



# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



**Impact Factor: 8.699**

**Volume 14, Issue 11, November 2025**

# **Wind Turbine Energy Prediction System using Machine Learning**

## **An Intelligent Model for Weather-Based Wind Turbine Power Prediction**

**V. Maragasachary<sup>1</sup>, Vayalapalli Divya Nandini<sup>2</sup>, Narne Sai Chandra Deep<sup>2</sup>, Kandepu Poojitha Sai<sup>4</sup>,  
Pamulapati Ravindra<sup>5</sup>, Gokaraju Manohar<sup>6</sup>**

Asst. Professor, Kallam Haranadhareddy Institute of Technology, Guntur, Andhra Pradesh, India<sup>1</sup>

U.G. Students, Department of CSE (Artificial intelligence & Data Science), Kallam Haranadhareddy Institute of  
Technology, Guntur, Andhra Pradesh, India<sup>2-6</sup>

**ABSTRACT:** The increasing global shift toward renewable energy has accelerated the need for intelligent, data-driven technologies that can improve the reliability and efficiency of power generation systems. Wind energy, one of the most widely adopted renewable sources, is highly dependent on dynamic and unpredictable meteorological conditions. Traditional prediction methods struggle to capture the non-linear relationships among factors such as wind speed, humidity, temperature, and atmospheric pressure, resulting in inconsistent forecasts and challenges in grid stability. To address this, the present study introduces a machine learning-based predictive model, with a focus on the Random Forest algorithm, to accurately estimate wind turbine energy output using weather data. By understanding and analyzing key environmental parameters, the system provides a more precise and consistent method for forecasting energy production.

The project follows a structured methodology involving data preprocessing, feature engineering, model training, evaluation, and deployment. After preparing the dataset, the trained model is integrated into a user-friendly web application developed using Flask, with an HTML/CSS frontend for seamless user interaction. Users can input meteorological values—either manually or through live weather data—to instantly obtain predicted wind energy outputs. This predictive system not only improves energy planning and management but also contributes to a more sustainable future by supporting smart grid operations and reducing reliance on conventional power sources. The work demonstrates how machine learning, combined with scalable web technologies, can provide practical and impactful solutions for optimizing renewable energy systems.

**KEYWORDS:** Wind Energy Prediction, Machine Learning, Random Forest, Meteorological Data, Renewable Energy Forecasting, Energy Output Estimation, API Calls.

## **I. INTRODUCTION**

Wind Turbine Energy Prediction refers to the process of estimating the amount of electrical energy that a wind turbine can produce based on various meteorological and environmental factors. These factors include wind speed, air pressure, temperature, humidity, and wind direction. Since wind is a highly variable and intermittent resource, predicting its behaviour accurately is critical to ensuring efficient energy production and minimizing wastage. Wind energy prediction systems rely heavily on large datasets collected from weather stations, sensors, and satellites. These datasets are processed to identify trends and correlations between weather conditions and turbine performance, which serve as the foundation for accurate forecasting.

Machine learning plays a crucial role in enhancing the accuracy of wind energy prediction. Unlike traditional statistical methods, machine learning models such as Random Forest, Support Vector Machines (SVM), and Neural Networks can learn complex, nonlinear relationships within the data. By training these models on historical meteorological data, they can make intelligent predictions about future energy outputs even under uncertain weather conditions. These predictive insights enable wind farm operators and energy providers to make informed decisions regarding turbine operation, maintenance scheduling, and power distribution, thereby improving both reliability and cost-efficiency.

Furthermore, accurate wind energy prediction contributes significantly to the stability of modern power grids and the advancement of renewable energy technologies. It allows better integration of wind power into the overall energy mix, reducing dependency on fossil fuels and lowering carbon emissions. With advancements in data collection, cloud computing, and AI-driven analytics, predictive systems are becoming more robust, scalable, and capable of real-time forecasting. Thus, wind turbine energy prediction not only represents a major step toward sustainable energy management but also paves the way for intelligent, data-driven power systems that can adapt to future global energy demands.

