



DEPARTMENT OF ELECTRONIC AND
TELECOMMUNICATION ENGINEERING

UNIVERSITY OF MORATUWA
SRI LANKA

An End-to-End Framework for
Predictive Maintenance
with AI-powered Digital Twins

FINAL PROJECT REPORT
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE COURSE
MODULE EN4202
MAY 6, 2024

SUPERVISORS:	MEMBERS:	GROUP 23
Dr. Subodha Charles	Adikari A.M.A.D.	190021A
Mr. Shalitha Pathiranage	Peirisipulle T.A.	190443T
	Priyashan B.W.S.	190476V
	Ranatunga R.G.S.M.	190504H

Approval of the Department of Electronic & Telecommunication Engineering

.....
Head, Department of Electronic &
Telecommunication Engineering

This is to certify that I/we have read this project and that in my/our opinion it is fully adequate, in scope and quality, as an Undergraduate Graduation Project.

Supervisor: Dr. Subodha Charles

Signature: . 

Date: . 6 May 2024

Supervisor: Mr. Shalitha Pathiranaage

Signature:

Date: . 6 May 2024

Declaration

This declaration is made on May 6, 2024.

We declare that the dissertation titled An End-to-End Framework for Predictive Maintenance with AI-powered Digital Twins and the work presented in it are our own. We confirm that:

- this work was done wholly or mainly in candidature for a B.Sc. Engineering degree at this university,
- where any part of this dissertation has previously been submitted for a degree or any other qualification at this university or any other institute, has been clearly stated,
- where we have consulted the published work of others, is always clearly attributed,
- where we have quoted from the work of others, the source is always given.
- with the exception of such quotations, this dissertation is entirely our own work,
- we have acknowledged all main sources of help,
- parts of this dissertation have been published.

6 May 2024
.....
DATE

.....
ADIKARI A.M.A.D.
(190021A)

.....

PEIRISPULLE T.A.
(190443T)

.....
PRIYASHAN B.W.S.
(190476V)

.....
RANATUNGA R.G.S.M.
(190504H)

Preface

This report was composed as partial fulfillment of the requirements of Module EN4202 - Project, in the curriculum of B.Sc. in Engineering (Electronic and Telecommunication) at the University of Moratuwa, Sri Lanka. The experience and knowledge that we gained during the final year project were used and were the inspiration to create this report.

Acknowledgment

We would like to express our deepest gratitude to everyone who contributed to the successful completion of this project. First and foremost, we extend our sincere thanks to our supervisor, Dr. Subodha Charles and co-supervisor, Mr. Shalitha Pathiranage, whose expertise and insightful guidance were invaluable throughout the research process. Their patience and profound knowledge in the field encouraged us tremendously.

Our heartfelt appreciation goes out to “Alta Vision Solar”, with whom we had the privilege to collaborate. We are particularly grateful to Mr. Chamodh Madushanka and Mr. Supun Wijesiriwardhana at Alta Vision Solar, who provided insight, resources, and support crucial for our research.

We are also grateful to the faculty and staff of the Department of Electronic and Telecommunication Engineering at University of Moratuwa for providing us with the necessary resources and an encouraging environment to conduct our research.

Additionally, we would like to thank our families, friends and our fellow batchmates for their understanding and endless support during the course of this project.

Lastly, we acknowledge each other’s efforts in this project. The collaboration, commitment, and exchange of ideas among group members were crucial in overcoming challenges and completing this dissertation.

Abstract

This study presents a comprehensive end-to-end framework for Predictive Maintenance (PdM) for components in a Microgrid, with a focus on Vertical Axis Wind Turbines and Energy Storage Systems. The proposed framework integrates the development of tailored hardware for data acquisition, predictive model development, and a Digital Twin (DT) that provides real-time data visualization and future operating condition predictions. We integrate all components of the PdM system, including the data collection hardware, predictive models, and the DT, into a unified application using Microsoft Azure as the backend system. This comprehensive framework serves as a scalable and robust solution for implementing PdM, providing a clear pathway to reduce maintenance costs and improve system reliability in industrial settings. The results indicate that this comprehensive framework can substantially reduce maintenance costs while enhancing system resilience and sustainability, as demonstrated through a real-world use case for wind turbines.

Contents

1	INTRODUCTION	1
1.1	Main Objectives	2
1.2	Organization of the Report	3
2	LITERATURE REVIEW	4
2.1	Data driven approach for Predictive Maintenance	4
2.1.1	Anomaly Detection	4
2.1.2	Fault Identification	5
2.1.3	Remaining Useful Life Prediction	5
2.2	Digital Twins and Cloud Integration	6
3	METHODOLOGY	8
3.1	Introduction	8
3.2	Wind Energy System	8
3.2.1	Hardware Development	8
3.2.1.1	Overview	8
3.2.1.2	Component Selection	10
3.2.1.3	Schematic Design	10
3.2.1.4	PCB Design	13
3.2.1.5	Enclosure Design	14
3.2.1.6	Implementation	15
3.2.2	Firmware Development	15
3.2.3	Machine Learning Model Development	17
3.2.3.1	Anomaly Detection	17
3.2.3.2	Fault Classification	20
3.2.3.3	Remaining Useful Life Prediction	21
3.3	Energy Storage System	26
3.3.1	Hardware Design	26
3.3.1.1	Overview	26
3.3.1.2	Schematic Design	27
3.3.1.3	PCB Design	30
3.3.1.4	Enclosure Design	31
3.3.1.5	Implementation	31
3.3.2	Firmware Development	31
3.3.3	Machine Learning Model Development	33
3.3.3.1	Remaining Useful Life Prediction	33

3.3.3.2	State of Health Prediction	35
3.4	System Integration	37
3.4.1	Azure Integration	37
4	RESULTS	40
4.1	Wind Energy System	40
4.1.1	Anomaly Detection	40
4.1.1.1	Anomaly Detection using Vibration Sensor Data	40
4.1.1.2	Anomaly Detection using SCADA parameters	43
4.1.2	Fault Classification	45
4.1.3	Remaining Useful Life Prediction	45
4.2	Energy Storage System	46
4.2.1	Remaining Useful Life Prediction	46
4.2.2	State of Health Prediction	46
4.3	Digital Twin	48
5	DISCUSSION & CONCLUSION	52
5.1	Discussion	52
5.1.1	Wind Energy System	52
5.1.1.1	Hardware Development	52
5.1.1.2	Machine Learning Model Development	52
5.1.2	Energy Storage System	53
5.1.2.1	Hardware Development	53
5.1.2.2	Machine Learning Model Development	53
5.2	Conclusion	54

List of Figures

2.1 Predictive maintenance road map [1]	4
2.2 Capabilities of a DT [2]	7
3.1 System breakdown of the structure of VAWT	9
3.2 Structure of the VAWT used in the research	9
3.3 Rotation equipment fault vibration artifacts	10
3.4 Power - wind turbine	11
3.5 ESP32 microcontroller - wind turbine	11
3.6 ICM-20948 - Wind Turbine	11
3.7 IIM-42352 - wind turbine	12
3.8 USB Connector - wind turbine	12
3.9 USB to Serial converter - wind turbine	13
3.11 Wind turbine PCB fitted to the enclosure	14
3.12 Final implementation on the wind turbine	15
3.13 Wind turbine firmware architecture	17
3.14 LSTM AE model	18
3.15 VRAE model used	20
3.16 Model implementation for online monitoring	21
3.17 Faults occurred in 2016	22
3.18 Fault distribution among training and testing data [3]	23
3.19 RUL model for each component	24
3.20 1D-CNN model	24
3.21 Equations to calculate costs	25
3.22 Costs of the components	25
3.23 Battery System connected to inverter	26
3.24 Final implementation on the battery system	27
3.25 ESP32 micro-controller - Battery System	27
3.26 USB Connector - Battery System	28
3.27 USB to Serial converter - Battery System	28
3.28 Hex inverter - Battery System	29
3.29 RS485 transceiver - Battery System	29
3.30 RJ45 socket - Battery System	30
3.31 Battery PCB	30
3.32 Enclosure design for battery PCB	31
3.33 Connection to the BMS	31
3.34 RUL prediction model implementation	34

3.35	RMSE (cycles) for the training, primary and secondary test sets	35
3.36	SoH degradation of CS2 batteries	36
3.37	SoH prediction model architecture	36
3.38	Azure services integration	37
3.39	Battery dashboard in Azure Data Explorer	39
3.40	Digital twin for the VAWT and Battery	39
4.1	Vibration signals of set 01	41
4.2	Reconstruction error vs Time	41
4.3	Vibration signals of set 02	42
4.4	Reconstruction error vs Time	42
4.5	Vibration signals of set 03	43
4.6	Reconstruction error vs Time	43
4.7	Training reconstruction loss	44
4.8	Bearing fault reconstruction loss	44
4.9	t-SNE plot	45
4.10	Observed cycle life vs Predicted cycle life	46
4.11	SoH predictions of models	47
4.12	Digital Twin Main Screen	48
4.13	(a) Wind turbine selected (b) Battery system selected	49
4.14	(a) Wind turbine interface when no fault (b) Wind turbine interface during a fault	49
4.15	Battery Screen	50
4.16	Dyamic weather in the DT	50

List of Tables

3.1	Vibration sensor parameters	10
3.2	Dataset split of 2016 T07	19
4.1	Results of anomaly detection	44
4.2	Results of RUL prediction	45
4.3	Cost savings	45
4.4	SoH prediction - Model Evaluation	47

List of Abbreviations and Acronyms

AE Auto-encoders.

AI Artificial Intelligence.

API Application Programming Interface.

ARIMA AutoRegressive Integrated Moving Average.

BMS Battery Management System.

CNN Convolution Neural Network.

DNN Deep Neural Network.

DT Digital Twin.

EDP Energias De Portugal.

FMEA Fault Mode and Effects Analysis.

GAN Generative Adversarial Networks.

GRU Gated Recurrent Units.

IoT Internet of Things.

KNN K Nearest Neighbour.

LSTM Long Short-Term Memory Network.

MG Micro-Grid.

MLP Multi Layer Perceptron.

MSE Mean Squared Error.

PCB Printed Circuit Board.

PdM Predictive Maintenance.

RF Random Forest.

RMSE Root Mean Squared Error.

RNN Recurrent Neural Network.

RUL Remaining Useful Life.

SCADA Supervisory Control and Data Acquisition.

SoC State of Charge.

SoH State of Health.

SVM Support Vector Machine.

SVR Support Vector Regression.

UI User Interface.

VAWT Vertical Axis Wind Turbine.

VRAE Variational Auto-encoders.

Chapter 1

INTRODUCTION

In many industries, unexpected equipment failure is a critical problem that leads to unplanned downtime, labor wastage, increased maintenance costs and loss of productivity. Traditional maintenance strategies, whether scheduled or reactive, are proving inadequate in addressing this issue. Scheduled maintenance also known as preventive maintenance, involves pre-scheduled time intervals to perform equipment checks and replacements which often leads to unnecessary servicing and higher costs. On the other hand, reactive maintenance approach, involves maintenance activities and repairs after an asset breaks down, which fails to detect early signs of impending failures, resulting in costly emergency repairs and under-performing assets.

Predictive Maintenance (PdM) is a better strategy that uses data and analytics to identify and predict when an equipment is likely to fail and then take actions to prevent the failure from occurring. PdM can result in longer asset life, reduced equipment downtime, reduced maintenance costs, increased productivity and worker safety. While the recent development of technologies such as Internet of Things (IoT) is making PdM systems more applicable for monitoring industrial equipment, a significant portion around 30% still remains reliant on the traditional methods of periodic inspections to identify faults [4].

