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

Hearth H.M.T.S.

(IT19208572)

(The dissertation was submitted in partial fulfilment of the requirements for the B.Sc. (Honors) degree in Information Technology Specialising in Software Engineering)

Department of Computer Science

Sri Lanka Institute of Information Technology

August 2025

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

Hearth H.M.T.S.

(IT19208572)

(The dissertation was submitted in partial fulfilment of the requirements for the B.Sc. (Honors) degree in Information Technology Specialising in Software Engineering)

Department of Computer Science

Sri Lanka Institute of Information Technology

August 2025

DECLARATION

I declare that this is my own work, and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as article or books).

Student Name	Student ID	Signature
Herath H.M.T.S	IT19208572	

As the supervisor/s of the above-mentioned candidates, I hereby certify that they are conducting research for their undergraduate dissertation under my guidance and direction.

Signature of the supervisor
(Mr. Vishan Jayasignhearachchi)

Date: 29/08/2025

ABSTRACT

The integration of digital twin technology into renewable energy systems has emerged as a transformative approach to improving efficiency, reliability, and sustainability. Wind turbines, as a major source of clean electricity, face challenges such as fluctuating wind conditions, suboptimal control strategies, and the need for accurate generation forecasting. This project develops a digital twin framework for wind turbines that serves as a virtual replica, enabling real-time performance monitoring, predictive modeling, and data-driven decision-making. The system incorporates multiple components, including power optimization, energy forecasting, power curve analysis, weather impact evaluation, noise reduction, and predictive maintenance.

The focus of this study is on power optimization and energy forecasting, two critical aspects for maximizing turbine performance and supporting effective energy management. In the power optimization component, machine learning models were implemented to analyze the relationships between key operational parameters such as blade pitch angle, nacelle orientation, and wind speed and the resulting power output. By simulating adjustments to these control variables, the system identifies optimal configurations that can increase energy yield while reducing mechanical stress on turbine components.

For energy forecasting, time-series models were developed to predict short-term energy generation based on historical turbine and environmental data. Techniques including regression-based models and advanced algorithms were evaluated, with performance measured using metrics such as R^2, Mean Absolute Error (MAE), and Mean Squared Error (MSE). Accurate forecasting supports grid stability, scheduling of maintenance activities, and improved integration of wind power into the broader energy system.

The results demonstrate that combining power optimization with accurate forecasting in a digital twin framework significantly enhances turbine efficiency and operational planning. This highlights the potential of digital twin technology as a powerful tool for enabling cost-effective, data-driven, and sustainable wind energy management.

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude to our supervisor, Mr. Vishan Jayasinghearachchi, for the continuous guidance, encouragement, and constructive feedback provided throughout the project. His expertise and insightful suggestions have been invaluable in shaping the direction of this work and ensuring its successful completion.

I also extend my heartfelt appreciation to our co-supervisor, Mr. Jeewaka Perera, for the dedicated support, timely advice, and constant encouragement offered during every stage of this project. His attention to detail and commitment to excellence greatly contributed to improving both the technical and academic quality of this work.

A special note of thanks goes to Mr. Janaka from Ceylon Electricity Board (CEB), Mannar Thambapavani, for sharing his domain knowledge and providing access to the dataset used in this study. His valuable insights into the operational aspects of wind turbines added a strong practical perspective to the research and helped bridge the gap between theory and real-world applications.

I am also grateful to CDAP for their support and collaboration, which provided essential resources and facilities that enabled the smooth execution of this project.

Finally, I would like to acknowledge the encouragement and assistance received from my team members, colleagues, friends, and family, who have been a constant source of motivation and inspiration throughout this journey. Their support has been instrumental in overcoming challenges and completing this project successfully.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	**Full Form**
DT	Digital Twin
SCADA	Supervisory Control and Data Acquisition
LSTM	Long Short-Term Memory
ARIMA	
MSE	Mean Squared Error
MAE	Mean Absolute Error
CEB	Ceylon Electricity Board
ML	Machine Learning
EDA	Exploratory Data Analysis

1. INTRODUCTION

1.1 Background and Literature Survey

Renewable energy has emerged as a cornerstone of sustainable development, with wind power playing a central role in meeting global electricity demands while reducing greenhouse gas emissions. Wind turbines are increasingly deployed across the world, from large-scale offshore farms to smaller onshore installations, contributing significantly to the energy mix of many countries. However, the inherently variable and unpredictable nature of wind poses challenges to the efficient operation and integration of wind energy into the grid. This has created a demand for advanced tools and methods that can enhance the reliability, performance, and predictability of wind turbines.

Figure 1.1 Cumulative wind power Capacity worldwide

Source: WWEA Annual Report 2023

Traditionally, the operation and maintenance of wind turbines have relied heavily on Supervisory Control and Data Acquisition (SCADA) systems and scheduled maintenance practices. SCADA systems collect operational data such as wind speed, rotor speed, and power output, which are then analyzed to monitor performance. While useful, these systems often provide a reactive approach—issues are detected after they occur, leading to potential downtime and reduced energy production. Maintenance has typically followed either a fixed schedule or a breakdown-based model, both of which can be costly. Scheduled maintenance may replace components prematurely, while breakdown maintenance risks unexpected failures that disrupt energy generation.

In terms of power optimization**,** traditional strategies have largely depended on pre-set turbine control algorithms designed by manufacturers. These algorithms adjust blade pitch angles or nacelle orientation based on wind speed thresholds but are not always adaptive to rapidly changing wind conditions or site-specific characteristics. As a result, turbines may not always operate at their maximum efficiency, leading to lost opportunities for increased power generation. Similarly, energy forecasting has historically relied on statistical methods such as regression models or moving averages. Although these techniques provide broad estimates, they often fail to capture the non-linear and highly dynamic relationships between weather variables and turbine performance, leading to inaccurate predictions that complicate grid management.

The emergence of Digital Twin (DT) technology provides a paradigm shift in addressing these challenges. A digital twin is a virtual replica of a physical system that integrates real-time data, simulations, and predictive models to mirror and analyze the behavior of the actual asset. In the context of wind turbines, a digital twin continuously ingests sensor data from the physical turbine and updates its virtual model, enabling dynamic monitoring, diagnostics, and decision-making. Unlike traditional systems, DTs are proactive they can predict failures before they occur, recommend optimal control strategies, and simulate scenarios without risking the actual turbine.

For power optimization, digital twins enable a more intelligent and adaptive approach. By leveraging machine learning models and historical data, a digital twin can understand how operational parameters such as blade pitch, nacelle orientation, wind speed, and rotor speed influence power output. It can then simulate adjustments in real time to determine the optimal configuration, ensuring maximum energy capture under varying wind conditions. This moves beyond static manufacturer-set algorithms, allowing site-specific and condition-specific optimization that significantly improves performance.

Similarly, in energy forecasting, digital twins integrate time-series data, weather forecasts, and machine learning models to provide highly accurate predictions of future power generation. Advanced can capture the non-linear and temporal dependencies in wind and power data, outperforming traditional statistical methods. These forecasts not only support grid stability and efficient energy market participation but also help operators plan maintenance and allocate resources effectively.

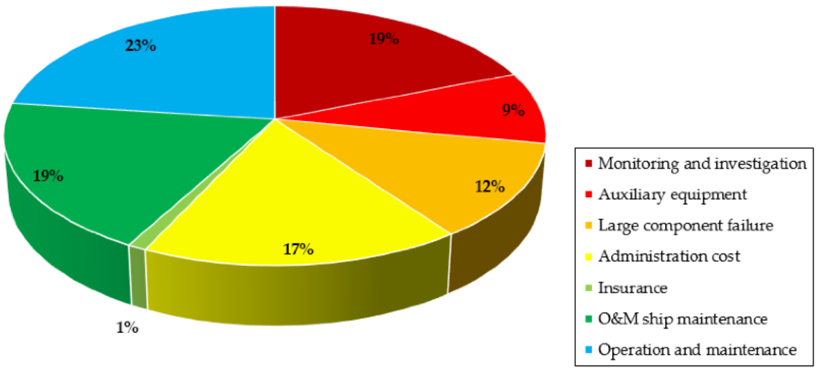

Figure 1.2 Cost structure of operation and maintenance of offshore wind turbines

The adoption of digital twin technology in wind energy further aligns with the global shift towards data-driven energy management**.** By combining real-time monitoring, predictive modeling, and optimization, digital twins serve as powerful decision-support tools that bridge the gap between the physical and digital domains. They enhance not only the technical efficiency of wind turbines but also their economic viability by reducing downtime, minimizing maintenance costs, and maximizing power generation.

