**IECDT**

**Renewable Energy Laboratory**

December 2024

The purpose of this course is to learn and practice wind farm power output predictions using both a theoretical method and a simple machine learning method. The machine learning part of this workshop aims to forecast wind power generation based on historical measurements and additional wind information. The key objectives are to understand the dataset, analyse and preprocess data, apply feature engineering, train and fine-tune the model, and evaluate its performance compared to the theoretical model based on the operational curve.

## Learning Outcomes

After attending the course and performing the required work you should:

1. Understand the performance and operational curves for wind turbines, and how power is generated.
2. Have a general appreciation of low-order engineering model to predict wind turbine and farm power output.
3. Have a broad understanding of how machine learning models can be applied to solve a general engineering problem.
4. Understand the approaches to apply machine learning in wind farm power forecasting.

**References**

[1] Burton et al., "Wind Energy Handbook", 2nd Ed., Sec. 2.3-2.4 (Wind Resource), 3.5 (Theory) and A3.2 / A3.8 (Lift and Drag).

[2] Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., & Hyndman, R. J. (2016). Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. *International Journal of forecasting*, *32*(3), 896-913.

[3] https://github.com/Mo-Saif/Wind-Power-Forecasting

[4] Bengio, Y., Goodfellow, I., & Courville, A. (2017). *Deep learning* (Vol. 1). Cambridge, MA, USA: MIT press.

# 1. Theoretical Background

A demonstrator will check your preparation. **Section 1 must be completed before your workshop session, but also read through the remainder of the handout beforehand so that you understand the course. In particular take note of section 2.2 which discusses how to use the Excel spreadsheet and the data that you will need to use to calculate the wind turbine performance using the expressions developed in section 1.**

## 1.1 Wind turbine characteristics

A wind turbine is a device that transforms the kinetic energy of wind into electrical energy. It consists of several essential components: blades, a rotor, a tower, a generator, and a control system as Fig. 1 shows. The blades are designed to harness the wind's energy, causing the rotor to spin. As the rotor turns, it drives the generator, which converts mechanical energy into electrical power. The control system ensures the turbine operates efficiently by adjusting the blade pitch and orientation to capture the most wind energy.

Wind turbines are generally categorized into two main types: horizontal-axis wind turbines (HAWTs) and vertical-axis wind turbines (VAWTs). HAWTs are the most common, with blades mounted on a horizontal axis that rotates with the wind. VAWTs have a vertical axis of rotation and are more suitable for urban or turbulent wind environments.

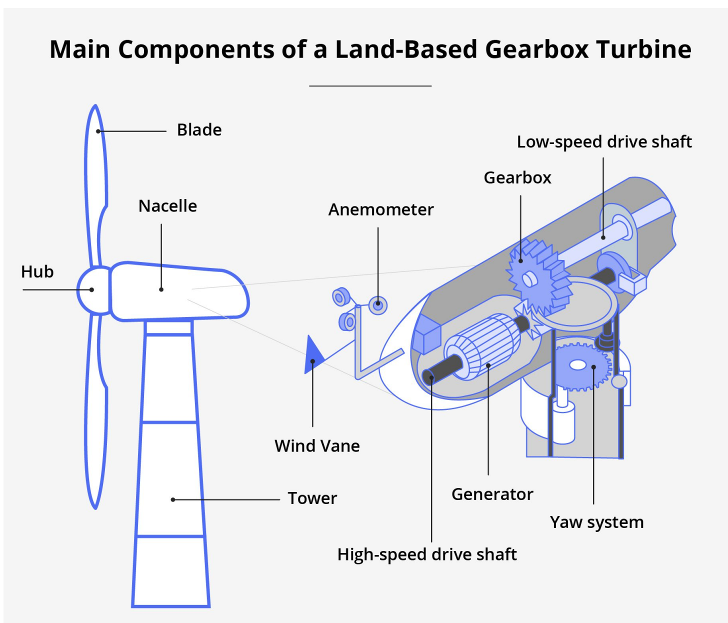



Fig. 1 Wind turbine components

The performance of a wind turbine depends on several factors, including wind speed, blade design, and the height of the tower. The optimal wind speed for generating power is typically between 10 and 15 meters per second. Higher towers capture stronger, more consistent winds, increasing energy production.

Wind turbines are designed to operate at maximum efficiency under certain working conditions. The thrust and power coefficient curves are essential for understanding its performance under varying operational conditions, particularly in relation to the rotational speed (or tip speed ratio, TSR). The TSR is the ratio of the tangential speed of the rotor blade tips to the wind speed, which determines the angle of attack for the flow passing the local airfoil, thus determining the local aerodynamic forces and efficiency of energy capture. Fig. 2 is an example of the thrust and power coefficient curves ($C_T$ and $C_P$) for a virtual wind turbine (based on the DTU 10MW reference wind turbine).