The energy industry is a critical sector where equipment downtime can have severe consequences, impacting not only the economy but also public safety and energy supply stability. As the industry transitions to renewable energy sources like wind, solar, and hydroelectric power, maintenance needs have become more complex and specialized. Efficient maintenance strategies are crucial for minimizing downtime and ensuring the reliable delivery of renewable energy, ultimately reducing costs and promoting sustainability in the long run. A Microgrid (MG) is a localized energy system that typically consists of various distributed energy resources, such as solar panels, wind turbines, and battery storage, along with conventional energy sources like diesel generators. MGs provide a flexible and scalable method of managing energy and are being utilised more and more to facilitate the integration of renewable energy sources, lower energy prices, and improve grid resilience [5]. Efficient maintenance procedures such as PdM contributes to the sustainability and cost-effectiveness of MGs. By addressing issues early and minimizing unplanned downtime, MGs can operate more efficiently, reducing energy waste and lowering operational costs resulting a more sustainable energy ecosystem.

Recent advancements in Artificial Intelligence (AI), Machine Learning and IoT in Indus-

try 4.0 have enabled the creation of digital mirrors of components or systems that offer a detailed view on their current and future operating conditions [6]. This concept is known as a Digital Twin (DT) and can be defined as “*software-based abstractions of complex physical systems that are connected to the real system via a communication link to continuously exchange data with the real environment and establish a dynamic digital mirror with a constantly running modeling engine*” [7]. A DT serves as an exceptional tool for monitoring MGs, offering a comprehensive and detailed perspective on their operations. With the integration of IoT technology, collecting data becomes more accessible and efficient, allowing for seamless data acquisition from various components within the MG. This data forms the backbone of a data-driven model, which accurately represents different physical entities in the system. By leveraging this virtual representation, one can closely monitor the MG’s performance, identify trends, and predict potential issues, leading to more informed decision-making and enhanced system reliability.

This research aims to develop a comprehensive end-to-end framework for PdM applied to MG components; a Vertical Axis Wind Turbine (VAWT) and an Energy Storage System. The backend system integration will be built on Microsoft Azure. The project includes creating a DT to visualize real-time data from MG components and to predict their future operating conditions. This DT will facilitate the monitoring and analysis of the components, enabling timely maintenance and improved system reliability.

1.1 Main Objectives

We introduce an end-to-end framework for PdM combined with DTs leveraging the power of AI, ultimately transforming MG components into more reliable and cost-effective energy solutions.

Primary objectives of the research will be as follows,

1. **Development of Customized Hardware :** Involves creating hardware specifically designed to gather data from the VAWT and Energy storage system tailored to the needs of a PdM system.
2. **Development of PdM Algorithms and Models :** Entails creating machine learning algorithms and predictive models capable of identifying patterns and predicting potential failures in the given machinery.
3. **Demonstration of Cost Reduction through PdM :** Showcase how a PdM application can lead to significant cost savings compared to traditional maintenance procedures.
4. **Development of an End-to-End Framework with a DT :** Building a comprehensive framework that integrates all aspects of the PdM system, from data collection to predictive analysis. Additionally, this framework will feature a DT, providing a real-time visualization and the future operating conditions of the components.

1.2 Organization of the Report

The subsequent chapters of this report are structured as follows: Chapter 2 presents an in-depth review of existing literature related to PdM, DTs, and Machine Learning concepts, highlighting key insights and developments in these fields. Chapter 3 describes the methodology, divided into two main parts: the implementation on wind energy systems and energy storage systems. Chapter 4 presents the results of the study, while Chapter 5 discusses these results and provides conclusions drawn from the research.

Chapter 2

LITERATURE REVIEW

In this section, we explore recent advancements, focusing on developments in machine learning algorithms, the integration of sensors and IoT devices, and the use of DT for PdM.

2.1 Data driven approach for Predictive Maintenance

Data-driven PdM in industrial applications typically follows a sequential four-stage process [1]. The initial phase involves monitoring the system to identify any deviations from normal operational patterns, which is Anomaly detection. Once an anomaly is detected, the next step is to diagnose its nature. Prognosis in PdM focuses on estimating the Remaining Useful Life (RUL) of machinery components after an anomaly has been detected and diagnosed. Finally, once an anomaly is detected, its cause diagnosed, and its future impact prognosticated, sufficient information is available to undertake early maintenance actions to mitigate the risk of failure.

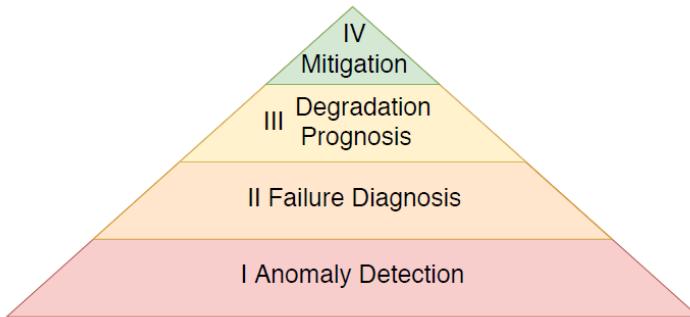


Figure 2.1: Predictive maintenance road map [1]

2.1.1 Anomaly Detection

In real-world scenarios where labeled data is unavailable, unsupervised machine learning models are well-suited for application. Anomaly detection can be categorized into two types based on the features being measured: *univariate anomaly detection*, which examines individual features for anomalies, and *multivariate anomaly detection*, which considers

patterns across multiple features. Traditional methods for *univariate anomaly detection* generally employ handcrafted features to model normal and anomalous event patterns [8], using techniques such as wavelet analysis [9], Singular Value Decomposition (SVD) [10], and AutoRegressive Integrated Moving Average (ARIMA) [11].

Models for *multivariate anomaly detection* can be categorized into Forecasting-based Models and Reconstruction-based Models [12]. A forecasting-based model identifies anomalies by evaluating prediction errors [13]. Gugulothu et al. combine non-temporal dimensionality reduction techniques with Recurrent Autoencoders for time-series modeling in an end-to-end learning framework [14]. Study [15] proposes a Recurrent Neural Network model, specifically a bidirectional Long Short-Term mMemory (bi-LSTM) model to forecast a time series of gearbox shaft temperature from a healthy state, enabling the detection of anomalies by comparing the predicted values with the actual measurements on the EDP wind dataset [16].

A reconstruction-based model captures the representation of an entire time series and learns to reconstruct the original input using a collection of latent variables. Study [17] proposes an encoder-decoder framework based on Long Short-Term Memory (LSTM) to learn representations from normal time-series data for the purpose of anomaly detection. The authors of [18] used a denoising autoencoder and novel convolutional neural networks for condition monitoring and performance forecasting on EDP wind dataset [16]. Auto-encoder (AE) models learn to compress and then reconstruct the input data, allowing the model to capture the underlying structure of normal instances and highlight anomalies through reconstruction errors. This capability makes AE particularly valuable for continuous monitoring and detection of subtle changes that may indicate potential failures or security breaches in various systems [1]. The key advantage of AEs is that they do not require labeled classification or failure data, making them ideal for unsupervised learning environments. This capability facilitates online condition monitoring, allowing for real-time insights into equipment status.

2.1.2 Fault Identification

Once an anomaly has been detected, the next step is to identify the nature of the anomaly. This task requires solving a classification problem. Results can be interpreted by grouping the detected anomalies based on their similarity, achieved using clustering models [1]. Machine learning algorithms such as Random Forest (RF), Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (k-NN) algorithms have been used in the classification of faults facilitating PdM [19]. Also by using the Supervisory Control and Data Acquisition (SCADA) data from a PV Cell plant, Alessandro Betti et al. used an unsupervised clustering algorithm and a pattern recognition Neural Network in generic status prediction of the cells and specific fault class predictions. case [20].

2.1.3 Remaining Useful Life Prediction

After detecting an anomaly and diagnosing the fault type, the next step involves estimating how much longer the component will remain operational. This task is typically performed by RUL models, which estimate the remaining time or cycles until a failure occurs [21]. Y. LI,

T.R. Kurfess *et al.* estimate the future lifespan of a rolling element bearing by introducing randomness into a previously deterministic model of defect propagation. This is achieved by incorporating a log-normal random variable, making the prediction more flexible and probabilistic in nature [22]. Specifically more advanced deep learning approaches, such as Recurrent Neural Networks (RNNs), Convolution Neural Networks (CNNs) and Generative Adversarial Networks (GANs) have been utilized in PdM literature. RNNs can be used for time series prediction tasks, making them suitable for predicting equipment failures or maintenance needs. Wu *et al.* utilized Long Short-Term Memory Networks (LSTMs) to carry out the RUL estimation for aircraft turbofan engines [23]. The study [4] presents a method for predicting the RUL in the EDP wind dataset, aiming to forecast component faults over the next 60 days. Various machine learning models are evaluated for this task, including RF, Multi-Layer Perceptron (MLP), and Support Vector Regression (SVR). The study also proposes a maintenance strategy that can potentially reduce the maintenance cost of wind turbine systems. The studies [24], [25] present a method for predicting RUL of a battery in terms of a number of cycles by analyzing the State of Health (SoH) values of the battery. In these studies, a statistical model combined with a machine learning model has been used to make predictions.

2.2 Digital Twins and Cloud Integration

In existing literature, three primary approaches stand out when it comes to designing a DT: the physics-based model, the data-driven model, and the hybrid model.

(a) Physics-based Approach

A physics-based DT is created using mathematical equations and physical laws to simulate its behavior and characteristics. Selçuk, Şahan Yoruç *et al.* used a physics based DT model developed using MATLAB/ Simulink for a motor and a gearbox to generate synthetic vibration data [26]. A physics-based Fast Run Model was developed for real-time power plant simulations, accurately replicating air-path pulsations and offering detailed, crank angle-resolved combustion simulations. This Fast Run Model seamlessly integrates with a comprehensive electrical system model, including battery storage components, creating a unified simulation framework for power plant operations [27].

(b) Data-driven Approach

A data-driven model use large amounts of historical data to capture and formulate hidden physics of a system on their own. These models can predict from data even if the underlying physics are unknown. This versatility significantly broadens their range of applicability, making them suitable for scenarios where traditional physics-based models fall short. Moreover, their efficiency shines through in their capacity to deliver real-time predictions, far surpassing the speed of conventional physics-based models [28].

(c) Hybrid Approach

Hybrid-based approaches represent a fusion of the strengths of both physical and digital models, capitalizing on their respective advantages to achieve superior performance and versatility. Werner *et al.* proposed a comprehensive PdM strategy in their work, which involves a hybrid approach that combines both data-driven and physics-based modeling techniques to estimate the RUL of systems or components [29].

The study [30] highlights the diverse capabilities of a DT. As shown in figure 2.2 a DT can be standalone, offering a basic representation of a system; descriptive, providing detailed visualization and real-time monitoring; and predictive, using data and analytics to forecast future states and system behaviors. Study [31] introduces an architecture that integrates MATLAB as a DT and Azure micro-services to implement the DT within a PdM framework. A DT platform architecture for monitoring and controlling wind farms using Unity3D and OPC-UA is discussed in [32]. The proposed framework in the study [33] connects a real Festo Cyber-Physical Factory for Industry 4.0 with its digital representation, enabling the implementation of a Predictive Maintenance model. This connection compensates for the lack of run-to-failure data by conducting error simulations on both the physical asset and its Digital Twin.

