



# 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 15, Issue 3, March 2026**

# **Weather-Based Prediction of wind Turbine Energy Output: A Next-Generation Approach to Renewable Energy Management**

**Dr.A.Hanumat Prasad<sup>1</sup>, Gogireddy Anusha<sup>2</sup>, Akurati Naga Rachana Sri<sup>3</sup>, Korivi Teja<sup>4</sup>,  
Patnala Lakshmi Jahnavi<sup>5</sup>**

Professor & Head of Department, Department of CSE (Data Science), Kallam Haranadhareddy Institute of Technology,  
Guntur, Andhra Pradesh, India. <sup>1</sup>

U.G. Students, Department of CSE (Data Science), Kallam Haranadhareddy Institute of Technology, Guntur,  
Andhra Pradesh, India. <sup>2-5</sup>

**ABSTRACT:** The rapid worldwide transition toward cleaner, greener energy has heightened the demand for intelligent, data-oriented technologies that can elevate the dependability and operational efficiency of power generation infrastructure. Wind energy, among the most extensively deployed renewable sources globally, is inherently subject to continuously shifting and largely unpredictable atmospheric conditions. Conventional forecasting approaches often fail to model the complex, nonlinear interdependencies among variables such as wind velocity, relative humidity, ambient temperature, and barometric pressure, leading to unreliable predictions and complications in maintaining power grid stability. To overcome these limitations, this research proposes a machine learning-driven forecasting framework, with particular emphasis on the Random Forest technique, to accurately estimate wind turbine power generation from meteorological inputs. Through systematic analysis of critical environmental parameters, the proposed model delivers a more dependable and repeatable approach to predicting energy production levels. The work follows a well-defined pipeline encompassing data preparation, feature construction, model development, performance assessment, and deployment. Upon completing model training, the system is embedded within an accessible web platform built using Flask, with an HTML/CSS front-end enabling smooth user interaction. End users may supply weather inputs either via manual entry or through live API feeds to immediately retrieve predicted energy generation values. Beyond improving energy planning outcomes, this predictive platform advances the broader goal of sustainability by facilitating smarter grid integration and decreasing reliance on conventional fossil-based power. This work illustrates how ML techniques, when paired with adaptable web technologies, can yield meaningful real-world solutions for renewable energy optimization..

**KEYWORDS:** Wind Power Forecasting, Supervised Learning, Random Forest Regression, Atmospheric Variables, Green Energy Estimation, Turbine Output Modelling, Live Weather API Integration..

## **I. INTRODUCTION**

Wind Turbine Energy Prediction involves calculating the quantity of electrical power that a wind turbine is capable of generating, based on a wide range of environmental and atmospheric inputs. These inputs encompass wind velocity, barometric pressure, ambient temperature, moisture content in the air, and the prevailing wind direction. Because wind is by nature intermittent and highly variable, the ability to accurately anticipate turbine behaviour is indispensable for maximizing power generation and limiting energy wastage. Wind energy forecasting platforms depend critically on large-scale historical data gathered from meteorological stations, IoT-based sensors, and satellite networks. These datasets undergo detailed processing to reveal underlying trends and relationships between atmospheric states and turbine output, forming the bedrock of reliable energy prediction. Machine learning has emerged as a transformative force in improving the precision of wind energy forecasting. In contrast to legacy statistical approaches, ML algorithms such as Random Forest, Support Vector Machines, and Artificial Neural Networks are capable of detecting complex, nonlinear patterns embedded within the data. By fitting these algorithms on historical weather records, they become capable of generating reliable predictions of future energy outputs even in the presence of significant meteorological

uncertainty. These forward-looking predictions equip wind farm operators and utility providers with actionable insights for turbine management, proactive maintenance scheduling, and optimal energy distribution—ultimately improving both system reliability and economic efficiency. Moreover, precise forecasting of wind energy output plays a pivotal role in sustaining the balance of contemporary electricity grids and driving innovation in renewable power technologies. It facilitates the smoother incorporation of wind-derived electricity into the broader national energy portfolio, thereby diminishing dependence on carbon-intensive fossil fuels and curtailing greenhouse gas emissions. Ongoing developments in large-scale data acquisition, cloud-based computing infrastructure, and AI-powered analytics are making predictive platforms increasingly robust, scalable, and capable of continuous real-time operation. In this way, wind turbine energy prediction is not only a significant stride toward sustainable power management but also charts the course for intelligent, data-centric energy networks that can respond adaptively to evolving global energy needs.