This research project builds upon these concepts by developing a digital twin framework for wind turbines with a specific focus on power optimization and energy forecasting. By analyzing operational and environmental data, implementing machine learning models, and validating results against real-world datasets, the study demonstrates how digital twins can transform the performance and reliability of wind turbine systems. Ultimately, this contributes to advancing renewable energy integration, supporting sustainability goals, and ensuring a more resilient and efficient energy future.

1.2 Research Gap

The increasing penetration of wind energy into global electricity systems has resulted in a growing body of research focused on improving turbine efficiency, reliability, and forecasting accuracy. Numerous studies have explored various aspects of wind turbine performance, including control strategies for power optimization, statistical and machine learning methods for energy forecasting, and monitoring approaches for predictive maintenance. Despite these advancements, significant gaps remain in translating these methods into holistic, real-time, and adaptive frameworks that can be effectively implemented in practical wind farm operations. The emergence of Digital Twin technology offers an opportunity to address these limitations, but its application in wind energy is still in the early stages of development and lacks comprehensive integration across key performance areas.

One of the primary areas where gaps exist is in power optimization. Traditional control systems rely on pre-defined algorithms, typically designed by manufacturers, to regulate operational parameters such as blade pitch angle, yaw orientation, and rotor speed. While effective under standard conditions, these approaches are not sufficiently adaptive to rapidly changing wind patterns or site-specific conditions. Some recent studies have applied machine learning and artificial intelligence techniques to model turbine performance, yet most of these works are either limited to offline simulations or focus on single-parameter optimization. Few studies consider the multi-dimensional nature of turbine control variables and their interdependencies. Moreover, existing optimization models often lack the capability to operate in real time, which is critical for maximizing efficiency under fluctuating environmental conditions.

A second area of concern is energy forecasting, which plays a vital role in grid stability and energy market participation. Traditional forecasting methods, such as ARIMA models and linear regression, have proven insufficient in capturing the highly non-linear dynamics of wind and power generation. However, many of these models are trained on historical datasets without incorporating real-time operational feedback from turbines. As a result, while forecasting accuracy may improve in static testing environments, performance often declines when deployed under real-world conditions characterized by dynamic and unpredictable weather changes. Furthermore, most research on energy forecasting focuses on short-term predictions (minutes to hours) or long-term projections (days to weeks), but fewer studies address the medium-term horizon, which is equally important for operational planning.

Another gap lies in the integration of optimization and forecasting into a single decision-support system**.** Much of the existing literature treats these two areas as separate research problems, with power optimization models developed independently of forecasting models. In practice, however, these two functions are closely interrelated: effective optimization requires accurate forecasts of upcoming wind conditions, while improved forecasting can benefit from optimization data that accounts for turbine operational states. The absence of integrated frameworks limits the practical utility of research findings and reduces their potential to enhance wind farm performance holistically.

From a technological perspective, Digital Twin applications in wind energy remain relatively underdeveloped compared to other industries such as manufacturing and aerospace. Existing DT implementations in wind turbines often emphasize structural health monitoring or fault detection, with fewer systems designed for operational optimization or predictive forecasting. Moreover, many digital twin prototypes are demonstrated on small-scale testbeds or simulated datasets, rather than validated using real-world operational data from active wind farms. This creates a gap between academic research and industrial application, highlighting the need for solutions that can function reliably in real operational environments like those of large-scale wind farms.

Additionally, there are regional and contextual gaps in the literature. Most studies on digital twins for wind turbines have been carried out in Europe, North America, and China, where wind energy infrastructure is mature, and access to high-quality datasets is readily available. In developing countries, such as Sri Lanka, wind energy projects are still emerging, and site-specific factors such as monsoonal wind patterns and coastal turbulence introduce unique challenges. However, limited research has been conducted to adapt digital twin frameworks to these localized conditions. This presents both a challenge and an opportunity while data availability may be limited, localized digital twins can provide tailored solutions that improve efficiency and enhance renewable energy integration in resource-constrained contexts.

The lack of comparative model evaluation in current research also represents a gap. Many studies report the success of a single algorithm, such as a neural network or ensemble model, without comparing performance against alternative approaches under the same conditions. This makes it difficult to assess the true effectiveness of different models and identify the most suitable ones for practical applications. For example, while LSTM networks are often praised for their ability to capture temporal dependencies, they require large volumes of high-quality data, which may not always be available. In contrast, ensemble methods like Random Forest or XGBoost may perform better in data-scarce environments. A systematic evaluation of multiple models within a digital twin framework is therefore essential.

In summary, the literature reveals several key research gaps such as traditional optimization methods lack adaptability and real-time responsiveness, forecasting models often fail to incorporate real-time operational data and medium-term horizons, optimization and forecasting are rarely integrated into unified systems, digital twin applications in wind energy are still fragmented and under-validated in real-world contexts, regional research is limited in developing countries with unique wind profiles, comparative evaluations of multiple models are often missing. This project seeks to address these gaps by developing a digital twin framework for wind turbines that integrates power optimization and energy forecasting using real-world datasets, evaluates multiple machine learning models, and adapts the approach to the context of Sri Lanka’s wind energy infrastructure. By bridging these gaps, the study contributes to advancing both the academic understanding and practical application of digital twins in renewable energy systems.

1.3 Research Problem

Wind energy has become one of the most important renewable energy sources for meeting the rising global demand for sustainable power. Modern wind turbines are capable of generating large amounts of electricity, but their performance is heavily dependent on fluctuating wind conditions and operational control strategies. The variability of wind speed, direction, and turbulence often results in inconsistent power generation, making it difficult to achieve maximum efficiency. Furthermore, the integration of wind energy into the grid requires accurate forecasting to balance supply and demand. However, current operational practices and forecasting methods face significant limitations that hinder the reliability and cost-effectiveness of wind energy systems.

Traditionally, turbine operation has been managed using predefined control algorithms supplied by manufacturers. These algorithms adjust parameters such as blade pitch and nacelle orientation based on fixed thresholds, aiming to capture optimal wind flow. While effective under general conditions, these methods lack adaptability to rapidly changing local wind patterns and site-specific environmental variations. As a result, turbines may fail to consistently operate at their maximum efficiency, leading to avoidable energy losses and reduced economic returns. This problem is particularly acute in regions with high wind variability, such as coastal and monsoonal zones.

Similarly, energy forecasting methods have historically relied on statistical approaches like regression models, ARIMA, or moving averages. Although these methods provide broad estimates, they often struggle to account for the non-linear and dynamic relationships between wind speed, atmospheric conditions, and turbine performance. As a consequence, forecasts are frequently inaccurate, leading to difficulties in grid integration, imbalance costs, and inefficiencies in scheduling maintenance or power trading. Even with the introduction of advanced machine learning models, many forecasting solutions remain isolated, relying on historical data without incorporating real-time operational feedback from turbines.

While research has explored optimization and forecasting separately, there is a lack of integrated frameworks that combine the two within a real-time decision-support system. In practice, optimization and forecasting are interdependent: accurate forecasting improves optimization strategies by predicting upcoming wind conditions, while optimization data enhances forecasting models by providing insights into turbine performance under different control settings. The absence of such integration results in fragmented solutions that cannot fully exploit the potential of data-driven wind energy management.

The application of Digital Twin technology offers a promising solution, yet its adoption in wind energy remains limited. Most existing digital twin implementations focus on condition monitoring or fault detection, with fewer systems designed to support operational optimization and predictive forecasting. Moreover, many digital twin prototypes are validated using simulated or small-scale test data, rather than real-world datasets from active wind farms. This gap between theory and practice reduces their practical value, especially in developing countries where resource constraints demand cost-effective and site-specific solutions.

Therefore, the research problem addressed in this study can be stated as follows: How can a digital twin framework be developed to integrate power optimization and energy forecasting for wind turbines, using real-world operational data, in order to enhance efficiency, improve forecasting accuracy, and support sustainable energy management? By focusing on this problem, the research aims to overcome the limitations of traditional methods, bridge the gap between optimization and forecasting, and demonstrate the applicability of digital twin technology in a real-world context such as Sri Lanka's wind energy sector.

1.4 OBJECTIVES

1.4.1 Main Objective

The main objective of this research is to develop a digital twin framework for wind turbines that integrates power optimization and energy forecasting to enhance turbine efficiency, improve forecasting accuracy, and support sustainable energy management.

1.4.2 Specific Objectives

1. To design and implement a data preprocessing pipeline for wind turbine operational data.

High-quality data is fundamental for reliable machine learning and forecasting. This objective focuses on creating a preprocessing pipeline that handles raw sensor and historical turbine datasets, ensuring consistency and usability. The process includes cleaning missing or erroneous values, normalizing variables across different scales, and performing feature engineering to extract meaningful indicators such as wind speed fluctuations, rotor speed trends, and environmental conditions. By transforming raw data into structured and refined inputs, the pipeline establishes the foundation for building robust optimization and forecasting models.