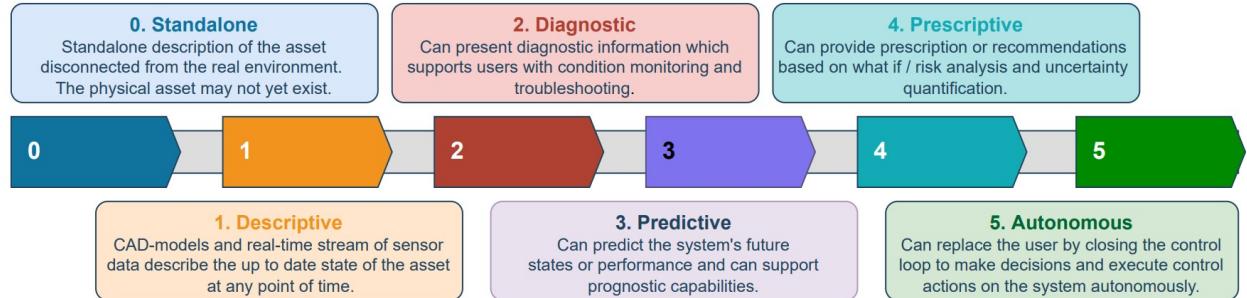


Figure 2.2: Capabilities of a DT [2]

Chapter 3

METHODOLOGY

3.1 Introduction

The proposed framework integrates a comprehensive PdM strategy within a MG that consists of a VAWT, an energy storage system featuring 10 ROSEN batteries connected to an inverter, and a solar energy system. The primary focus of our project is on the PdM of the VAWT and the energy storage system within the MG. The VAWT, with its variable-speed, gearless, direct-drive generator, requires precise maintenance strategies that can predict and prevent potential failures before they affect system performance. Similarly, the energy storage system, comprised of 10 the batteries, is monitored through advanced algorithms that analyze critical parameters such as charge cycles and temperature. This targeted approach allows for detailed insights into the health of the batteries, enabling preemptive actions to mitigate wear and extend battery life.

3.2 Wind Energy System

3.2.1 Hardware Development

3.2.1.1 Overview

One objective of this research is to create custom hardware for the wind turbine system and the energy storage system. This hardware must be designed to collect data that is relevant for a PdM application. To identify which components of the VAWT are prone to failure and understand the root causes, a Fault Mode and Effects Analysis (FMEA) was conducted. FMEA is a widely used engineering technique for designing, identifying and eliminating potential/known system failures. This analysis helped pinpoint the most vulnerable subsystems and provided insight into potential sources of failure, guiding the development of the customized hardware. Figure 3.1 shows the typical system breakdown of a VAWT [34]. The VAWT used in this research is much simpler in design and is of type direct driven generator, hence does not contain a gearbox (Figure 3.2). Mechanical, electrical, and thermal faults were identified as the primary types of faults that could occur in this wind turbine [35]. However, due to the unique structure of the VAWT, it was not feasible to install

temperature sensors, which would typically be used to detect thermal faults. Vibration data was chosen as the key parameter for monitoring mechanical faults. To address electrical faults, the current and voltage generated by the VAWT will be collected and analyzed. By focusing on vibration, current, and voltage data, the system aims to detect and address potential faults effectively, providing a comprehensive approach to PdM. A custom Printed Circuit Board (PCB) was designed to facilitate the acquisition of vibration sensor data.

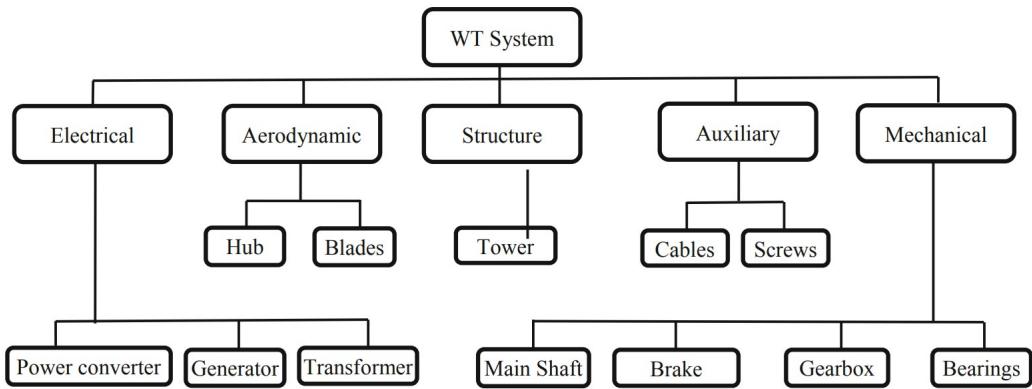


Figure 3.1: System breakdown of the structure of VAWT

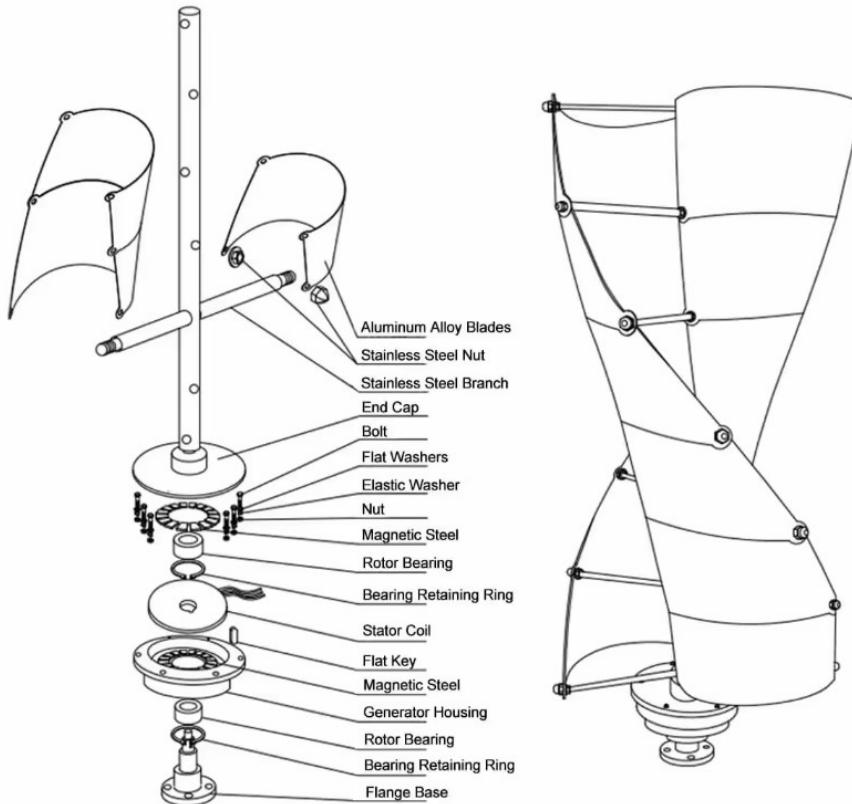


Figure 3.2: Structure of the VAWT used in the research

3.2.1.2 Component Selection

Selecting a suitable vibration sensor for a PdM application is crucial for accurate data collection and fault detection. To find the best sensor, a thorough search was conducted, focusing on key selection criteria such as noise floor, bandwidth, and range. These factors are critical in determining the sensor's sensitivity, frequency response, and ability to capture a wide range of vibrations [36]. A sensor with a low noise floor ensures high signal clarity, while a broader bandwidth allows for detecting a variety of vibration frequencies. The range criterion ensures that the sensor can measure both low and high levels of vibration. Figure 3.3 represents the frequency range of vibrations on different faults in rotating machines [37]. This shows most faults can occur at lower frequencies, especially in VAWT and similar low RPM applications.

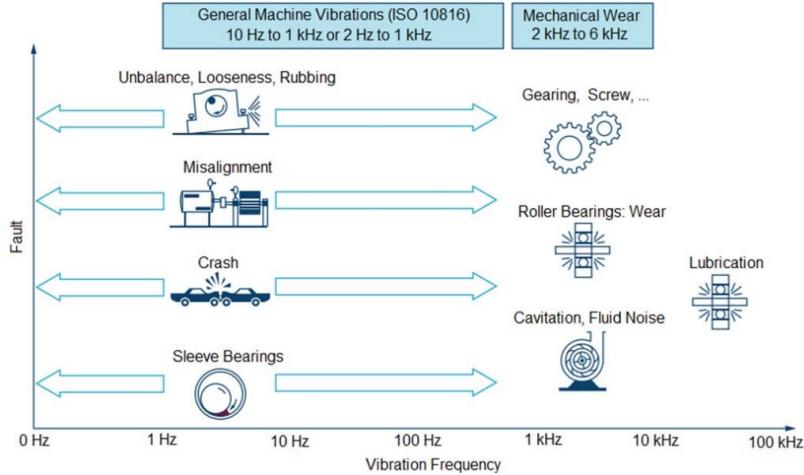


Figure 3.3: Rotation equipment fault vibration artifacts

Taking into account all these factors, including the ability to detect low-frequency vibrations, cost-effectiveness, and overall performance, two vibration sensors were ultimately selected for the PdM application: STMicro's IIS3DWBTR and TDK's IIM-42352. IIM-42352 is used as the vibration sensor in the PCB. Along with this sensor a ICM-20948 9-axis MotionTracking sensor is used to calculate the tilt angles of the VAWT's pole.

Selected vibration sensors		
Parameters	IIS3DWBTR	IIM-42352
Bandwidth (kHz)	6	4
Noise ($\mu\text{g}/\sqrt{\text{Hz}}$)	75	70

Table 3.1: Vibration sensor parameters

3.2.1.3 Schematic Design

The schematic design for the PCB, created using Altium Designer, consists of several modules, each tailored to fulfill specific functions within the system.

- Power - PCB is powered by 230V AC power which will be converted 5V by HLK-5M05 transformer and then 5V to 3.3V by using AMS1117 regulator.

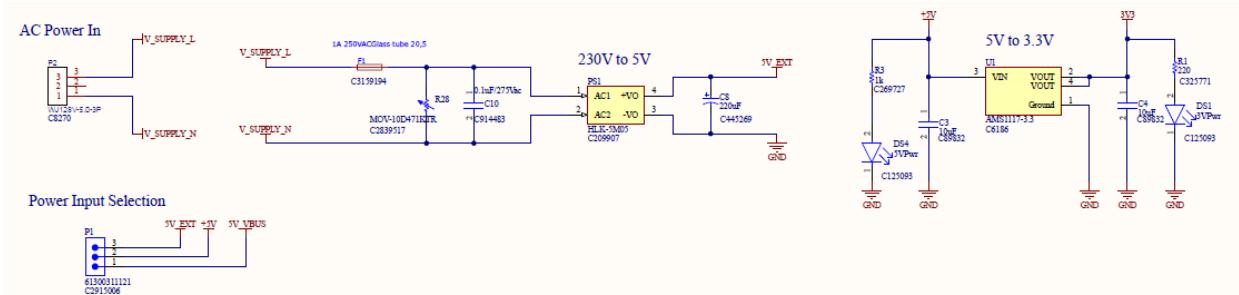


Figure 3.4: Power - wind turbine

- ESP32 - The main micro-controller unit used in the PCB is an ESP32-WROOM-32E.

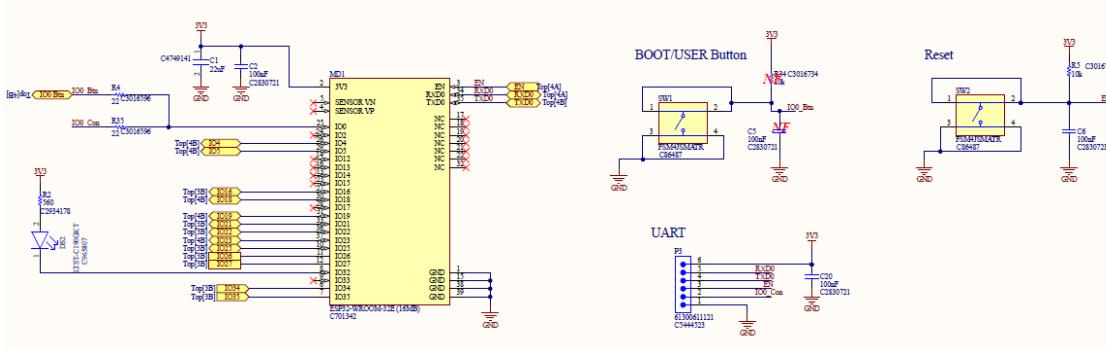


Figure 3.5: ESP32 microcontroller - wind turbine

- ICM Vibration Sensor - One ICM-20498 vibration sensor is used, powered by 1.8V obtained by converting 3.3V to 1.8V using a TC1017 regulator.

