# Real-Time Optimization and Maintenance of Wind Turbine Performance Using Digital Twin Technology

# Meet Our Team !

Dilmini N.A.C.

IT21836954

Jenojan P.

IT 21355714

Dhanushikan V.

IT18149890

Herath H.M.T.S

IT19208572

# INTRODUCTION

## Real-Time Optimization and Maintenance of Wind Turbine Performance Using Digital Twin Technology

- Wind energy is one of the most rapidly growing renewable energy sources
- reducing greenhouse gas emissions, providing a clean, sustainable source of power.
- Mainitaning turbines efficiently is a challenge due to their size, complexity, and remote locations.
- A **Digital Twin** is a virtual replica of a physical asset, system, or process.

# RESEARCH QUESTION

**Weather Impact Analysis**
How can we use real-time and forecasted weather data to detect risky wind turbine conditions early and recommend suitable control actions to reduce those risks?

**Noise Effect Investigation**
How can a digital twin with machine learning optimize wind turbine noise and performance in real time?

**Maintenance Optimization**
Predict and plan maintenance efficiently.

**Power Efficiency Improvement**
How can a Digital Twin system improve the **performance monitoring, energy forecasting, and power optimization** of a wind turbine using real-time data and machine learning?

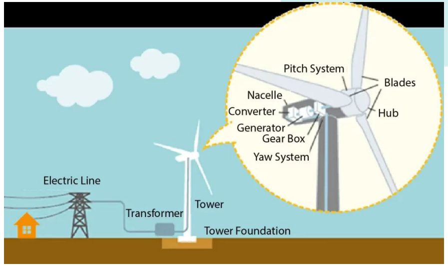
# OBJECTIVES

## Main Objective

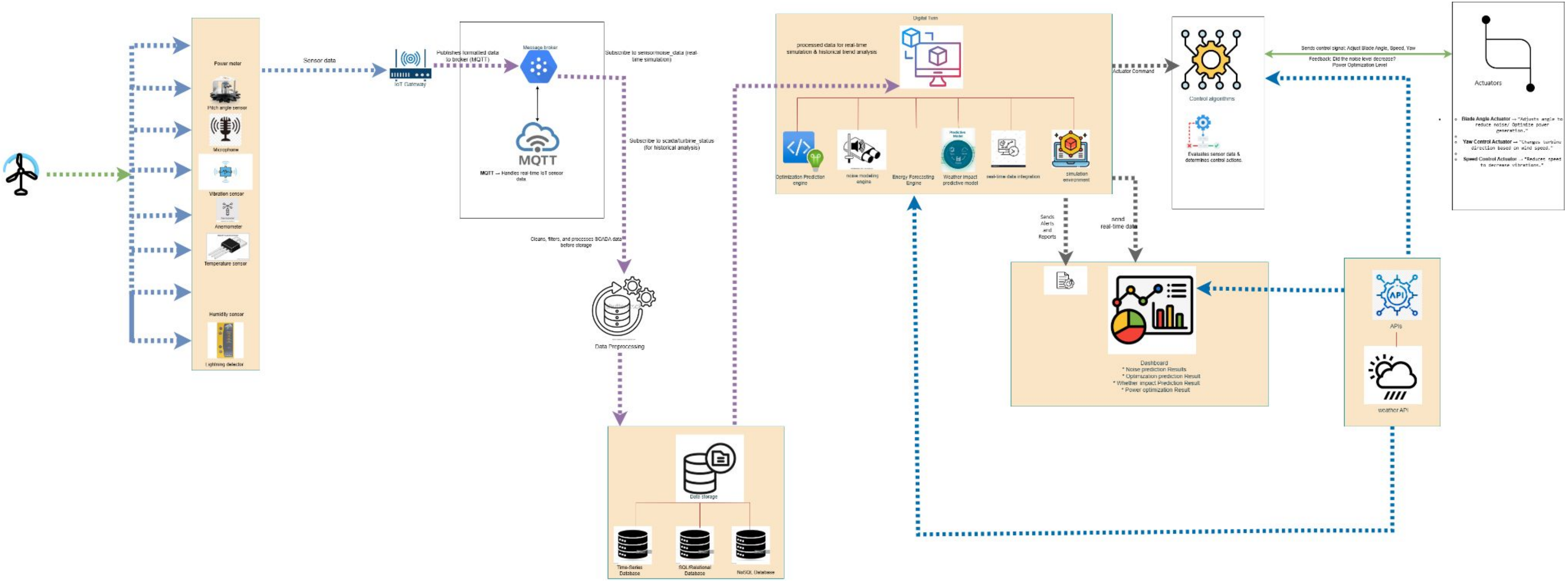
Real-Time Optimization and Maintenance of Wind Turbine Performance Using Digital Twin Technology.

## Sub Objectives

1. Analyzing weather impacts on turbines.
2. Enhancing power generation efficiency.
3. Investigating noise effects in wind turbines.
4. Optimizing wind turbine maintenance strategies.



# SYSTEM ARCHITECTURE DIAGRAM



IT21836954 | Dilmini N.A.C

Specialization – Software Engineering

INTRODUCTION

# Background

- Wind power is a clean and harmless source of renewable energy.

- Wind turbines can be damaged by strong wind storms, heavy rain, and lightning.

- In March 2025, a Vineyard wind turbine with blade failure was later destroyed by a lightning strike.

- There was no proper digital twin system to detect risks early and take safety precautions.

- This leads to physical damage, turbine downtime, and increased maintenance costs.

INTRODUCTION

# Research Question

How can we detect risky wind turbine conditions early using real-time and forecasted weather data?

What control actions can be taken to reduce those risks based on weather analysis?

## INTRODUCTION

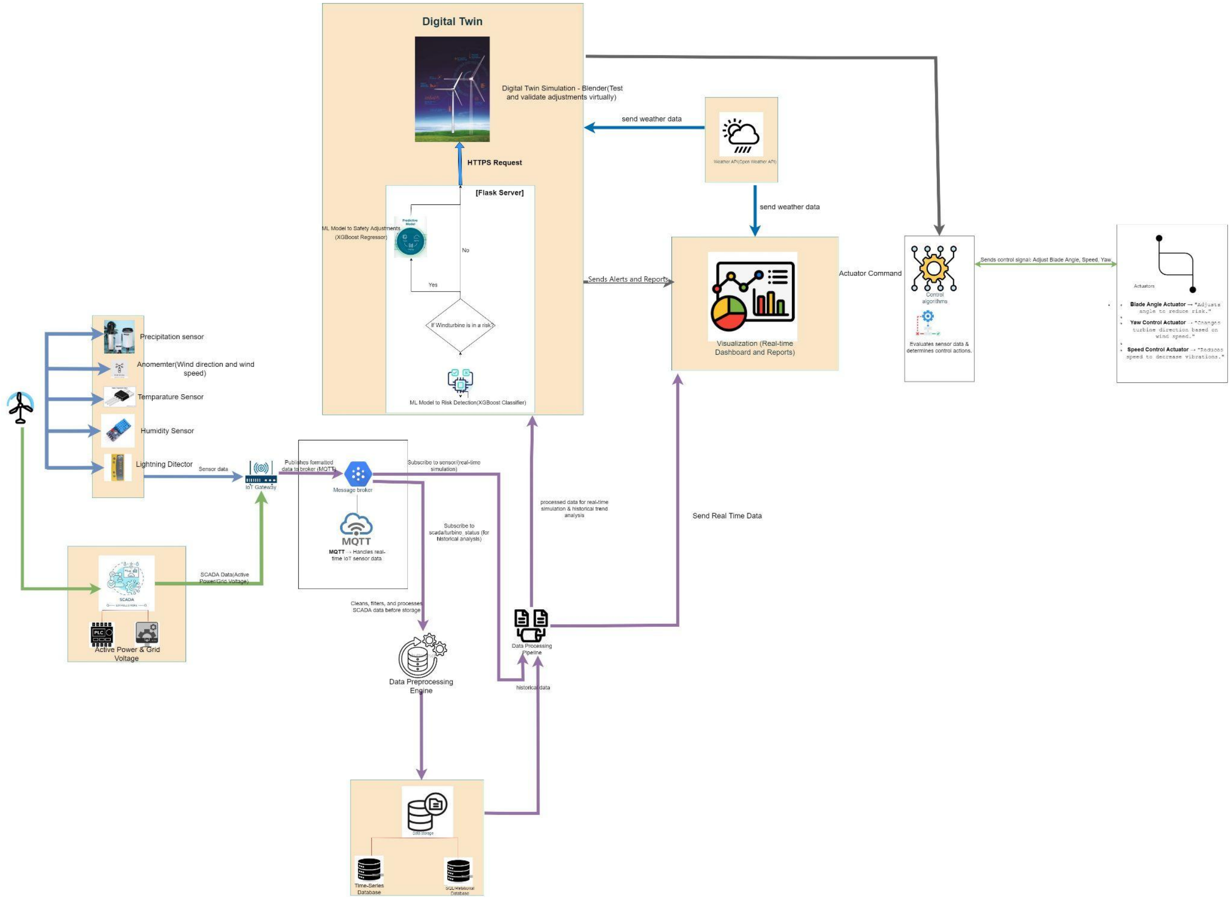
# Specific and Sub Objective

Analysing weather impact on wind turbine using digital twin technology

1. Identify the most suitable machine learning algorithm and train the model for detecting risky operating conditions in wind turbines using weather and turbine data and train the model
2. Identify the best machine learning algorithm and train model for predicting appropriate control actions under identified risk conditions
3. Integrate the trained models into a Flask server to provide real-time and forecast-based predictions through a web interface.
4. Develop a digital twin model using Blender, and integrate it with the Flask server to virtually simulate and test the predicted control adjustments