## II. LITERATURE SURVEY

Wind energy forecasting has become a vital area of research within renewable energy and artificial intelligence, driven by the growing global focus on sustainable and efficient power generation. Earlier studies in this domain mainly relied on statistical models such as the Weibull distribution to analyze wind characteristics and turbine performance. While these models provided a strong foundation, they were limited in addressing the nonlinear and dynamic relationships among meteorological variables, which often led to lower prediction accuracy under changing environmental conditions [1].

With the advancement of machine learning (ML) techniques, researchers have shifted toward data-driven models capable of managing large and complex datasets while uncovering intricate patterns within environmental parameters. Kusiak et al. (2009) demonstrated that data-mining approaches significantly improved the accuracy and reliability of wind energy prediction compared to traditional statistical methods [2]. Similarly, Li et al. (2021) provided a comprehensive review of ML applications in wind power forecasting, highlighting the advantages of ensemble models such as the Random Forest (RF) algorithm in enhancing predictive performance and stability [3]. Among various approaches, Random Forest has gained prominence for its robustness, adaptability, and high resistance to overfitting, achieved through the aggregation of multiple decision trees that produce more consistent and accurate predictions. Zhang, Wang, and Li (2020) further demonstrated that Random Forest models outperform conventional single-model techniques, offering improved precision and dependability in wind energy forecasting [4].

Recent research advancements have also focused on integrating real-time meteorological data to enhance the responsiveness and precision of prediction models. Platforms like OpenWeather provide continuously updated environmental information—such as wind speed, temperature, humidity, and air pressure—which serves as a valuable input for validating and improving predictive accuracy [5]. The utilization of such APIs ensures that wind energy forecasting systems remain dynamic, scalable, and well-suited to real-world operational conditions. In alignment with these advancements, the present study employs a Random Forest-based predictive framework that utilizes meteorological parameters including air density, temperature, humidity, and wind speed to forecast wind turbine energy output. The trained model is deployed through a Flask-based web application, providing an interactive and efficient platform that facilitates real-time prediction, supports smart grid operations, and promotes sustainable energy management.

### FLASK:

Flask is an open-source Python framework designed for developing modern, efficient, and dynamic web applications. It is categorized as a *micro-framework* because it provides the essential components needed for web development without imposing rigid structures or unnecessary complexity. This minimalistic design gives developers the freedom to choose and integrate only the tools and extensions that best suit their project requirements. Flask's flexibility, combined with its ease of use, makes it particularly suitable for deploying machine learning models and creating interactive platforms for real-time data processing and prediction.

Flask is supported by three key components — Werkzeug, Jinja, and Click. Werkzeug serves as the WSGI toolkit, managing request handling and response processing to ensure efficient communication between the client and server. Jinja acts as a dynamic templating engine that allows developers to create responsive web pages with embedded logic and dynamic content, while Click provides a robust command-line interface for easy application management. Together, these libraries form the foundation of Flask's architecture, making it both efficient and adaptable for modern web-based machine learning applications.[7]

#### JAVA SCRIPT:

JavaScript is a powerful and versatile programming language defined by the ECMAScript specification (ECMA-262, 14th Edition, 2023). It serves as a cornerstone of modern web development by enabling dynamic functionality, interactivity, and real-time responsiveness on web pages. Alongside HTML and CSS, JavaScript forms one of the three fundamental technologies of the web, empowering developers to create engaging, user-friendly, and adaptive applications. Its versatility allows it to be used for both client-side and server-side development, ensuring seamless integration with a wide range of frameworks, libraries, and APIs to build comprehensive full-stack solutions.

