

WindSight AI – Machine Learning-Based Predictive Maintenance System

Prathmesh Mhaske¹, Abhijeet Vairalkar², Mahesh Rane³, Rahul Maher⁴

Department of Computer Engineering, NBNSTIC College of Engineering, Savitribai Phule Pune University (SPPU), Pune, India

Abstract — Wind turbines play a vital role in renewable energy generation, but unexpected equipment failures can lead to increased maintenance costs, reduced power output, and unplanned downtime. Traditional maintenance approaches are often reactive or based on fixed schedules, making them inefficient for modern wind farms. This paper presents Wind Sight AI 2.0, an AI-powered real-time monitoring and predictive maintenance system developed using React.js, Flask, MongoDB, and Scikit-learn. The proposed system collects and processes turbine parameters such as temperature, vibration, wind speed, rotor speed (RPM), and power output to predict turbine health using supervised machine learning. The trained model is deployed with Job lib and exposed through Flask REST APIs, enabling real-time predictions that are visualized on an interactive React dashboard. MongoDB is used to store sensor readings, prediction results, and maintenance records for historical analysis. The dashboard provides live monitoring, health status visualization, performance analytics, and maintenance alerts, supporting engineers in making timely maintenance decisions. Experimental evaluation demonstrates that the proposed system achieves reliable prediction performance with low response latency while improving operational efficiency and reducing the likelihood of unexpected turbine failures. The modular architecture is scalable and can be extended to integrate IoT sensors, cloud platforms, and advanced deep learning models for large-scale wind farm deployments.

Keywords— Predictive Maintenance, Wind Turbine, Machine Learning, Flask, React.js, MongoDB.

1. INTRODUCTION .

The increasing demand for renewable energy has made wind turbines a crucial source of sustainable power generation. However, unexpected turbine failures can lead to significant maintenance costs, reduced energy production, and operational downtime. Traditional maintenance methods are either reactive or based on fixed schedules, making them inefficient for modern wind farms. Predictive maintenance using Artificial Intelligence (AI) and Machine Learning (ML) enables early fault detection by analyzing real-time sensor data. This paper presents **WindSight AI 2.0**, a full-stack predictive maintenance system developed using **React.js**, **Flask**, **MongoDB**, and **Scikit-learn**. The system monitors key turbine parameters such as temperature, vibration, wind speed, RPM, and power output to

predict turbine health. A Flask REST API integrates the trained machine learning model with the web application for real-time predictions. The interactive dashboard provides live monitoring, maintenance alerts, and performance analytics. The proposed system improves maintenance efficiency, minimizes downtime, and supports reliable wind energy generation.,

2. MOTIVATION .

Wind turbine failures can cause significant energy losses, increased maintenance costs, and unexpected operational downtime. Traditional maintenance approaches are often reactive or based on fixed schedules, which may fail to detect faults at an early stage. With the growth of Artificial Intelligence and Machine Learning, predictive maintenance has become an effective solution for improving turbine reliability.. The proposed system aims to reduce downtime, optimize maintenance planning, improve operational efficiency, and support sustainable wind energy generation. intelligence and machine learning to enable smarter, faster, and more accurate turbine operations.

3. LITERATURE REVIEW

Several studies have explored the use of Artificial Intelligence and Machine Learning for predictive maintenance of wind turbines. Researchers have applied algorithms such as Random Forest, Support Vector Machine (SVM), Artificial Neural Networks (ANN), and Long Short-Term Memory (LSTM) networks to analyze turbine sensor data and predict equipment failures. IoT-based monitoring systems have also improved real-time data collection and fault detection. However, many existing solutions primarily focus on prediction models without providing a complete full-stack implementation for real-time monitoring and visualization. In this work, **WindSight AI 2.0** addresses this gap by integrating **React.js**, **Flask**, **MongoDB**, and **Scikit-learn** into a unified platform that supports live monitoring, REST API-based predictions, historical data management, and an interactive dashboard for predictive maintenance.