# Methodology

# System Diagram



# Used Techniques and Technologies

## Techniques:

- Data Preprocessing
- Supervised Machine Learning
- Model Evaluation
- Hyperparameter Tuning

## Technologies:

- Python
- Jupyter Notebook
- Pandas, NumPy, Matplotlib, Seaborn
- Scikit-learn, XGBoost
- Anaconda
- OpenWeatherMap API
- Flask API

Methodology

# Evidence of Completion

- Data Collection
- Data Preprocessing
- Model Selection
- Model Training & Hyperparameter Tuning
- Model Evaluation
- Display Results Using Web UI

# Data Collection & Pre-Processing

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ActivePow | AirPress1 | EGDInRotc | EGDYawPc | Hum1 | PitchBlade | PitchBlade | Precipitati | Temp1 | WD187m | Windspeec | risk_label | TurbineOperat |
| 2 | 343.7458 | 80.63289 | 14.9159 | 113 | 35.41194 | -1.91799 | -1.83665 | 0.003661 | 1.153391 | 182.9935 | 7.108178 | 0 | 1 |
| 3 | 310.6304 | 80.87608 | 14.5667 | 113 | 35.7648 | -1.91799 | -1.83757 | 0.002808 | 1.526489 | 182.9944 | 6.92009 | 0 | 1 |
| 4 | 314.0022 | 80.62978 | 14.7018 | 113 | 35.62122 | -1.91749 | -1.83494 | 0.00347 | 1.392387 | 182.9939 | 6.443538 | 0 | 1 |
| 5 | 322.9313 | 80.64644 | 14.8452 | 113 | 35.95006 | -1.9185 | -1.83551 | 0.003616 | 1.849462 | 182.9971 | 6.344331 | 0 | 1 |
| 6 | 330.4503 | 80.75833 | 14.8459 | 113 | 35.76834 | -1.9185 | -1.82712 | 0.003636 | 1.545831 | 183.006 | 6.217314 | 0 | 1 |
| 7 | 311.1798 | 80.74252 | 14.5113 | 113 | 36.00096 | -1.91749 | -1.84238 | 0.00378 | 1.842736 | 182.996 | 6.014656 | 0 | 1 |
| 8 | 298.8521 | 81.02656 | 14.3372 | 113 | 36.20271 | -1.919 | -1.84275 | 0.003927 | 2.095228 | 182.9936 | 6.486429 | 0 | 1 |
| 9 | 273.7643 | 80.95069 | 13.8969 | 113 | 36.19559 | -1.91799 | -1.83518 | 0.003579 | 2.009316 | 182.993 | 6.551162 | 0 | 1 |
| 10 | 266.2126 | 80.6739 | 13.9025 | 113 | 35.3634 | -1.9185 | -1.83226 | 0.003165 | 0.761545 | 183.0008 | 6.139479 | 0 | 1 |
| 11 | 283.4081 | 81.08177 | 14.3536 | 113 | 36.07639 | -1.91799 | -1.83614 | 0.003902 | 1.816432 | 182.9956 | 6.252147 | 0 | 1 |
| 12 | 314.6266 | 80.76262 | 14.6752 | 113 | 35.85959 | -1.91698 | -1.83409 | 0.003101 | 1.638364 | 182.9912 | 6.23578 | 0 | 1 |
| 13 | 295.7377 | 80.92244 | 14.2362 | 113 | 35.7165 | -1.9185 | -1.83278 | 0.003292 | 1.400688 | 183.0044 | 6.430126 | 0 | 1 |
| 14 | 275.174 | 80.95329 | 13.971 | 113 | 35.96233 | -1.9185 | -1.8368 | 0.003326 | 1.962328 | 182.999 | 5.995057 | 0 | 1 |
| 15 | 258.7599 | 80.86171 | 13.5802 | 113 | 36.19309 | -1.91799 | -1.83673 | 0.003723 | 1.986448 | 182.999 | 6.143929 | 0 | 1 |
| 16 | 220.6878 | 80.8397 | 12.7783 | 113 | 35.82029 | -1.9185 | -1.83839 | 0.003368 | 1.464101 | 182.9949 | 5.842479 | 0 | 1 |
| 17 | 185.1244 | 80.80806 | 12.1388 | 113 | 36.0962 | -1.92153 | -1.84 | 0.003316 | 1.902195 | 183.0011 | 6.004678 | 0 | 1 |
| 18 | 167.6885 | 80.6598 | 11.8568 | 113 | 35.85323 | -1.92002 | -1.84423 | 0.003573 | 1.354867 | 182.9921 | 5.120638 | 0 | 1 |
| 19 | 164.9725 | 80.80104 | 11.9024 | 113 | 35.8271 | -1.92052 | -1.84252 | 0.003615 | 1.584449 | 182.9984 | 5.803294 | 0 | 1 |
| 20 | 166.1365 | 80.73285 | 11.8493 | 113 | 36.02318 | -1.91951 | -1.84555 | 0.003402 | 1.829016 | 182.9942 | 5.303623 | 0 | 1 |
| 21 | 161.224 | 80.88924 | 11.7037 | 113 | 36.08129 | -1.92002 | -1.84439 | 0.003163 | 2.01007 | 183.0033 | 5.223141 | 0 | 1 |
| 22 | 158.0153 | 80.88007 | 11.6583 | 113 | 36.12362 | -1.92052 | -1.83835 | 0.003567 | 1.947171 | 183.0023 | 5.821279 | 0 | 1 |
| 23 | 159.2959 | 80.65858 | 11.6472 | 113 | 36.10008 | -1.92103 | -1.84441 | 0.003531 | 1.889914 | 182.997 | 6.010152 | 0 | 1 |
| 24 | 133.7037 | 80.6557 | 11.1968 | 113 | 35.86815 | -1.92153 | -1.84465 | 0.004103 | 1.588836 | 182.9952 | 5.526567 | 0 | 1 |
| 25 | 118.1784 | 80.47148 | 11.1512 | 113 | 35.80004 | -1.92204 | -1.83824 | 0.003051 | 1.317287 | 182.9989 | 5.868557 | 0 | 1 |
| 26 | 161.7001 | 80.95877 | 11.3183 | 113 | 35.80097 | -1.92103 | -1.84181 | 0.003819 | 1.063603 | 183.002 | 6.376671 | 0 | 1 |
| 27 | 180.8578 | 80.21044 | 11.4972 | 113 | 35.54414 | -1.92052 | -1.84313 | 0.003488 | 1.022536 | 182.9971 | 6.361147 | 0 | 1 |

- **Data Set**

[1]:
```python
import pandas as pd

df = pd.read_csv("turbine_weather_filtered.csv")

df.head()
```

[1]:

| | ActivePower | AirPress1 | EGDInRotorSpd | EGDOpCtlTurbineStatus | EGDYawPositionToNorth | Hum1 | PitchBlade1 | PitchBlade2 | Precipitation | Temp1 | TurbineOpe |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 343.745848 | 80.632891 | 14.9159 | 2.0 | 113.0 | 35.411942 | -1.917993 | -1.836653 | 0.003661 | 1.153391 | |
| 1 | 310.630445 | 80.876080 | 14.5667 | 2.0 | 113.0 | 35.764805 | -1.917993 | -1.837575 | 0.002808 | 1.526489 | |
| 2 | 314.002250 | 80.629776 | 14.7018 | 2.0 | 113.0 | 35.621216 | -1.917487 | -1.834939 | 0.003470 | 1.392387 | |
| 3 | 322.931256 | 80.646442 | 14.8452 | 2.0 | 113.0 | 35.950058 | -1.918499 | -1.835506 | 0.003616 | 1.849462 | |
| 4 | 330.450311 | 80.758325 | 14.8459 | 2.0 | 113.0 | 35.768344 | -1.918499 | -1.827118 | 0.003636 | 1.545831 | |

[3]:
```python
df.isnull().sum()

df.dropna(inplace=True)

df.shape
```

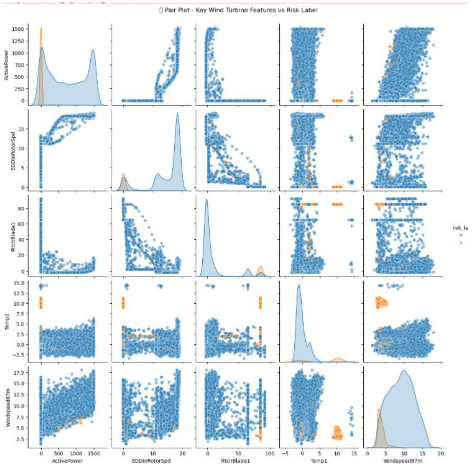
[3]: (14776, 15)

[5]:
```python
df['risk_label'] = df['EGDOpCtlTurbineStatus'].apply(lambda x: 1 if x == 7 else 0)
df['risk_label'].value_counts()
```