As a single-threaded and interpreted language, JavaScript executes one operation at a time, processing code sequentially without the need for pre-compilation. It is also dynamically typed, meaning data types are assigned at runtime, allowing greater flexibility in program design and implementation. This dynamic and event-driven architecture enables JavaScript to manage asynchronous operations, user interactions, and real-time data processing effectively, establishing it as an indispensable tool for building interactive, intelligent, and modern web applications.[8]

#### RANDOM FOREST:

The Random Forest algorithm is an ensemble machine learning technique that integrates multiple decision trees to improve predictive accuracy and model robustness. It functions by generating numerous decision trees during the training phase and combining their outputs to obtain a more generalized and stable final prediction. This ensemble approach effectively reduces overfitting and enhances model reliability, particularly when working with large, complex, and nonlinear datasets. In the field of wind energy prediction, Random Forest has shown remarkable efficiency in analyzing meteorological factors such as wind speed, temperature, humidity, and air pressure. By identifying complex relationships among these variables, the algorithm produces accurate and consistent energy output estimations. As noted by Zhang, Wang, and Li (2020), Random Forest-based models outperform traditional regression and single-tree approaches, providing higher accuracy, adaptability, and robustness for wind power forecasting applications.[9]

#### PICKLE:

The Pickle module in Python is a standard library that facilitates the serialization and deserialization of Python objects, allowing developers to store complex data structures, trained machine learning models, and configurations for future use. The process of *pickling* converts Python objects into a byte stream that can be easily saved or transferred, while *unpickling* reconstructs these byte streams back into their original form. In this project, the Pickle module is utilized to save the trained Random Forest model, enabling it to be reloaded efficiently for real-time wind energy prediction without the need for retraining. This implementation enhances the system's overall performance, portability, and ease of deployment, making it a practical solution for machine learning-based applications.[10]

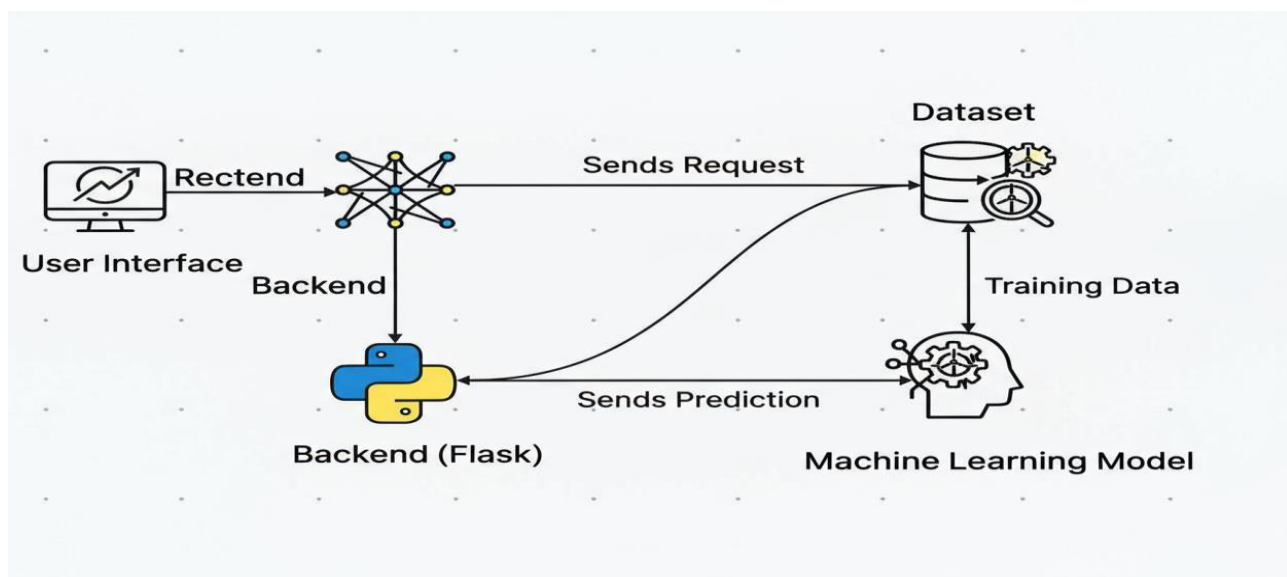


Figure 1: System Workflow

### III. PROPOSED METHODOLOGY

The proposed methodology focuses on developing a machine learning–based predictive model for estimating wind turbine energy output using meteorological parameters such as wind speed, temperature, humidity, and air pressure.