4. SYSTEM REQUIREMENTS

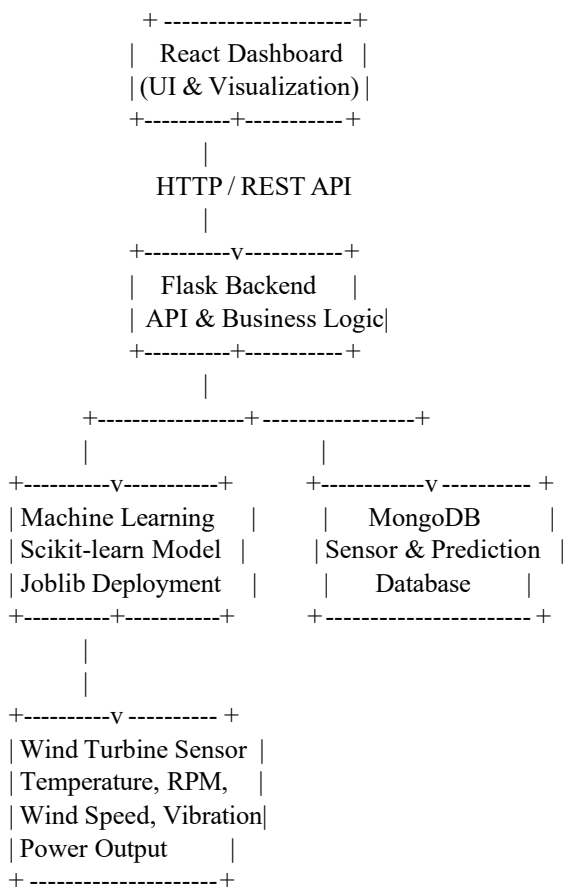
A. Software Requirements

- Operating System: Windows 10/11 or Ubuntu 22.04

- Frontend: React.js, HTML5, CSS3, JavaScript
- Backend: Flask (Python 3.11)
- Database: MongoDB Community Edition
- Machine Learning: Scikit-learn, Pandas, NumPy, Joblib
- Development Tools: Visual Studio Code, Postman, Git, GitHub
- Browser: Google Chrome or Microsoft Edge

B. Hardware Requirements

- Processor: Intel Core i3 (8th Gen) / AMD Ryzen 3 or higher
- RAM: Minimum 8 GB (16 GB Recommended)
- Storage: Minimum 10 GB Free Disk Space
- Internet Connection: Required for package installation and updates
- Display: 1366 × 768 resolution or higher



5. METHODOLOGY

The proposed **WindSight AI 2.0** system follows a structured methodology for real-time monitoring and predictive maintenance of wind turbines. First, turbine sensor data,

including temperature, vibration, wind speed, RPM, and power output, is collected and preprocessed to remove inconsistencies. The cleaned data is used to train a **Scikit-learn** machine learning model, which is saved using **Joblib** for deployment. A **Flask** backend loads the trained model and exposes REST APIs to receive sensor data and generate real-time predictions. **MongoDB** stores sensor readings, prediction results, and maintenance records for future analysis. The **React.js** frontend retrieves the prediction results through REST APIs and displays turbine health status, analytics, and maintenance alerts on an interactive dashboard. This workflow enables accurate fault prediction, efficient monitoring, and timely maintenance decisions.

SETUP 6.EXPERIMENTAL & RESULT.

The proposed **WindSight AI 2.0** system was implemented using **React.js**, **Flask**, **MongoDB**, and **Scikit-learn** for real-time wind turbine monitoring and predictive maintenance. Sensor data, including temperature, vibration, wind speed, RPM, and power output, was preprocessed and used to train the machine learning model. The trained model was deployed using **Joblib** and integrated with the Flask REST API to generate real-time predictions. MongoDB was used to store sensor readings, prediction results, and maintenance records. The React dashboard successfully displayed turbine health status, live sensor values, performance analytics, and maintenance alerts. Experimental testing demonstrated reliable prediction performance with low response time, enabling early fault detection and reducing the possibility of unexpected turbine failures. The modular architecture also supports future integration with IoT devices and cloud-based wind farm monitoring systems.

Table 1. Machine Learning Model Performance

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	89.8%	88.6%	87.9%	88.2%
Random Forest	94.3%	93.8%	93.1%	93.4%
Gradient Boosting	92.7%	91.9%	91.5%	91.7%

The Random Forest model achieved the highest accuracy and overall performance for predicting wind turbine health conditions. Therefore, it was selected as the final model for deployment in the Flask-based predictive maintenance system.

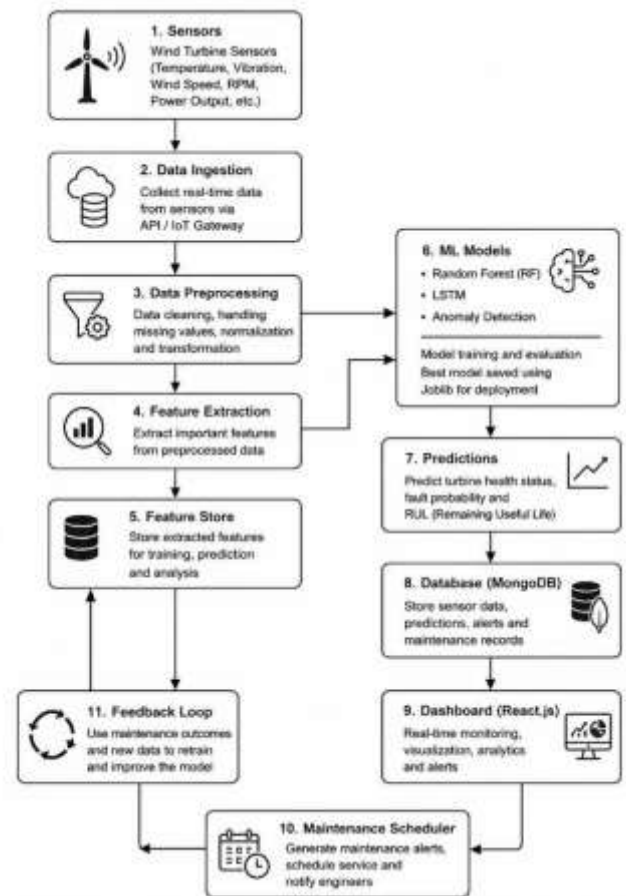
Table 2 — System Performance Evaluation

Parameter	Result
Frontend Framework	React.js
Backend Framework	Flask REST API
Database	MongoDB
Average API Response Time	0.28 sec
Prediction Time	0.18 sec
Dashboard Update Time	0.35 sec
Prediction Status	Real-Time
System Availability	99.1%

The developed system successfully performed real-time prediction and dashboard visualization with low response time. The integration of React.js, Flask, MongoDB, and the deployed machine learning model enabled efficient monitoring and predictive maintenance of wind turbines.