[5]:
```
risk_label
0    13504
1     1272
Name: count, dtype: int64
```

[7]:
```python
feature_cols = [
    'ActivePower', 'AirPress1', 'EGDInRotorSpd', 'EGDYawPositionToNorth',
    'Hum1', 'PitchBlade1', 'PitchBlade2', 'Precipitation',
    'Temp1', 'WD187m', 'Windspeed87m'
]
```

- **Data pre-processing implemetation**




- **Pair Plot - Key Wind Turbine Features vs Risk Label**

# ML Algorithms to identify risks in Wind Turbines

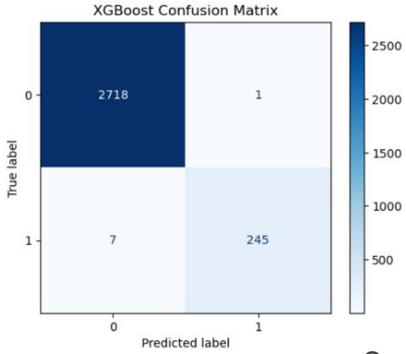
## Evidence of Completion

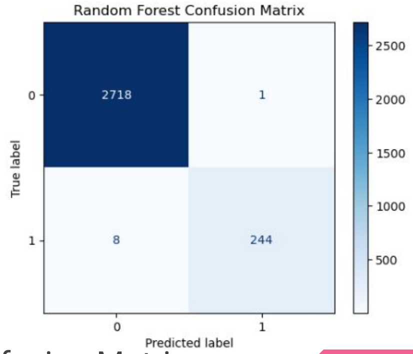
**Supervised Learning Algorithms**

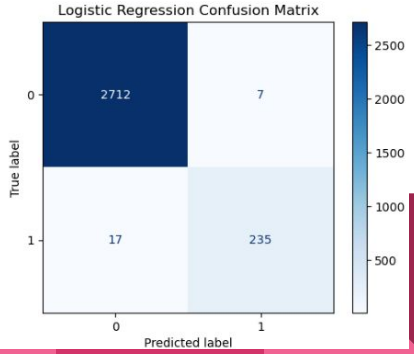
- **XGBoost Classifier Algorithm** ✓
- **RandomForest Classifier Algorithm**
- **Logistic Regression Algorithm**

| Metric | XGBoost | Random Forest | Logistic Regression |
|---|---|---|---|
| Accuracy | 0.9973 | 0.997 | 0.9919 |
| F1 Score | 0.9839 | 0.9819 | 0.9514 |
| Precision | 0.9959 | 0.9959 | 0.9711 |
| Recall | 0.9722 | 0.9683 | 0.9325 |
| ROC-AUC | 0.9999 | 0.9977 | 0.9918 |

- Classification Report



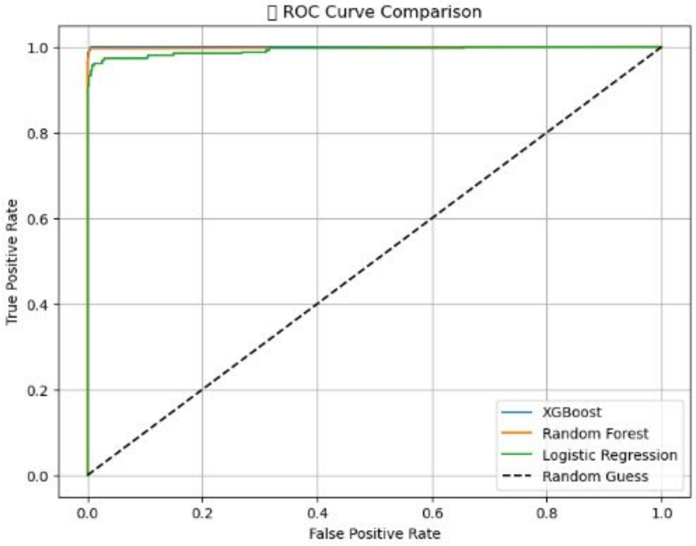



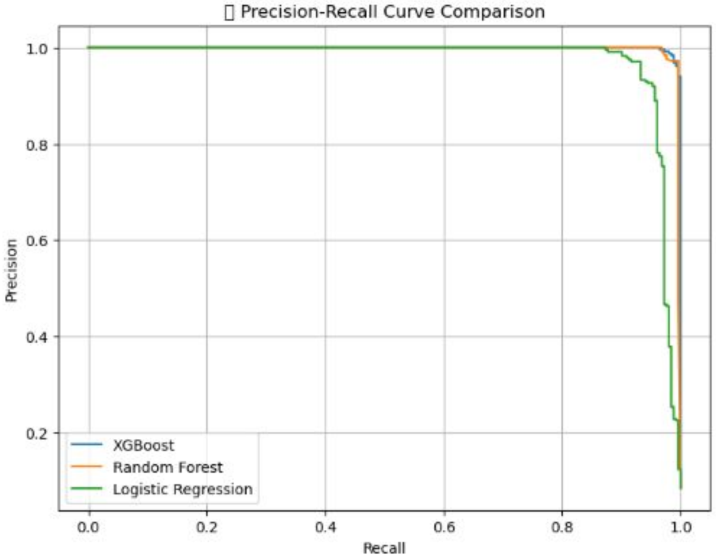

- Confusion Matrix

# ML Algorithms to identify risks in Wind Turbines

## Evidence of Completion




- ROC Curve Comparison




- Precision-Recall Curve Comparison

# ML Algorithms to Recommend Optimal Control Actions During Risky Conditions

## Evidence of Completion

**Supervised Learning Algorithms**

- **XGBoost Regressor Algorithm** ✓
- **RandomForest Regressor Algorithm**
- **Linear Regression Algorithm**

Rotor Speed Prediction :

| Metric | XGBoost Regressor | Random Forest Regressor | Linear Regression |
|---|---|---|---|
| $R^2$ Score | 0.819 | 0.8048 | 0.5763 |
| MAE | 0.6454 | 0.66 | 1.3713 |
| RMSE | 1.1051 | 1.1476 | 1.6907 |

Yaw Angle Prediction :

| Metric | XGBoost Regressor | Random Forest Regressor | Linear Regression |
|---|---|---|---|
| $R^2$ Score | 0.8578 | 0.8803 | 0.0927 |
| MAE | 3.9698 | 3.2818 | 11.8325 |
| RMSE | 5.9163 | 5.4279 | 14.9425 |

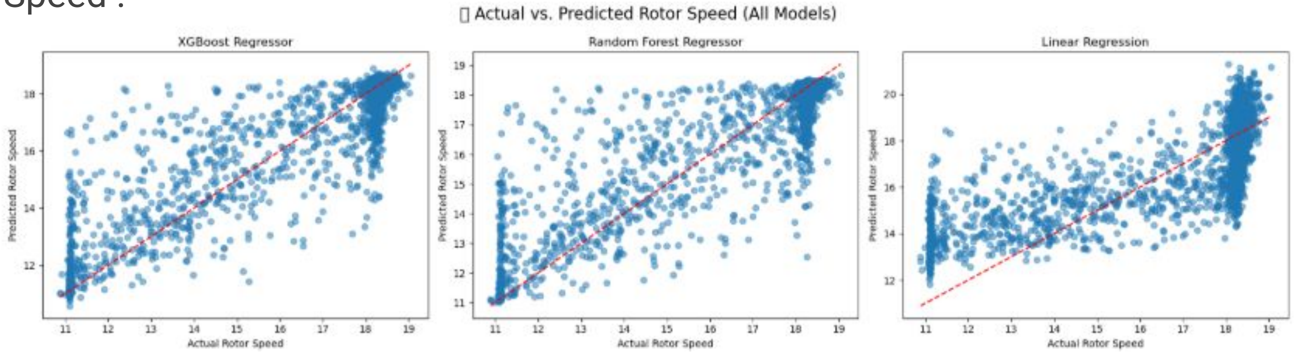
Pitch Angle Prediction :

| Metric | XGBoost Regressor | Random Forest Regressor | Linear Regression |
|---|---|---|---|
| $R^2$ Score | 0.6391 | 0.614 | 0.4187 |
| MAE | 1.3432 | 1.3541 | 2.2597 |
| RMSE | 2.3694 | 2.4505 | 3.0069 |

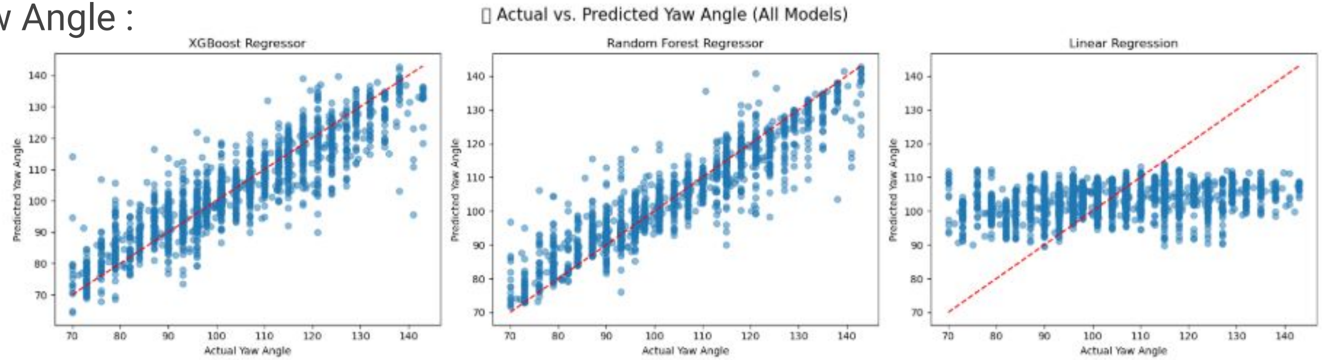
# ML Algorithms to Recommend Optimal Control Actions During Risky Conditions