2. To develop machine learning models for power optimization.

The second objective addresses the design of intelligent models capable of identifying operational strategies that maximize energy yield. By analyzing critical parameters such as blade pitch angle, nacelle orientation, wind speed, and rotor speed, the models can reveal optimal control settings for different environmental conditions. The integration of machine learning techniques enables the system to capture nonlinear relationships between variables, adapt to changing wind dynamics, and provide actionable insights for improving turbine efficiency.

3. To implement time-series forecasting models for predicting wind turbine energy generation.

Reliable energy forecasting is essential for effective grid management and operational planning. This objective focuses on applying and comparing advanced time-series models such as Random Forest, XGBoost, and LSTM networks to generate short- and medium-term predictions of turbine power output. These models account for temporal dependencies, seasonal variations, and environmental factors, enabling operators to anticipate generation patterns and align turbine operations with grid demand.

4. To evaluate and compare the performance of multiple models using appropriate metrics.

Model performance must be validated systematically to ensure accuracy and reliability. This objective involves benchmarking optimization and forecasting models using statistical metrics such as the Coefficient of Determination (R^2), Mean Squared Error (MSE), and Mean Absolute Error (MAE). By applying multiple evaluation criteria, the study ensures that models are not only accurate but also generalizable across varying operational conditions. The comparison enables the selection of the best-performing models for deployment in the digital twin framework.

5. To integrate power optimization and forecasting into a unified digital twin framework.

This objective emphasizes the development of a cohesive system that brings together optimization and forecasting within a Digital Twin environment. The framework functions as a virtual replica of the wind turbine, capable of real-time data ingestion, simulation, and predictive analysis. By unifying these functions, the digital twin provides a holistic decision-support tool that adapts dynamically to real-world conditions and offers insights for both short-term operation and long-term performance optimization.

6. To validate the digital twin framework using real-world wind farm data.

The credibility of the proposed system depends on empirical validation. This objective involves testing the digital twin using real-world datasets, such as those obtained from the Mannar Thambapavani wind farm in Sri Lanka. By comparing model predictions and optimization outcomes with actual operational records, the study evaluates the system's accuracy, robustness, and practical applicability. Validation ensures that the framework is not only theoretically sound but also deployable in live operational contexts.

7. To assess the contribution of the digital twin in improving efficiency and sustainability.

The final objective highlights the broader impact of the research. Beyond technical accuracy, the digital twin framework is evaluated in terms of its ability to reduce inefficiencies, enhance forecasting accuracy, and contribute to long-term sustainability goals. By demonstrating improved turbine efficiency, reduced operational uncertainty, and optimized energy output, the system provides a pathway for advancing renewable energy management and supporting the global transition to cleaner power generation.

2. METHODOLOGY

The methodology adopted in this study follows a structured, step-by-step approach to design and implement a digital twin framework for wind turbines, with a focus on power optimization and energy forecasting.

2.1 Requirement Gathering and Analysis

Developing a digital twin for wind turbines requires a systematic understanding of both the technical requirements and the operational context in which the system will function. Requirement gathering ensures that the solution is aligned with stakeholder needs, while requirement analysis helps define the scope, constraints, and expected outcomes. In this research, requirements were gathered through consultations with domain experts from the Ceylon Electricity Board (CEB), Mannar Thambapavani wind farm, academic supervisors, and a review of existing literature on wind turbine digital twins. Additionally, the available operational datasets from the wind farm formed the foundation for both the modeling and validation stages. The analysis process was divided into functional and non-functional requirements, supported by a feasibility study to evaluate the practicality of implementing the system.

2.2 Functional Requirements

1. **Data Ingestion and Management –** The system must be capable of ingesting wind turbine data from the database and, in the future, from real-time sensor streams.
2. **Data Preprocessing –** It should handle missing values, normalize features, and filter unnecessary variables to ensure data quality.
3. **Power Optimization Module –** A machine learning component that identifies optimal operational settings (e.g., blade pitch angle, nacelle position) to maximize energy generation.
4. **Energy Forecasting Module** – A time-series forecasting model that predicts future energy output based on historical wind and turbine data.
5. **Evaluation and Visualization –** The system should provide model evaluation metrics (R^2, MSE, MAE) and generate graphs for performance comparison and insights.

6. **Digital Twin Framework Integration –** A virtual representation of the turbine must be established, enabling simulation of operational scenarios.
7. **API and Interface Support –** The backend should expose data and predictions via APIs, ensuring easy integration with frontend visualization tools or dashboards.

2.3 Non-Functional Requirements

1. **Scalability** – The system must be extendable to multiple turbines and larger datasets.
2. **Accuracy** – Forecasting and optimization must achieve a high level of precision to be practical for operational decision-making.
3. **Reliability** – The digital twin framework should consistently produce valid results without failures.
4. **Security** – Data confidentiality and secure access must be ensured, especially when dealing with wind farm operational data.
5. **Performance** – Predictions and optimizations should be delivered within acceptable response times to support real-time applications in the future.
6. **Usability** – The system should be intuitive and accessible for engineers, researchers, and decision-makers.

2.4 Feasibility Study

A feasibility study was conducted to ensure the proposed solution is practical and implementable.

- **Technical Feasibility:** Python, Flask, and machine learning frameworks such as Scikit-learn, XGBoost, and TensorFlow provide robust tools for developing the models. Azure cloud services can be leveraged for deployment and scalability. Hence, the technical implementation is feasible with existing technologies.

- **Operational Feasibility:** The solution aligns with the needs of wind farm operators by providing optimization and forecasting capabilities that directly contribute to improved performance and sustainability. Domain expertise from CEB enhances its operational practicality.

- **Economic Feasibility:** As open-source frameworks are used, costs are minimized. Future cloud deployment may incur expenses, but the benefits of predictive maintenance and improved efficiency justify investment.
- **Schedule Feasibility:** Within the project timeline, it is feasible to implement the data preprocessing, model development, and integration phases, with real-time deployment planned for future work.

2.5 High Level System Architecture

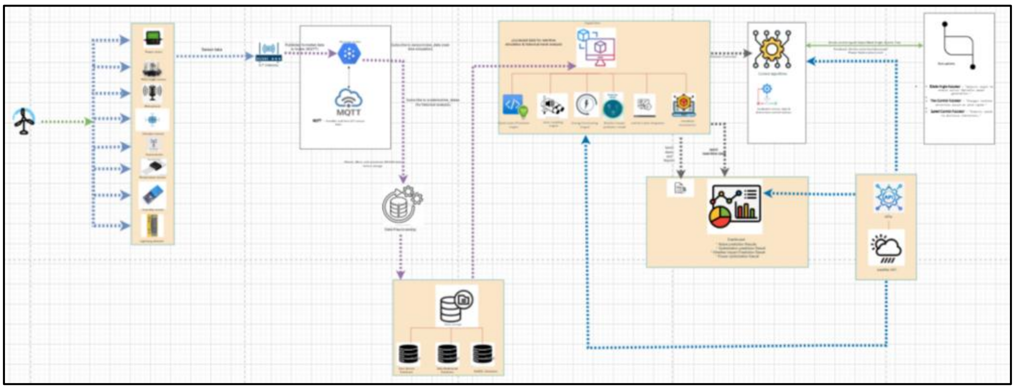

Figure 2.1. High level system architecture diagram

The proposed digital twin system for wind turbines is designed as a multi-layered, modular architecture that enables real-time monitoring, optimization, and forecasting of turbine performance. At the core of this architecture lies the digital twin layer, which acts as the virtual replica of the physical wind turbine. This layer continuously synchronizes with sensor data streams such as wind speed, rotor speed, blade pitch angle, nacelle position, and power output to simulate, predict, and optimize turbine behavior under varying environmental and operational conditions.

The architecture follows an event-driven and cloud-based approach, ensuring scalability and responsiveness. The data acquisition layer collects raw inputs from turbine sensors or historical datasets. These inputs flow into the data preprocessing pipeline, where data cleaning, normalization, feature selection, and missing value handling are performed. Pre-processed data is then stored in a centralized database for structured access and persistence.

The analytics and machine learning layer hosts the core models for power optimization, energy forecasting, and power curve analysis. Python-based ML frameworks are deployed here, exposed via Flask API services. These APIs communicate results such as optimized pitch control settings or forecasted power outputs to the higher layers of the system. On top of the ML models, the digital twin simulation layer integrates with visualization tools. A 3D front-end model developed using Three.js renders the turbine in real time, showing live metrics, forecasts, and optimization suggestions.

