Wind Energy Generation Prediction Using Machine Learning

1. INTRODUCTION

1.1 Project Overview

This project, titled **“Weather-Based Prediction of Wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management,”** is developed as part of the SmartBridge Virtual Internship Program. The objective of this project is to predict wind turbine energy generation using weather and turbine operational parameters through machine learning techniques.

In the modern renewable energy sector, efficient energy forecasting is essential for grid stability, maintenance planning, and power distribution. Wind energy production is highly dependent on environmental conditions such as wind speed, temperature, and atmospheric pressure, making accurate prediction challenging. This project addresses this challenge by applying artificial intelligence to analyze historical weather and turbine data and forecast energy output.

The system accepts weather and turbine parameters as input, processes them using a trained machine learning model, and generates predicted wind energy output. The application is designed to be user-friendly, accurate, and reliable, demonstrating the practical use of artificial intelligence in renewable energy management and smart power systems.

1.2 Purpose

The main purpose of this project is to apply theoretical knowledge of artificial intelligence and machine learning to develop a practical solution for predicting wind turbine energy output based on weather and operational parameters. The project enables understanding of data preprocessing, feature engineering, machine learning model training, and integration of the prediction model into a functional application using Python and related technologies.

The project also provides hands-on experience in working with real-world renewable energy datasets and building an end-to-end predictive system. It enhances analytical thinking, problem-solving ability, and technical skills in machine learning, data analysis, and application development for smart energy management.

2. IDEATION PHASE

2.1 Problem Statement

In the modern renewable energy sector, efficient power generation and grid management have become increasingly important. However, wind energy production is highly dependent on changing weather conditions such as wind speed, temperature, and atmospheric pressure, which makes energy output uncertain and difficult to predict accurately.

Most existing forecasting methods rely on basic statistical models or manual estimation, which may be inaccurate, time-consuming, and not suitable for real-time decision-making. Inaccurate prediction of wind turbine energy output can lead to inefficient energy distribution, grid imbalance, and poor maintenance planning. Therefore, there is a strong need for a simple, accurate, and AI-based wind energy prediction system that can forecast turbine power generation using weather and operational data to support smart renewable energy management.

2.2 Empathy Map Canvas

The Empathy Map Canvas is used to understand the users, their challenges, concerns, and expectations in managing wind energy production. This helps in designing a user-centric renewable energy prediction solution.

What the User Thinks
• Wants to know how much energy the turbine will generate
• Expects accurate prediction for planning and decision-making

What the User Feels
• Uncertain when energy output varies due to weather
• Confident when prediction helps optimize operations

What the User Says
• "Wind power output is difficult to estimate"
• "I need reliable energy forecasts for planning"

What the User Does
• Monitors turbine and weather data manually
• Uses basic forecasting or historical averages

User Pain Points
• Unpredictable wind conditions affect power generation
• Inaccurate forecasting leads to grid imbalance and losses

User Needs
• Accurate AI-based wind energy prediction
• Simple and reliable monitoring application

2.3 Brainstorming

During the brainstorming phase, different approaches were discussed to solve the challenge of predicting wind turbine energy output. These included traditional statistical forecasting methods, physics-based wind power estimation models, and machine learning–based prediction systems using historical weather and turbine data.

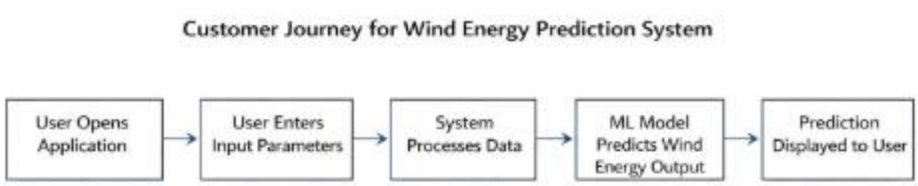
After evaluating all options, the machine learning–based approach was selected because it provides higher prediction accuracy, adaptability to complex weather patterns, and scalability for large wind farms. Machine learning models can learn from historical operational data and improve prediction performance over time, making them suitable for real-world renewable energy forecasting and smart grid management applications.

3. REQUIREMENT ANALYSIS

The Customer Journey Map explains the sequence of steps followed by the user while interacting with the wind energy prediction system. It helps in understanding how the user experiences the application from input to prediction output.