## Evidence of Completion

Rotor Speed :

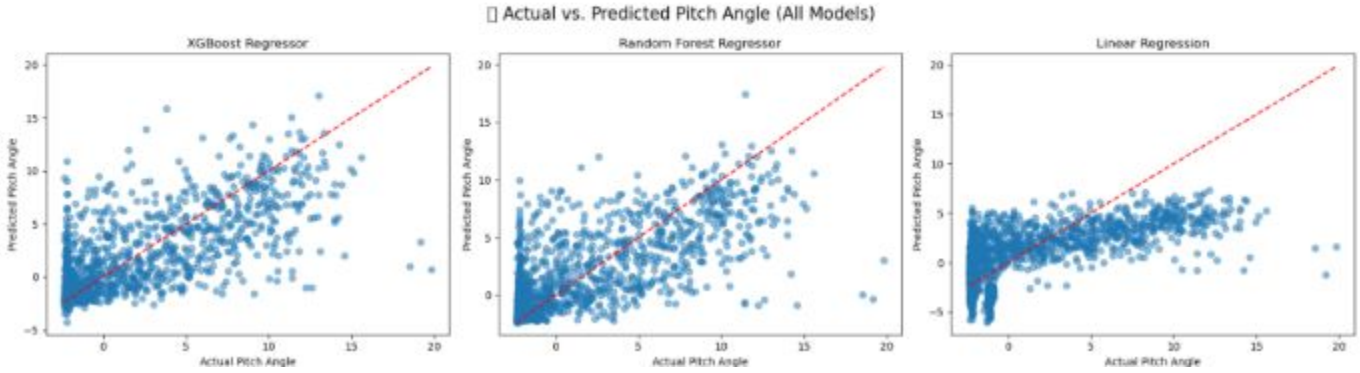



Yaw Angle :



# ML Algorithms to Recommend Optimal Control Actions During Risky Conditions

## Evidence of Completion



# Functional, Non-Functional and Personal Requirements

## Functional Requirements

- Detect risky turbine conditions using weather and turbine data
- Predict future risks using forecast data
- Suggest safety actions like adjusting yaw, pitch, or rotor speed
- Recommend shutdown if risk is too high
- Allow user input and show results clearly

## Non-Functional Requirements

- Should give accurate and reliable results
- Respond within a few seconds
- Easy to use interface
- Should work well with different data inputs
- Easy to update and expand in the future

## Personal Requirements

- Ceylon Electricity Board
- Thambapawani Wind Farm
- Mr.Ganushka(Chief Engineer of the wind farm)

# Completion and Future works

60%

**Completion of the component :**

Identify the most suitable machine learning algorithm and train a model to detect risky operating conditions in wind turbines using weather and turbine data. (Sub Objective 01)

Select and train models to predict optimal control actions under those identified risk conditions. (Sub Objective 02)

Integrate the trained models into a Flask server to deliver real-time and forecast-based predictions through a web interface. (Sub Objective 03)

40%

**Future Implementations :**

Collect local data from the Thambapawani Wind Farm and train models using actual real-time data

Develop a digital twin model using Blender and connect it with the Flask server to simulate and validate the recommended control adjustments. (Sub Objective 04)

# References


- U.S. Department of Energy, “Wind Energy Data Hub,” [Online]. Available: https://wdh.energy.gov/data/wind-energy. [Accessed: Apr. 9, 2025].
- T. Ackermann, Wind Power in Power Systems, 2nd ed. Hoboken, NJ: Wiley, 2012.
- J. Zhang, Z. Wang, J. Chen, and Y. Huang, “Data-driven wind turbine fault detection using machine learning algorithms,” Renewable Energy, vol. 160, pp. 185–198, 2020.
- M. Ghofrani, A. Arabali, M. Etezadi-Amoli, and M. S. Fadali, “A framework for optimal placement of wind turbines using digital twin modeling,” IEEE Transactions on Sustainable Energy, vol. 9, no. 3, pp. 1265–1274, Jul. 2018.
- A. Kusiak, H. Zheng, and Z. Song, “Wind farm power prediction: A data-mining approach,” Wind Energy, vol. 12, no. 3, pp. 275–293, May 2009.

IT19208572 | Herath H.M.T.S

Specialization - Software engineering

# Introduction - Background

How can a Digital Twin system improve the **power optimization, energy forecasting, and Power curve analysis** of a wind turbine using real-time data and machine learning?

- DTs optimize power, forecast energy, and analyze curves
- Essential for precision in dynamic environments
- Power Optimization
  - Monitors turbine parameters (blade pitch, rotor speed)
  - Suggests real-time adjustments using ML models
- Energy Forecasting
  - Combines sensor data, weather forecasts, and history
  - Predicts energy output (hours, days, weeks)
- Power Curve Analysis
  - Builds dynamic power curves from real-time data
  - Spots deviations (wear, damage, icing)

# Specific and sub objective

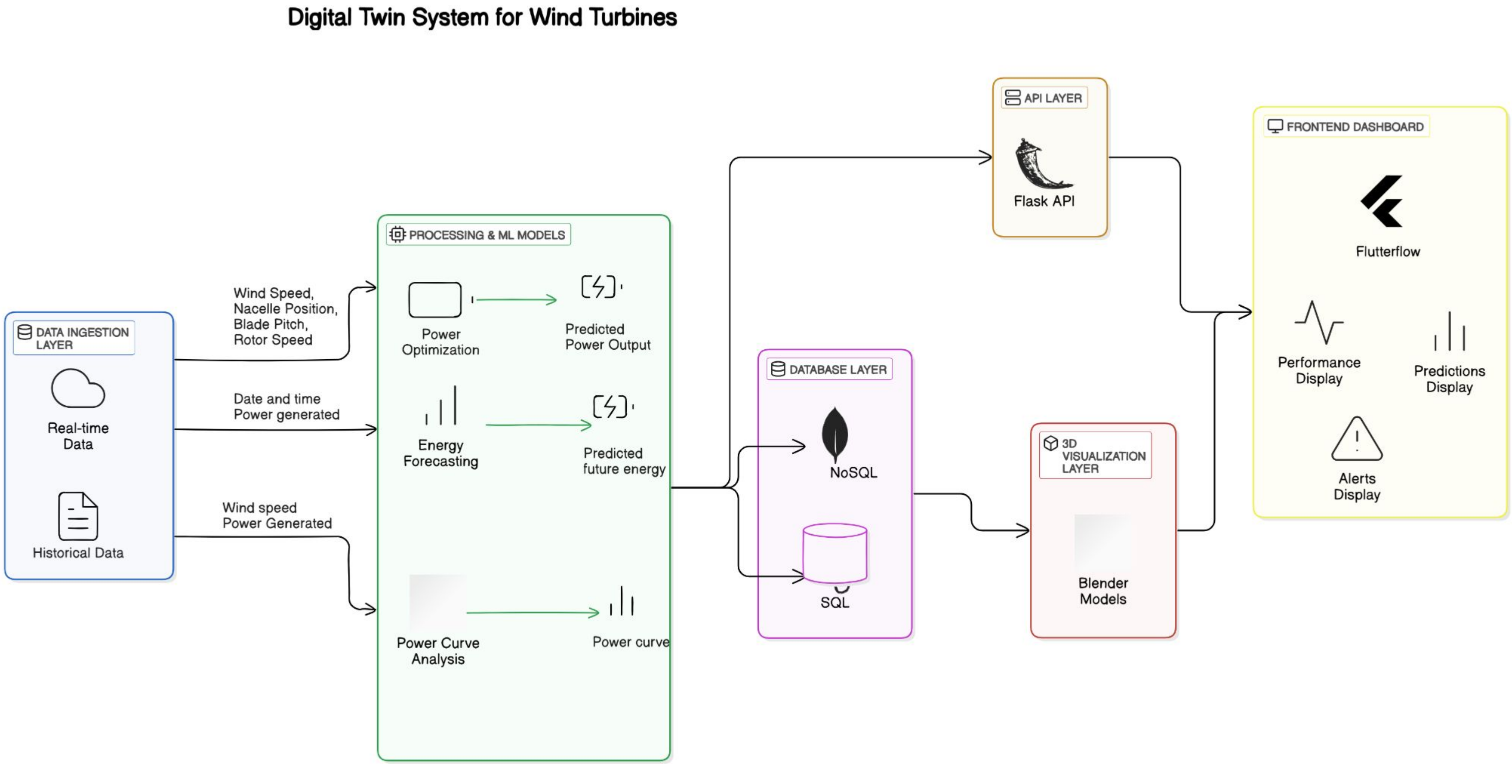
**Main Objective:**

- To develop a Digital Twin system that simulates and optimizes the behavior of a wind turbine in real-time.