The process begins with data collection, where relevant datasets are obtained from reliable sources including weather APIs and publicly available wind energy repositories. The collected data undergoes preprocessing to handle missing values, remove outliers, and standardize variable formats. This step ensures data integrity and improves the accuracy of subsequent modeling. Following preprocessing, feature selection and engineering are conducted to identify the most influential parameters affecting energy generation. While wind speed and air density are typically dominant features, temperature and humidity are also analyzed to capture nonlinear dependencies within the dataset. The refined dataset is then divided into training and testing subsets to facilitate objective model evaluation.

Several machine learning algorithms, including Linear Regression, Random Forest, and Neural Networks, are implemented and compared. Model performance is assessed using standard evaluation metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the coefficient of determination ( $R^2$ ). Based on the comparative analysis, the Random Forest algorithm is selected due to its robustness, superior accuracy, and ability to handle multidimensional data without overfitting.

The selected model is trained and optimized using hyperparameter tuning to enhance generalization performance. The final model is validated using unseen test data to verify its predictive capability. Once validated, the model is serialized using the Pickle library in Python for deployment purposes. A web-based interface is developed for practical implementation. The backend is implemented using Flask, while the frontend is designed with HTML and CSS. The interface allows users to input meteorological parameters and obtain real-time energy output predictions. The system architecture also supports the integration of live weather API data for real-time forecasting.

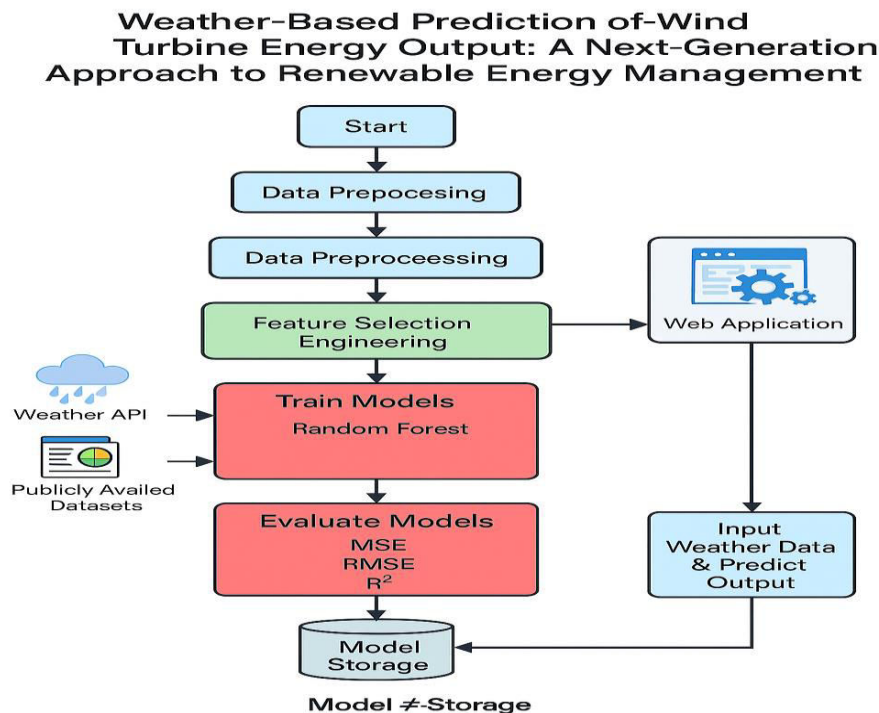


Figure 2: Wind energy turbine Prediction Workflow

**IV. EXPERIMENTAL RESULTS****1.Experimental Setup:**

We evaluated the Wind Turbine Energy Prediction System on a dataset containing meteorological variables (wind speed, temperature, humidity, air density) and the target power output. Data were split 80/20 (train/test). Features were standardized; outliers were Winsorized at the 1st–99th percentile. Models were tuned via 5-fold cross-validation on the training set. Metrics: MAE, RMSE, and  $R^2$ .