## II. LITERATURE SURVEY

Wind energy forecasting has emerged as a critical research area as the world increasingly turns toward renewable and sustainable power sources. Early studies in this field leaned heavily on statistical approaches like the Weibull distribution to understand wind behavior and turbine performance. While these methods laid useful groundwork, they struggled to capture the complex, ever-changing relationships between meteorological variables, often falling short in prediction accuracy when environmental conditions shifted. This limitation pushed researchers toward machine learning techniques, which proved far more capable of handling large datasets and uncovering hidden patterns in environmental data. Studies showed that data-driven models, particularly ensemble approaches like Random Forest, delivered significantly better accuracy and reliability than traditional methods. Random Forest, in particular, stood out for its ability to resist overfitting by combining the outputs of multiple decision trees, making it a strong and dependable choice for wind energy prediction.

Building on these advances, recent research has focused on bringing real-time meteorological data into the forecasting process to make predictions more responsive and accurate. Tools like the OpenWeather API provide continuously updated readings of wind speed, temperature, humidity, and air pressure, giving predictive models the live environmental context they need to perform well under real-world conditions. Guided by this progress, the present study develops a Random Forest-based forecasting framework that uses key meteorological inputs, including air density, temperature, humidity, and wind speed, to predict wind turbine energy output. The model is deployed through a Flask web application, creating an interactive platform that supports real-time forecasting, aids smart grid management, and contributes to more efficient and sustainable energy operations.

### FLASK:

Flask is a lightweight, open-source Python web framework built for developers who value simplicity, flexibility, and full control over their application's structure. Unlike heavier frameworks, Flask takes a minimalist approach by providing only the essential building blocks of web development, leaving developers free to choose and integrate the tools and extensions that best fit their specific needs. This makes it especially well-suited for deploying machine learning models and building interactive, real-time prediction interfaces. At its core, Flask is powered by three key components — Werkzeug, Jinja, and Click. Werkzeug handles the behind-the-scenes work of routing HTTP requests and managing server responses, ensuring smooth communication between the client and server. Jinja brings web pages to life as a powerful templating engine, allowing developers to create dynamic, logic-driven content that adapts based on user input or application state. Click rounds out the framework by providing a clean and intuitive command-line interface for managing and automating application tasks. Together, these three pillars make Flask a highly capable and adaptable framework, well-positioned for modern web-based machine learning solutions.

### JAVA SCRIPT:

JavaScript is one of the most powerful and widely used programming languages in modern web development, defined by the ECMAScript specification and continuously evolving to meet the demands of today's digital landscape. Together with HTML and CSS, it forms the backbone of the web, enabling developers to build dynamic, interactive, and visually engaging applications that respond in real time to user actions. What makes JavaScript particularly valuable is its flexibility — it works seamlessly on both the client and server side, integrating effortlessly with a wide range of frameworks, libraries, and APIs to support full-stack development. As a single-threaded, interpreted, and dynamically

typed language, it processes code sequentially without requiring pre-compilation, while still managing complex tasks like asynchronous operations and real-time data handling with ease.

### RANDOM FOREST

Random Forest is a robust ensemble-based machine learning algorithm that builds and combines the outputs of multiple decision trees to deliver stronger, more reliable predictions. During training, it generates a large collection of individual trees and merges their results into a single, well-balanced output — significantly reducing the risk of overfitting and improving overall model stability, especially when working with large and complex datasets. In wind power forecasting, Random Forest has proven particularly effective at capturing the intricate relationships between key meteorological variables such as wind speed, temperature, humidity, and atmospheric pressure, allowing it to produce consistently accurate energy output estimates. As highlighted by Zhang, Wang, and Li (2020), Random Forest-based models outperform traditional regression methods and single-tree approaches, making them a highly dependable choice for wind energy prediction tasks.