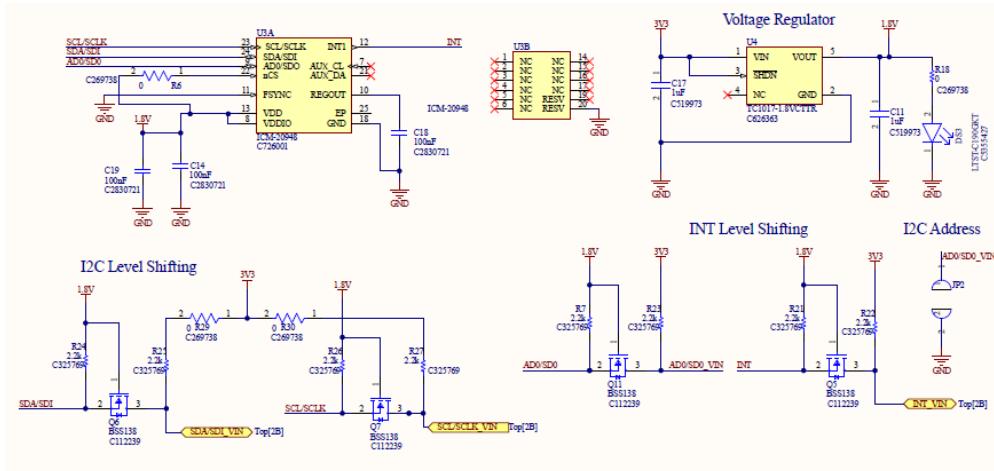


Figure 3.6: ICM-20948 - Wind Turbine

4. IIM Vibration Sensor - Two IIM-42352 vibration sensors are used, powered by 3.3V

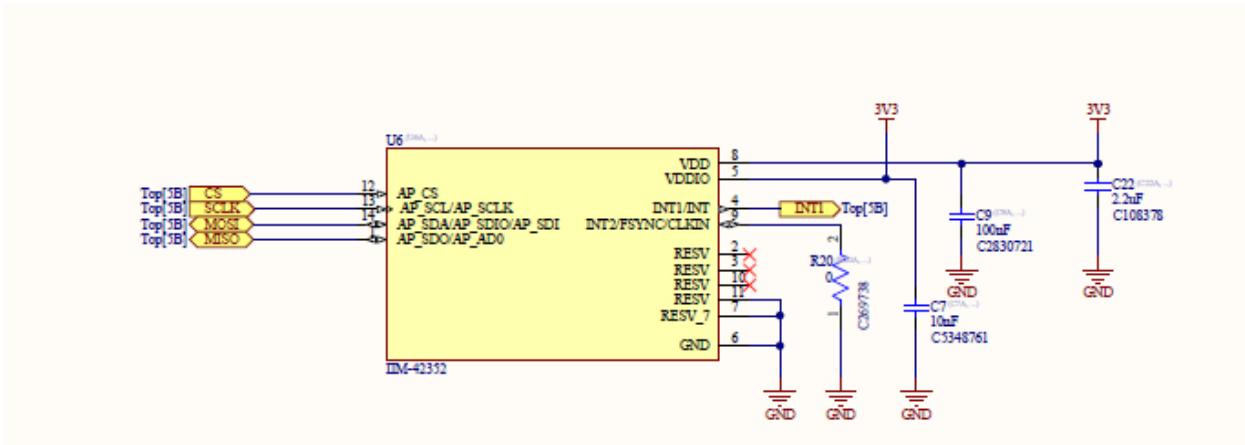


Figure 3.7: IIM-42352 - wind turbine

5. USB Connector

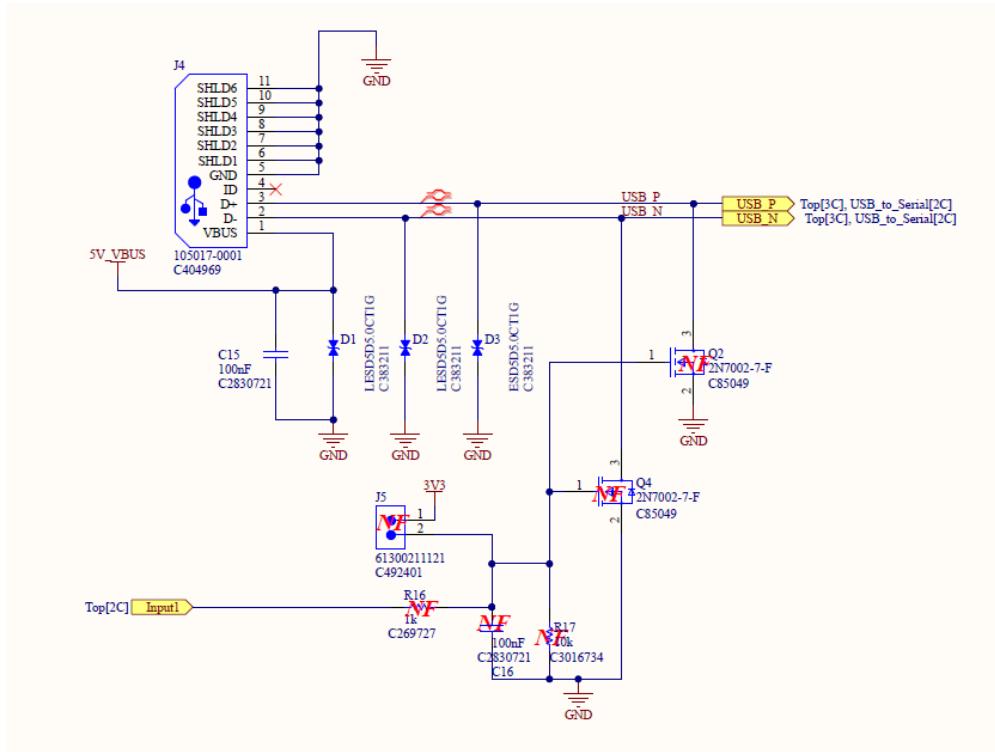


Figure 3.8: USB Connector - wind turbine

6. USB to Serial

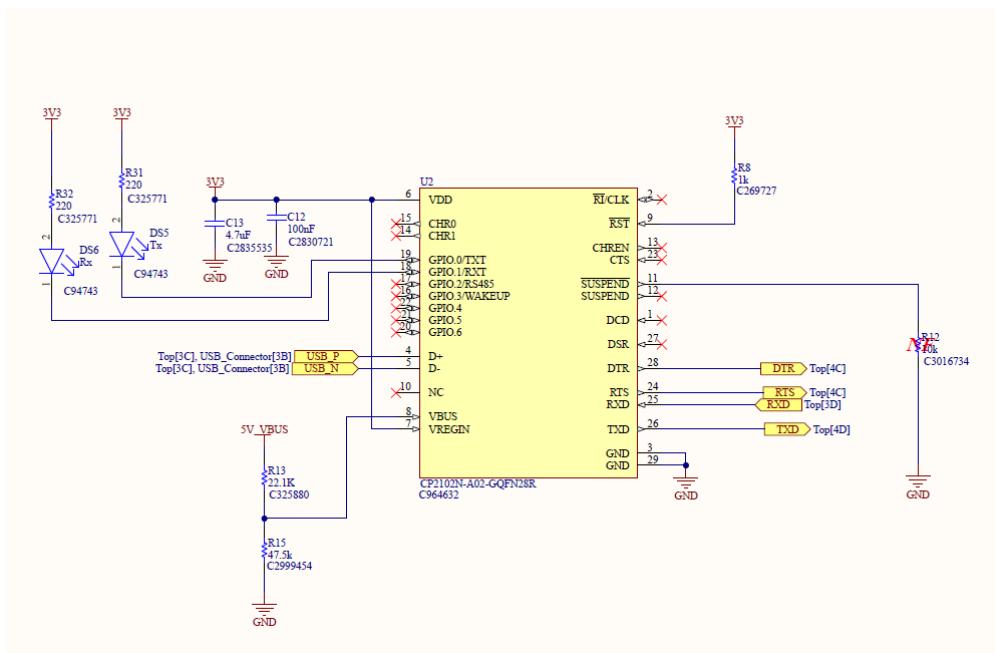


Figure 3.9: USB to Serial converter - wind turbine

3.2.1.4 PCB Design

The PCB is a carefully designed 4-layer board, built to endure the harsh conditions typical of wind turbine environments.

Layer stackup - 4 layers

- Top Layer (Signal)
- Inner Layer 2 (Ground)
- Inner Layer 1 (Power)
- Bottom Layer (Signal)
- Microcontroller and Sensor Placement:

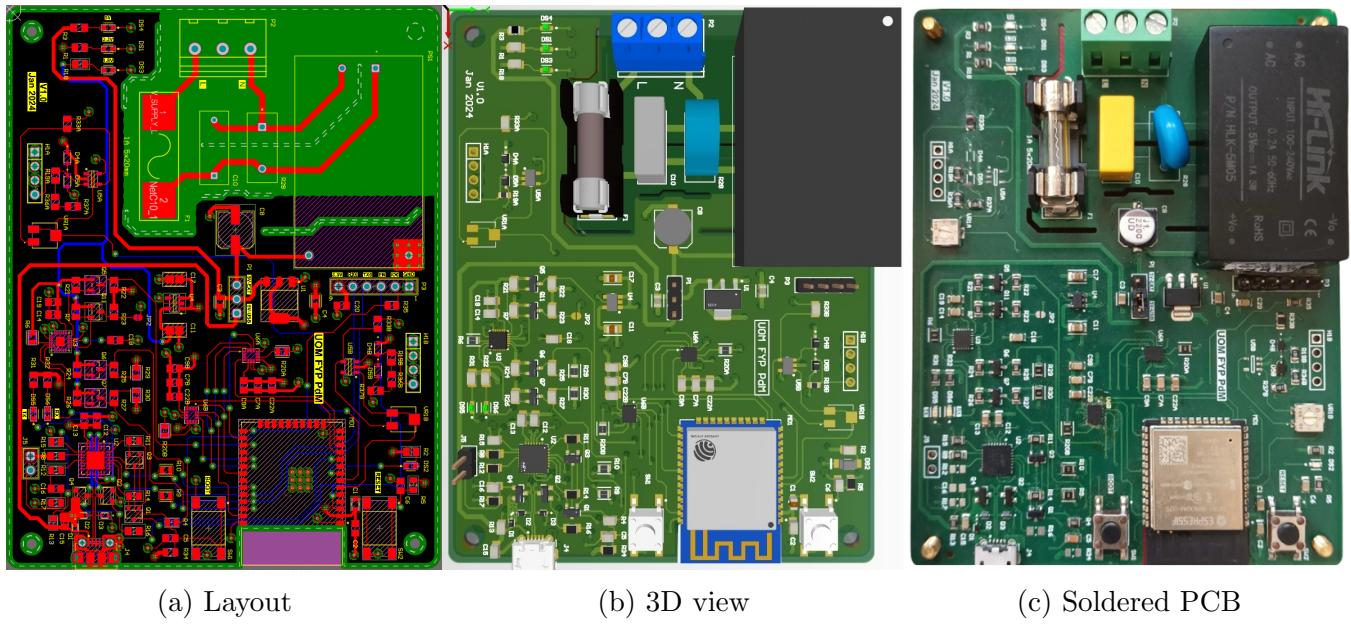
The heart of the data acquisition module is an ESP32 microcontroller, strategically placed at a corner of the PCB. This placement optimizes the ESP32's connectivity with peripheral devices while allowing for efficient heat dissipation and easier access during maintenance. Adjacent to the microcontroller are the vibration sensors, positioned to optimize signal pathways and minimize latency and noise in data capture. The optimal placement of these sensors is critical for accurate vibration analysis, as it directly impacts the quality of the data collected.

- High and Low Power Separation:

PCB functions with a 230V power supply and to mitigate interference and potential risks, the PCB layout distinctly separates high-power and low-power sections. This

separation is crucial in preventing noise and heat transfer from high-power components affecting the performance of low-power circuits, particularly the sensitive signal pathways associated with the vibration sensors.

- Differential Pair Routing and USB Connectivity: Given the necessity for reliable data communication, the PCB incorporates differential pair routing to enhance the signal integrity of the USB connections. This routing technique is essential for reducing electromagnetic interference and crosstalk, thereby ensuring stable and clear data transmission.