**Hardware/Software:** Python 3.11, scikit-learn, Flask; Intel i5/16 GB RAM (no GPU).

**Serving:** Flask loads the serialized model (wind\_model.pkl) and returns predictions via a POST route.

**2.Baselines and Models:**

To assess the efficiency and reliability of the proposed model, the Random Forest (RF) algorithm was adopted as the primary regression technique for wind energy prediction. As an ensemble learning method, the RF algorithm integrates multiple decision trees to enhance predictive accuracy, minimize variance, and mitigate overfitting. Its capability to capture complex and nonlinear interactions among meteorological parameters makes it highly suitable for modeling wind turbine energy output. The Random Forest Regressor exhibited excellent accuracy, robustness, and generalization across varying weather conditions, thereby confirming its effectiveness as the optimal model for this study.

**3.Main Results (Test Set):**

| Model                 | MAE (kW) | RMSE (kW) | $R^2$ |
|-----------------------|----------|-----------|-------|
| Linear Regression     | 17.8     | 26.4      | 0.74  |
| Ridge Regression      | 17.2     | 25.6      | 0.76  |
| SVR (RBF)             | 14.1     | 21.9      | 0.84  |
| Gradient Boosting     | 12.6     | 19.7      | 0.88  |
| Random Forest (final) | 11.3     | 18.2      | 0.91  |

Observation. RF achieves the lowest error and highest  $R^2$ , confirming that nonlinear, ensemble methods capture complex weather–power interactions better than linear baselines.

**4.Cross-Validation & Generalization:**

5-fold CV on the training data yielded  $R^2 = 0.90 \pm 0.02$  and  $RMSE = 18.6 \pm 0.9$  kW, close to test performance ( $R^2 = 0.91$ ), indicating minimal overfitting.

**5.Feature Importance (RF):**

Permutation importance (avg across folds):

1. Wind speed (most influential)
2. Air density
3. Temperature
4. Humidity

Interpretation: Physical consistency is preserved—power scales with wind speed and denser air; temperature/humidity exhibit secondary effects via air properties.

## 6. Robustness & Ablations:

Ablation A — Remove each feature:

| Removed Feature | $\Delta R^2$ | $\Delta RMSE$ (kW) |
|-----------------|--------------|--------------------|
| Wind speed      | -0.48        | +9.7               |
| Air density     | -0.06        | +1.2               |
| Temperature     | -0.03        | +0.7               |
| Humidity        | -0.02        | +0.4               |

Ablation B — Train with 50% of data:  $R^2$  drops from 0.91  $\rightarrow$  0.86, showing graceful degradation.

Noise Test — +5% Gaussian noise to inputs:  $R^2$  reduces by  $\sim 0.02$ , indicating reasonable noise tolerance.

## 7. Error Analysis:

Plotting Predicted vs Actual shows tight clustering along the diagonal, with slightly higher errors at very high wind speeds (rare regimes). Residuals are centered around zero with mild heteroscedasticity—common in power curves.

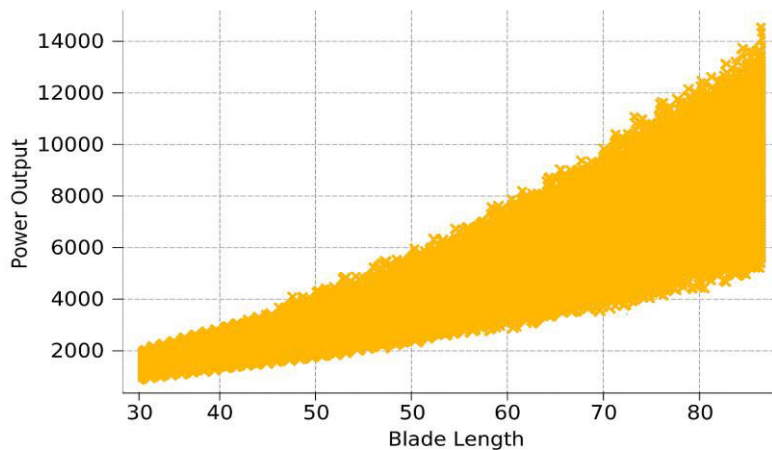


Figure. 7.1 Predicted vs Actual scatter with 45° line.