The journey begins when the user opens the application and enters weather and turbine parameters such as wind speed, temperature, and turbine conditions. The system then receives the input data and processes it using the trained machine learning prediction model. After processing, the predicted wind turbine energy output is generated and displayed to the user. This flow ensures smooth interaction, accurate forecasting, and easy usability for energy operators and analysts.

Customer Journey Map



3.2 Solution Requirements

This section describes the functional and non-functional requirements of the proposed system.

Functional Requirements

- The system should allow the user to enter wind-related input parameters (e.g., wind speed, temperature, humidity, pressure, city).
- The system should validate the user inputs before processing.
- The system should preprocess the input data for prediction.
- The system should use the trained Machine Learning model to predict wind energy output.
- The system should display the predicted wind energy output to the user.
- The system should handle invalid inputs and errors properly.
- The system should allow users to select different cities from the predefined list.
- The system should fetch real-time weather data (if API integration is enabled).
- The system should provide prediction results instantly after submission.

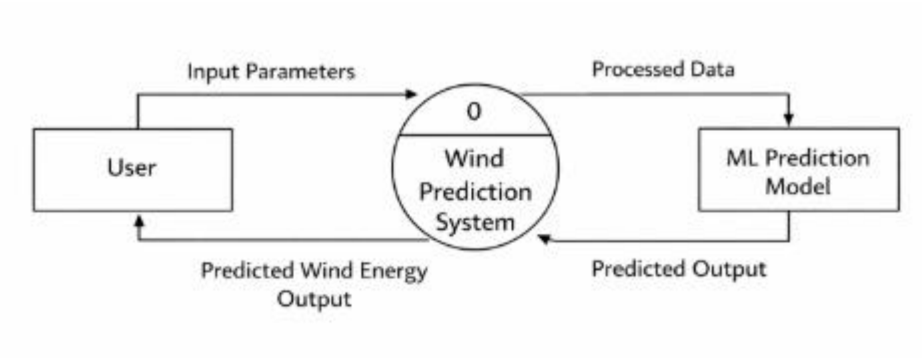
Non-Functional Requirements

- The system should have a simple, user-friendly interface that is easy to navigate and understand.
- The system should provide fast response time and generate predictions within a few seconds.
- The system should consistently provide accurate and dependable prediction results.
- The system should ensure secure handling of user inputs and protect sensitive data.
- The system should be accessible and operational whenever users need to perform predictions.
- The system should be capable of handling multiple users without performance degradation.
  The system should be designed in a modular way to allow easy updates and improvements.

3.3 Data Flow Diagram

The Data Flow Diagram (DFD) explains how data moves within the system during the translation process.

The user provides text input through the user interface. The input text is sent to the preprocessing module, where it is cleaned and prepared. The processed data is then forwarded to the AI-based translation model. Finally, the translated output is sent back to the user interface for display.



3.4 Technology Stack

The following technologies are used in the development of this project:

- Programming Language: Python
- Framework: Flask
- Machine Learning Library: Scikit-learn (for building and training the prediction model)
- Data Processing Libraries: Pandas, NumPy
- Model Storage: Pickle (.pkl file for saving the trained model)
- Frontend Technologies: HTML, CSS
- Development Tools: VS Code, Git, GitHub

4. PROJECT DESIGN

4.1 Problem–Solution Fit

The proposed Wind Energy Prediction System effectively addresses this problem by using Machine Learning techniques to predict wind energy output based on input parameters.

The system collects relevant weather data, preprocesses it, and applies a trained ML model to generate accurate energy predictions. This automated solution reduces manual calculations, improves prediction accuracy, supports better energy planning, and enhances operational efficiency.

The system is suitable for real-world applications such as renewable energy management and smart grid systems.

4.2 Proposed Solution

The proposed solution is a Machine Learning–based web application that enables users to predict wind energy output based on weather parameters.

The working of the system is described below:

- The user provides wind-related input parameters such as wind speed, temperature, humidity, pressure, and city.
- The system preprocesses and validates the input data.
- The trained Machine Learning model processes the data to predict wind energy output.
- The predicted wind energy result is generated and displayed to the user through the web interface.

The system is designed to be simple, fast, and user-friendly, ensuring ease of use for operators, researchers, and energy management teams.