3.2.1.5 Enclosure Design

The PCB was designed to fit into a commercially purchased enclosure, which is IP65 - water resistant and is made from durable ABS plastic. Metal fittings were used to securely attach the enclosure to the highest possible position on the wind turbine's pole.



Figure 3.11: Wind turbine PCB fitted to the enclosure

3.2.1.6 Implementation

Following the design phase, the PCB underwent a series of tests to validate its performance. The testing phase focused on evaluating the PCB's electrical safety, signal integrity, and operational stability, ensuring that the board meets all required specifications and industry standards.



Figure 3.12: Final implementation on the wind turbine

3.2.2 Firmware Development

The main purpose of this firmware is to collect data from the sensors and send them to the cloud. Arduino IDE was used to develop the firmware. The main components of the firmware are as follows.

Data Reading

In the wind turbine PCB, an ICM20948 and two IIM42352 sensors were used to obtain tilt angles and vibrations respectively.

- ICM20948 data reading: "SparkFun 9DoF IMU Breakout - ICM 20948" Arduino library was used to read data from this sensor with I^2C communication protocol. This sensor reads three euler angles: roll, pitch and yaw.

- IIM42352 data reading: This sensor was completely accessed by reading and writing to the sensor registers. Vibrations were obtained by reading acceleration X,Y and Z data from *ACCEL_REG* acceleration data registers. SPI communication was used to read and write data.

Data collecting and Data string creation

Data from the above sensors is collected at a rate of 30 Hz and sent to the cloud at 5-second intervals. In a 5-second interval, there are 150 data rows, each with 9 parameters (3 from ICM20948 and 6 from two IIM42352 sensors). Each 5-second interval begins with the generation of an empty data string, which is then filled with the data from each data row. Following the interval, the data string is sent to the cloud.

WiFi Handling

The WiFi connection of ESP32 is used for the data communication between the ESP32 and the cloud. "WiFiManager" Arduino library was used for this purpose. The connection is established at the beginning of a chip restart. If the WiFi connection is lost, then the ESP32 continuously attempts to reconnect to the network.

MQTT Telemetry Message Transmission

The MQTT protocol was used to send telemetry messages from the ESP32 to the cloud. ESP32 communicates directly with Azure IoT Hub service via this MQTT protocol. Here, "Azure SDK for C" Arduino library was used. When the ESP32 restarts, it establishes an IoT Hub client and a MQTT client after connecting to WiFi. If the WiFi connection is lost, this MQTT connection will also be dropped. If the MQTT connection is lost without a WiFi disruption, the ESP32 will attempt to reconnect in 10-second intervals and if this fails for 1 minute, the ESP32 will restart and establish all connections from the beginning. If the WiFi and MQTT connections are successful, the MQTT client sends the created data string to the Azure IoT Hub at 5-second intervals. This MQTT client employs a QoS (Quality of Service) value of 1 to avoid requiring an acknowledgment message confirming the successful transmission of telemetry messages. This helps to maintain a stable connection.

Data string buffer

A buffer is used to store data strings in case of WiFi or MQTT connection failure. This buffer can store up to 10 data strings and if there are more to store, it eliminates the old data strings and stores the latest ones.

Over-the-air (OTA)

The over-the-air method enables remote firmware uploading to the ESP32 over WiFi, allowing firmware upgrades or changes in credentials.

The ESP32 contains two cores: core 0 and core 1. In this firmware, both cores are used to perform tasks. The core 1 is responsible for collecting data from the sensors and generating the data string. After a 5-second interval, it enqueues the data string in the data string buffer. Then if the WiFi and MQTT connections are successful, core 0 will dequeue the data strings in the buffer and transmit them to the Azure IoT Hub. Fig.3.13 shows the architecture of the firmware of wind turbine PCB.

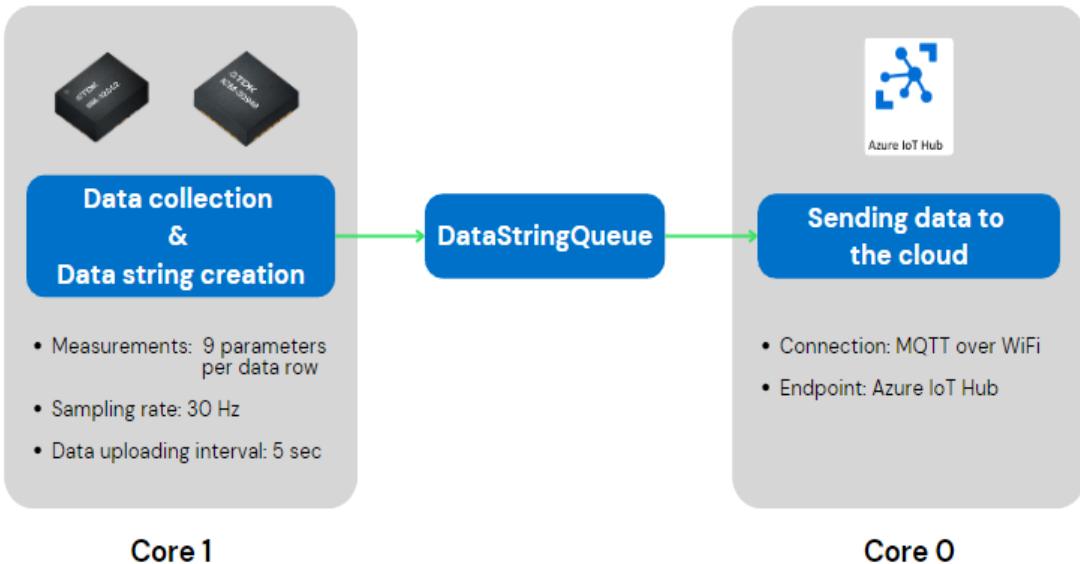


Figure 3.13: Wind turbine firmware architecture

3.2.3 Machine Learning Model Development

3.2.3.1 Anomaly Detection

The initial phase of the PdM process is anomaly detection. This crucial step involves identifying early signs of abnormal conditions in turbine components, potentially leading to failures if unaddressed. Effective anomaly detection provides a proactive means to schedule maintenance, reduce downtime, and extend the operational lifespan of the turbines.

Given the criticality of early fault detection, this section specifically focused on using vibration sensor data and SCADA parameters for anomaly detection.

NASA Bearing Vibration Dataset

Vibration analysis is a proven method for early-stage anomaly detection in rotating machinery like wind turbines, where abnormal vibrations can be indicative of issues such as imbalance, misalignment, or mainly bearing faults.

Although the specific vibration data from our custom-designed PCB was not used in the initial model training phase, we employed the NASA bearing vibration dataset for Machine Learning Model Development. This dataset provided three sets of run to failure test data collected over three different periods of time. [38]

- Data from Center for Intelligent Maintenance Systems (IMS), University of Cincinnati.
- Vibration data of four bearings installed on a shaft
- File Recording Interval: 1-second vibration signal snapshots recorded at every 10 minutes intervals. Each file consists of 20,480 points with the sampling rate set at 20 kHz.

Model Development for Vibration Data

To address the challenge of anomaly detection, we developed a generalized machine-learning model capable of processing and analyzing vibration data. This model is designed to be adaptable and can be implemented once data collection commences.

For anomaly detection using Vibration signals, the LSTM AE model shown in Figure 3.14, is trained only on normal operational data. This approach allows the model to establish a baseline of typical behavior for the machinery. The primary focus during training is on minimizing the reconstruction loss, which measures the model's ability to reproduce the input data after compressing and decompressing it.

Upon completion of the training, the threshold for anomaly detection is set based on the maximum reconstruction loss observed in the training dataset. This threshold acts as the critical value, with any data exceeding this threshold considered anomalous, signaling potential issues or failures. By implementing this threshold, the model can effectively identify deviations from normal operations well before an actual failure might occur. This capability facilitates timely maintenance interventions, potentially preventing more severe malfunctions and reducing operational downtime.

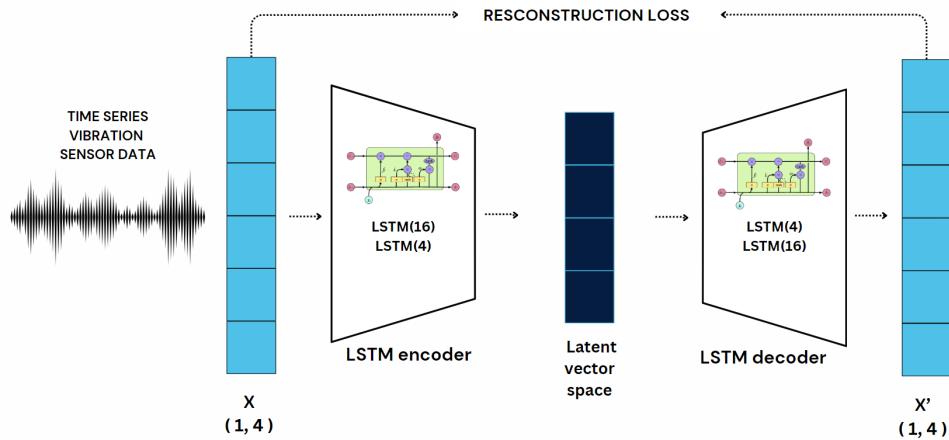


Figure 3.14: LSTM AE model

EDP Wind Dataset

In this section we utilize a public dataset from Energias De Portugal (EDP), derived from the Wind Turbine Failure Detection Challenge, to support our research [39]. This dataset contains SCADA records spanning two full years, from 2016 to 2017. The current version

of the dataset contains data from four wind turbines, collected at 10-minute intervals. The dataset also provides time steps where the faults have occurred in the corresponding wind turbines. The dataset contains 81 monitored parameters from major components including gearbox, generator, generator bearing, transformer, and hydraulic group. As the data comes from actual wind turbines, the dataset reflects real-life challenges, providing a solid base for developing and testing PdM models. The results obtained will be reliable and suitable for real-world applications, like wind turbines or similar equipment. Since the dataset does not have any labeled data, unsupervised machine learning methods will be used to analyze it. Here we show how Anomaly detection models can be used to extract features of fault data.

Data Preprocessing

Initially, the 2016 dataset was divided into separate datasets for each wind turbine. For each turbine's dataset, the data was further segmented based on fault times, using fault time as breakpoints to split the raw data into distinct subsets, as illustrated in Table 3.2. To create training and test datasets, each subset was split such that the first 70% of the data was designated as the training set, while the remaining 30% was designated as the test set which is same method used in [18].

Dataset	Starting time	Ending time	Fault logs		
			Component	Timestamp	Remark
1	2016-01-01 00:00:00	2016-04-30 23:50:00	Generator Bearing	2016-04-30 12:40:00	High temperature in generator
	2016-05-01 00:00:00	2016-08-23 04:00:00	Transformer	2016-08-23 02:21:00	High temperature transformer

Table 3.2: Dataset split of 2016 T07

The training set of each dataset is assumed to contain only normal data. This data was used to train a Variational Auto-encoder Model (VRAE). Out of the 81 features, 38 were selected as per the literature [18]. Any outlier removal technique was not applied as it may cause the loss of information which might be useful for the anomaly detection. Missing values can be a greater issue than outliers. Fortunately, there were no Not a Number (NaNs) in the 2016 dataset, and only one NaN in the 2017 dataset across the selected 38 features. To address this single instance of missing data, forward fill was used.

Data normalization is an essential step when training neural network models. It involves scaling the data to ensure that all features are on a similar scale, which helps in stabilizing and accelerating the training process [40]. MinMaxScaler from sklearn library was used to scale the 38 features to the range 0 and 1.

Model Development for SCADA Parameters

As mentioned in the literature review, anomaly detection falls within the domain of unsupervised machine learning. This approach does not rely on labeled data but instead

seeks to find patterns or irregularities within the dataset. An AE model was used for this purpose because it can handle multidimensional data, making it suitable for complex datasets with many features. AEs are designed to reconstruct their input after encoding it into a compressed form, known as the latent space. If an anomaly occurs in any of the features, the AE may struggle to reconstruct the data accurately, resulting in a higher reconstruction loss. This discrepancy allows the model to identify where the anomaly might be. Due to this reason reconstruction error based anomaly detection is implemented compared to forecasting based, discussed in the literature review.