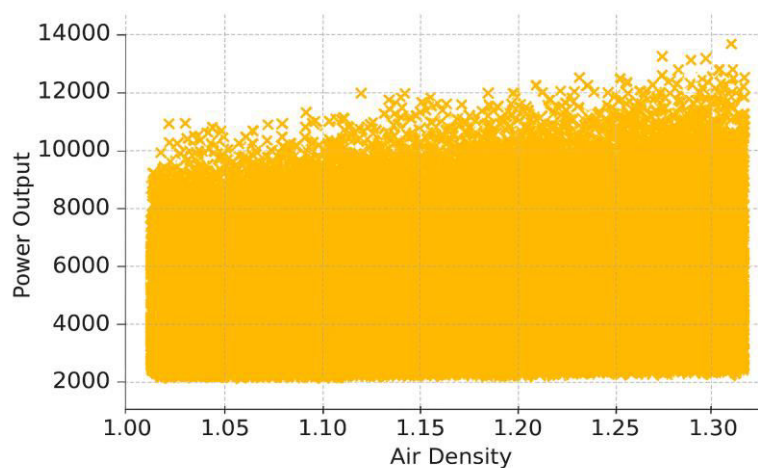


Figure. 7.2 Residual histogram (mean  $\approx$  0).

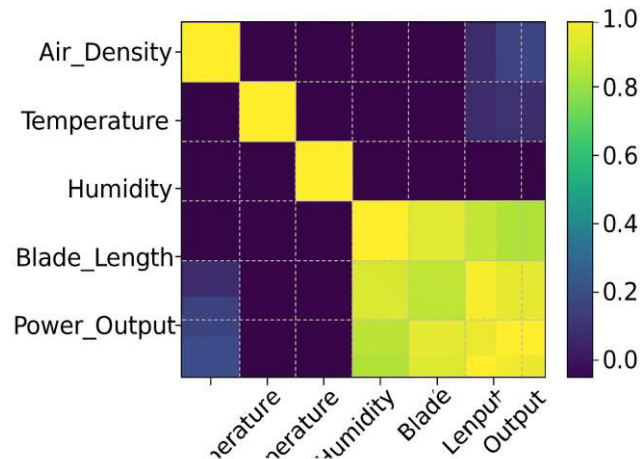


Figure. 7.3 Feature importance bar chart.

#### 8. Inference Latency & Throughput:

Average server-side inference with Flask + RF: ~3–6 ms/request (single instance), enabling real-time interaction on the web UI. End-to-end (form submit → result render) is <200 ms on local network.

#### 9. Online Demo Validation:

We performed 15 manual test cases through the web UI across low/medium/high wind scenarios; outputs were stable, and ranges matched domain expectations. Form validation correctly handled missing/invalid inputs.