**Sub Objectives:**

1. To forecast **energy output** using machine learning algorithms.
2. To analyze the **power curve** for identifying operational efficiency.
3. To optimize **blade pitch angle** and other parameters to maximize power output.
4. To expose functionality via **backend APIs** for integration with a user interface.

# System diagram



# Methodology - Tasks Completed

- Data Collection, data pre-processing  and data augmentation
- Selecting the best ML model for accurate power optimization, Power curve analysis
- Data visualization and tuning hyperparameters to create the best fitting model
- Deploy the finalized model and python algorithms to the flask server

# Data Collection and Pre- processing

```
df = pd.read_csv("wind_data1.csv")


cols = ['Wind speed (m/s)', 'Rotor speed (RPM)', 'Nacelle position ', 'Blade angle', 'Power (kW)']
df = df[cols]

# Drop missing/duplicate values
df.dropna(inplace=True)
df.drop_duplicates(inplace=True)

df.head()
```

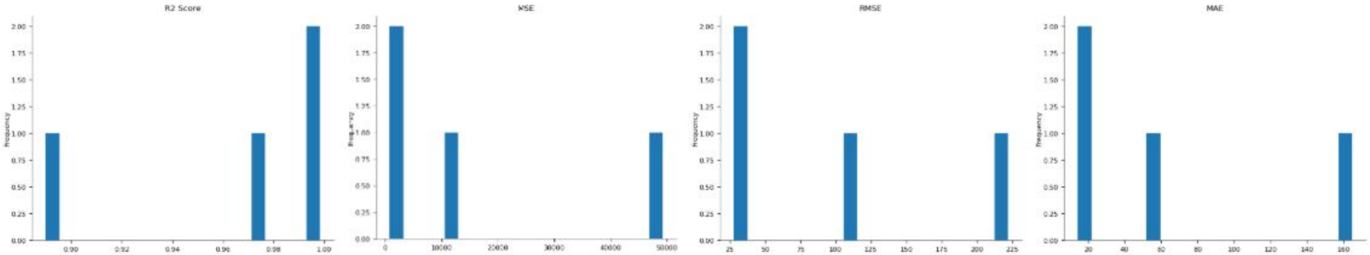
```
X = df.drop('Power (kW)', axis=1)
y = df['Power (kW)']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

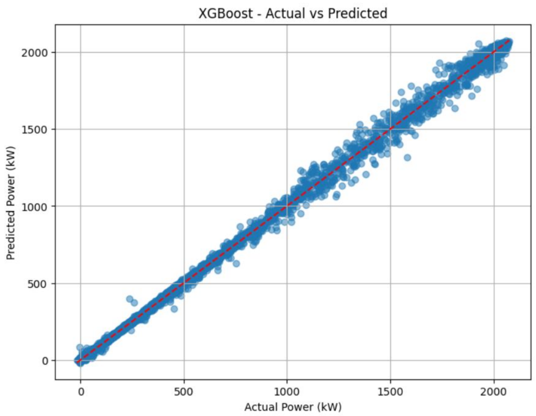
# ML Algorithms

## Power Optimization

| | R2 Score | MSE | RMSE | MAE |
|---|---|---|---|---|
| **XGBoost** | 0.9983 | 778.20 | 27.90 | 14.56 |
| **Random Forest** | 0.9982 | 822.70 | 28.68 | 15.06 |
| **SVR** | 0.9729 | 12141.14 | 110.19 | 59.36 |
| **Linear Regression** | 0.8900 | 49322.73 | 222.09 | 164.77 |

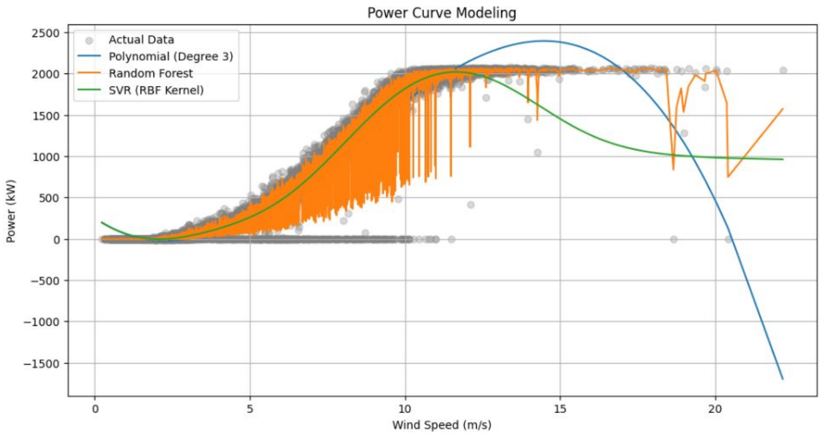
**Distributions**