The training set, which contains only normal condition data, was used to train the model illustrated in figure 3.15. During this process, the described VRAE model was trained to learn a compact representation of the input data. Once the model was trained, the maximum training reconstruction loss was calculated and set as the threshold for identifying anomalies. The model was trained for 150 epochs with early stopping strategy to prevent overfitting. Adaptive moment estimation method (Adam) was used as the preferred optimizer for training. The latent space dimension, learning rate was tuned to 20 and 0.0005.

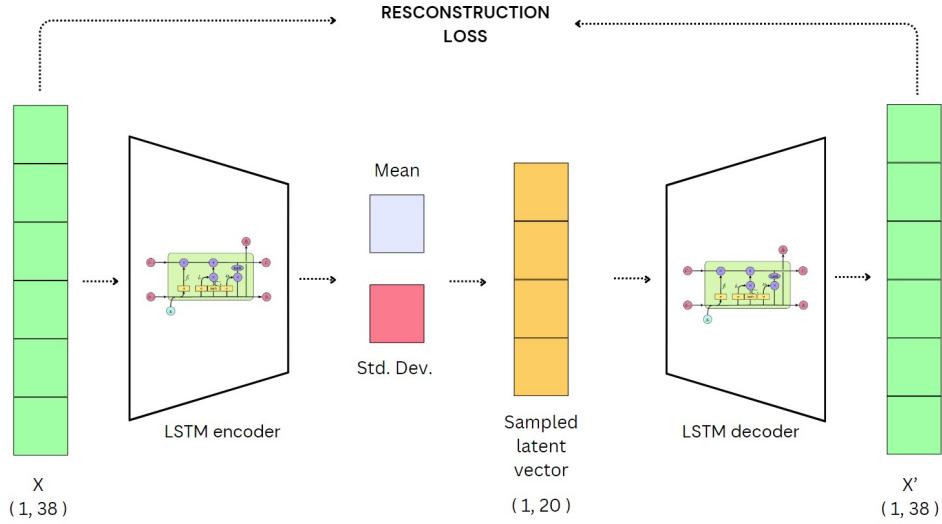


Figure 3.15: VRAE model used

3.2.3.2 Fault Classification

An anomaly detection model helps identify when a fault is occurring, but it doesn't provide specific information about the type of fault or the component affected. To gain more insight, the indexes of the datapoints identified as anomalies by the anomaly detection model were extracted, creating a new dataset in which each anomaly has a unique label.

To visualize potential clusters within the anomalies, the latent space embeddings were retrieved at these anomaly indexes and plotted using t-distributed Stochastic Neighbor Embedding (t-SNE), a technique for visualizing high-dimensional data. The t-SNE plot provides a visual representation of the data structure, which helps to identify clusters or patterns among the anomalies.

Following this visualization, a K Nearest Neighbour (k-NN) model was trained on the latent space embeddings. k-NN is a clustering method that identifies groups of similar datapoints based on their proximity to one another in the feature space. By training a k-NN model, we can cluster the anomalies into distinct groups, providing more detailed insights into the underlying patterns of the identified faults. This approach enables further analysis to classify the type of fault and determine which components are most likely to experience issues.

The integration of anomaly detection with clustering methods like k-NN facilitates a more targeted approach to PdM, allowing a single model to identify anomalies and determine where in the system the issue is happening. Final implementation of the anomaly detection and fault classification model is depicted in figure 3.16. The results of this method will be discussed in the Results section.

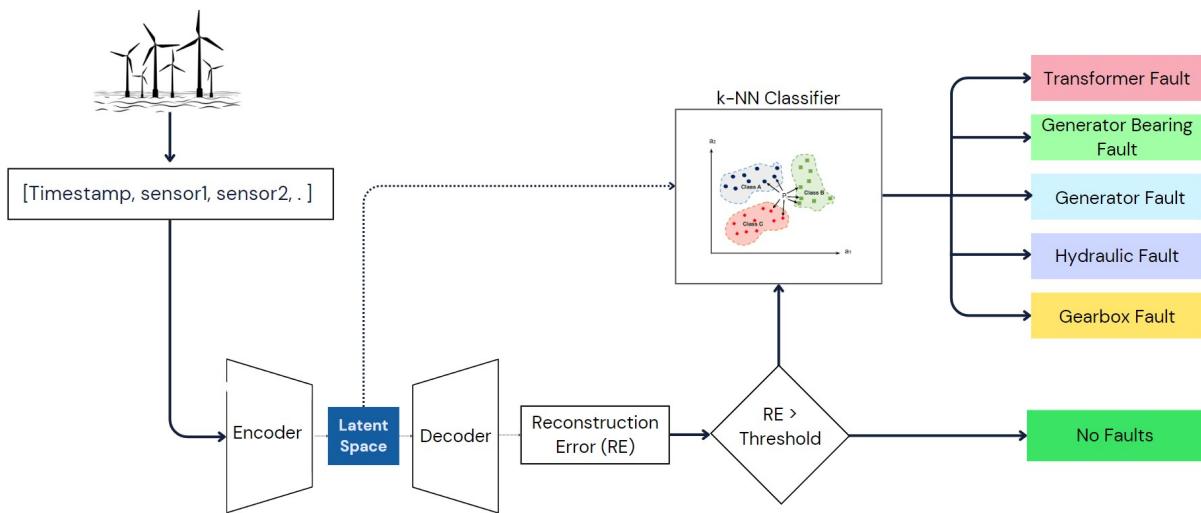


Figure 3.16: Model implementation for online monitoring

3.2.3.3 Remaining Useful Life Prediction

Although our anomaly detection method provides fairly good results in early fault identification (Table 4.1), it generally detects faults about 2 days before they occur. This lead time may not be sufficient for wind farm operators to effectively plan and schedule maintenance tasks, as it can take more time to coordinate resources, arrange for spare parts, and ensure equipment availability. Predicting wind turbine failure as early as possible is crucial to minimize downtime and reduce the risk of costly repairs or equipment replacement. To address this, a method that can identify faults over a longer time interval is needed. This extended lead time would enable wind farm operators to schedule maintenance tasks well in advance. In this section, we present a method designed to predict whether any component of the wind turbine, is likely to fail within the next 60 days.

Fault Data Analysis

In this section, we use the same EDP wind dataset for our analysis [39]. The current version of this dataset contains data from four wind turbines. This dataset has two full years of operational data (2016 and 2017) from a wind farm. The dataset includes critical information on four major components: the gearbox, generator, generator bearing, transformer, and hydraulic group. Additionally, it provides breakage reports from these components, detailing instances of failure and other significant issues. The dataset focuses on four turbines, identified as T01, T06, T07, T11. The number of faults occurred per each turbine is shown in figure 3.17.

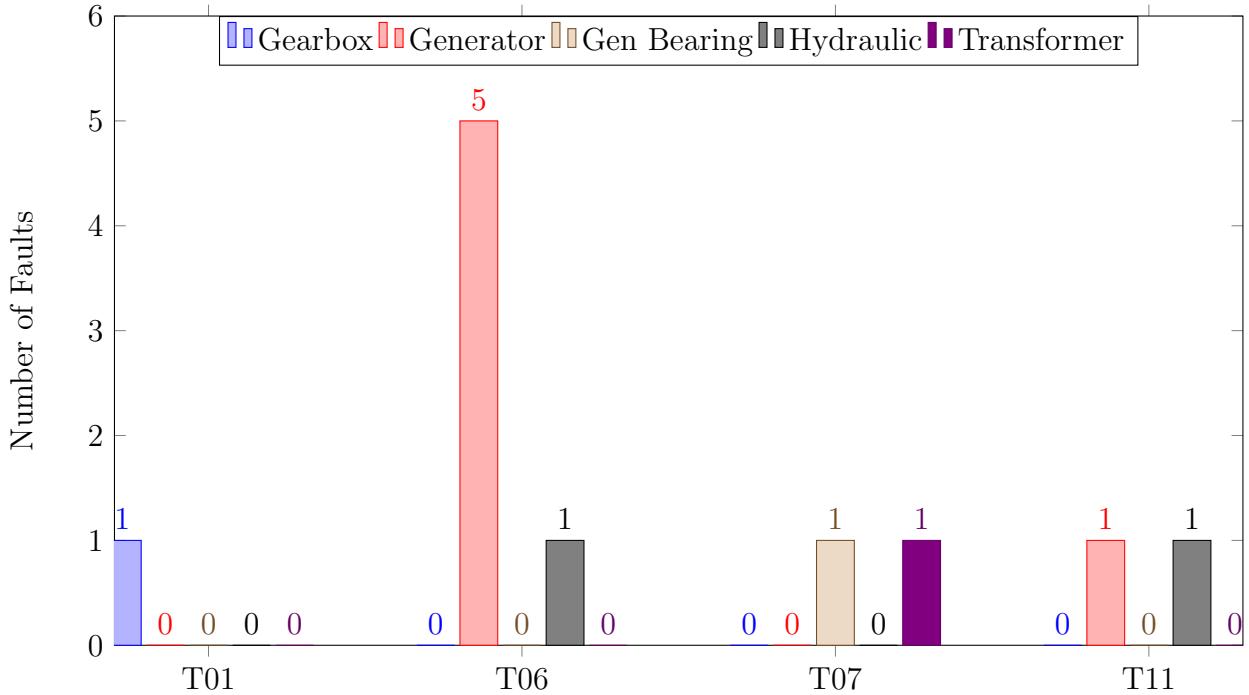


Figure 3.17: Faults occurred in 2016

It can be seen that some faults occurred in some wind turbines. As the dataset contains two years of data, 2016 data is used as training data, 2017 data is used as testing data. If we look at the fault distribution among the training and testing set for each wind turbine, there are faults that did not occur for the wind turbine in training set, which have occurred in the testing set and vice versa (marked in yellow in figure 3.18). To assess whether a predictive model can generalize, it must successfully predict these unseen faults in the testing set. Since these faults are unique to the testing data, they serve as a benchmark for evaluating the model's adaptability and robustness.

	T01		T06		T07		T09		T11	
Component	Tr	Ts								
Gearbox	1	0	0	1	0	0	1	1	0	0
Gen	0	0	5	0	0	1	0	0	1	0
Gen_Bear	0	0	0	0	1	1	3	1	0	0
Trans	0	1	0	0	2	0	0	0	0	0
Hydr_Grp	0	0	1	1	0	2	0	1	1	2

Figure 3.18: Fault distribution among training and testing data [3]

Data Preprocessing

The 2016 dataset did not contain any NaN values, while any NaNs found in the 2017 dataset were handled using forward filling. The dataset underwent a feature selection process to remove features with low variance. A variance threshold of 0.1 was used to identify features that had minimal variability, indicating that they contributed little information. Hence features ‘*Prod_LatestAvg_ActPwrGen2*’, ‘*Prod_LatestAvgReactPwrGen2*’, ‘*Grd_Prod_CosPhi_Avg*’, ‘*Grd_Prod_Freq_Avg*’ were removed. Outliers were not removed for the same reason explained in section 3.2.3.1. Finally dataset is scaled using MinMaxScaler from sklearn library.

Feature Engineering and Model Development

A label should be added to the dataset in consideration of the model to be built. To properly train a classifier that predicts whether a component will break within the next 60 days, a label indicating the “*remaining_useful_lifetime*” was added to the dataset. This additional column provides a way to quantify the time left before a component fails, allowing for supervised learning. For a given fault timestamp provided in the dataset, we calculated the time difference between that fault timestamp and timestamps before it. Then column “*label_60_days*” was added such that if the “*remaining_useful_lifetime*” is less than 60 days a unique integer label was assigned to each fault. Using one-hot encoding and this column we would be able to handle multiple failures.

