Renewable Energy Systems
Wind Turbine Power Output Prediction

Wilmer H. Jacobsson & João Domingos

1. Introduction & Motivation

1.1 Problem Statement

Weather is a chaotic system that changes from day to day. Wind, being a part of this system, is also inherently chaotic. This makes the wind, as a natural source of power, highly inconsistent and hard to predict. Due to this, wind turbine's energy output is challenging to integrate into the power grid [1]. This inconsistency also creates a potential issue of overproduction of energy at times of low demand [2].
A possible solution to these problems is to use a machine learning algorithm that helps to predict the power output of wind turbines, so it is easier to integrate into the grid and plan for the right amount of production needed.

1.2 Motivation

Climate change has affected the modern world with more frequent natural disasters like floods, hurricanes and wildfires threatening the lives and living places of millions of people. The current global economy relies on burning fossil fuels for energy production, which emits huge amounts of CO2, contributing to the intensification of climate change. The transition from fossil fuels to renewable energies is evident and urgent. Wind is a perennial source of power that, although inconsistent, can never be depleted. Therefore wind power is a promising alternative of clean energy. Wind turbines can harness this power and have the potential to play a big role into a sustainable, renewable energy future. [3]

1.3 Dataset & Method

In order to predict energy output, two supervised machine learning algorithms were used. Supervised, because the data is already labeled, not choosing a deep learning algorithm because of the features already being quite few.
The most commonly picked algorithms for wind power prediction used in relevant papers are: the linear regression and random forest regression [5]. These were trained on the mentioned dataset and performance tested against each other.

2. Methods & Implementation

2.1 Data Acquisition

The dataset used was "Wind Turbine SCADA Dataset" [4] and was acquired, as title suggests, using Supervisory Control and Data Acquisition (SCADA). The dataset was uploaded by Berk Erisen to Kaggle in 2018 and the uploader claims it was collected in Turkey over a one year span of that same year. The dataset contains 5 features:

- Date/Time in 10 minute intervals,
- LV ActivePower(kW): Power generated by turbine for that interval,
- Wind Speed (m/s),
- Theoretical_Power_Curve (kW): Theoretical power that is generated by the turbine,
- Wind Direction (°): Wind turbines turn to this direction automatically.

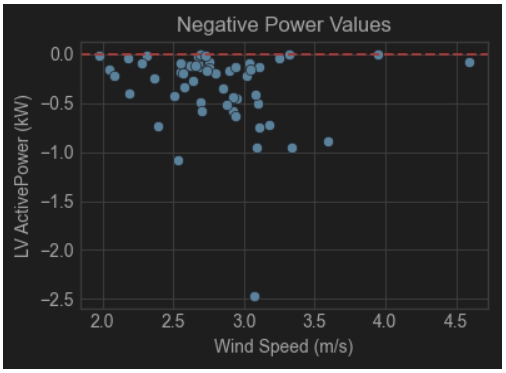
2.2 Preprocessing

To generate reliable and relevant output it is necessary that the input data is of high-quality and tailored to the models that were selected. Thus, a preprocessing pipeline was developed to transform the raw data into an optimal format to fit both Linear Regression and Random Forest Regression. This includes data cleaning steps such as handling missing values and outliers, and other model specific steps like encoding, feature engineering like extraction, scaling and transformation, and data splitting according to the time series nature of the data to increase the predictive capacity of the models.

2.2.1 Outlier handling

There were no missing values on the dataset but there were some negative values on the active power. Since we are predicting the power generation output, negative values in Active Power act as noise for the model and need to be handled. There are two scenarios for these values that needed further investigation: If the wind speed is low when the active power is negative then the most probable scenario was that the blades were not spinning and the turbine was consuming electricity. The other possibility is that if the wind speed is high then the blades are spinning and is probably a sensor malfunction.

Depending on which scenario the handling of the outliers would be different, so then the Wind Speed and Active Power variables were plotted.



We can observe that most of the negative values occur below the 3.5 m/s wind speed, we conclude that it is most probably not sensor malfunction and that the turbine is truly consuming more energy than it generates. Since the usual threshold for the blades to start spinning and producing electricity is between 3 and 4 m/s [6].
In this case we set the negative power values to zero because for power production forecasting, a negative value means zero power is generated.

2.2.2 Feature Engineering

Feature Engineering is a critical step in the preprocessing phase, transforming raw data into features that represent the problem to the machine learning model. In this case we used feature extraction, encoding, scaling and feature transformation.

2.2.2.1 Extraction

In the feature extraction phase the original Date/Time feature was used to extract month, day of the month, hour, minutes, season and day/night features. To extract the seasons a dictionary was used to map the seasons according to the month. For the day/night feature, since the sunset and sunrise times shift throughout the year, for a simpler extraction and considering Turkey's latitude, we choose to define a static selection of day between 6am and 6pm and night on the remaining ones.

2.2.2.2 Encoding

Following the extraction process the season and night/day features were one hot encoded and binary encoded respectively. Seasons were one-hot encoded because they are categorical (Spring, Summer, Fall, Winter) and don't have a specific rank. The night/day feature was binary encoded because there are only two possible states, making it more efficient to represent as a single column (0 or 1) without increasing the dimensionality of the dataset.

2.2.2.3 Scaling

Since we are applying a linear regression model it is best practice to scale the variables to prevent the model from giving false importance to the larger numerical values. In this case, it was applied the standard scaler to the wind speed and theoretical power curve utilizing the following formula, where μ represents the mean of the feature and σ represents the standard deviation.