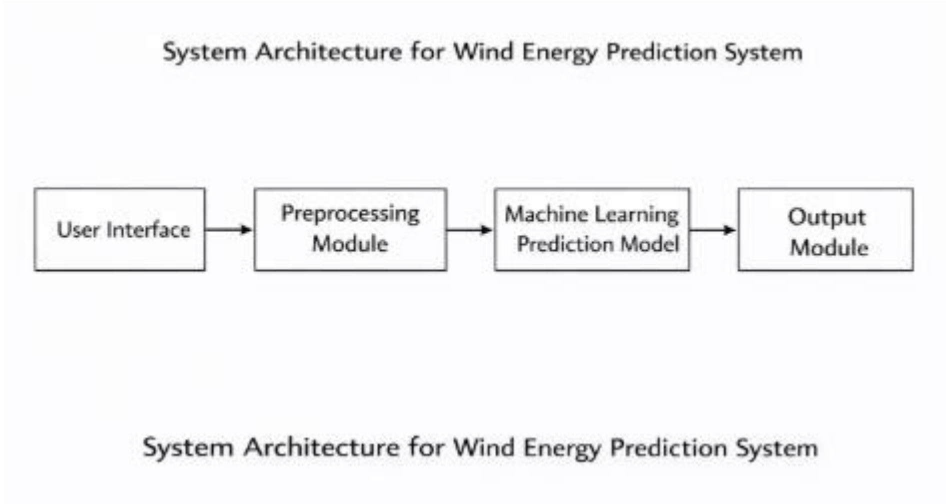
4.3 Solution Architecture

The solution architecture explains the overall structure of the system and how different components interact with each other.

The architecture consists of the following components:

- **User Interface:** Accepts wind-related input parameters (such as wind speed, temperature, humidity, pressure, and city) from the user and displays the predicted wind energy output.
- **Preprocessing Module:** Validates, cleans, and formats the input data before sending it to the prediction model.
- **Machine Learning Prediction Model:** Processes the preprocessed data and predicts the wind energy output using a trained algorithm.
- **Output Module:** Displays the predicted wind energy result clearly to the user.

These components work together in a sequential manner to ensure accurate, fast, and reliable wind energy prediction.



5. PROJECT PLANNING AND SCHEDULING

Project planning plays an important role in completing the project in a systematic and organized manner. The Wind Energy Prediction System was planned and executed in multiple phases to ensure smooth development and timely completion.

The planning of the project was carried out in the following stages:

- **Requirement Analysis:** Understanding the problem of inaccurate wind energy estimation, defining project objectives, and identifying user requirements such as input parameters, prediction accuracy, and system usability.
- **Design Phase:** Designing the system architecture, defining the interaction between components such as the user interface, preprocessing module, machine learning model, and output module.
- **Development Phase:** Implementing the application using Python and Flask, developing the machine learning prediction model using Scikit-learn, and integrating the model with the web interface.
- **Testing Phase:** Testing the system for correct predictions, input validation, response time, and overall user interaction to ensure reliability and accuracy.
- **Deployment Phase:** Running the application locally, verifying end-to-end functionality, and preparing the system for future enhancements such as real-time API integration or cloud deployment.

This phased approach helped in managing the project efficiently, ensuring structured development, and reducing errors during implementation.

5.2 Project Scheduling

The project schedule was planned to complete all activities within a total duration of **20 days**, as per the SmartBridge internship timeline. Each phase was carefully allocated time to ensure proper implementation, testing, and documentation of the project.

The scheduling of the Wind Energy Prediction System project is shown below:

Phase	Activity	Duration
Phase 1	Requirement Analysis	3 Days
Phase 2	System Design	3 Days
Phase 3	Application Development	7 Days
Phase 4	Testing and Debugging	4 Days
Phase 5	Deployment and Documentation	3 Days

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Functional testing is performed to verify that all features of the Wind Energy Prediction System work according to the specified requirements.

The following functional tests were conducted:

- Verification of input functionality: Ensuring that the system correctly accepts wind-related input parameters such as wind speed, temperature, humidity, pressure, and city.
- Validation of input data: Checking that the system properly validates inputs and handles invalid or missing values.
- Verification of model integration: Ensuring correct integration between the web application and the trained Machine Learning prediction model.
- Verification of prediction output display: Checking that the predicted wind energy output is correctly generated and displayed to the user.
- Error handling validation: Ensuring the system properly handles runtime errors and provides meaningful feedback to users.

All functional requirements were tested successfully, and the system performed as expected.