For the model building we follow the same approach used by TIDRIRI et. al. [3]. We use the entire preprocessed and feature-engineered 2016 dataset for model training. The 2016 data from all turbines is combined to create a general training set, and a separate model is built for each component in the wind turbine system. To train each model, a dataset is built for each one. Data specific to the relevant component’s faults is extracted from the general training set, while data involving other faults and normal operations is randomly sampled. The goal is to ensure that one-fourth of the training data for each model consists of faults related to the specific component. The training process is shown in figure 3.19.

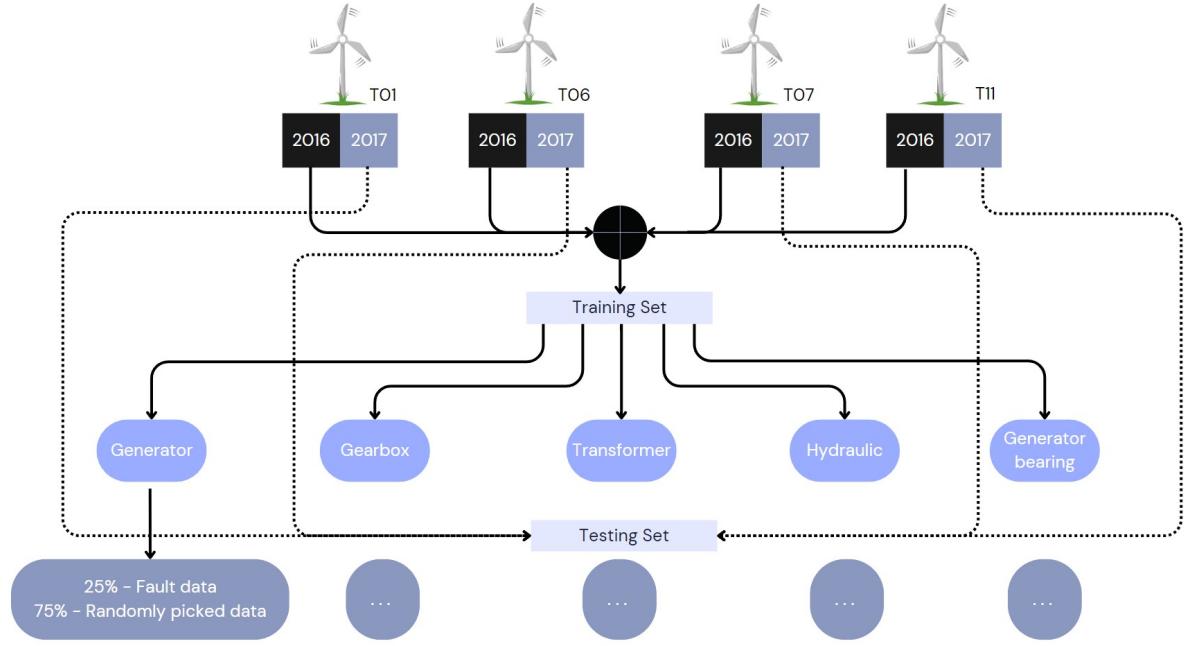


Figure 3.19: RUL model for each component

Various deep learning classification models were tested, including a Deep Neural Network (DNN), a LSTM classifier, and a one-dimensional Convolutional Neural Network (1D-CNN) classifier. After evaluating these models using validation accuracy, the 1D-CNN model was chosen as the best-performing model for the classification task. The model is shown in figure 3.20. Given that the dataset for each model is unbalanced, a stratified cross-validation approach (with $k = 10$) was used to assess validation accuracy and detect potential overfitting to the training data. Stratified cross-validation ensures that each fold has a proportional representation of each class, addressing the issue of imbalance by preserving the distribution of classes across training and validation sets.

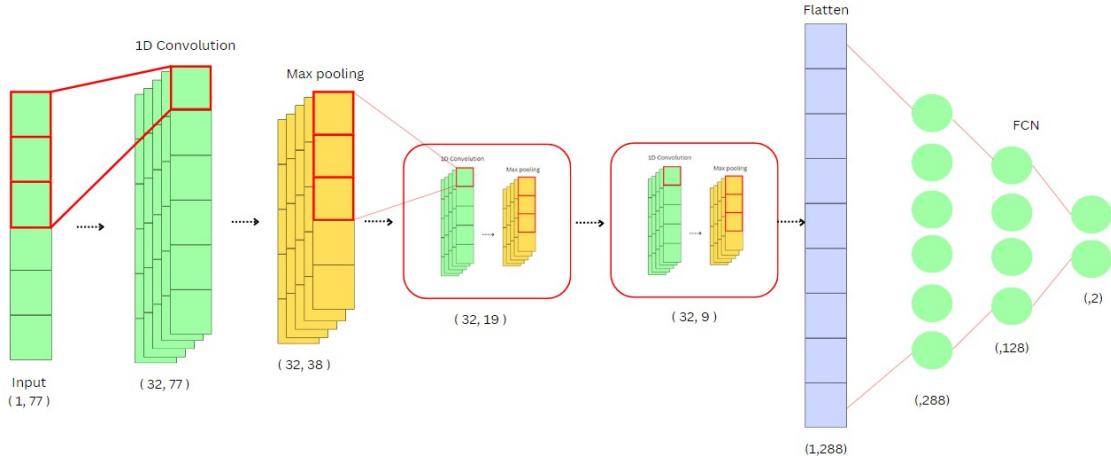


Figure 3.20: 1D-CNN model

Maintenance strategy

In the testing set, each model generates a label for each 10-minute timestamp, indicating whether a fault is likely. Given the high frequency of these predictions, the number of warnings could be substantial, raising concerns about the practicality of sending a technician to inspect a component every time the model signals a potential issue. In real-life scenarios, sending a technician for each warning is impractical. Instead, it's more likely that if a component has been examined and determined to be functioning properly, subsequent warnings might be disregarded for a certain period before another inspection is considered. Hence in this system, once the model issues a warning, a technician is supposed to be sent for inspection. If the technician could find any fault, the output from the model will be taken as a true positive (TP). These true positives will incur repair costs. If there is no fault a technician could find, the output of the model will be taken as a false positive (FP). In such a case it will incur inspection costs. Subsequently, all the warnings from the model will be disregarded for 30 days. This 30 day choice is made with the understanding that even if a false positive occurs, more than half of the 60-day interval (which is the timeframe for predicting a fault) remains for the model to act on. Thus, this 30-day buffer allows for a reasonable margin before deciding to re-inspect or take further action. Finally, a false negative (FN) is where no warnings were sent at all during the 60 days leading up to a failure. False negatives will incur replacement costs.

$$TP_{Savings} = \sum_{i=\#TP} (Replacement - (Repair + (Replacement - Repair)(1 - \frac{t_i}{60})))$$

$$FP_{Cost} = \sum_{i=\#FP} (Inspection)$$

$$FN_{Cost} = \sum_{i=\#FN} (Replacement)$$

$$Total_{Savings} = \sum_{Turbine} \sum_{Component} (TP_{Savings} - FP_{Cost} - FN_{Cost})$$

Figure 3.21: Equations to calculate costs

The costs for each component is given as follows,

Component	Replacement Cost (RP)	Repair Cost (RC)	Inspection Cost (IC)
Gearbox	100 000 €	20 000 €	5 000 €
Generator	60 000 €	15 000 €	5 000 €
Generator Bearings	30 000 €	12 500 €	4 500 €
Transformer	50 000 €	3 500 €	1500 €
Hydraulic Group	20 000 €	3 000 €	2 000 €

Figure 3.22: Costs of the components

The results of the model will be discussed in the results section.

3.3 Energy Storage System

The battery system in the micro-grid is comprised of ten “ROSEN” battery packs which produce 4.8 kWh. One of these packs is designated as the master battery and connects directly to the inverter. The remaining nine batteries function as slaves, each connected to the master battery. All ten batteries are 48V, 100 Ah Lithium Iron Phosphate batteries and incorporate individual “PACE” battery management systems. Each battery consists of 15 cells connected in 15S2P composition.



Figure 3.23: Battery System connected to inverter

3.3.1 Hardware Design

3.3.1.1 Overview

One of the main challenges in the battery system was the inability to acquire individual data from each battery. As nine batteries were connected to the master battery and only the master battery was connected to the inverter, only overall data related to the battery system could be obtained from the inverter. As a solution, our custom hardware was designed to acquire data from each battery with data related to each cell in real time.

Hardware design for the Energy Storage System will henceforth be referred as the “Battery PCB”. This PCB was designed according to the dimensions of the purchased enclosure explained later.



Figure 3.24: Final implementation on the battery system

3.3.1.2 Schematic Design

The schematic design for the Battery PCB done using Altium Designer is given below.

1. ESP32

- The micro-controller used in the PCB is an ESP32-WROOM-32E. Boot button is used to enter the ESP32 into flashing mode. Reset button is used to restart the ESP32 microcontroller.

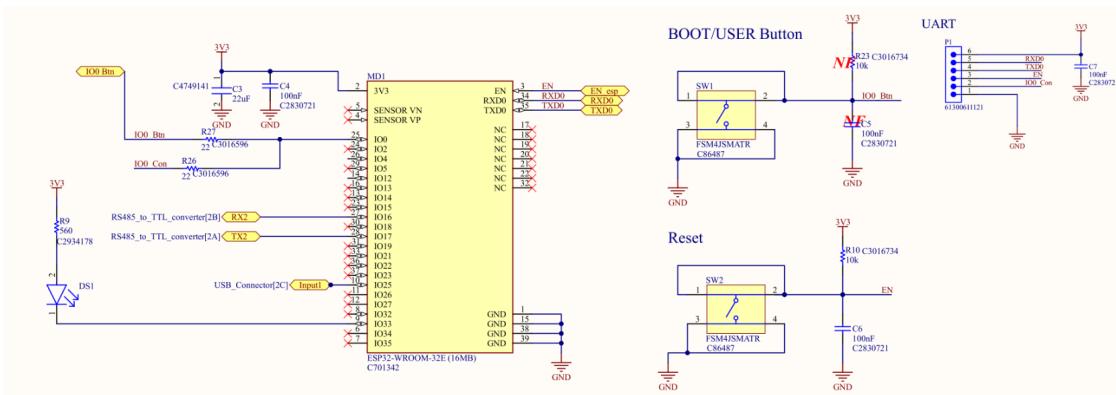


Figure 3.25: ESP32 micro-controller - Battery System

2. USB Connector

- Differential pair routing was adhered when drawing traces for USB communication. Impedance profile for USB communication is 90Ω .