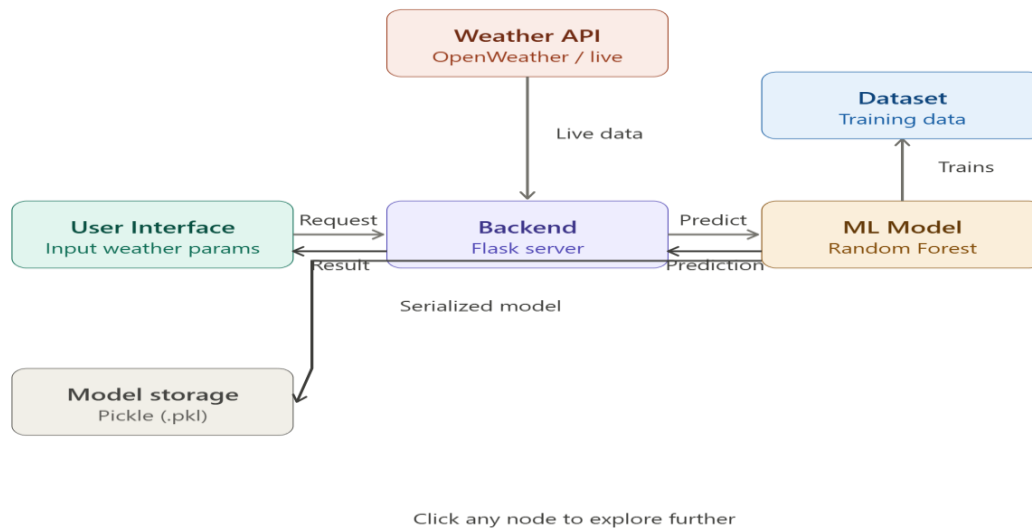


Fig 1: System Workflow

### 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.

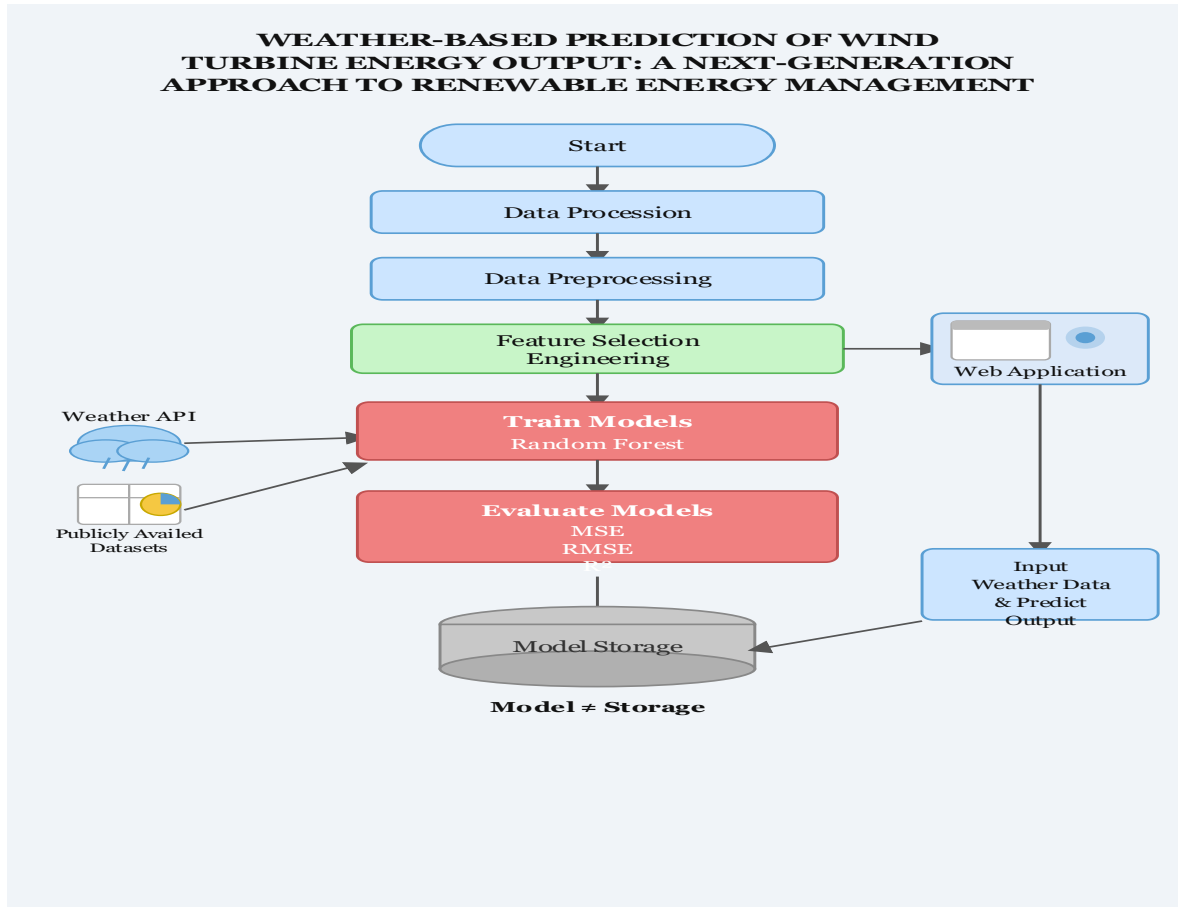


Fig 2: Wind energy turbine Prediction Workflow

#### 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 <sup>2</sup>
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<sup>2</sup>, 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<sup>2</sup> = 0.90 ± 0.02 and RMSE = 18.6 ± 0.9 