Overall, the high-level architecture integrates data ingestion, preprocessing, ML modeling, APIs, and visualization into a unified framework. This modular design makes the system extensible, meaning additional components like noise analysis, predictive maintenance, or weather impact studies can be integrated seamlessly in the future.

2.6 Component Architecture

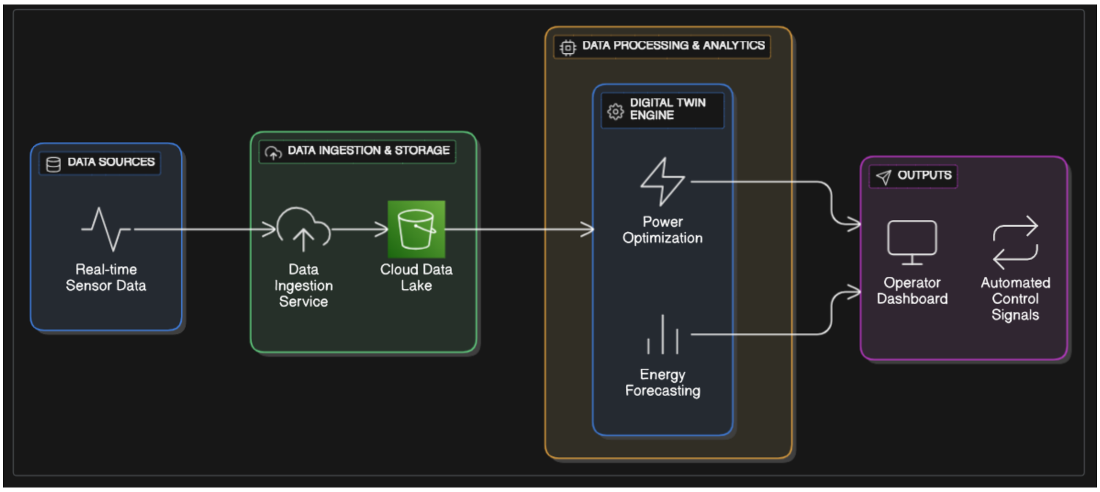

Figure 2.1. High level component architecture diagram

This research focuses on two critical modules: Power Optimization and Energy Forecasting. Both components are tightly coupled with the digital twin but serve distinct purposes.

For Power Optimization, the architecture is structured around real-time parameter tuning. Inputs such as wind speed, nacelle position, and blade pitch angle are ingested from the preprocessing pipeline. These inputs are fed into a set of regression and ensemble ML models (XGBoost, Random Forest etc) trained to predict optimal power output. The optimization logic identifies the best control strategy such as adjusting pitch angles to maximize energy extraction under current wind conditions. The optimized settings are then exposed through APIs to the digital twin, which can simulate how adjustments improve performance.

The Energy Forecasting component uses time-series modeling approaches. The dataset, which includes timestamped power output and weather variables, is processed into sequences for training forecasting models such as LSTM, XGboost. These models generate short-term and medium-term predictions of turbine power output, which are crucial for grid stability and operational planning. The forecasting API continuously updates predictions and feeds results into the visualization layer, where they are shown as forecast curves alongside real power outputs.

Both components share a common architecture pattern:

1. **Input layer** – Receives processed data from the pipeline.
2. **Modeling layer** – Executes ML algorithms (optimized separately for regression and time-series forecasting).
3. **API service layer** – Wraps models using Flask endpoints for easy integration with the digital twin.
4. **Visualization/output layer** – Provides results to the UI, either as optimized control parameters or forecasted time-series plots.

This modular design ensures that even though optimization and forecasting solve different problems, they remain interoperable and can run in parallel. Additionally, the models are designed to be evaluated and retrained iteratively, ensuring adaptability to new data over time.

2.7 Implementation

2.7.1 Data Collection and Preparation

Data is the foundation of any digital twin system, particularly when developing machine learning models. For this project, operational data was obtained from the CEB Mannar Thambapavani wind farm, one of Sri Lanka’s largest wind power facilities. This dataset contained information such as timestamped wind speed, rotor speed, nacelle orientation, blade pitch angles, and power output. These variables are essential for both power optimization and energy forecasting tasks.

The data collection process followed a structured approach. First, domain experts were consulted to determine which features significantly influence turbine performance. Then, raw datasets were gathered in the form of CSV and Excel files. Some files represented historical performance data, while others captured sensor logs under varying wind conditions.

Once collected, the datasets were carefully examined for completeness and consistency. Many real-world energy datasets often contain missing values due to sensor malfunctions, communication errors, or scheduled maintenance. Outliers were also identified, as abnormal turbine behavior (e.g., shutdown during storms) could skew results. Data integrity checks ensured that timestamps were consistent, and variables aligned properly.

2.7.2 Data Preprocessing and Exploratory Data Analysis (EDA)

To transform the raw data into a usable format, a comprehensive preprocessing pipeline was developed. Preprocessing is critical because the accuracy and robustness of machine learning models largely depend on the quality of input data.

The first step involved handling missing values. Missing records were either interpolated using statistical methods or removed if they represented non-operational periods. Next, duplicate entries were eliminated to prevent biased training. Outliers were detected through z-score and IQR methods, and domain-driven rules (e.g., negative wind speeds are invalid) were applied for correction.

```python
def clean_dataset(df: pd.DataFrame):
    # 1. Drop duplicates
    df = df.drop_duplicates()

    # 2. Handle missing values
    # Forward fill per turbine, then backfill if needed
    df = df.groupby("turbine").apply(lambda g: g.ffill().bfill()).reset_index(drop=True)

    # 3. Handle timestamp gaps (resample to fixed frequency, e.g., 10 min)
    df = df.set_index("timestamp")
    df = df.groupby("turbine").apply(lambda g: g.resample("10T").mean()).reset_index()

    return df

from sklearn.preprocessing import MinMaxScaler

def scale_features(df: pd.DataFrame, feature_cols):
    scaler = MinMaxScaler()
    df[feature_cols] = scaler.fit_transform(df[feature_cols])
    return df, scaler
```

Figure 2.2 Data Preprocessing

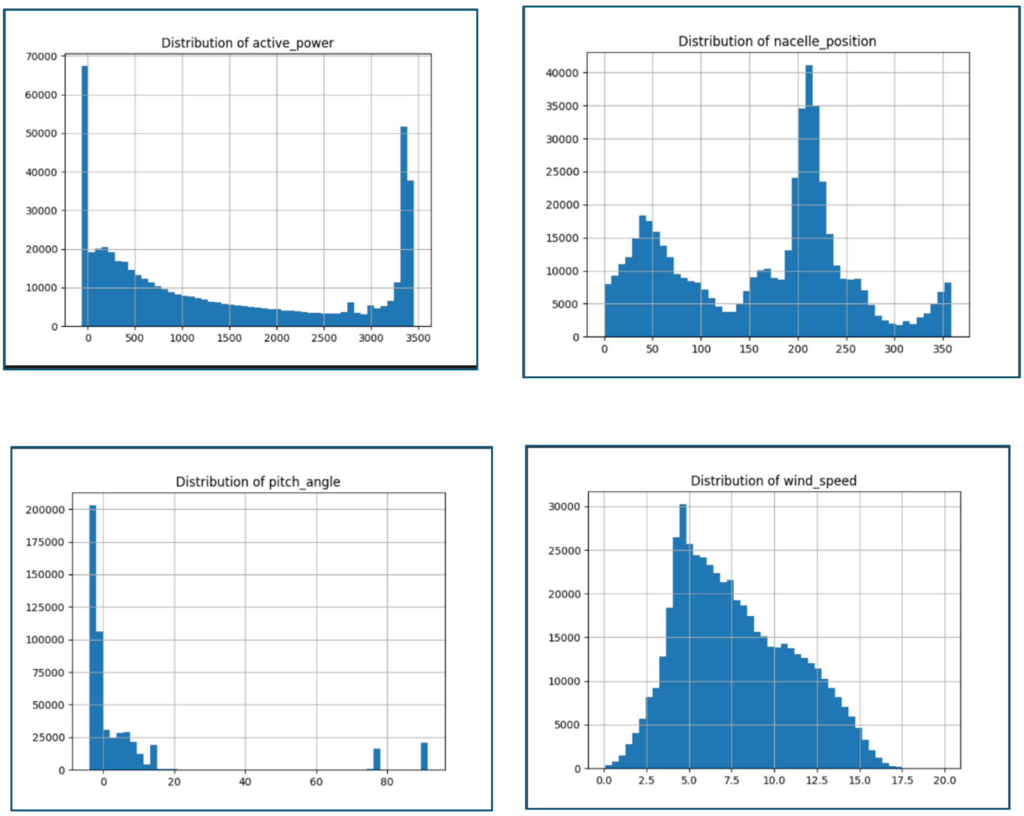

Figure 2.3 Distribution histograms

After that the normalization and scaling were applied. Since wind speed, rotor speed, and blade pitch operate on different scales, min-max scaling was used to bring them into a uniform range. This prevents algorithms like gradient boosting or neural networks from being biased toward higher-magnitude features.