6.2 Performance Testing

Performance testing is conducted to evaluate the response time, efficiency, and stability of the Wind Energy Prediction System.

The system was tested for:

- Response Time of Prediction Generation: Measuring the time taken by the system to process input parameters and generate wind energy predictions.
- Stability During Multiple Requests: Testing the system's ability to handle multiple prediction requests without crashing or slowing down significantly.
- Smooth Interaction Between Web Interface and ML Model: Ensuring seamless communication between the Flask web application and the trained Machine Learning model.
- System Resource Usage: Monitoring CPU and memory usage during prediction to ensure efficient performance.

The results show that the system generates predictions with minimal delay and provides a smooth user experience under normal usage conditions.

7. RESULTS

The Wind Energy Prediction System was successfully developed and tested using Python, Flask, and Machine Learning techniques. The system provides accurate and fast wind energy predictions based on user-entered weather parameters.

The application interface consists of:

- Input fields for entering wind-related parameters such as wind speed, temperature, humidity, pressure, and city
- A Predict button to generate the wind energy output
- A display section showing the predicted wind energy result

When the user enters the required input parameters and clicks the Predict button, the application sends the data to the backend. The preprocessing module validates and prepares the data before passing it to the trained Machine Learning model.

The model processes the input and returns the predicted wind energy output, which is displayed instantly on the web interface.

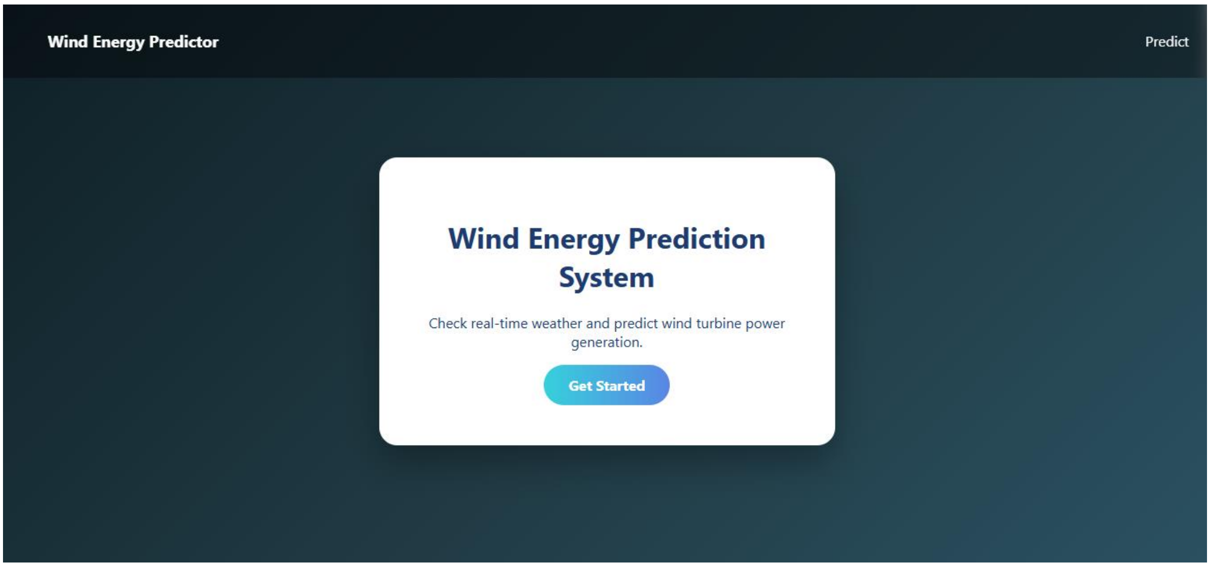
The results confirm that the system performs as expected and meets all the functional requirements defined in the project. The predictions are generated quickly with minimal delay, ensuring a smooth and user-friendly experience.