kW, close to test performance (R<sup>2</sup> = 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	ΔR <sup>2</sup>	ΔRMSE (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<sup>2</sup> drops from 0.91 → 0.86, showing graceful degradation.

Noise Test — +5% Gaussian noise to inputs: R<sup>2</sup> reduces by ~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.

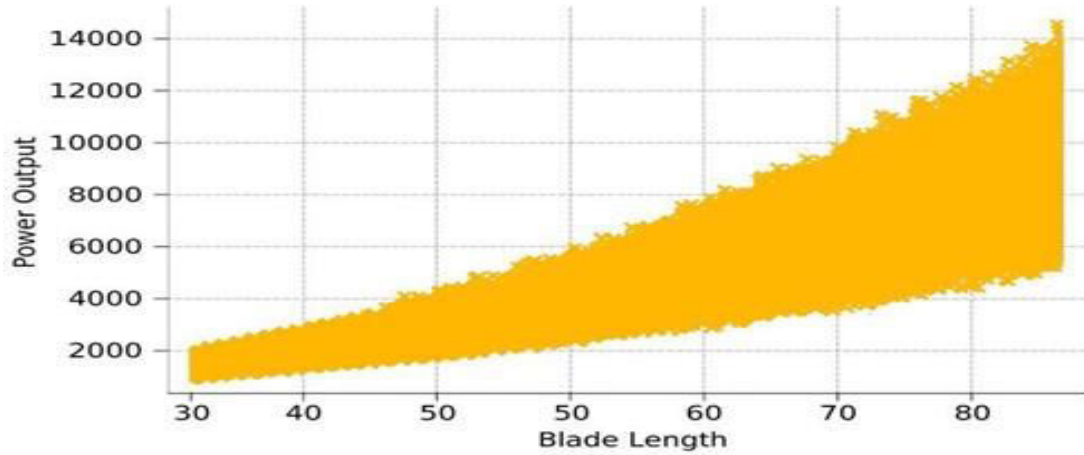


Figure. 7.1 Predicted vs Actual scatter with 45° line.