The power coefficient (Cp) curve illustrates the efficiency of a turbine in converting the wind's kinetic energy into electrical power. As TSR increases, Cp rises, reaching a maximum value at an optimal TSR, typically around 6 to 8 for horizontal-axis wind turbines. After this point, Cp starts to decrease, indicating that the turbine blades are moving too fast to efficiently capture the energy from the wind.

The thrust coefficient (Ct) curve represents the aerodynamic force exerted on the turbine blades, which is related to the wind's pressure and speed. At lower TSRs, the thrust coefficient increases rapidly, reaching a peak before declining as TSR increases. This reflects the turbine's interaction with the wind, where higher TSRs can reduce the load on the blades, but also decrease power capture.

Both curves are important for designing and optimizing wind turbines, as they guide the selection of operating TSRs that maximize efficiency and minimize mechanical stresses.

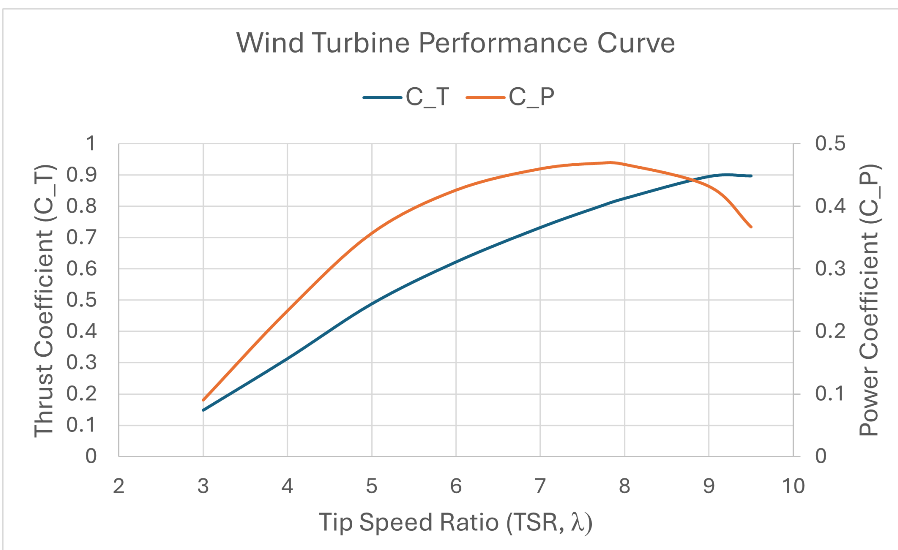



Fig. 2 Performance curve for a virtual wind turbine.

## 1.2 Wind turbine operating strategy

To maximize the energy production, wind turbines are expected to operate under the TSR that it has been designed for, when the wind speed is sufficiently high for operation; and normally with pitched blades at wind speeds above the rated wind speed to shed loads off from the blade. Fig. 3 presents an example of a wind turbine operation curve, which is reflected by power against wind speed. The lowest wind speed that a wind turbine can operate at is referred to as the cut-in speed, and the rotation speed of the turbine increases with the wind speed so that the turbine is operating at or close to maximum power coefficient (optimum TSR). The loads on the blades are increasing with wind speed and rotation speed. When the loads on the blades reach the design limit, the blades are pitched to reduce the loads. The wind speed before the blades start to pitch is called the rated wind speed. When the wind speed is greater than the rated speed, the turbine is working at a constant power (rated power) and reduced efficiency. Once the wind speed moves outside the the designed speed range, the turbine stops working and the blades are locked mechanically. The maximum wind speed before the wind turbine top working is referred to as the cut-out speed.

The control strategy of a virtual wind turbine (based on the DTU 10MW RWT) rated at 10MW is presented in Fig. 4, the cut-in and cut-out speeds are 4 m/s and 25 m/s, with rated speed of about 11 m/s.

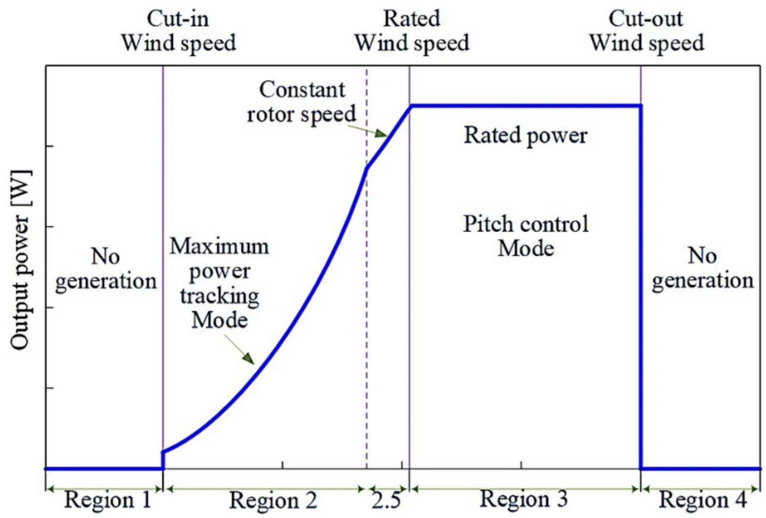



Fig. 3 Example of a wind turbine operation curve (Elkodama et al. 2023, reused under cc by 4.0).