```
Optimal configuration to maximize power:
Wind speed (m/s)          5.386314
Rotor speed (RPM)        10.033944
Nacelle position        330.578620
Blade angle              92.570000
Name: 2299, dtype: float64
Expected Power Output: 415.15 kW
```

# Power curve analysis




Model Performance Comparison (Power Curve Analysis)

| | Model | R2 Score | MSE | MAE |
|---|---|---|---|---|
| 0 | Random Forest | 0.991200 | 3786.880000 | 27.460000 |
| 1 | SVR (RBF Kernel) | 0.957300 | 18290.600000 | 53.040000 |
| 2 | Polynomial (Degree 3) | 0.944800 | 23632.620000 | 80.180000 |

# References

[1] M. Fahim, V. Sharma, T.-V. Cao, B. Canberk, & T. Q. Duong."Machine Learning-Based Digital Twin for Predictive Modeling in Wind Turbines," IEEE Access, vol. 10, pp. 14184–14194, 2022.

[2] Ibrahim, Mahmoud, Anton Rassõlkin, Toomas Vaimann, Ants Kallaste, Janis Zakis, Van Khang Hyunh, & Raimondas Pomarnacki."Digital Twin as a Virtual Sensor for Wind Turbine Applications," Energies, vol. 16, no. 17, p. 6246, 2023.

[3] Dai, J., Zeng, H., Zhang, F., Chen, H., & Li, M."Study on Obtaining Real Power Curve of Wind Turbines Using SCADA Data," Frontiers in Energy Research, vol. 10, 2022, Art. no. 916355.

[4] Ballinan, Jane, Lu, Sean, Pajalla, Jovener, Tagsip, Early, & Tubola, Orland."Wind Energy Potential of High-Rise Buildings in Makati Using Digital Twins," 2024 IEEE International Symposium on Computing and Technology (ISCT), 2024, pp. 597–603.

[5] Schneider, Janina, Klüner, André, & Zielinski, Oliver."Towards Digital Twins of the Oceans: The Potential of Machine Learning for Monitoring the Impacts of Offshore Wind Farms on Marine Environments," Sensors, vol. 23, no. 10, p. 4581, 2023.

# IT21355714

# Jenojan P

Prediction and Optimizing Wind Turbine Noise Effect

# Background

Wind turbines are a key source of renewable energy, but noise pollution from turbine operations poses environmental challenges.

This research focuses on predicting and optimizing wind turbine noise using machine learning, aiming to balance noise control with power efficiency.

# Research Question

How can a digital twin system integrated with machine learning models be used to predict and optimize wind turbine noise levels while maintaining efficient power output and rotor speed under varying wind conditions?

# Objectives

**Specific Objectives:**

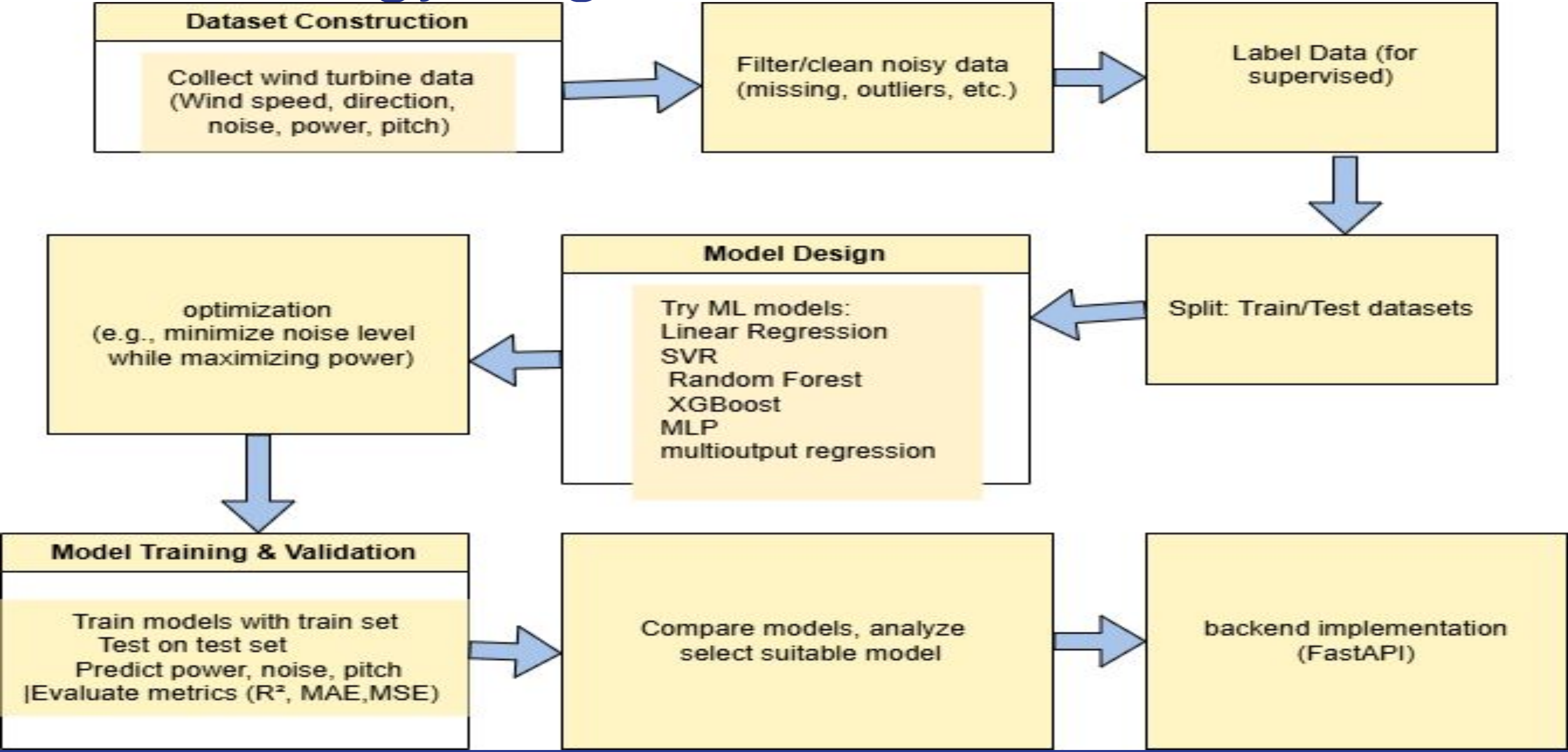
- To identify and analyze major sources of wind turbine noise.
- To develop ML models that predict noise levels, rotor speed, power output, and pitch angle.
- To build a Digital Twin system that simulates turbine behavior based on live or manual input.
- To implement noise mitigation strategies (e.g., pitch angle adjustment)

# Objectives

**Sub-Objectives:**

- Train and evaluate different ML models (Linear, MLP, RF, XGBoost,SVR)
- Compare model performances (MAE, MSE, $R^2$).
- Integrate a backend (FastAPI) for predictions.
- Store, retrieve, and manage predictions in a database.
- Visualize results through a web interface or dashboard.

# Methodology Diagram



# Methodology

**Data Collection**

•Gathered sensor-based data (wind speed, wind direction, pitch angle, power output, rotor speed, noise level).

**Preprocessing**

•Handled missing values and outliers.

•Normalized input/output features using StandardScaler.

•Split data into training and test sets.

# Methodology

**Model Training**

•Trained multiple regression models:

- Linear Regression
- Multi-layer Perceptron (MLP)
- Random Forest Regressor
- XGBoost Regressor
- SVR
- Multi_output regressor

.

# Methodology

**Model Comparison**

| Model Evaluation | | | | | | |
|---|---|---|---|---|---|---|
| | mlp | linar | random_forest | multi_output | svr | xgboost |
| $R^2$ | 0.55 | 0.4 | 0.54 | 0.6 | 0.53 | 0.56 |
| MSE | 6.98 | 9.33 | 7.04 | 6.31 | 7.32 | 6.82 |
| MAE | 1.43 | 1.5 | 1.39 | 1.25 | 2.34 | 1.59 |

# Methodology

**Pitch Angle Optimization**

•Developed a noise-aware optimization function.

