Weather-Based Prediction of Wind Turbine Energy Output A Next-Generation Approach to Renewable Energy Management (Project Report)

1. INTRODUCTION

1.1 Project Overview

Wind energy is one of the most important renewable energy sources for sustainable development. However, wind turbine energy output depends heavily on changing weather conditions such as wind speed, wind direction, temperature, humidity, and atmospheric pressure.

This project focuses on predicting wind turbine energy output using Machine Learning models based on weather data. By analyzing historical weather conditions and turbine power generation records, the system can forecast energy production more accurately.

The prediction system can help renewable energy companies optimize power generation, improve grid stability, and reduce energy wastage. A trained ML model is integrated with a web-based interface where users can input weather conditions and get predicted turbine energy output.

1.2 Purpose

The purpose of this project is:
- To predict wind turbine energy output based on weather parameters.
- To help renewable energy management with accurate forecasting.
- To reduce energy loss and improve power grid planning.
- To develop a real-time prediction system using Machine Learning.
- To provide a user-friendly interface for prediction.

2. IDEATION PHASE

2.1 Problem Statement

Wind turbine power output varies due to unpredictable weather conditions. Energy production forecasting is challenging, leading to inefficient energy distribution and management.

Problem:

To build a Machine Learning system that can accurately predict wind turbine energy output using weather-related input parameters.

2.2 Empathy Map Canvas

User: Renewable Energy Company / Grid Operator

- Thinks: Energy prediction should be accurate for better planning.
- Feels: Concerned about sudden fluctuations in power generation.
- Says: 'We need a system to forecast turbine output efficiently.'
- Does: Monitors turbine performance and manages energy distribution.
- Pain Points: Sudden weather changes reduce prediction accuracy.
- Gains: Accurate forecasting helps improve renewable energy utilization.

2.3 Brainstorming

Ideas discussed:
- Collect historical weather and wind turbine energy output data.
- Apply regression algorithms for power prediction.
- Use deep learning models like LSTM for time-series forecasting.
- Build a web interface for user input and prediction.
- Integrate ML model with Flask for real-time predictions.

Final chosen idea:
Develop a Machine Learning based wind turbine energy output prediction system integrated with a web application.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

1. User opens the wind energy prediction website.
2. User enters weather conditions such as wind speed, temperature, and humidity.
3. User clicks on 'Predict Output'.
4. Backend sends data to the ML model.
5. ML model predicts energy output.
6. Result is displayed on the UI.

3.2 Solution Requirement

Functional Requirements:
- User should input weather parameters.
- System should predict wind turbine energy output.
- Output should be displayed in numeric form (kW/MW).
- Model should load automatically for prediction.

Non-Functional Requirements:
- High accuracy and fast response time.
- Easy-to-use interface.
- Should handle multiple user inputs.
- Must work on local server environment.

3.3 Data Flow Diagram (DFD)

Level 0 DFD:
User → Web UI → Flask Server → ML Model → Predicted Output → User

Level 1 DFD:
1. User enters weather data.
2. Flask backend receives input.
3. Data preprocessing (scaling/cleaning).
4. Model predicts energy output.
5. Output displayed to user.

3.4 Technology Stack

Frontend:
- HTML
- CSS
- Bootstrap

Backend:
- Python Flask

Machine Learning:
- Pandas
- NumPy
- Scikit-learn
- Matplotlib

Model Saving:
power_prediction.sav

Development Tools:
- VS Code
- Jupyter Notebook

4. PROJECT DESIGN

4.1 Problem Solution Fit

Accurate wind turbine energy output prediction is essential for renewable energy management. Machine Learning models can learn patterns from historical data and provide accurate predictions based on real-time weather inputs. This improves power distribution planning and reduces energy wastage.

4.2 Proposed Solution

The proposed system is a Machine Learning-based prediction model integrated into a Flask web application.

- Weather and turbine output dataset is collected.
- Data is preprocessed and cleaned.
- ML model is trained using regression algorithms.
- Trained model is saved using Pickle.
- Flask loads the model and predicts output from user input.
- UI displays the predicted turbine energy output.

4.3 Solution Architecture

Architecture Flow:
1. User enters weather parameters in UI.
2. Flask backend receives input.
3. Input data is preprocessed.
4. Trained model predicts energy output.
5. Output is returned and displayed.

Components:
- User Interface (HTML form)
- Flask Server
- Preprocessing Module
- Machine Learning Model
- Output Display Module

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Phase 1: Data Collection
- Collect wind turbine dataset and weather data.
- Study dataset features.

Phase 2: Data Preprocessing
- Handle missing values.
- Normalize/scale features.
- Remove noise.

Phase 3: Model Training
- Split dataset into training and testing sets.
- Train regression model.
- Evaluate model accuracy.

Phase 4: Model Deployment
- Save trained model.
- Create Flask backend.
- Connect model with frontend.

Phase 5: Testing
- Test model with different weather inputs.
- Validate output accuracy.

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

The system was tested with different weather conditions and wind speed values.

Testing Results:
- Model predicts output quickly.
- UI provides real-time prediction.
- Prediction accuracy depends on dataset and model used.

Performance Metrics used:
- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- R2 Score

7. RESULTS

7.1 Output Screenshots

The output page displays:
- Input weather parameters
- Predicted wind turbine energy output (kW/MW)

Screenshots should include:
- Home page form
- Predicted output page
- Flask server running output

8. ADVANTAGES & DISADVANTAGES

Advantages

- Helps forecast renewable energy generation.
- Improves power grid planning.
- Reduces energy wastage.
- Real-time prediction system.
- Easy to use web interface.

Disadvantages

- Weather data accuracy affects prediction.
- Requires frequent retraining for better results.
- Sudden climate changes may reduce accuracy.
- Needs large dataset for best performance.

9. CONCLUSION

This project successfully implements a Weather-Based Wind Turbine Energy Output Prediction system using Machine Learning. The system predicts turbine energy generation based on weather inputs like wind speed, temperature, humidity, and pressure. Integration with a Flask web application provides a user-friendly interface for real-time predictions. This system helps improve renewable energy management and supports efficient power distribution.

10. FUTURE SCOPE

Future enhancements can include:
- Implement Deep Learning models such as LSTM for time-series forecasting.
- Integrate real-time weather API for live prediction.
- Deploy the project on cloud platforms.
- Add dashboard for turbine monitoring.
- Improve prediction accuracy using advanced feature engineering.
- Add database for storing historical predictions.

11. APPENDIX

Source Code:
- app.py
- train_model.py
- HTML files for UI

GitHub & Project Demo Link:shaikshakeera1705/Weather-Based-Prediction-of-Wind-Turbine-Energy-: Weather-Based Prediction of Wind Turbine Energy Output is an ML project that predicts wind turbine power generation using weather data like wind speed, temperature, humidity, and pressure. It helps in accurate energy forecasting, better grid management, and efficient renewable energy planning.