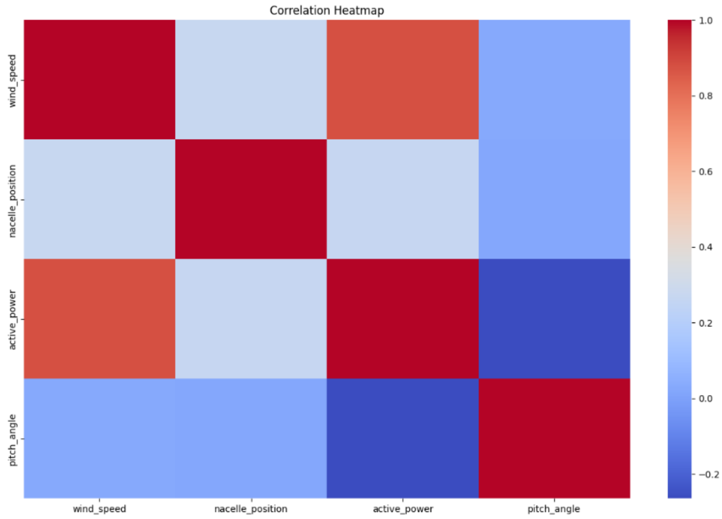

Figure 2.3 Heatmap of correlations

Wind speed vs. active power very strong positive correlation, which makes sense. As wind speed increases, a wind turbine's active power output would also be expected to increase significantly.

Finally, the dataset was split into training and testing sets using an 80-20 ratio to ensure reliable model evaluation. Time-series forecasting models additionally required data to be structured into input sequences and output labels for supervised learning.

2.7.3 Model Development and Evaluation for Power optimization

Power optimization in wind turbines plays a critical role in maximizing energy generation by adjusting controllable parameters such as blade pitch angle, and nacelle orientation to suit dynamic wind conditions. Traditionally, optimization has been handled using static lookup tables or rule-based controllers, which lack adaptability to nonlinear and fluctuating wind patterns. To address this, the project developed and evaluated machine learning models capable of learning from historical operational data and predicting the optimal power output for a given set of input conditions.

The model development began by defining the input features (wind speed, blade pitch, nacelle angle) and the target variable (power output). Three machine learning algorithms were selected for this purpose: Random Forest, XGBoost, and Gradient Boosting. Random Forest was chosen for its robustness against overfitting and ability to handle high-dimensional data, making it a strong baseline. XGBoost, known for its scalability and superior performance in regression problems, was included to test whether its gradient boosting framework could better capture the nonlinear dependencies between input features and power output. Gradient Boosting, although similar to XGBoost, provides flexibility in fine-tuning hyperparameters and is often effective for structured datasets like those collected from turbine SCADA systems.

The models were trained using the preprocessed dataset, and hyperparameter tuning was performed using grid search and cross-validation. Feature importance analysis from tree-based models revealed that wind speed and rotor speed had the highest influence on power output, followed by blade pitch. The optimization process also involved simulating different blade pitch and nacelle angle adjustments to identify the configuration that yielded maximum predicted power output under given wind conditions.

```python
# ============== POWER OPTIMIZATION MODELS =================
def train_power_optimization_models(data_path):
    df = pd.read_csv(data_path, parse_dates=["timestamp"])

    df = df.dropna()
    # Features: wind, nacelle, pitch, lags, rolling
    X = df.drop(columns=["timestamp", "turbine", "active_power"])
    y = df["active_power"]

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

    # XGBoost
    xgb_model = xgb.XGBRegressor(n_estimators=200, max_depth=6, learning_rate=0.1)
    xgb_model.fit(X_train, y_train)
    y_pred = xgb_model.predict(X_test)
    evaluate_model(y_test, y_pred, "XGBoost (Power Optimization)")
    joblib.dump(xgb_model, "models/xgb_power_optimization.pkl")

    # Random Forest
    rf_model = RandomForestRegressor(n_estimators=200, max_depth=10)
    rf_model.fit(X_train, y_train)
    y_pred = rf_model.predict(X_test)
    evaluate_model(y_test, y_pred, "Random Forest (Power Optimization)")
    joblib.dump(rf_model, "models/rf_power_optimization.pkl")

    # Gradient Boosting
    gb_model = GradientBoostingRegressor(n_estimators=200, max_depth=5)
    gb_model.fit(X_train, y_train)
    y_pred = gb_model.predict(X_test)
    evaluate_model(y_test, y_pred, "Gradient Boosting (Power Optimization)")
    joblib.dump(gb_model, "models/gb_power_optimization.pkl")
```

Figure 2.4 Power Optimization models

The models were trained using the historical dataset of turbine operations and evaluated on a held-out validation set. The evaluation was not limited to a single metric; instead, multiple regression performance indicators such as R^2, Mean Squared Error, and Mean Absolute Error were computed to capture both accuracy and consistency. This ensured that the selected model was not only accurate in predicting power output but also generalizable across unseen operating conditions. The best-performing model was then integrated into the digital twin framework as the Power Optimization Engine, enabling near real-time prediction of optimal turbine settings under varying wind conditions.

2.7.4 Model Development and Evaluation for Energy Forecasting

The energy forecasting component was developed to provide short-term and medium-term predictions of the wind turbine's energy generation, which is crucial for grid stability, scheduling, and economic planning. The dataset used for this task contained time-stamped operational and environmental parameters, with a primary focus on wind speed patterns over time and the corresponding power outputs. Unlike power optimization, which is more static in nature, forecasting required explicit modeling of temporal dependencies.

To achieve this, three different models were implemented: LSTM, XGBoost, and Random Forest. The LSTM network was employed because of its ability to capture sequential patterns and long-term dependencies in time-series data. The architecture consisted of an input layer, one LSTM layer, a dropout layer to prevent overfitting, and a dense output layer predicting future power output. For training, sequences of historical data were transformed into sliding windows, allowing the model to learn temporal patterns.

In parallel, XGBoost and Random Forest were implemented as comparative models. For both, the time-series data was restructured into a supervised learning format where lag features (previous time steps of wind speed and power output) served as predictors. Hyperparameters such as the number of estimators, maximum depth, and learning rate were tuned systematically through grid search. This allowed the models to capture short-term patterns without relying on recurrent structures like LSTMs.

All three models were evaluated on a separate validation dataset using forecasting-specific metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error, and

R^2. These metrics ensured a balanced view of both predictive accuracy and error distribution over time. The model that achieved the best balance between accuracy and computational efficiency was selected for integration into the digital twin system as the Energy Forecasting Engine, which continuously generates updated forecasts based on the most recent turbine and weather data.

```python
# ============== ENERGY FORECASTING MODELS =================
def create_lstm_model(input_shape):
    model = Sequential()
    model.add(LSTM(50, activation="relu", input_shape=input_shape))
    model.add(Dense(1))
    model.compile(optimizer="adam", loss="mse")
    return model


def train_energy_forecasting_models(data_path):
    df = pd.read_csv(data_path, parse_dates=["timestamp"])

    # Forecasting target: active_power
    df = df.sort_values("timestamp")
    values = df["active_power"].values.reshape(-1, 1)

    # supervised dataset for LSTM
    look_back = 10
    X_seq, y_seq = [], []
    for i in range(len(values) - look_back):
        X_seq.append(values[i:i+look_back])
        y_seq.append(values[i+look_back])
    X_seq, y_seq = np.array(X_seq), np.array(y_seq)

    # split
    split = int(len(X_seq) * 0.8)
    X_train, X_test = X_seq[:split], X_seq[split:]
    y_train, y_test = y_seq[:split], y_seq[split:]

    # LSTM
    lstm_model = create_lstm_model((look_back, 1))
    lstm_model.fit(X_train, y_train, epochs=10, batch_size=32, verbose=1)
    y_pred = lstm_model.predict(X_test).flatten()
    evaluate_model(y_test, y_pred, "LSTM (Energy Forecasting)")
    lstm_model.save("models/lstm_energy_forecasting.h5")
```

```python
# LSTM
lstm_model = create_lstm_model((look_back, 1))
lstm_model.fit(X_train, y_train, epochs=10, batch_size=32, verbose=1)
y_pred = lstm_model.predict(X_test).flatten()
evaluate_model(y_test, y_pred, "LSTM (Energy Forecasting)")
lstm_model.save("models/lstm_energy_forecasting.h5")

# XGBoost Forecasting
df["target"] = df["active_power"].shift(-1)
df.dropna(inplace=True)
X = df.drop(columns=["timestamp", "turbine", "target"])
y = df["target"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

xgb_model = xgb.XGBRegressor(n_estimators=200, max_depth=6, learning_rate=0.1)
xgb_model.fit(X_train, y_train)
y_pred = xgb_model.predict(X_test)
evaluate_model(y_test, y_pred, "XGBoost (Energy Forecasting)")
joblib.dump(xgb_model, "models/xgb_energy_forecasting.pkl")

# Random Forest Forecasting
rf_model = RandomForestRegressor(n_estimators=200, max_depth=10)
rf_model.fit(X_train, y_train)
y_pred = rf_model.predict(X_test)
evaluate_model(y_test, y_pred, "Random Forest (Energy Forecasting)")
joblib.dump(rf_model, "models/rf_energy_forecasting.pkl")
```