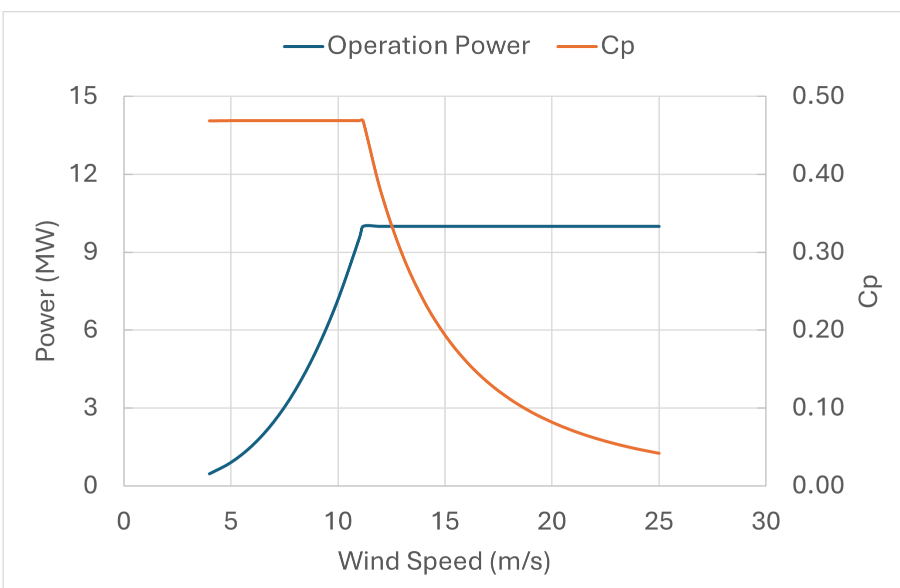



Fig. 4 Operation curve for a virtual wind turbine.

# 2. Tool and Database Guidance

**1. Understand the Spreadsheet (Wind_Turbine_Calculator)**

1.1 Wind turbine design

The DTU 10MW reference wind turbine (RWT) is selected as the demonstrative wind turbine for the calculation of performance and energy yield. The FFA-W3 airfoil family was selected by DTU for the 10MW RWT blade design, as it is frequently used by the Mega Watt wind turbine designs. To challenge the aerodynamics and increase stiffness, DTU decided to use a minimum relative thickness of $24.1\%$ at the tip and increase the relative thickness when moving towards the hub, up to a relative thickness of $60\%$ and then transition into a cylinder at $9.2\%$ of blade span.

The chord ($c$) and blade geometry twist angle ($\beta$) are pre-determined by the DTU team and we are not going to change it in this course. The DTU 10MW RWT blade consists of 5 different aerofoil sections with different thickness-to-chord ratios. Hence, two-dimensional lift and drag coefficients for each of the aerofoil sections are to be obtained so that the blade can be simulated in reduced order models following the blade element theory.

**The blade design is presented in the BLD tab of the provided spreadsheet. The tab is protected and should not be modified.**

1.2 Airfoil data

To simplify the modelling process in this course, we pick the lift and drag table for the FFA-W3-241 airfoil, which has a $24.1\%$ thickness-to-chord ratio and is used from $67\%$ of blade span onwards, provided by the DTU 10MW RWT project repository. The lift and drag coefficients are created by EllipSys2D simulations on the 2D airfoil at local Reynolds number of 12 million for angles of attack between $-32°$ and $32°$, while other angles are estimated using simple formulars.

The lift and drag coefficients for the FFA-W3-241 airfoil provided by DTU are presented in the tab named **ARFL**, with visualised lift and drag coefficients in the interested angle of attack range. Those lift and drag coefficients are used to predict the performance of the wind turbine.

**The airfoil aerodynamic performance is presented in the ARFL tab of the provided spreadsheet. The tab is protected and should not be modified.**

1.3 Wind turbine performance simulation

The tab **Performance** is implemented with a BEM solver following Ning's approach. The majority of the cells are locked to protect the simulation. The cells **in orange** (Solver Parameters) are the variables and equation to solve. When clicking the "**Reset**" button, the flow angle $\phi$ will be set to an initial guess of $\pi/9$, which is a reasonable starting point for iterations. When the "**Solve**" button is clicked, the flow angle $\phi$ will be solved to make $f(\phi)$ as close as possible to $0$. The cells **in yellow** are modifiable, to calculate turbine performance at different operating conditions.

The results of the turbine performance are presented in blue and green cells under "BEM Results" and "Force Distributions" block. The blade local force and coefficient distributions are visualized in the plots above.

