
237081.6	237081.6	237081.6	0	0
356665.9775	337562.3781	357030.8438	1.8519	0.8475
158825.0781	158825.0781	158825.0781	0	0
237081.6	237081.6	237081.6	0	0
337562.3781	337562.3781	337562.3781	0	0
540282.0238	540282.0238	540282.0238	0	0

CONCLUSIONS

Python program was written that allows you to receive data on the wind speed forecast. The MAE (Mean Absolute Error) method was used to estimate the forecast error. Forecast errors for output power are calculated based on wind speed data from the site and after processing using the exponential smoothing method. A comparison of the forecast errors before and after data processing shows that the exponential smoothing method makes it possible to obtain more accurate forecast data on wind turbine power. The average error on the site is 5.11%, and the average error after calculation is 4.57%. This suggests that after applying the exponential smoothing method, the forecast accuracy increased, and the average error decreased by 0.54%

ACKNOWLEDGMENT

This study is financed by the European Union-NextGenerationEU, through the National Recovery and Resilience Plan of the Republic of Bulgaria, Project BG-RRP-2.013-0001-C01.