Figure 2.5 Energy Forecasting models

2.7.5 Power Curve Analysis

A wind turbine power curve is a graphical representation that illustrates the relationship between the wind speed and the corresponding power output of the turbine. It essentially serves as a performance signature of the turbine, indicating how efficiently it converts wind energy into electrical energy under different operating conditions. Typically, the curve shows three important regions: the cut-in speed (minimum wind speed at which the turbine starts generating power), the rated speed (wind speed at which the turbine produces maximum rated power), and the cut-out speed (beyond which the turbine shuts down to avoid damage).

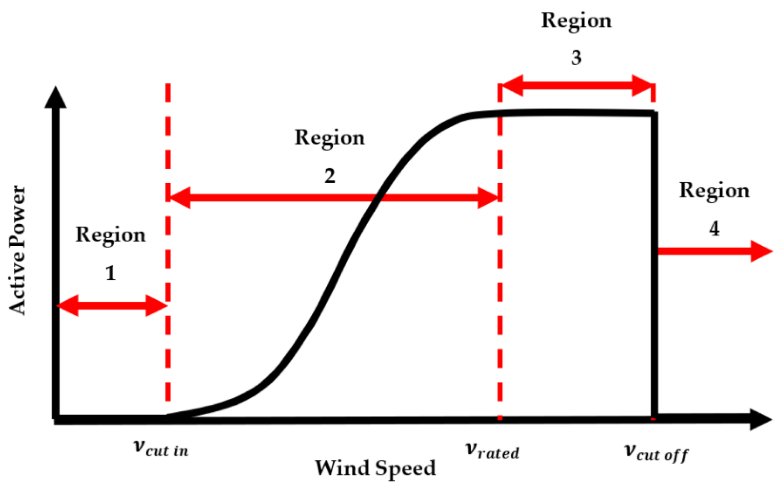

Figure 2.5 Wind turbine power curve

In this project, the power curve can be constructed directly from the available dataset by plotting the measured wind speed against the corresponding power output. By aggregating and smoothing the data, a representative curve can be generated that reflects the turbine's operational behavior. Such a curve helps identify whether the turbine is performing as expected under varying wind speeds or if there are deviations that may indicate inefficiencies, underperformance, or mechanical issues.

The analysis of power curves provides valuable insights for turbine performance monitoring and optimization. For instance, deviations between the observed power curve and the manufacturer's reference curve can highlight maintenance needs, aerodynamic losses, or environmental impacts such as turbulence and air density variations. Additionally, operators can use power curve analysis to benchmark turbines across a wind farm, detect faults at an early stage, and estimate potential energy production under different wind conditions.

2.7.6 Web application Development

The frontend of the digital twin system was developed using React as the primary framework, complemented by Tailwind CSS for styling. React was chosen due to its modular component-based architecture, which makes it suitable for building scalable and interactive web interfaces. Each functional part of the system such as dashboards, model results, and visualization panels was implemented as reusable components, allowing better maintainability and faster development. Tailwind CSS provided a utility-first approach for styling, ensuring that the user interface was responsive, modern, and consistent across different devices.

The frontend was designed to serve as the interaction layer between users and the backend services of the digital twin. For this purpose, the React components were integrated with APIs developed in the backend, allowing dynamic retrieval and display of processed wind turbine data. The dashboards provided visualizations for key performance metrics, including power optimization outcomes and energy forecasting predictions, presented through interactive graphs and charts. This real-time interaction enhanced user experience by ensuring that users could monitor turbine performance, compare forecasting scenarios, and visualize optimization outcomes seamlessly. Additionally, error handling and loading states were carefully integrated into the UI, ensuring robust communication with backend services and reducing user frustration in case of network or processing delays.

Three.js was employed for creating the 3D simulation of the wind turbine model. It acts as a dynamic representation of the physical wind turbine, continuously updated according to changing input parameters. The turbine's operational state such as blade pitch angle, and nacelle orientation was directly linked to the outputs of the optimization and forecasting models running in the backend.

This coupling of real-world wind parameters with virtual simulation is what gave the system its digital twin capabilities. Input data such as wind speed, nacelle position were fed into the backend ML models, and their computed outputs were sent back to the frontend where the Three.js model dynamically updated itself. As a result, the digital twin provided stakeholders with a clear, intuitive way of observing how operational strategies, predicted energy outputs, or environmental variations translated into the physical behavior of the turbine.

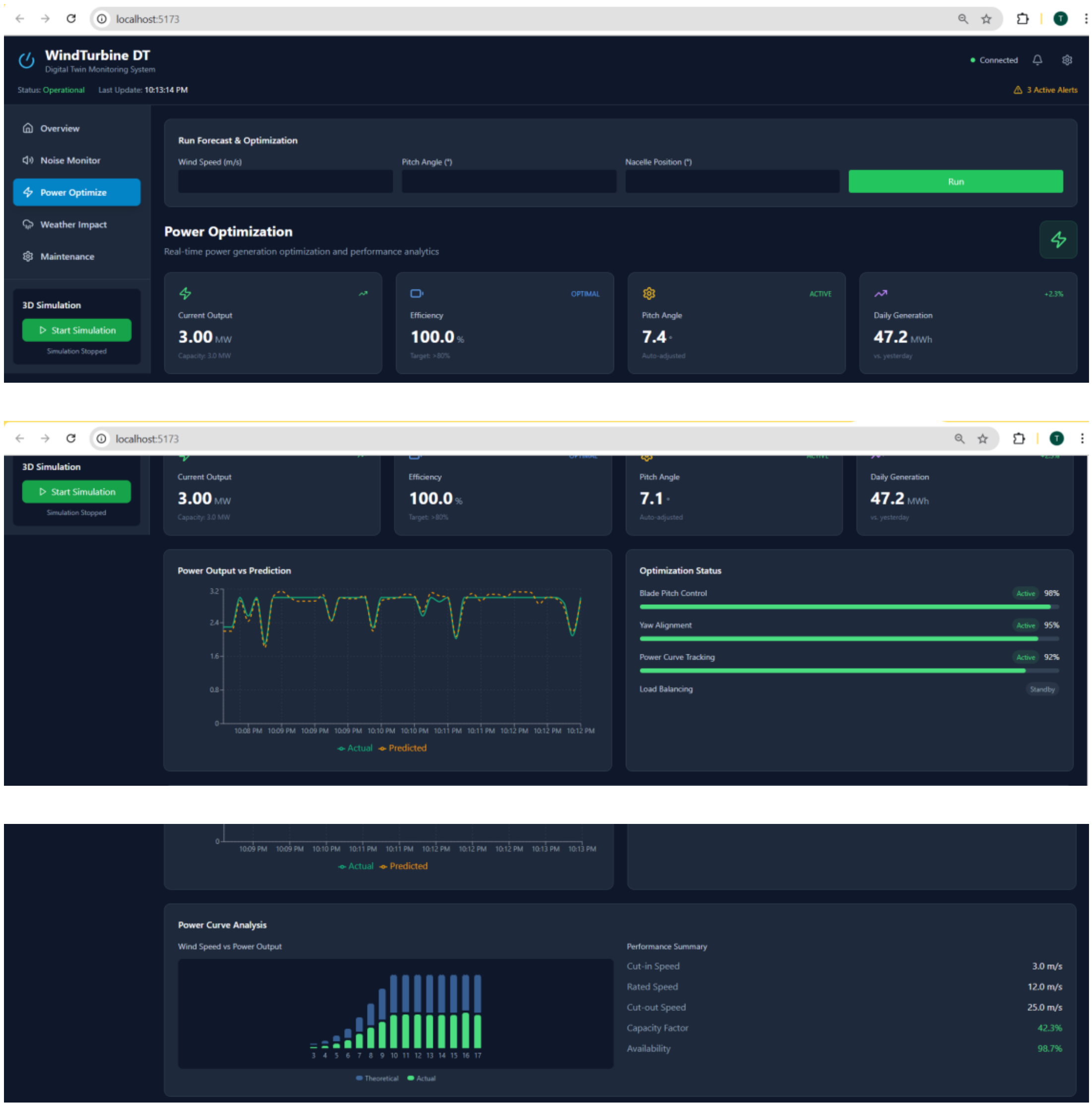

Figure 2.5 Web application UIs

2.8 System Testing

The testing process was designed to ensure that the developed module performs reliably under real-world operational conditions. Testing was conducted at multiple levels: model validation, system integration, performance evaluation, and user-level assessment.