The tab "**TipLoss**" is applying the Glauert tip loss correction to the BEM solver. This tab is **locked** and should not be modified unless given permission.

1.4 Wind turbine operation curve

The tab “**OpCurve**” presents the operation curves of the wind turbine, which is later used in the wind farm performance evaluation. The tab “**OpCurve**” will be updated following the simulation practice of applying different operation conditions (cells in **brown colour**) to the tab “**Performance**” and record the results in the column “**Power**”.

## 2. Understanding the wind farm power dataset for the machine learning part

The dataset analysed in this notebook comes from an energy forecasting competition (see code cells 2 and 3). The focus is on forecasting the hourly power generation at a particular zone, corresponding to a single wind turbine in Australia.

The data is available for periods ranging from the 1st hour of 01/01/2012 at 1:00 to 01/01/2014 at 00:00. The training period starts from 01/01/2012 at 1:00 to 01/12/2013 at 00:00, containing 16800 hourly records. The remainder of the datasets is used for evaluation and aimed at estimating real operational conditions, which start from 01/12/2013 at 01:00 to 01/01/2014 at 00:00, containing 744 hourly for each zone.

The input wind data include wind speed forecasts at two heights—10 meters and 100 meters above ground level—sourced from the European Centre for Medium-Range Weather Forecasts (ECMWF). These wind data are provided as eastward and northward wind vectors, which are represented by the variables ‘U’ and ‘V’ respectively. The U wind component is parallel to the x-axis (i.e. longitude). A positive u wind comes from the west, and a negative u wind comes from the east. The V wind component is parallel to the y-axis (i.e. latitude). A positive V wind comes from the south. These wind components will be later converted into wind speed and directions as additional input for the training process.

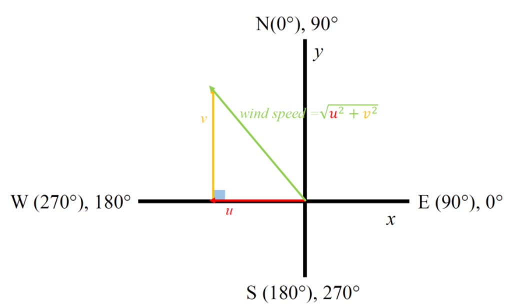



*Figure 1 Wind components: eastward (U) and northward (V) wind vectors*

Hourly power measurements from various wind farms were provided and standardized between 0 and 1 by normalizing each measurement with the nominal capacity of its respective wind farm, expressed as a value between 0 (no output) and 1 (maximum capacity).

# 3. Step-by-Step Guidance

### 1. Wind Turbine Performance Modelling

1.1. Performance and operational curve of a single wind turbine

1. **First download the spreadsheet from the platform.** Read this guidance and make sure you fully understand it before you change anything.
2. Go to the "**Performance**" tab and check the yellow cells are set with original values. Click the "**Reset**" button to reset the flow angle for the BEM solver.
3. Find out the optimal power coefficient for the provided wind turbine by setting the wind speed below rated (<11m/s). This power coefficient will be held constant when the wind speed is below rated.
4. Find out the exact wind speed that delivers 10MW power output at optimal Cp, this is the rated wind speed point on the operation curve. The pitch angle will begin to increase when wind speed is greater than this speed.
5. In the tab "**OpCurve**", observe the wind speeds from cut-in to cut-out, TSR, rotational speed and the pitch angle setup.
6. Take the working conditions at each wind speed from the tab "**OpCurve**" (**Wind speed, TSR, Pitch Angle**) to the tab "**Performance**" (fill in the **Yellow Cells**). Start calculating the power output by first clicking the button "**Reset**" and then clicking the button "**Solve**". Take the value in the cell "**Power**" from the region "**BEM Results**", and fill the yellow column "**Power**" in the tab "**OpCurve**". Repeat this step and fill all the cells in the "**Power**" column of the tab "**OpCurve**".

1.2. Wind farm power output prediction (will be practiced in the later section 2.5)

7. Take the operational curve generated by Steps 1-6 (wind speed and power) and upload it to the code.
8. Write a code to use the wind speed history provided and the generated operational curve to predict wind farm power output. Assume the wind farm has a total of 400MW (40 wind turbines in total).

### 2. Wind Farm Power Prediction Using Machine Learning Approach

#### 2.1. Understanding preprocessing and data analysis code

Due to time constrain, we have simplified the exercise and provided a sample code for each step.

1. Importing required packages
2. Load the target power generation dataset and convert the timestamp column to datetime format (Table shows the first 5 sets of the power generation dataset).