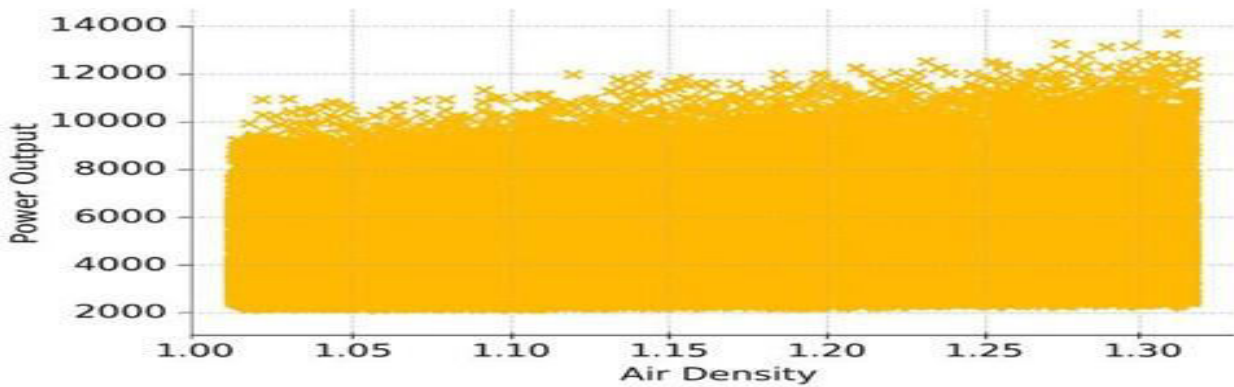


Figure 7.2 Residual histogram (mean ≈ 0).