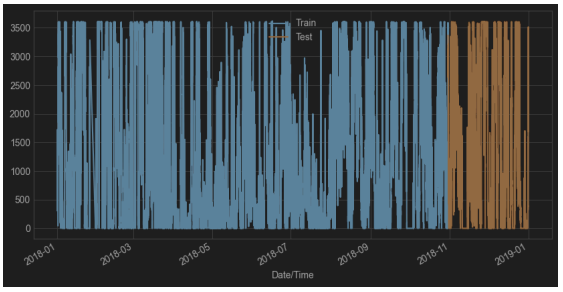
$$z = \frac{x - \mu}{\sigma}$$

2.2.2.4 Transformation

For the wind direction variable, because it is a directional and circular feature, it needed to be handled differently. Since standard numerical scales treat 0° and 360° as distant values, a linear model would fail to recognize that they actually represent a very similar wind direction.
To solve this issue, the wind direction was converted into two features: sine and cosine. This transformation maps the circular data onto a 2D coordinate system. The sine component captures the East-West orientation, and the cosine component captures the North-South orientation.
This ensures that values like 359° and 1° are treated as being next to each other, allowing the algorithm to process correctly the circular nature of wind direction.

2.2.4 Time Series Data Splitting

Since this is time series data, it is best practice to avoid random shuffling during the validation process. In traditional machine learning, data is often split randomly into training and testing sets, but this approach is not correct for temporal datasets. Random splitting would make the model fall in data leakage, meaning, in this case, it would have access to future information while predicting past events.
To maintain the order of the observations, a Time Series Split was utilized. In this method, the training set grows over time, always staying chronologically before the validation set. In the image below we can see the last cross validation split.



3. Experiments & Results

Cross-validation was used for a more representative measurement. To evaluate and compare the performance of the models, Root Mean Squared Error (RMSE) and the Coefficient of Determination (R squared) was utilized. RMSE was selected to measure the average magnitude of the prediction errors, specifically to penalize larger deviations. R squared was included to assess the proportion of variance in the target variable explained by the features, facilitating a comparison of

goodness-of-fit across both linear and non-linear approaches. From the table below it is possible to see that the Random Forest Regression model performed slightly better than the Linear Regression model.

Model	R^2	RMSE
Linear Regression	0.9065	367.1876
Random Forest Regression	0.9106	366.4178

4. Analysis & Discussion

The minor superior performance of the random forest model (R^2: 0.9106) over linear regression is physically consistent with wind power kinetics. Power generation exhibits a cubic dependence on wind speed up to the rated capacity, followed by a non-linear saturation plateau. Linear models inherently fail to capture these distinct operational regimes (cut-in, ramp-up, and rated power maintenance). The random forest algorithm, through decision-tree segmentation, effectively handles these non-linear thresholds. However, the results are very similar which could be because of the theoretical power curve having a larger impact on both models due to its linear relation and strong correlation to power generated. The observed RMSE (~370 kW) remains non-trivial. While acceptable for individual turbine monitoring, such variance could compound significantly if extrapolated across a wind farm, necessitating further error reduction for commercial grid dispatch.

One of the primary arguments against renewable energy is "intermittency." Grid operators often keep fossil-fuel-based "peaker plants" running in the background (spinning reserves) to compensate for sudden drops in wind power [7]. By implementing accurate forecasting models grid operators can reduce the required safety margin of fossil fuel backup. Uncertainty is also financial risk. If energy providers cannot guarantee output, they face penalties in the energy market [8]. High-accuracy machine learning models reduce this financial risk.

The model's generalizability is constrained by two primary factors. First, the absence of air temperature and pressure data prevents the model from accounting for air density fluctuations, which linearly scale kinetic energy transfer. Second, the single-year dataset captures only immediate seasonal cycles, failing to account for multi-year climatic oscillations, risking temporal overfitting. Ultimately, while this study confirms the efficacy of non-linear machine learning for power forecasting, integrating broader meteorological and temporal datasets remains the critical next step to transform wind from a variable resource into a reliable pillar of the global energy transition.

6. References

[1] 2023. V. S. Rao, P. S. Teja, N. Vamsi, P. V. S. R. Krishna. Prediction of Power Generation in Wind Turbines. IJSREM Journal.

[2] 2021. S. Preethi, H. Prithika, M. Pramila, S. Birundha. "Predicting the Wind Turbine Power Generation based on Weather Conditions," 5th International Conference on Electronics, Communication and Aerospace Technology. 10.1109/ICECA52323.2021.9676051.

[3] 2025. Braeschke, M. & Müller, S. Leveraging Shapley Values for Temperature Contributions to Power Deviations in Wind Turbines. https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1985686&dswid=-8466

[4] - Erisen, B. 2018. Kaggle - Wind Turbine Scada Dataset. https://www.kaggle.com/datasets/berkerisen/wind-turbine-scada-dataset/data

[5] Magesh, T., Samuel Franklin, F., Santhi, P. S., & Thiyagesan, M. (2026). Machine Learning-Driven Wind Energy Forecasting for Sustainable Development. https://doi.org/10.1051/matecconf/202439302003

[6] N. M. Nasab, J. Kilby, and L. Bakhtiaryfard, "Comparative Critique on Power Generation in Wind Turbines," *Chemical Engineering Transactions*, vol. 76, pp. 883-888, 2019. Available: https://www.aidic.it/cet/19/76/148.pdf

[7] 2012. G. Liu, & K. Tomsovic. Quantifying spinning reserve in systems with significant wind power penetration. IEEE Transactions on Power Systems.

[8] 2011. Ela, E., Milligan, M., & Kirby, B. Operating reserves and variable generation.