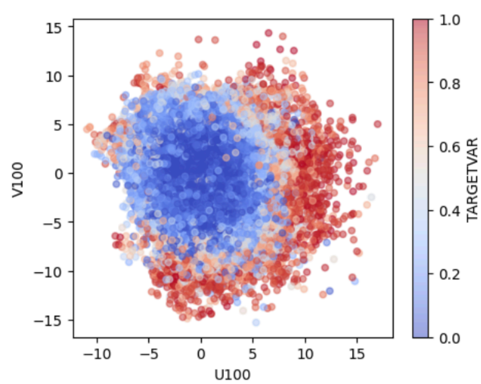
| | ZONEID | TIMESTAMP | TARGETVAR |
|---|---|---|---|
| 0 | 1 | 2013-12-01 01:00:00 | 0.844469 |
| 1 | 1 | 2013-12-01 02:00:00 | 0.795038 |
| 2 | 1 | 2013-12-01 03:00:00 | 0.809792 |
| 3 | 1 | 2013-12-01 04:00:00 | 0.550418 |
| 4 | 1 | 2013-12-01 05:00:00 | 0.496476 |

3. Load the training and test data. The timestamp column also needs to be converted. Note that the test data URL does not include the target value, hence, we need to merge "test_target" from step 2 to the test datatest (Table shows the first 5 sets of the train data for zone 1).

| | ZONEID | TIMESTAMP | TARGETVAR | U10 | V10 | U100 | V100 |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2012-01-01 01:00:00 | 0.000000 | 2.124600 | -2.681966 | 2.864280 | -3.666076 |
| 1 | 1 | 2012-01-01 02:00:00 | 0.054879 | 2.521695 | -1.796960 | 3.344859 | -2.464761 |
| 2 | 1 | 2012-01-01 03:00:00 | 0.110234 | 2.672210 | -0.822516 | 3.508448 | -1.214093 |
| 3 | 1 | 2012-01-01 04:00:00 | 0.165116 | 2.457504 | -0.143642 | 3.215233 | -0.355546 |
| 4 | 1 | 2012-01-01 05:00:00 | 0.156940 | 2.245898 | 0.389576 | 2.957678 | 0.332701 |

- **ZONEID:** Zone ID for each wind farm, ranging from 1 to 10. For this workshop, we only look at Zone1.
- **TIMESTAMP**: Date and time for each recorded observation.
- **TARGETVAR**: The normalized power output of the wind farm during the given hour, ranging from 0 and 1.
- **U10**: East-west component of the forecasted wind vector at 10 meters above ground level. Positive values indicate wind blowing eastward.
- **V10**: North-south component of the forecasted wind vector at 10 meters above ground level. Positive values indicate wind blowing northward.
- **U100**: East-west component of the forecasted wind vector at 100 meters above ground level. Positive values indicate wind blowing eastward.
- **V100**: North-south component of the forecasted wind vector at 100 meters above ground level. Positive values indicate wind blowing northward.

4. Visualize dataset – The relationship between the wind forecasts at 100 meters and the power output.



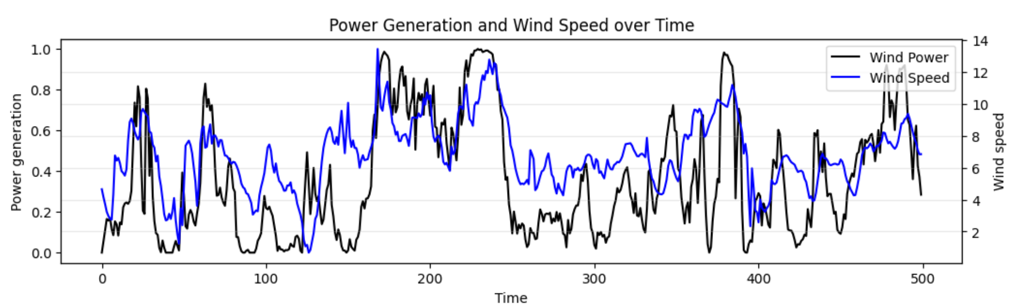

The west-east wind speed is plotted on the horizontal axis and the south-north wind speed is plotted on the vertical axis. The further a point from the center (0,0) the stronger the wind. The position of the point relative to the center reflects the wind direction. The colorbar corresponds to the power output where blue means no output and red means maximum output.

We can see that the wind is observed to be stronger towards the east and south. The power output varies even at the same wind speed and direction across different times. Additionally, there are instances of relatively high wind speed but low or no power generation. It should be noted that these inconsistencies can impact the predictive performance of the models.

5. To further explore the data, we will generate more features that can be later used to improve our model. From wind speed components U and V at 10m and 100m, we can generate its corresponding wind speed and directions. To give the model more information about the wind profile, we can also add the average of those speeds as an additional feature (Table shows the first 5 sets of the train data after adding the features).