•Iterate through possible pitch angles and calculate the power output and noise levels for each, selecting the pitch angle that optimizes power and minimizes noise.

# Methodology

**Backend Deployment**

•Implemented prediction API using **FastAPI**.

•Integrated SQLite for saving and retrieving optimized results.

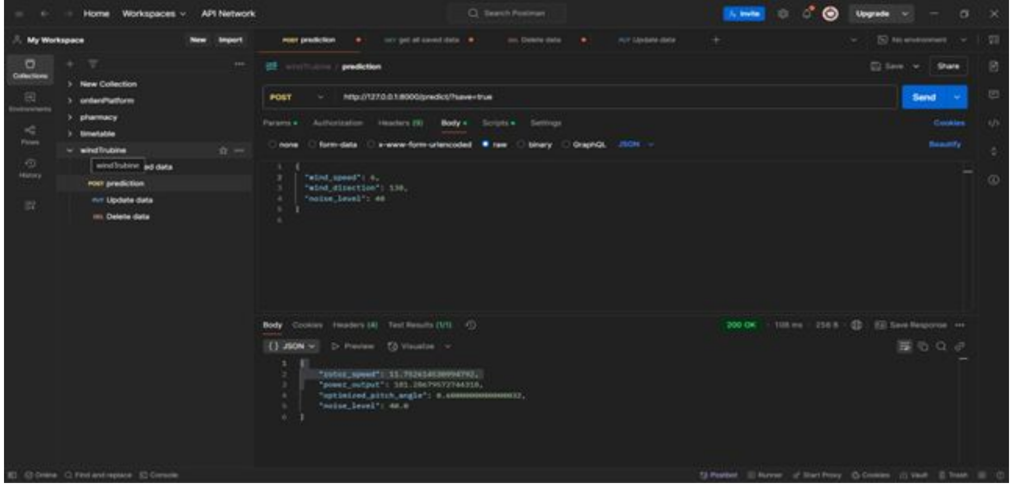
**Digital Twin Integration**

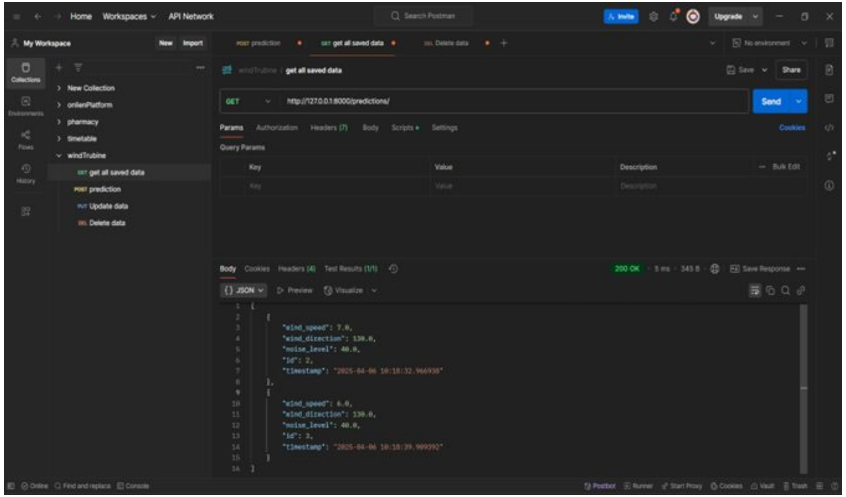
•Create a hybrid digital twin that accepts **manual** and **real-time inputs**.

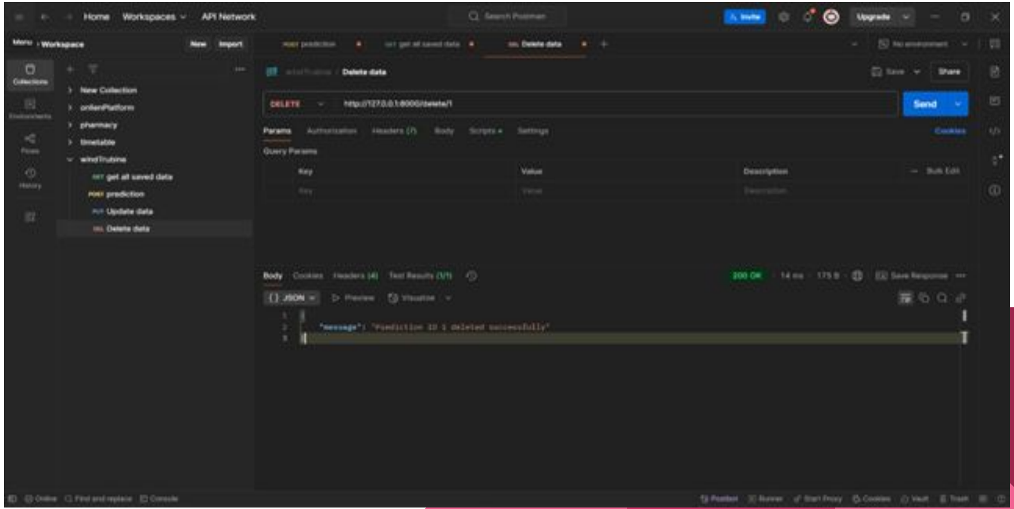
•Provides real-time feedback on pitch angle, noise, power, and rotor speed predictions.

# Methodology

Backend API - Postman Results



# Technologies, Techniques, Algorithms

## Technologies

- **Python**
- **FastAPI**
- **Pydantic**
- **Uvicorn**
- **SQLite / CSV**
- **ReactJS**
- **Blender**
- **Postman**

## Techniques

- **Data Normalization**
- **Digital Twin Simulation**
- **Threshold-based Optimization**
- **Visualization**

## Algorithms

- **MLP Regressor (Multi-Layer Perceptron)** – Predict Power, Rotor Speed, Noise
- **Loop-based Optimization** – Iterative pitch angle tuning to meet target noise
- **Custom Logic** – Match best pitch angle under noise constraints

# Challenges Faced

**Data Issues**:

•**Real-Time Data Availability**:

•Lack of access to real-time data from wind turbines.

•**Use of Existing and Foreign Datasets**:

•Relying on existing and foreign datasets for model training.

# Next Steps & Future Work

•**Model Refinement**: Plans for improving model accuracy and optimizing prediction results.

•

•**Further Data Collection**: Additional data points or real-time data integration for better model performance.

•**Frontend Implementation**:Develop a user-friendly frontend for input parameters and visualization of results (graphs, dashboards).

•**Digital Twin Integration**:Implement real-time synchronization with the digital twin to optimize pitch angle and reduce noise dynamically.

•**Blender (3D Modeling) or matlab**:Create 3D models to visualize turbine operation and noise reduction techniques in real-time.

# References


- Bakker, E., Pedersen, E., van den Berg, G. P., Stewart, R. E., Lok, W., & Bouma, J. (2012). "Wind turbine noise and health: A critical review of the literature." Journal of the Acoustical Society of America, 131(6), 493-507.
- Li, L., Wei, J., & Zhang, S. (2016). "Noise Reduction by Blade Design Optimization in Wind Turbines." Journal of Sound and Vibration, 377, 342-358.
- Feng, Z., Zhang, T., & Sun, H. (2019). "Active Noise Control Systems in Wind Turbines: A Review." Noise Control Engineering Journal, 67(3), 231-247.
- Gao, Z., Fan, L., & Zhang, Y. (2019). "A Novel Approach to Wind Turbine Performance Monitoring Using Digital Twin Technology." Renewable Energy, 131, 132-145.

# IT18149890 Dhanushikan V

Prediction and Optimizing Wind Turbine maintain Strategics

# Research Question

How can digital twin technology be utilized to predict the failure, enhance the reliability, efficiency, and cost-effectiveness of wind turbine maintenance in Sri Lanka's unique tropical climate?

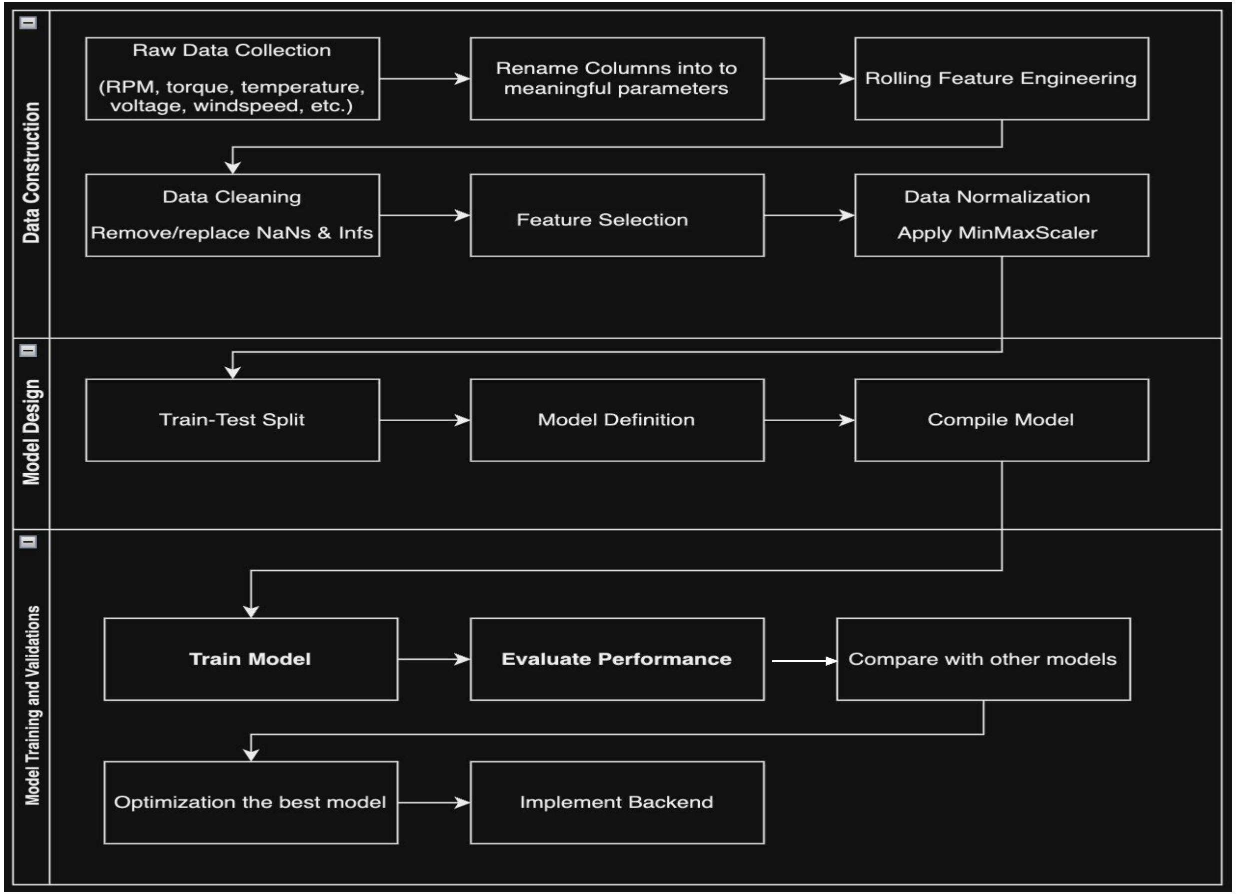
## Specific Objective:

**To design and implement a digital twin-driven maintenance strategy for wind turbines that enhances their reliability, optimizes performance, and minimizes downtime while adapting to Sri Lanka's tropical climate.**

Sub Objectives:

- Collect and integrate real-time sensor data into a centralized system for continuous turbine performance monitoring.
- Use digital twin simulations to model turbine behavior under different weather conditions and analyze impacts.
- Develop machine learning models to predict failures and identify early signs of wear or degradation.
- Create weather-specific protocols like speed adjustments and optimize operations dynamically.
- Build dashboards to display turbine health and generate maintenance reports.
- Evaluate cost savings and tailor solutions to Sri Lanka's climatic challenges.