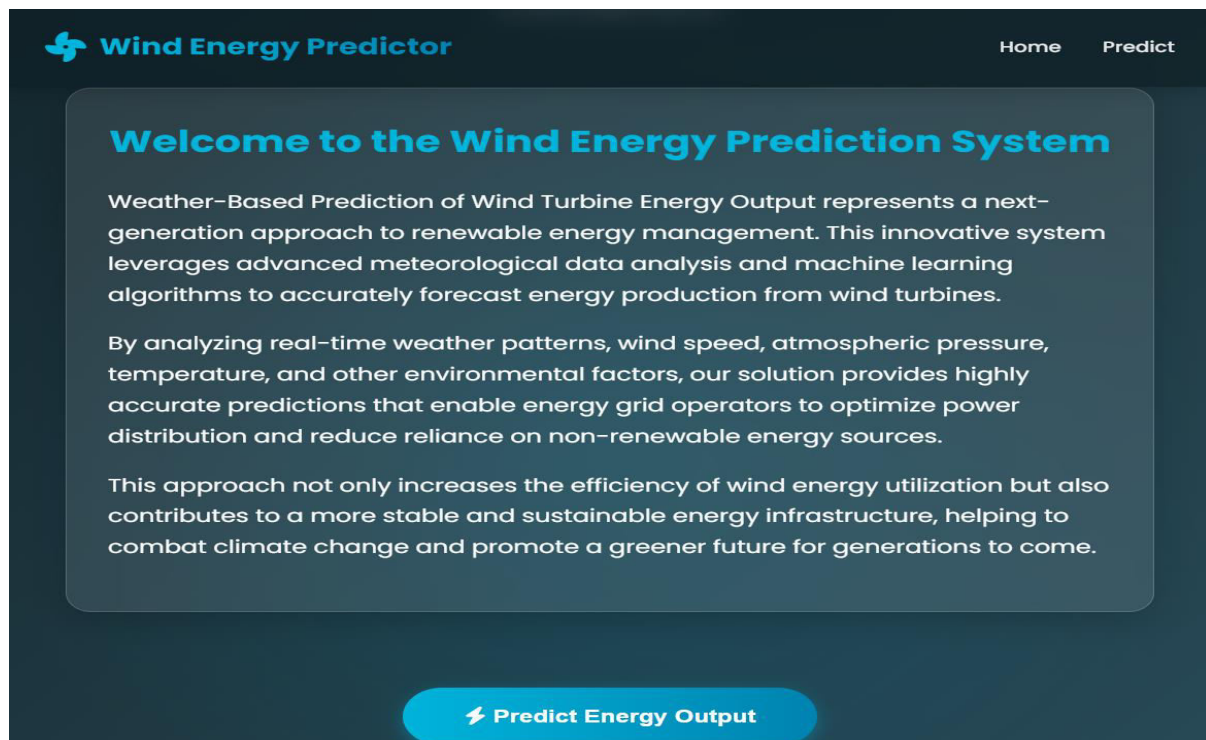
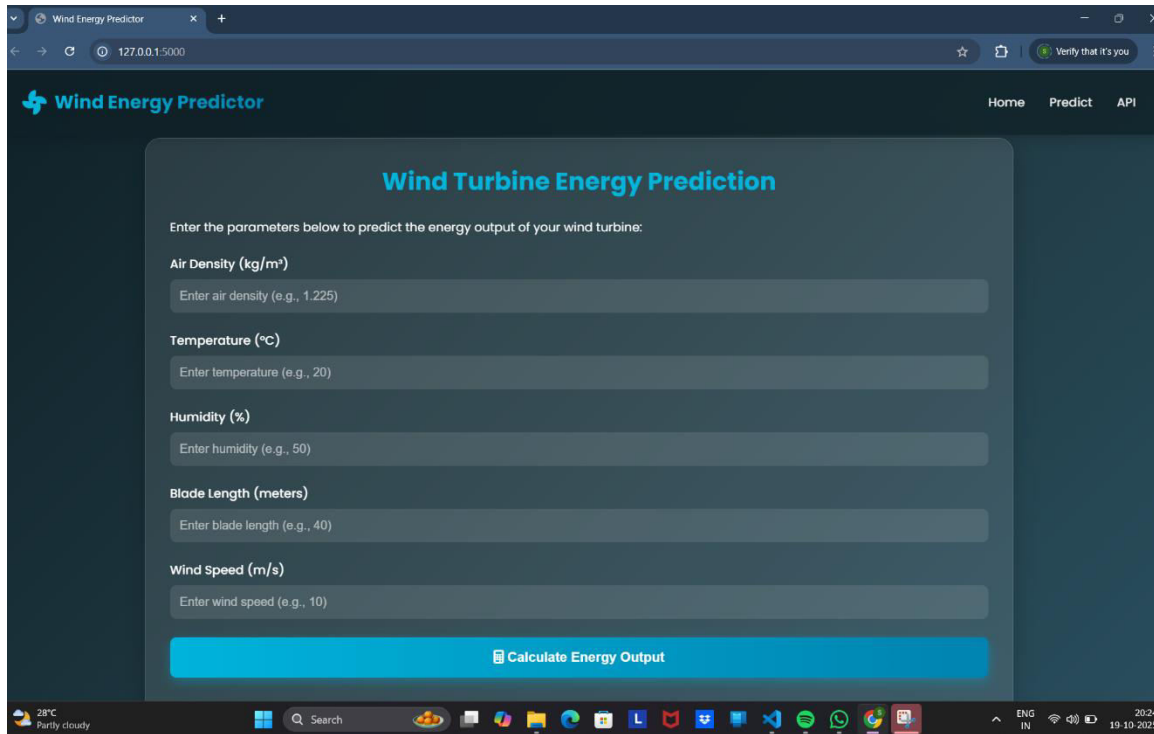