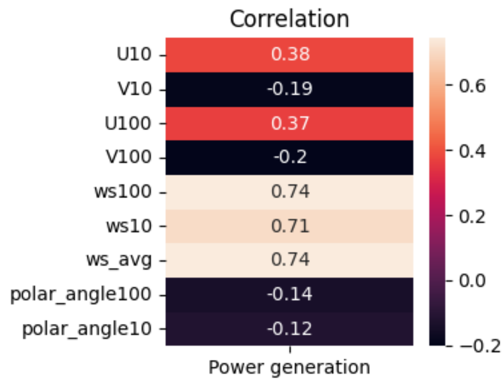
| | ZONEID | TIMESTAMP | TARGETVAR | U10 | V10 | U100 | V100 | ws100 | ws10 | ws_avg | polar_angle100 | polar_angle10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2012-01-01 01:00:00 | 0.000000 | 2.124600 | -2.681966 | 2.864280 | -3.666076 | 4.652334 | 3.421530 | 4.036932 | -51.999735 | -51.614439 |
| 1 | 1 | 2012-01-01 02:00:00 | 0.054879 | 2.521695 | -1.796960 | 3.344859 | -2.464761 | 4.154892 | 3.096451 | 3.625672 | -36.385781 | -35.473680 |
| 2 | 1 | 2012-01-01 03:00:00 | 0.110234 | 2.672210 | -0.822516 | 3.508448 | -1.214093 | 3.712577 | 2.795932 | 3.254255 | -19.088098 | -17.108562 |
| 3 | 1 | 2012-01-01 04:00:00 | 0.165116 | 2.457504 | -0.143642 | 3.215233 | -0.355546 | 3.234831 | 2.461699 | 2.848265 | -6.310236 | -3.345160 |
| 4 | 1 | 2012-01-01 05:00:00 | 0.156940 | 2.245898 | 0.389576 | 2.957678 | 0.332701 | 2.976332 | 2.279435 | 2.627884 | 6.418062 | 9.840676 |

6. Plot the power generation and wind speed at 100m over time




7. Plot the heatmap to check the correlation between input features and target power generation for different zones.

It can be seen that there is a stronger correlation between U components (towards the East direction) than V components (towards the south direction) for most cases due to the wind flowing to the east more than the west. The wind speeds are linearly correlated with the power output, and the wind speed at 100m is more correlated than 10m with the output. The more linearly correlated features are to the output, the easier it is to predict the output.

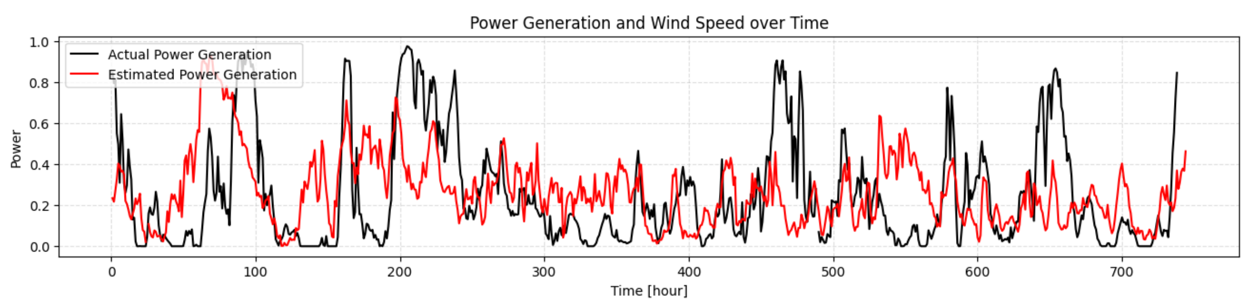



## 2.2. Machine learning Modelling

1. Benchmark

To evaluate prediction accuracy, it is helpful to set a benchmark model. Here, the benchmark predicts the current month's value as the average of the same month last year and the previous month.

This simple approach accounts for seasonality and yearly variations. A straightforward benchmark like this is often enough for comparison. The error is measured using root mean square error (RMSE).




2. Multi-Layer Perceptron (MLP) model**.**

MLP is a basic type of neural network commonly used for supervised learning tasks, such as time series prediction. It serves as a universal approximation function that maps a set of input values x (e.g., wind speed, direction, etc) to output values y (wind power generation). The goal of the training process is to identify an approximation function f by optimising two sets of parameters: the weights w and the bias terms b, namely y = f(x ; w,b). A basic ANN consists of an input layer, one hidden layer, and an output layer. The input layer passed the input data to the hidden layer. Each layer comprises several neurons, and each neuron in the hidden layer is connected to every neuron in the subsequent layer. Neurons compute their values by applying a nonlinear activation function to the weighted sum of their inputs with a bias term. For more detailed descriptions of the MLP model architecture, see reference [4].