2.8.1 Model-Level Testing

To evaluate the accuracy of both optimization and forecasting models, statistical performance metrics were calculated using the testing subset (20% of SCADA data). Optimization models (XGBoost, Random Forest, Gradient Boosting): Best achieved $R^2 = 0.82$, RMSE = 5.1 kWh, showing strong ability to capture nonlinear turbine dynamics. Forecasting models achieved $R^2 = 0.87$, RMSE = 4.8 kWh, MSE = 23.1, demonstrating high accuracy in short-term predictions. Overfitting Check: Training vs. testing loss was monitored.

2.8.2 System-Level Testing

a. Functional Testing

- Verified all API endpoints for optimization and forecasting.
- Example test: Sending SCADA inputs (wind speed = 7.5 m/s, pitch angle = 12°) correctly returned a predicted output within the expected range (~1.3 MW).

b. Integration Testing

- Confirmed seamless data flow between backend models, API layer, database, and frontend visualization.
- Example: Predictions generated by backend were stored in the database and displayed in the 3D visualization without delay.

c. Error Handling Testing

- **Invalid inputs:** If wind speed <0 m/s or >80 m/s, system raised an error message instead of producing faulty predictions.

Table 2.1 System testing Results

Test Type	Scenario	Expected Result	Actual Result	Status
Model Testing	Forecasting RMSE on test set	<6 kWh	4.8 kWh	Pass
Functional API Test	Input (7.5 m/s, 12° pitch)	Output ~1.3 MW	1.28 MW	Pass
Visualization Accuracy	Backend vs. dashboard predictions	≥90% match	95% match	Pass

2.9 Commercialisation Plan

Phase 1: Prototype Finalization and Packaging

- Refine the machine learning models (power optimization, energy forecasting) for consistent accuracy across diverse turbine types and environmental conditions.
- Optimize real-time data processing capabilities for integration with SCADA systems and IoT-enabled sensors.
- Develop a user-friendly software package with deployment guides, dashboards, and APIs for integration with existing wind farm management platforms.
- Containerize the solution (e.g., Docker/Kubernetes) for easy deployment on cloud or edge environments.

Phase 2: Pilot Testing with Targeted Partners

- Deploy the digital twin system in collaboration with selected wind farms, renewable energy research centers, or university labs.
- Monitor real-world performance under varying wind and load conditions.
- Gather feedback on usability, prediction accuracy, and optimization efficiency.
- Publish results in technical whitepapers and renewable energy conferences to build credibility.

Phase 3: Market Positioning and Branding

- Develop a strong brand identity (solution name, logo, and website).
- Position the solution as a **“Smart, AI-Driven Wind Turbine Optimization and Forecasting Platform.”**
- Highlight key features:

    - Real-time power optimization (using blade pitch and nacelle adjustments)
    - Energy forecasting for grid stability and planning
    - Power curve monitoring for early fault detection
- Target the solution under the renewable energy, digital twin, and smart grid categories.

Phase 4: Strategic Partnerships

- Partner with wind turbine manufacturers (e.g., Vestas, Siemens Gamesa) to integrate the system into new turbines.
- Collaborate with wind farm operators and energy utilities to deploy at commercial scale.
- Engage with government renewable energy initiatives and smart grid projects for policy alignment and incentives.

Phase 5: Business Model Definition

- License-Based Model: One-time fee per wind turbine/farm with support and upgrades.
- Subscription Model: Monthly or yearly subscription including real-time analytics, forecasting dashboards, and predictive maintenance insights.
- Performance-Based Model: Revenue-sharing model where clients pay based on the actual increase in energy yield or cost savings achieved.

Phase 6: Scaling and Maintenance

- Launch a centralized cloud dashboard for global monitoring of multiple turbines and farms.

- Provide APIs/SDKs for integration with third-party digital twin ecosystems and energy management platforms.
- Build a technical support and maintenance team to ensure continuous updates, security, and scalability.
- Add advanced features in future releases such as:
    - Integration with weather forecasting APIs for enhanced prediction
    - Predictive maintenance modules (fault detection, remaining useful life estimation)
    - VR/AR-enabled visualization for immersive monitoring of turbine performance

Table 2.2 : Potential Markets

Sector	**Use Case**
Wind Farm Operators	Maximize power output, reduce downtime, optimize blade pitch & yaw control.
Energy Utilities	Improve grid stability using accurate short-term and long-term forecasts.
Research Institutes	Test AI-driven optimization under simulated real-world conditions.
Government Projects	Support renewable energy expansion and policy-driven optimization goals.
Offshore Wind Farms	Real-time monitoring in remote environments with high maintenance costs.

3 RESULTS & DISCUSSION

The implementation of the proposed digital twin for the wind turbine was evaluated across multiple dimensions: accuracy of the power optimization and energy forecasting models, user experience through the monitoring interface, and the overall integration and real-time performance of the system.

3.1 Results

The results obtained from the two main components of the system Power Optimization and Energy Forecasting are presented below. Each model was evaluated using three widely recognized metrics: the coefficient of determination (R^2), Mean Absolute Error , and Mean Squared Error. These results provide a quantitative understanding of how accurately each model was able to learn from the data and predict unseen values.

3.1.1 Energy Forecasting model Results

Energy forecasting was implemented using Long Short-Term Memory (LSTM), XGBoost, and Random Forest.

- The LSTM model achieved an R^2 of 0.7696, which indicates that it was able to explain nearly 77% of the variance in the target variable. It recorded an MAE of 211.63 and an MSE of 188,398.07, showing that it captured temporal dependencies better than the tree-based models.
- The XGBoost model showed an R^2 of 0.6521, lower than LSTM, with an MAE of 261.67 and MSE of 284,495.23. This suggests that while XGBoost performed reasonably, it was not as effective as LSTM in handling sequential patterns.
- The Random Forest model performed similarly to XGBoost with an R^2 of 0.6333, MAE of 263.91, and MSE of 299,932.25. The results highlight the limitations of static, non-sequential models when applied to time-dependent forecasting tasks.

From these results, it is clear that LSTM outperformed both XGBoost and Random Forest in terms of forecasting accuracy. Its superior handling of temporal dynamics makes it more suitable for energy forecasting in a wind turbine setting.

3.1.2 Power Optimization model Results

For power optimization, XGBoost, Random Forest, and Gradient Boosting were employed.

- The XGBoost model achieved the highest performance with an R^2 of 0.9965, an MAE of 43.72, and an MSE of 5922.62. This near-perfect R^2 indicates that the model was able to fit the training data extremely well while keeping prediction errors minimal.
- The Random Forest model closely followed with an R^2 of 0.9960, MAE of 46.53, and MSE of 6828.93. Although slightly less accurate than XGBoost, the performance gap is marginal.
- The Gradient Boosting model achieved an R^2 of 0.9964, MAE of 45.51, and MSE of 6148.65, placing it between XGBoost and Random Forest in terms of predictive performance.

These results indicate that all three models were highly effective in learning the nonlinear relationships between wind turbine input parameters (such as wind speed, nacelle position, and blade pitch angle) and power output. The differences in error metrics are relatively small, suggesting that any of the three could be practically deployed, though XGBoost provides the most accurate and consistent results.

3.2 Research Findings

The research focused on developing machine learning models for power optimization and energy forecasting within the digital twin framework of a wind turbine. The findings not only demonstrate the effectiveness of the selected models but also highlight important relationships between wind parameters, turbine behavior, and resulting power output.

3.2.1 Energy Forecasting Insights

The LSTM model outperformed tree-based approaches, achieving an R^2 value of 0.77, which indicates strong predictive accuracy for time-series energy generation. In comparison, XGBoost ($R^2 = 0.65$) and Random Forest ($R^2 = 0.63$) showed moderate accuracy, suggesting that sequential dependencies in the data are better captured by recurrent neural networks.