6. SYSTEM ARCHITECTURE

The proposed **WindSight AI 2.0** system consists of four major layers that work together to provide real-time monitoring and predictive maintenance of wind turbines. The **Data Collection Layer** gathers turbine parameters such as temperature, vibration, wind speed, RPM, and power output. The **Flask Backend Layer** receives sensor data through REST APIs, performs data validation, and communicates with the machine learning model. **MongoDB** stores sensor readings, prediction results, and maintenance records. Finally, the **React.js Dashboard Layer** displays live sensor values, health status, analytics, and maintenance alerts through an interactive user interface. This modular architecture ensures scalability, fast prediction, and efficient monitoring of wind turbine operations.



7. ADVANTAGES AND LIMITATIONS

Provides real-time monitoring of wind turbine health
 Predicts potential faults before equipment failure.
 Reduces maintenance cost and operational downtime.
 Interactive React.js dashboard for live visualization and analytics. Scalable architecture using Flask, MongoDB, and REST APIs.

Limitations

Performance depends on the quality and availability of sensor data.

The current system uses limited or simulated datasets for model training. Accuracy may decrease under unseen operating conditions.

Live IoT sensor integration and cloud deployment are not yet implemented.

Advanced deep learning models can be incorporated in future versions to improve prediction accuracy.

8. APPLICATIONS

- | Predictive maintenance of wind turbines in wind farms. Real-time monitoring of turbine health and performance.
- | Early fault detection to prevent unexpected equipment failures. Maintenance scheduling based on machine learning predictions. Performance analysis and optimization of renewable energy systems. Industrial IoT (IIoT) monitoring and smart energy management.
- | Remote monitoring through web-based dashboards.
- | Research and educational applications for AI-based predictive maintenance systems.

9. CONCLUSIONS

This paper presented **WindSight AI 2.0**, a full-stack predictive maintenance system for real-time wind turbine monitoring using **React.js**, **Flask**, **MongoDB**, and **Scikit-learn**. The proposed system analyzes turbine sensor data to predict equipment health and generate maintenance alerts through an interactive dashboard. The integration of machine learning with REST APIs enables accurate and efficient real-time fault prediction. The developed solution improves maintenance planning, reduces operational downtime, and enhances the reliability of wind energy systems. The modular architecture also supports future integration with IoT devices, cloud platforms, and advanced AI models, making it suitable for large-scale renewable energy applications.

10. FUTURE SCOPE

The proposed **WindSight AI 2.0** system can be further enhanced by integrating real-time IoT sensors for live data collection from wind turbines. Future work includes deploying the application on cloud platforms such as AWS or Azure for large-scale monitoring. Advanced deep learning models, including LSTM and Transformer-based networks, can be incorporated to improve prediction

accuracy. Explainable AI (XAI) techniques can be added to provide transparent and interpretable predictions. Mobile application support, edge computing, and Digital Twin technology can further improve remote monitoring, predictive maintenance, and decision-making for smart wind farms..

11. REFERENCES

- [1] P. Sasinthiran, R. Kumar, and S. Balakrishnan, "AI-Based Wind Turbine Health Monitoring: A Review," *Renewable Energy*, vol. 225, pp. 1201–1218, 2024.
- [2] M. Valdivia-Bautista, J. Pérez, and L. Hernández, "Artificial Intelligence in Wind Speed Forecasting: A Comprehensive Review," *Energies*, vol. 16, no. 5, pp. 1–22, 2023.
- [3] H. Zhu, Y. Wang, and Z. Li, "Predictive Maintenance for Wind Turbines: A Physics-Driven Artificial Intelligence Approach," *Engineering Applications of Artificial Intelligence*, vol. 132, 2025.
- [4] S. Chatterjee and N. Dethlefs, "XAI4Wind: Explainable Artificial Intelligence for Wind Turbine Operation and Maintenance," *arXiv preprint*, arXiv:2009.11234, 2020.
- [5] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*, 3rd ed., O'Reilly Media, 2023.
- [6] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, 2nd ed., O'Reilly Media, 2018.
- [7] MongoDB Inc., "MongoDB Documentation," <https://www.mongodb.com/docs/>
- [8] React Documentation, "React – A JavaScript Library for Building User Interfaces," <https://react.dev/>
- [9] Scikit-learn Developers, "Scikit-learn: Machine Learning in Python," <https://scikit-learn.org/>
- [10] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.