1) In this notebook, we will use a single validation subset instead of cross-validation to reduce computational cost. The training data will be split into a training subset and a validation subset, with the validation subset comprising the most recent 10% of the data (approximately two months). Table below shows random selected 5 set in training data

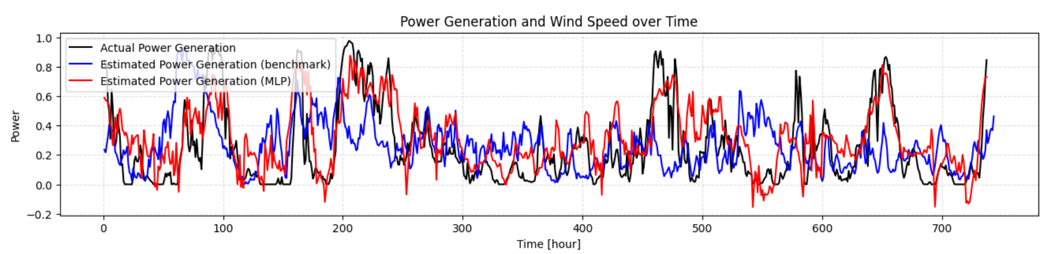
| | ZONEID | TIMESTAMP | TARGETVAR | U10 | V10 | U100 | V100 | ws100 | ws10 | ws_avg | polar_angle100 | polar_angle10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 11432 | 1 | 2013-04-21 09:00:00 | 0.658961 | -2.699554 | -5.952542 | -4.390311 | -10.176911 | 11.083517 | 6.536081 | 8.809799 | -113.335329 | -114.394899 |
| 3631 | 1 | 2012-05-31 08:00:00 | 0.000000 | 1.236654 | -1.919330 | 1.621899 | -2.227717 | 2.755591 | 2.283230 | 2.519411 | -53.943343 | -57.205699 |
| 6995 | 1 | 2012-10-18 12:00:00 | 0.018977 | -1.149674 | -2.009281 | -2.324246 | -5.992266 | 6.427237 | 2.314943 | 4.371090 | -111.199995 | -119.777404 |
| 8609 | 1 | 2012-12-24 18:00:00 | 0.096983 | 0.559041 | 2.601521 | 0.917978 | 4.854753 | 4.940780 | 2.660910 | 3.800845 | 79.292448 | 77.872135 |
| 10834 | 1 | 2013-03-27 11:00:00 | 0.539235 | 3.925016 | 0.726833 | 6.094070 | 1.199461 | 6.210990 | 3.991746 | 5.101368 | 11.134867 | 10.491168 |

The model's architecture, including the number of layers, neurons, and other hyperparameters, will be determined based on its performance on this validation set. We will use this validation error to help us select the features and fine tune the model; after that, we will evaluate the model on the test set and compare it to the theoretical model and the benchmark.

2) Build a 2-layer NN model with 64 neurons in each layer.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| dense (Dense) | (None, 64) | 704 |
| dense_1 (Dense) | (None, 64) | 4,160 |
| dense_2 (Dense) | (None, 1) | 65 |

3) Train and check the model performance on test data. You should see that this model already shows a significant improvement over the benchmark score. This improvement was achieved without any feature selection or fine-tuning. Next, we will perform these steps to explore how much more we can enhance the model's performance.




### 2.3. Feature engineering and selection

At this stage, we have already added some features in section 2 and gained insights from our analysis to identify which features contribute to improving the predictive power. Since our dataset is small and the computation time is efficient. We can just add or remove features to check if they improve performance.

1. Drop ZONEID as an input feature as we only use 1 zone in this workshop, and it is expected that it does not contribute to the predictive power.

2. Experiment with adding or removing different features to observe their effect on the RMSE for the validation set.

    Note that the RMSE may vary between runs due to the stochastic nature of the MLP model. The influence of individual features may not be immediately apparent, as it is affected by the small dataset and the simplicity of the model. This is just to showcase the process of feature selection and its effect on model performance.

3. In theory, incorporating the power output from the past few hours, along with forecasted changes in wind speed for the preceding and upcoming hours, could improve prediction accuracy. However, in practical applications, wind power forecasting is typically performed 24 to 48 hours ahead of real-time, making it impractical to use recent power output as an input. Instead, forecasted wind speeds from both previous and upcoming hours can be included as additional features to enhance the model's predictive performance.

    We can also try to include the past wind speeds as the input to the model. Here, we can check how many hours should we add.

    Before testing the effect on the past values, we can build a pipeline to streamline the process of evaluating different features and tuning hyperparameters. This will make it easier to compare performance across various configurations efficiently.

```
RMSE for  0 speed lag is 0.20462382177103075
RMSE for  1 speed lag is 0.21508062714977272
RMSE for  2 speed lag is 0.18424192416427784
RMSE for  3 speed lag is 0.18450162602890907
RMSE for  4 speed lag is 0.21586619450053662
RMSE for  5 speed lag is 0.19583267155889236
RMSE for  6 speed lag is 0.1994835140544181
RMSE for  7 speed lag is 0.18384216207329415
RMSE for  8 speed lag is 0.20052224974621805
RMSE for  9 speed lag is 0.18999280810369298
RMSE for 10 speed lag is 0.18387723896603322
The optimal number of speed lags to add is 7 and the corresponding RMSE is 0.1838
```

### 2.4. Fine-tuning the model

Now that we have identified the features to use, we will proceed to hyperparameter tuning. Instead of using scikit-learn `GridSearchCV`, which relies on cross-validation, we will use a simpler approach without cross-validation.

