

Project Design Phase

Proposed Solution

Date	13 February 2026
Team ID	LTVIP2026TMIDS90282
Project Name	Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management
Maximum Marks	

Proposed Solution:

The proposed solution involves developing a machine learning-based system capable of predicting wind turbine energy output based on weather-related inputs. The selected algorithm for this purpose is the Random Forest Regressor, which belongs to the ensemble learning family.

Random Forest works by constructing multiple decision trees during the training phase. Each tree is built using a subset of the training data and a random subset of features. This randomness improves model diversity and reduces overfitting, leading to better overall performance.

Wind energy output does not always follow a simple linear pattern with respect to wind speed. Random Forest can handle complex, non-linear relationships effectively, making it well-suited for this application. It also performs well with moderate-sized datasets and requires minimal feature engineering.

Another reason for choosing Random Forest is its robustness to noise and outliers in the dataset. Weather data can contain irregularities or sudden fluctuations, and Random Forest is capable of maintaining stable predictions under such conditions.

During training, each decision tree learns patterns between input features such as wind speed and theoretical power values and the target output, which is the actual energy generated. After training, predictions from all individual trees are averaged to produce the final output value.

This ensemble averaging mechanism helps in minimizing errors that may occur in individual decision trees. As a result, the model provides more stable and accurate predictions compared to a single tree model. This characteristic is particularly useful in handling variable weather data.

The trained Random Forest model is then serialized and saved as a .sav file using Joblib. This file contains the learned patterns and can be reused without retraining the model every time the application runs.

In the deployment phase, the Flask web application loads the saved model and accepts user input through a web interface. The input data is passed to the model, which computes the predicted wind energy output instantly.

Overall, the proposed solution integrates data preprocessing, machine learning model training, model saving, and web deployment into a single system that provides real-time energy predictions based on weather conditions.