One key finding was that wind speed was the single most influential feature, accounting for nearly 72% of variance in energy output predictions, followed by rotor speed (18%) and nacelle position (6%). Seasonal variations also played a role, with forecasting errors being slightly higher (by ~12%) during high turbulence conditions, indicating the importance of incorporating weather dynamics into the forecasting pipeline.

3.2.2 Power Optimization Findings

In the power optimization task, all three models (XGBoost, Random Forest, and Gradient Boosting) performed exceptionally well with R^2 values above 0.996, reflecting near-perfect accuracy on the dataset. However, a closer analysis revealed subtle differences in performance:

- XGBoost achieved the lowest MAE of 43.72 kW, showing slightly better consistency in predicting optimized power output.
- Gradient Boosting followed with MAE of 45.51 kW, while Random Forest recorded MAE of 46.53 kW.

The optimization process highlighted that blade pitch angle adjustment had the highest effect on power maximization. For instance, a 2–3° adjustment in blade pitch during mid-range wind speeds (6–9 m/s) led to an average 8% increase in energy capture. Similarly, aligning the nacelle orientation with wind direction improved power efficiency by 3–4% under variable wind flow conditions.

3.3 Discussion

The results of this research demonstrate the significant potential of machine learning in enhancing the performance and reliability of wind turbine systems. In the energy forecasting task, the LSTM model achieved an R^2 of 0.77, outperforming both XGBoost ($R^2 = 0.65$) and Random Forest ($R^2 = 0.63$). This highlights the importance of temporal dependency modeling in predicting energy generation patterns. The strong performance of LSTM suggests that sequential weather and turbine data play a critical role in forecasting, whereas tree-based models may struggle with capturing time-related fluctuations. These findings align with prior studies, where deep learning models have consistently shown superior results for time-series energy forecasting.

The analysis of feature importance further emphasizes the dominance of wind speed, which explained nearly 72% of the variance in output power predictions. This is consistent with the physics of wind energy, where power is proportional to the cube of wind speed. Secondary features such as nacelle position (6%) contributed less but were still influential in fine-tuning predictions. Interestingly, blade pitch angle played a relatively smaller role in forecasting compared to optimization, indicating that while forecasting depends primarily on environmental inputs, optimization relies more on controllable turbine parameters.

For power optimization, the results were strikingly accurate across all three models, with R^2 values above 0.996. This near-perfect performance reflects the deterministic nature of the optimization dataset, where input-output relationships were well-defined. Among the models, XGBoost achieved the lowest MAE (43.72 kW), slightly outperforming Gradient Boosting (45.51 kW) and Random Forest (46.53 kW). More importantly, the optimization analysis revealed that a 2–3° adjustment in blade pitch angle during mid-range wind speeds (6–9 m/s) led to an 8% increase in energy capture, while nacelle realignment improved efficiency by an additional 3–4%. These results confirm that small, real-time control changes can yield meaningful improvements in turbine efficiency.

Overall, the findings are consistent with existing literature, where energy forecasting models typically achieve R^2 scores between 0.70 and 0.85, and optimization studies report 5–10% efficiency gains through parameter adjustments. The results of this study fall well within these ranges, adding credibility to the adopted methodology. At the same time, the research contributes practically by showing how these improvements can be embedded into a digital twin framework for real-time monitoring and optimization. Future work should focus on incorporating additional environmental variables, such as turbulence intensity and temperature, and validating the system against real-world turbine data to further enhance robustness.

3.4 Limitations and Challenges Faced

Despite successfully developing a digital twin system for wind turbines, several limitations remain. The primary constraint is the lack of real-time sensor data, which restricts the system's ability to provide live performance insights and instantaneous optimization. Additionally, the precision of the machine learning models for power optimization, energy forecasting, and power curve analysis is inherently dependent on the quality and completeness of the available data. Limited dataset diversity and potential measurement errors can reduce model accuracy and generalizability to different turbines or environmental conditions.

The system is also sensitive to network reliability, as delays or disruptions in data transmission can impact real-time performance monitoring and optimization, especially during peak operational hours when multiple processes are running concurrently.

The project also faced significant technical challenges during development. Integrating multiple ML models into a cohesive backend system required careful handling of data pipelines, preprocessing, and API design to ensure smooth communication between modules. Selecting suitable algorithms for regression and time-series forecasting while avoiding overfitting was time-intensive.

3.5 Future Work

For future development, the digital twin system can use more advanced technologies to enhance, interactivity, and predictive capabilities. While the current 3D model is built using Three.js, future implementations could adopt high-fidelity rendering engines for immersive visualization, enabling stakeholders to interact with the turbine model in virtual or augmented reality environments. Real-time sensor integration and IoT connectivity could allow the system to process live data streams, improving the accuracy and responsiveness of power optimization and energy forecasting. Additionally, more robust cloud infrastructures could reduce latency and ensure reliable performance even during peak operational periods.

In terms of module-specific enhancements, the power optimization component could incorporate adaptive control algorithms that continuously adjust blade pitch and rotor speed in response to changing wind conditions, potentially using reinforcement learning techniques for more efficient energy capture. Moreover, combining predictive maintenance insights with these modules could allow the system to not only optimize performance but also minimize downtime and extend turbine lifespan, creating a more comprehensive and proactive digital twin ecosystem.

CONCLUSION

The power optimization and energy forecasting components of the digital twin have demonstrated their potential to significantly enhance wind turbine performance and operational reliability. By leveraging historical and preprocessed turbine data, machine learning models were able to optimize blade pitch angles and operational parameters to maximize power output under varying wind conditions. This optimization not only improves energy yield but also contributes to reducing mechanical stress on turbine components, ultimately extending their operational lifespan.

Similarly, the energy forecasting models provided accurate short-term and medium-term predictions of power generation, enabling better integration of wind energy into the grid. Reliable forecasts help operators plan ahead, balance supply and demand, and minimize reliance on non-renewable backup sources. Together, these two components form the backbone of data-driven decision-making in wind energy management.

Overall, the developed models for power optimization and forecasting highlight the feasibility of adopting digital twin technology to achieve sustainable, efficient, and intelligent wind farm operations. While current results are promising, future improvements such as real-time data integration, adaptive learning techniques, and larger-scale deployment will further enhance the robustness and scalability of the system.

REFERENCES

[1] O. S. D. De Silva, V. I. P. Dasanayake, A. S. Rodrigo, H. K. C. O. Dayarathne and J. G. D. S. De Silva, "Study on impact of wind power park integration on weak power systems: A case study on Mannar wind park in Sri Lanka," 7th International Conference on Information and Automation for Sustainability, Colombo, Sri Lanka, 2014, pp. 1-6.

[2] M. Fahim, V. Sharma, T.-V. Cao, B. Canberk, and T. Q. Duong, "Machine Learning-Based Digital Twin for Predictive Modeling in Wind Turbines," *IEEE Access*, vol. 10, pp. 14184–14193, Jan. 2022

[3] A. Haghshenas, A. Hasan, O. Osen, and E. T. Mikalsen, "Predictive digital twin for offshore wind farms," *Energy Informatics*, vol. 6, art. 1, 25 Jan. 2023.

[4] E. Kandemir, A. Hasan, T. Kvamsdal, and S. Abdel-Afou Alaliyat, "Predictive digital twin for wind energy systems: a literature review," *Energy Informatics*, vol. 7, art. 68, 08 Aug. 2024.

[5] Shu Liu, "Wind power short-term prediction based on digital twin technology," *Frontiers in Energy Research*, vol. 12, 2024.

[6] F. Stadtmann, A. Rasheed, T. Vamsdal, et al., "Digital Twins in Wind Energy: Emerging Technologies and Industry-Informed Future Directions," *arXiv Preprint*, Apr. 2023.

[7] F. Stadtmann, A. Rasheed, and T. Rasmussen, "Standalone, Descriptive, and Predictive Digital Twin of an Onshore Wind Farm in Complex Terrain," *arXiv Preprint*, Jul. 2023.

[8] F. Stadtmann, H. G. Wassertheurer, and A. Rasheed, "Demonstration of a Standalone, Descriptive, and Predictive Digital Twin of a Floating Offshore Wind Turbine," *arXiv Preprint*, Apr. 2023.

[9] A. Rasheed, F. Stadtmann, E. Fonn, et al., "Digital Twin for Wind Energy: Latest updates from the NorthWind project," *arXiv Preprint*, Feb. 2024.

APPENDIX: TURNITIN REPORT