1. To proceed to hyperparameter tuning, we need to modify the **run_nn_pipeline** function to include the number of layers in the function.

2. Experiment with different hyperparameters, such as the number of neurons, layers, activation function, optimizer, batchsize, etc.

3. After optimizing the model, evaluate its performance on the test dataset.

### 2.5. Compare ML model prediction with the theoretical model

Using the provided operational curve (data provided in the code), interpolate the curve to estimate the power output for the wind speeds in the test data. Assume the wind farm has an operational capacity of 400 MW, based on 40 wind turbines. Scale both the theoretical predictions from the curve and the ML model predictions to 400 MW for comparison.

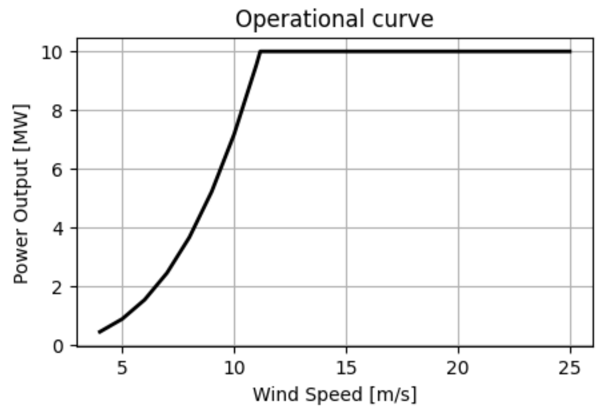



*Figure 2 The operational curve of wind turbine.*

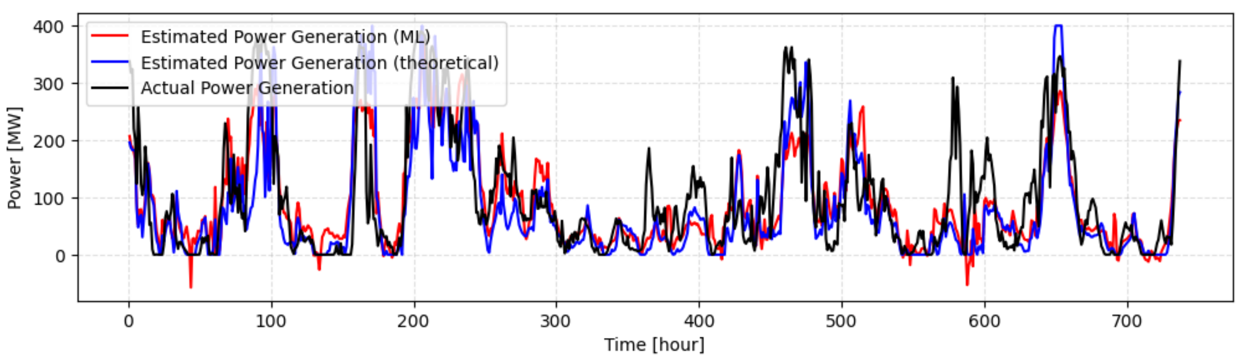



*Figure 3 Comparison between the ML and the theoretical model with RMSE of 0.1613 and 0.1825, respectively.*

### 2.6. Further exploration

From the above analysis, you may observe that tuning parameters or adding additional features has only a minor impact on prediction performance. Additionally, the predicted power output occasionally drops below zero, which is unrealistic. These limitations primarily come from the small size of the training dataset provided and the use of a very basic neural network model. Given the time constraints of this workshop, the primary goal is to provide an overview of preprocessing, feature engineering, and fine-tuning rather than achieving minimal prediction error. However, there are several ways to improve the model's performance, which can be explored further. Here are some ideas:

- **Expand the Training Dataset:** The current dataset is limited, but more data can be accessed by changing the file name from Train_W_Zone1.csv to "Train_W_Zone2.csv", up to Zone10. The same applies to the test set. Including additional data will enhance the ML model's performance and help eliminate issues like negative power predictions

- **Include Additional Input Features:** Features such as changes in wind speed and forecasted wind speeds, which are available a few hours before real-time from sources like ECMWF, can provide valuable information to improve model predictions.

- **Explore Advanced Modeling Approaches:** Consider more sophisticated models like Long Short-Term Memory (LSTM) networks, which are specifically designed for time-series predictions. Additionally, as low wind speeds are generally more frequent than high wind speeds, separate models could be developed to predict power output under low and high wind speed conditions. Or a mix of different models can be applied and combined to improve overall accuracy.

## 3. Discussion

Discuss with a demonstrator the change in wind speed history in different regions and how it relates to the changes in wind turbine and farm power output. Discuss your predicted wind farm power output curves and discuss the differences between the theoretical method, machine learning prediction and real-world history.