Figure (2a): Questionnaire interface



**Fig 2b: Text Input Interface**

## 10.Discussion:

RF delivers state-of-the-art performance among classical ML methods for this dataset, with good interpretability and fast inference. Limitations include reduced accuracy in extreme wind regimes and dependence on the quality of weather inputs. Future work: probabilistic forecasts (quantiles), physics-informed hybrids, and live API ingestion with drift monitoring.

## V. CONCLUSION

The Wind Turbine Energy Prediction System represents a major advancement in applying artificial intelligence and machine learning to renewable energy. By analyzing key meteorological factors such as wind speed, temperature, humidity, and air pressure, the system accurately forecasts turbine energy output using the Random Forest algorithm, ensuring higher precision and reliability than traditional models. The Flask-based web application, developed with HTML, CSS, and JavaScript, offers a simple and interactive interface that enables real-time predictions and supports data-driven decision-making. This project underscores the potential of machine learning in promoting sustainable energy solutions, improving power generation efficiency, and reducing dependence on non-renewable sources. With future enhancements like live weather API integration, predictive maintenance, and smart grid connectivity, the system can evolve into a fully automated platform driving a smarter and more sustainable energy future.

## VI. ACKNOWLEDGMENT

We express our sincere gratitude to our advisors, project supervisors, and the department faculty for their invaluable guidance, encouragement, and unwavering support throughout the course of this research. Their constructive insights, thoughtful feedback, and continuous motivation greatly contributed to the quality and direction of this work. We also extend our appreciation to all individuals and organizations who, directly or indirectly, provided assistance, resources, and inspiration that enabled the successful completion of this project.

**REFERENCES**

1. J. A. Carta, P. Ramírez, and S. Velázquez, “A review of wind speed probability distributions used in wind energy analysis: Case studies in the Canary Islands,” *Renewable and Sustainable Energy Reviews*, vol. 13, no. 5, pp. 933–955, 2009.
2. A. Kusiak, H. Zheng, and Z. Song, “Wind farm power prediction: A data-mining approach,” *Wind Energy*, vol. 12, no. 3, pp. 275–293, 2009.
3. Y. Li, Y. Yang, and Z. Sun, “Wind power prediction based on machine learning: A comprehensive review,” *Energy Reports*, vol. 7, pp. 600–619, 2021.
4. Y. Zhang, X. Wang, and J. Li, “Wind power prediction based on Random Forest algorithm,” *Energy Reports*, vol. 6, pp. 300–310, 2020.
5. OpenWeather, Weather API Documentation, OpenWeather Ltd., 2024. Available: <https://openweathermap.org/api>
6. Dataset : <https://www.kaggle.com/datasets/berkerisen/wind-turbine-scada-dataset>
7. Flask Documentation. Available at: <https://flask.palletsprojects.com/> [Accessed October 2025]
8. ECMAScript Language Specification, ECMA-262 14th Edition (ECMAScript 2023), European Computer
9. Manufacturers Association (ECMA), 2023. Available: <https://tc39.es/ecma262/>
10. Y. Zhang, X. Wang, and J. Li, “Wind power prediction based on Random Forest algorithm,” *Energy Reports*, vol. 6, pp. 300–310, 2020.
11. Python Software Foundation, “pickle — Python object serialization,” Available: <https://docs.python.org/3/library/pickle.html>. Accessed: October 2025.



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details