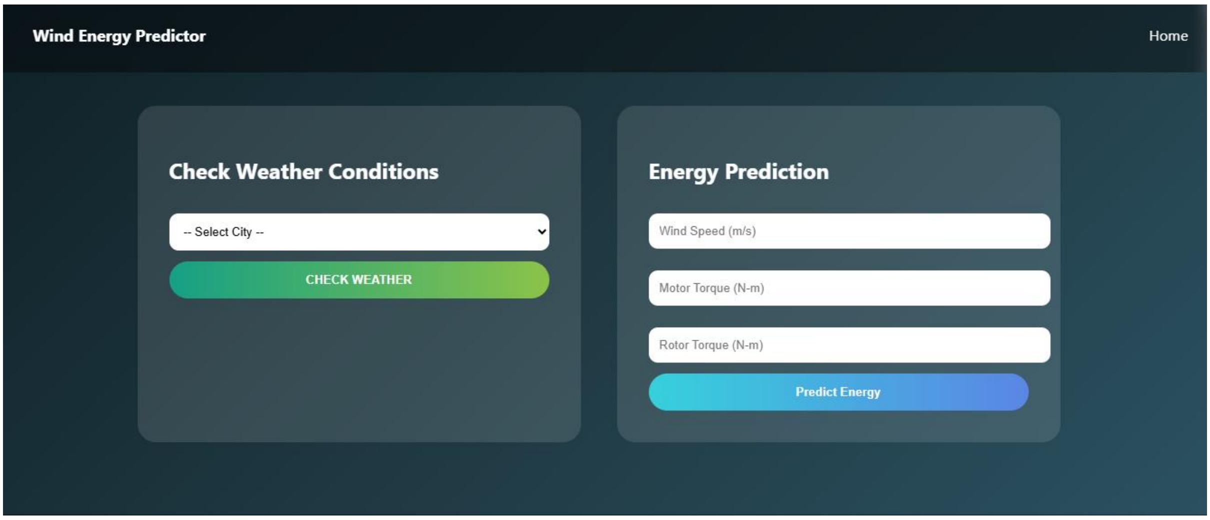
7.2 Output Screenshots

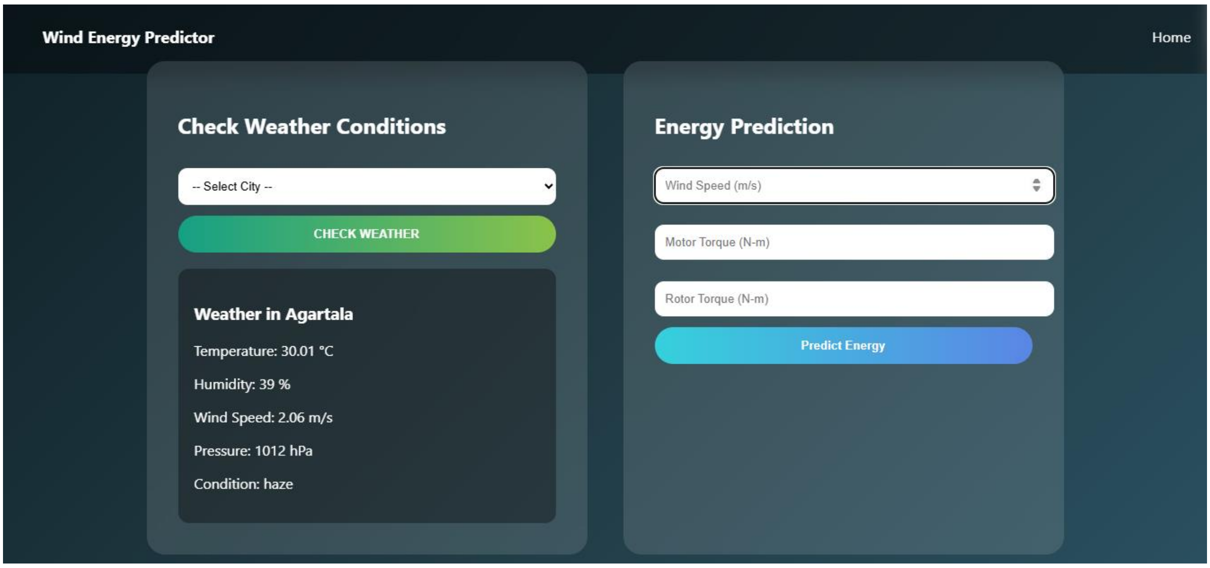
The output screenshots demonstrate the working of the Wind Energy Prediction System. The screenshots show:

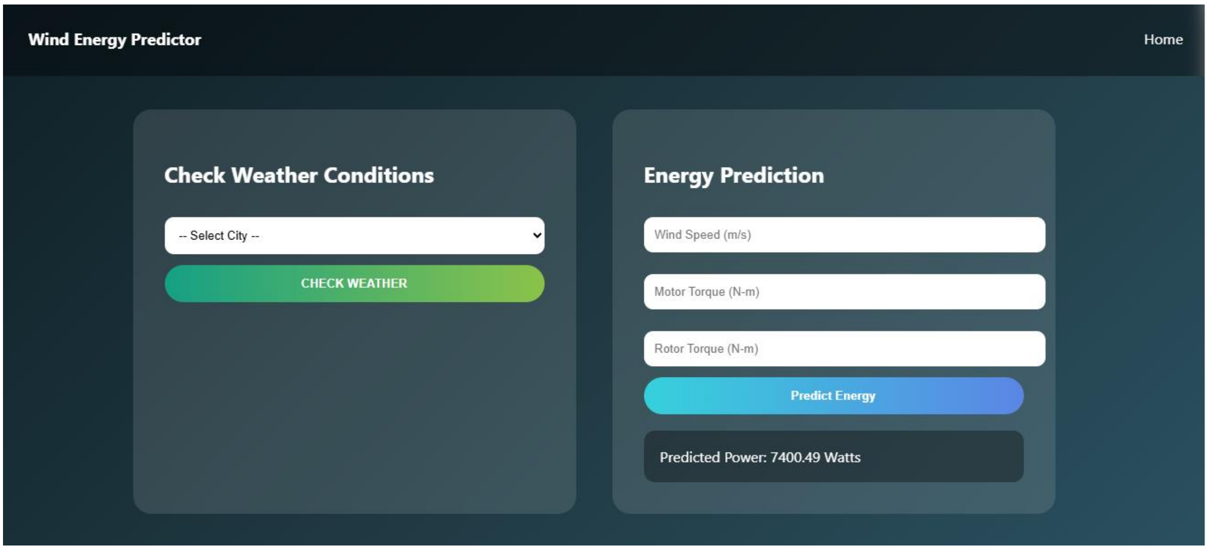
- User-entered input values such as wind speed, temperature, humidity, pressure, and city
- Weather condition details displayed after clicking the "Check Weather Conditions" button
- The calculated theoretical power based on wind speed
- The predicted wind energy output generated after clicking the "Predict Energy" button

These screenshots confirm that the system correctly accepts user inputs, processes the data, and displays accurate prediction results through the web interface.




Wind Energy Predictor
Home
Check Weather Conditions
-- Select City --
CHECK WEATHER
Energy Prediction
Wind Speed (m/s)
Motor Torque (N-m)
Rotor Torque (N-m)
Predict Energy


Wind Energy Predictor
Home
Check Weather Conditions
-- Select City --
CHECK WEATHER
Weather in Agartala
Temperature: 30.01 °C
Humidity: 39 %
Wind Speed: 2.06 m/s
Pressure: 1012 hPa
Condition: haze
Energy Prediction
Wind Speed (m/s)
Motor Torque (N-m)
Rotor Torque (N-m)
Predict Energy


Wind Energy Predictor
Home
Check Weather Conditions
-- Select City --
CHECK WEATHER
Energy Prediction
Wind Speed (m/s)
Motor Torque (N-m)
Rotor Torque (N-m)
Predict Energy
Predicted Power: 7400.49 Watts

8. ADVANTAGES AND DISADVANTAGES

8.1 Advantages

The Wind Energy Prediction System provides several advantages that make it effective and practical for real-world applications:

- Easy to use and user-friendly web interface
- Fast prediction of wind energy with minimal response time
- Accurate estimation based on wind speed and weather parameters
- Web-based application with simple deployment
- Lightweight and scalable system design
- Helps in renewable energy planning and analysis
- Useful for students, researchers, and renewable energy developers
- Supports multiple cities for broader usability

8.2 Disadvantages

Despite its advantages, the Wind Energy Prediction System has a few limitations:

- Requires an active internet connection (if real-time weather API integration is used)
- Prediction accuracy depends on the quality and quantity of training data
- The current model may not consider all real-world factors such as turbine efficiency losses, terrain effects, or mechanical constraints
- Performance may vary depending on system resources and server configuration

9. CONCLUSION

The Wind Energy Prediction System successfully demonstrates the application of Machine Learning techniques to estimate wind turbine power generation based on weather parameters. The project provides an efficient and user-friendly solution for predicting wind energy using a web-based interface built with Python and Flask.