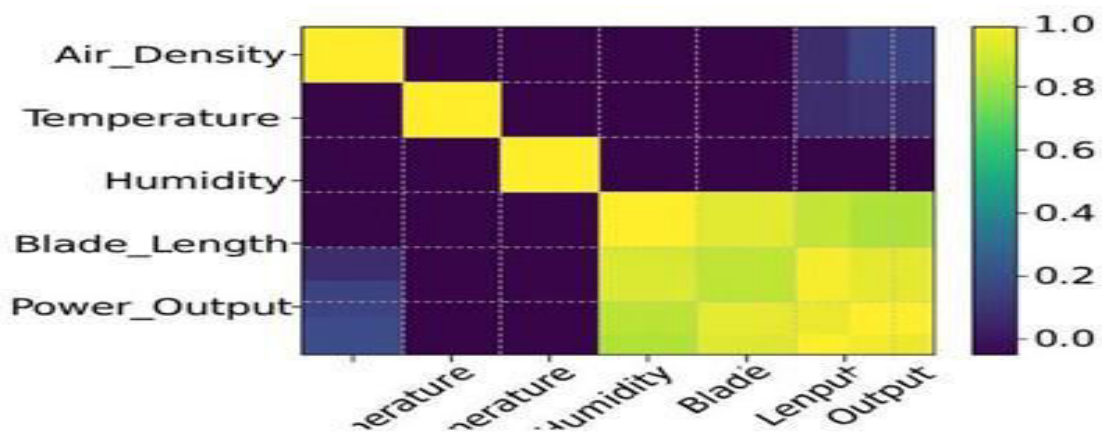


Figure.7.3 Feature importance bar chart

### 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.

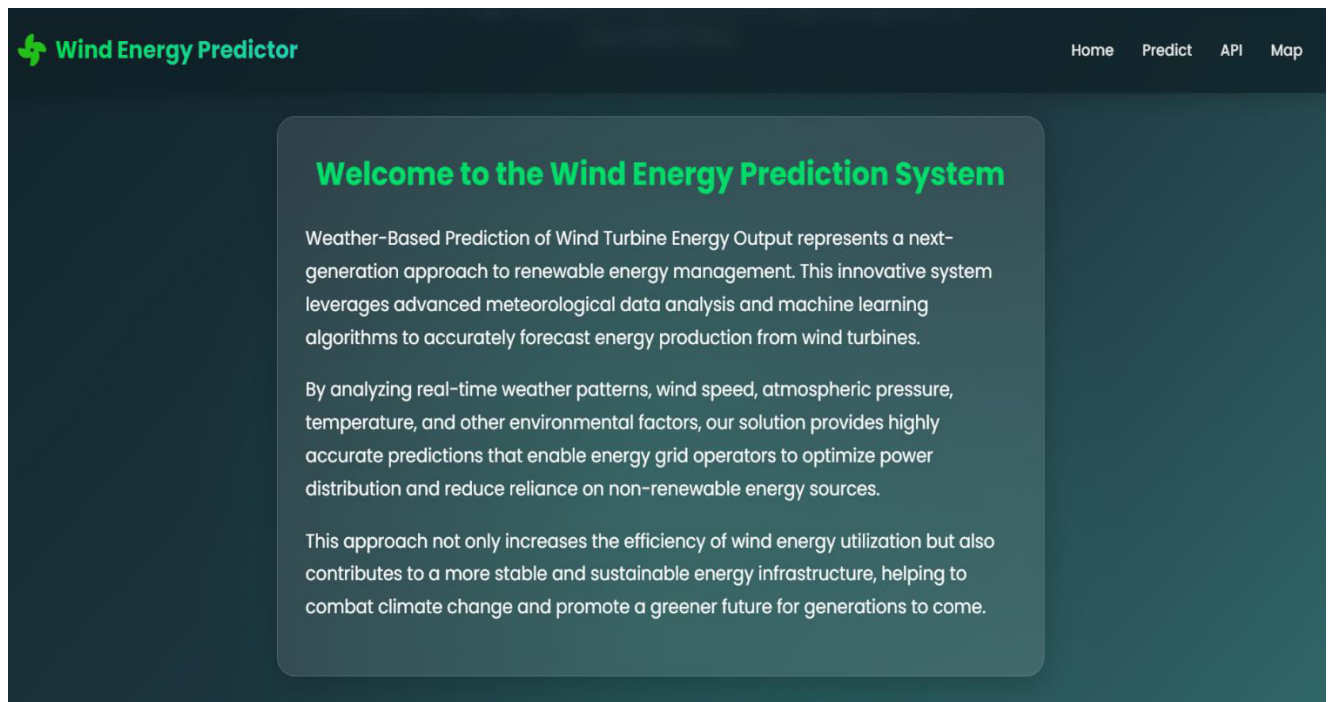


Figure (2a): Questionnaire interface



Fig 2b: Text Input Interface

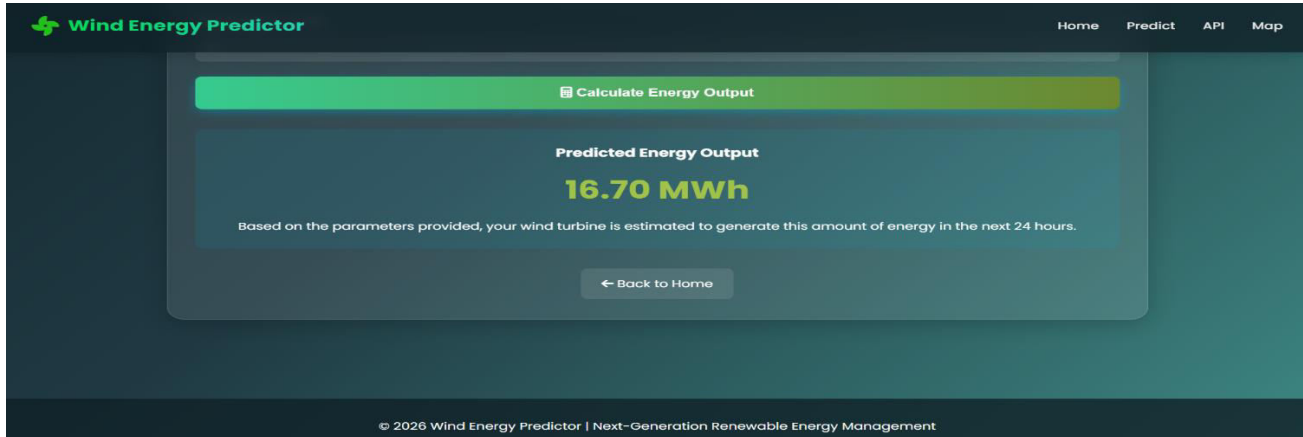


Fig 2c: Calculate energy Output

## 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.

## REFERENCES

1. J. A. Carta, P. Ramírez, and S. Velázquez, "A review of wind speed probability distributions used in wind energy analysis: Case studies 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