# Research Methodology



# Methodology

**Data Collection**

Collection Required Raw Data such as RPM, Torque, Power Out, Event count, Windspeed, Voltage, Current Out,Event Status , Turbine Status

**Preprocessing**

- Renaming the column names into understandable format.
- Rolling Feature
- Handling NAN and infinity values
- Feature Selection
- Data Norrmalization

# Models

Models Trained

- Random forest
- Logistic Regression
- XGBoost
- LSTM

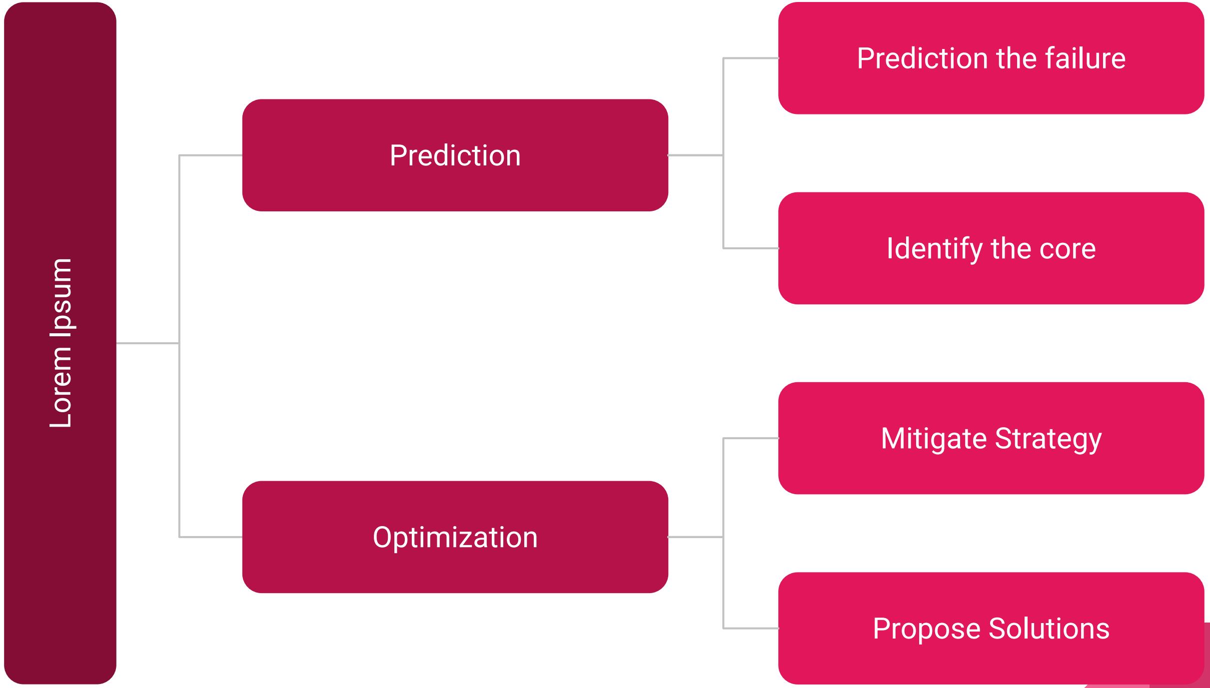
Model selected **LSTM**

Why this model?

Why this better than other model?

Can we achieve the goal with this model?

# Methodology



# Methodology

**Backend Deployment**

•Implemented prediction API using **FastAPI**.

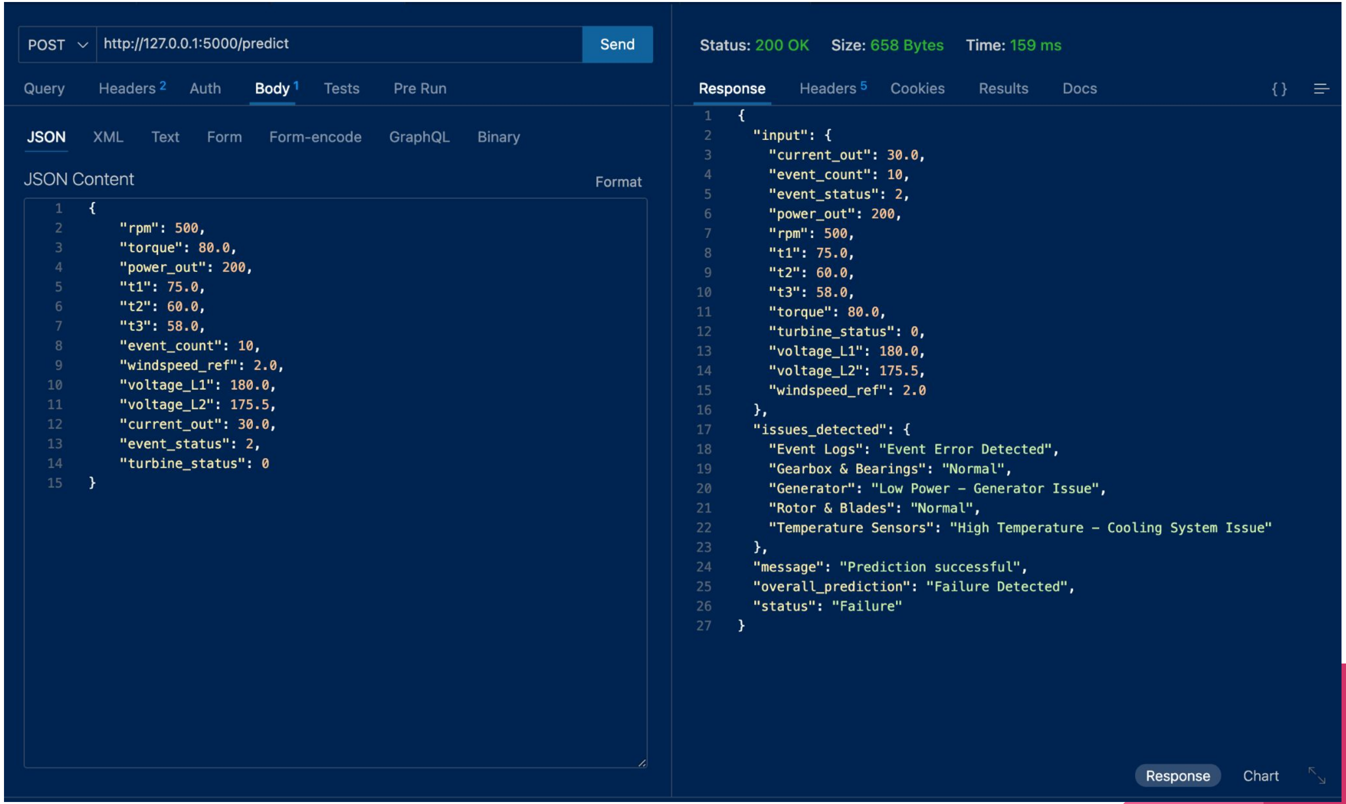
•Integrated SQLite for saving and retrieving optimized results.

**Digital Twin Integration**

•Create a hybrid digital twin that accepts **manual data** and **real-time inputs**.

•Provides real-time feedback required parameters.

# Sample result predicted



# Techniques & Technology Followed in the Journey

## Techniques

- Data Preprocessing
- Normalization
- Rolling Feature
- SMOTE

## Technology

- Anaconda
- Python
- Libraries like Tenserflow, Numpy, Pandas, Matplotlib, Seaborn
- Google Colab
- Flask API

# Challenges Faced

**Overfitting Model:**

- Due to the large imbalance Data set the models were often overfitting

**Limited Access to Real-Time Data**:

- Lack of real-time data from wind turbines makes it hard to update models and track performance in real-time.

**Reliance on External Datasets**:

- Using external datasets can cause issues with data quality, affecting model accuracy in real-world conditions.

**Mitigation Strategy Based on Predictions**:

- The strategy will rely on predictions about power, noise, and weather to reduce risks effectively.

# Next Steps & Future Work

**Model Enhancement**: Strategies for enhancing model accuracy and optimizing the prediction outcomes.

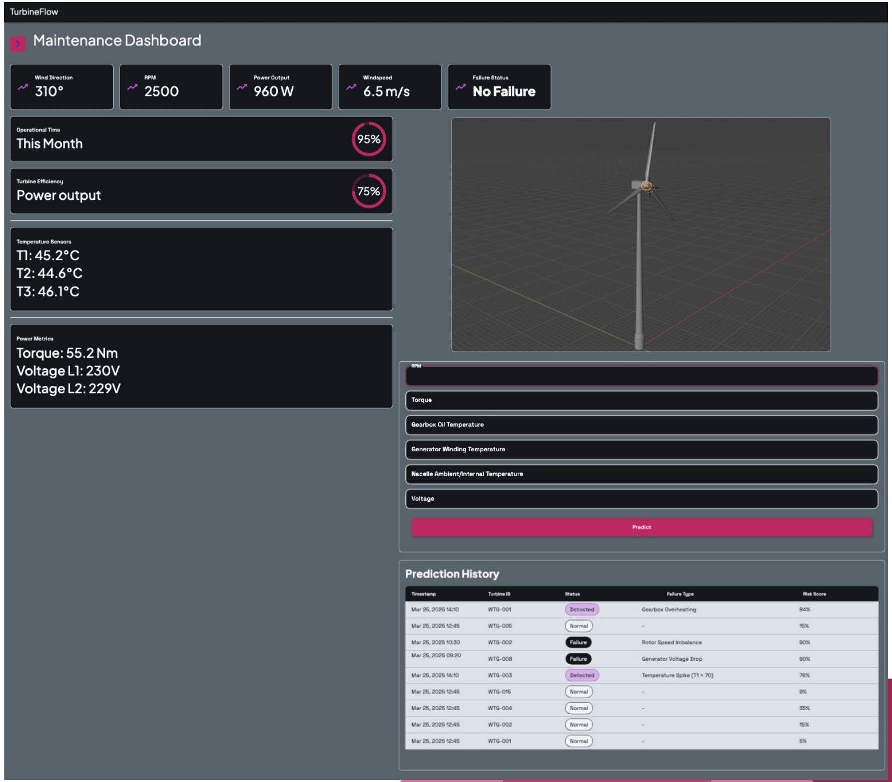
**Expanded Data Acquisition**: Gathering additional data or integrating real-time data for improved model performance.

**User Interface Development**: Build an intuitive user interface for inputting parameters and visualizing results through graphs and dashboards.

**Digital Twin Integration**: Establish real-time data synchronization with the digital twin model to dynamically optimize pitch angle and minimize noise.

**3D Visualization**: Develop 3D models to visually represent turbine operations and noise reduction techniques in real-time.

## A Conceptualized Dashboard Design



# References

[1] A. Pujana, M. Esteras, E. Perea, E. Maqueda, and P. Calvez, “Hybrid-Model-Based Digital Twin of the Drivetrain of a Wind Turbine and Its Application for Failure Synthetic Data Generation,” Energies, vol. 16, 2023.

[2] H. Luo, W. Sun, L. Wang, and C. Lu, “Intelligent Monitoring and Maintenance of Wind Turbine Blades Driven by Digital Twin Technology,” 2023 3rd International Conference on New Energy and Power Engineering (ICNEPE), 2023.

[3] J. Daniel, S. Bhardwaj, H. R. Kulkarni, H. Rahman, S. Mahajan, and S. Agrawal, “Enhancing Efficiency of Wind Farm Energy Using Digital Twin Technology,” 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2024.

[4] M. M. H. Sifat and S. K. Das, “Proactive and Reactive Maintenance Strategies for Self-Healing Digital Twin Islanded Microgrids Using Fuzzy Logic Controllers and Machine Learning Techniques,” IEEE Transactions on Power Systems, vol. 14, 2024.

[5] J. Nikitina, “Caught in the Net: How Digital Twins Work in the Electric Power Industry,” RBC News, 2022.

[6] F. Castellani et al., “Digital Twins in Wind Energy: Emerging Technologies and Industry-Informed Future Directions,” Journal of Renewable Energy Research, 2023.

[7] Ildo Luis Sauer, Alcantaro Lemes Rodrigues, Welson Bassi , Dataset on SCADA Data of an Urban Small Wind Turbine Operation in São Paulo, Brazil”

# GITHUB LINK

https://github.com/it21355714/wind-turbine-optimizer.git

# Thank You !