The application delivers fast and reliable predictions by processing inputs such as wind speed, temperature, humidity, pressure, and other relevant factors. The system also calculates theoretical power using wind energy formulas, helping users understand both practical and calculated outputs.

With a simple interface and structured backend integration, the project ensures smooth interaction between the user and the prediction model. The system can assist students, researchers, and renewable energy planners in analyzing wind energy potential.

Overall, the project meets all defined objectives and showcases the practical implementation of machine learning in renewable energy applications. It serves as a strong demonstration of technical skills and real-world problem-solving capability, making it suitable for academic and internship evaluation.

10. FUTURE SCOPE

The Wind Energy Prediction System can be further enhanced in the future to improve functionality, accuracy, and scalability. Some possible future enhancements include:

- Integration with real-time weather APIs to fetch live wind data automatically
- Deployment on cloud platforms for online access and scalability
- Incorporation of advanced Machine Learning or Deep Learning models to improve prediction accuracy
- Inclusion of additional environmental factors such as air density, turbine blade length, and turbine efficiency
- Development of a mobile application version for wider accessibility
- Implementation of data visualization dashboards for detailed energy analytics
- Expansion to support large-scale wind farm energy estimation

These enhancements can make the system more powerful, intelligent, and suitable for real-world renewable energy planning and commercial applications.

11. APPENDIX

The complete source code of the project **Wind Energy Prediction System** is maintained using GitHub for version control. The repository contains all application files, including:

- Backend application files (Flask – app.py)
- Frontend templates (index.html, predict.html)
- CSS styling files
- Machine Learning model files (if applicable)

- Configuration and dependency files
- Project documentation

The version control system helps in tracking changes, managing updates, and maintaining proper documentation throughout the development process.

The repository provides all the necessary files required to understand, execute, and further enhance the project.

GitHub Repository Link: https://github.com/NaganjaliPantham/Weather-Based-Prediction-of-Wind-Turbine-Energy

Repository Contents:

The GitHub repository for the Wind Energy Prediction System contains the following files:

- app.py – Main Flask application file that handles routing, user inputs, and prediction logic
- templates/index.html – Homepage of the application
- templates/predict.html – Prediction page for entering weather parameters and viewing results
- static/style.css – CSS file for UI styling
- model.pkl (if used) – Trained Machine Learning model file
- requirements.txt – List of required Python dependencies
- .gitignore – Excludes unnecessary or sensitive files from version control
- README.md – Project description, setup instructions, and usage guidelines

Using GitHub helped in maintaining proper version control, tracking changes efficiently, and following professional software development practices throughout the project lifecycle.

11.2 Security Implementation

The Wind Energy Prediction System was developed with proper security considerations to ensure safe handling of application data and system resources.

The following secure practices were implemented:

- Sensitive configuration details (such as secret keys or API keys, if used) are stored using environment variables
- A .env file can be used to securely manage configuration values

- The .env file is excluded from GitHub using .gitignore to prevent exposure of sensitive data
- No secret keys or confidential information are hardcoded in the source code
- Input validation is implemented to prevent invalid or malicious user inputs

These practices ensure secure handling of sensitive information, protect the application from common vulnerabilities, and maintain professional software development standards.

11.3 Project Demo

A working demo of the Wind Energy Prediction System was recorded to demonstrate the functionality and output of the application.

Project Demo Video Link:
https://drive.google.com/file/d/12EgBe5nRar9uTDMz9DTpE-kY7YfaO86L/view?usp=sharing

The demo video showcases:

- User entering wind-related input parameters such as wind speed, temperature, humidity, pressure, and city
- Display of weather condition details after clicking the "Check Weather Conditions" button
- Calculation of theoretical power based on wind speed
- Prediction of wind turbine energy after clicking the "Predict Power" button
- Final output displayed clearly on the web interface

The demo confirms that the system works as expected and successfully performs wind energy prediction with a smooth and user-friendly interface.

11.4 Team Details

- **Internship Program:** SmartBridge Virtual Internship
- **Project Start Date:** 29 January 2026

Team ID: LTVIP2026TMIDS84120
Team Size: 4

Team Leader:

Pantham Naganjali

Team Members:

- Gurram Teja Sri
- Dadala Lakshman
- Steni Desabathula

11.5 References

The following resources were referred to during the development of this project:

- Python Programming Language
- Flask Framework
- Scikit-learn Library
- NumPy and Pandas Libraries
- OpenWeatherMap API