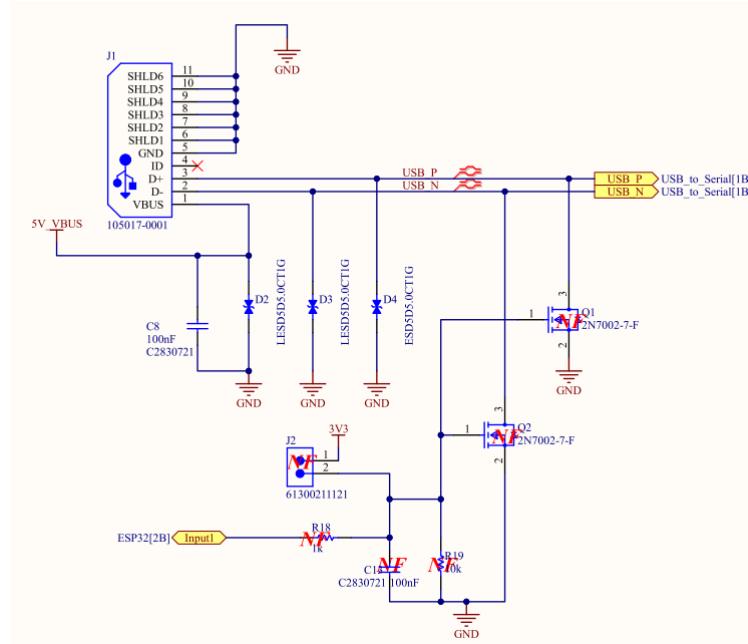


Figure 3.26: USB Connector - Battery System

3. USB to Serial Converter

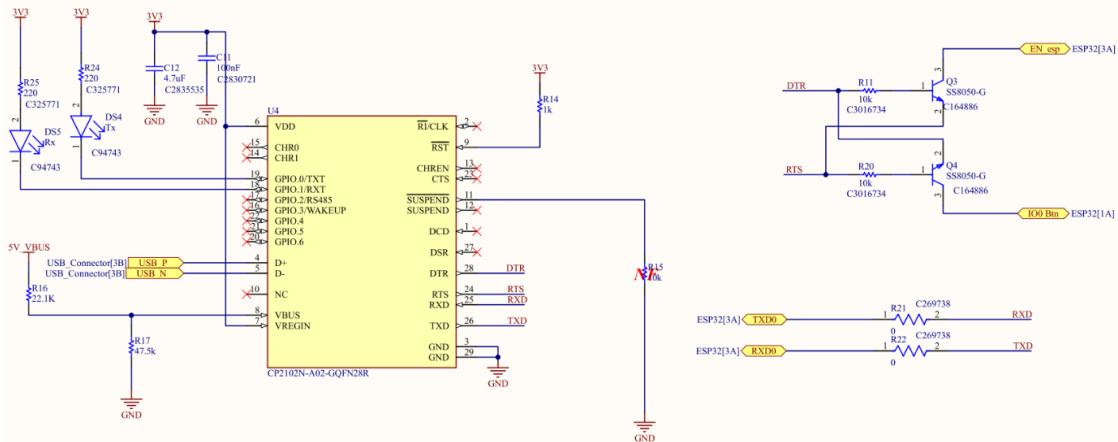


Figure 3.27: USB to Serial converter - Battery System

4. Hex inverter

- Contains two LEDs to indicate data communication both ways. RX, TX pins are connected to RX, TX pins of the ESP32 micro-controller respectively.

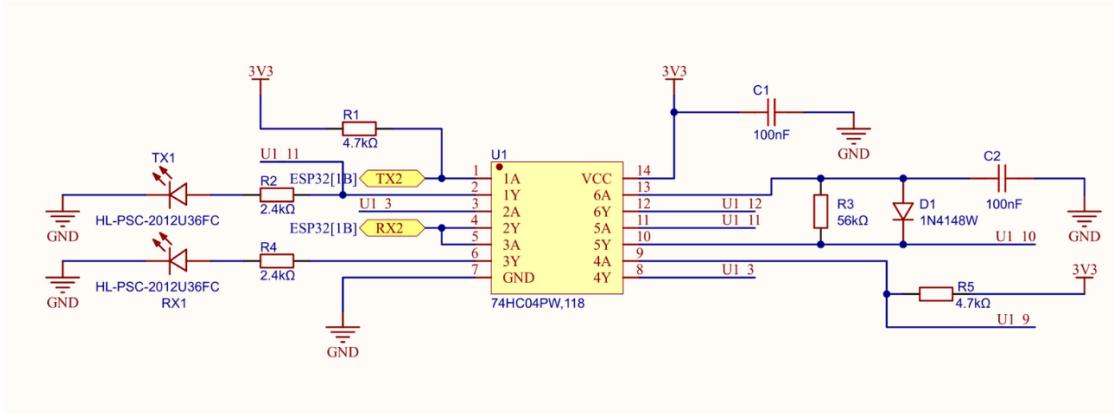


Figure 3.28: Hex inverter - Battery System

5. RS485 transceiver

- Half-duplex transceiver that meets the specifications of RS-485 serial protocols with enhanced Electrostatic Discharge (ESD) performance. It meets the requirements of the RS-485 protocols upto 10Mbps under load. Includes a $120\ \Omega$ termination resistor to match the characteristic impedance of the transmission line, reduce signal reflections, and improve signal integrity. Differential pair routing have been followed when drawing data communication traces from the RJ45. Impedance profile for RS485 communication is $100\ \Omega$.

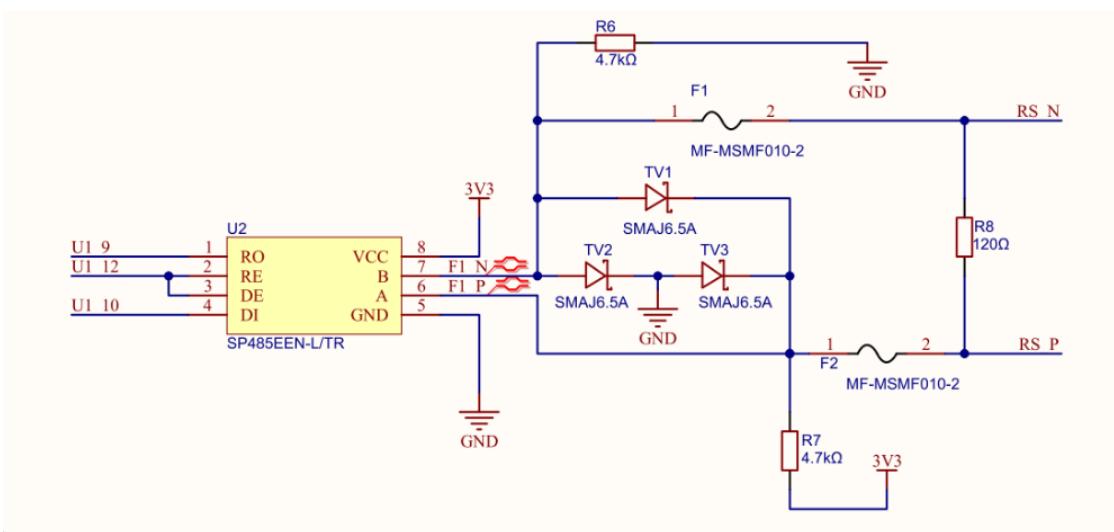


Figure 3.29: RS485 transceiver - Battery System

6. RJ45

- Output from the battery system is connected to this RJ45 socket using an Ethernet cable. Data communication lines are drawn adhering to differential pair routing.

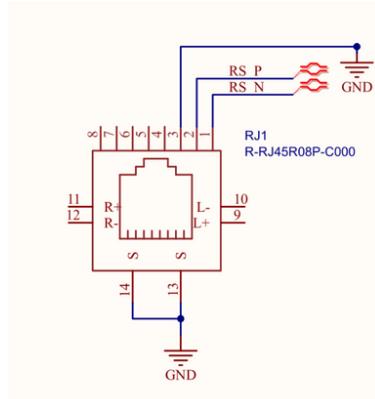


Figure 3.30: RJ45 socket - Battery System

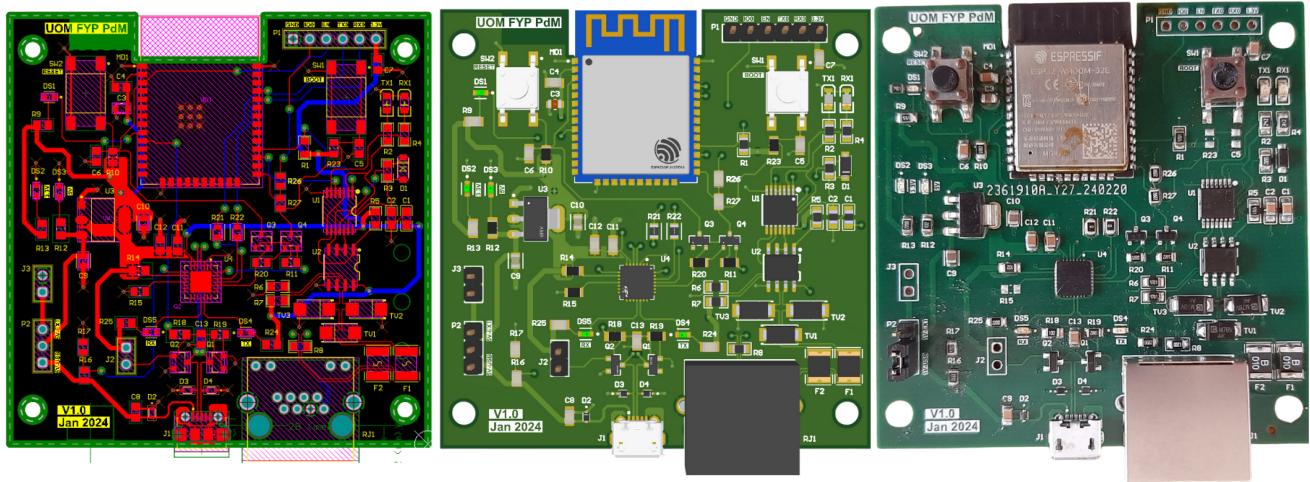
3.3.1.3 PCB Design

1. Layer stackup - 4 layers

- Top Layer (Signal)
- Inner Layer 1 (Ground)
- Inner Layer 2 (Ground)
- Bottom Layer (Signal)

2. Main features

- Differential pair routing for RS485 protocol
- Differential pair routing for USB communication
- Controlled impedance traces and length matching for USB and RS485 signals
- 7628 Layer stackup according to JLCPCB guidelines



(a) Layout

(b) 3D view

(c) Soldered PCB

Figure 3.31: Battery PCB

3.3.1.4 Enclosure Design

The PCB was designed to fit into a commercially purchased enclosure, which has IP54 protection and is made from durable ABS plastic. These enclosures along with the PCB were fitted close to the battery in the energy storage system.



Figure 3.32: Enclosure design for battery PCB

3.3.1.5 Implementation

Following the design phase, the PCB underwent a series of tests to validate its performance. The testing phase focused on evaluating the PCB's electrical safety, signal integrity, and operational stability, ensuring that the board met all required specifications and industry standards.

Our PCB connects to the battery through the RS485 port of the Battery Management System (BMS) using an RJ45 ethernet cable. Using the firmware in the ESP32 micro-controller and RS485 to TTL conversion, battery data can be acquired from the BMS.

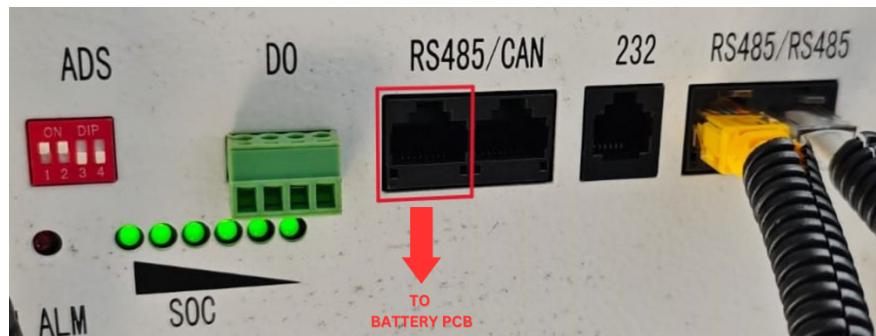


Figure 3.33: Connection to the BMS

3.3.2 Firmware Development

The main firmware component in the ESP32 micro-controller is the esphome-pace-bms. esphome-pace-bms is a project that provides an ESPHome component for monitoring and

controlling a PACE Battery Management System (BMS) using the RS485 Modbus protocol. It is an open-source platform for creating custom firmware for ESP32 micro-controllers.

Our project is specifically implemented for the PACE Battery Management System (BMS). PACE is a manufacturer of Battery Management Systems, which are electronic systems that monitor critical parameters of a battery, such as voltage, current, temperature, and health.

The firmware is specifically configured to measure the following parameters from the BMS;

- Current
- Voltage of pack
- State of Charge (SoC)
- State of Health (SoH)
- Remaining Capacity
- Full Capacity
- Charging cycles count
- Warning flag
- Protection flag
- Status flag
- Cell voltages of cells 1 - 15
- Battery temperature from sensors 1 - 4

The following custom components were built for the esphome-pace-bms;

MQTT Telemetry Message Transmission

The MQTT protocol was used to send telemetry messages from the ESP32 to the cloud. ESP32 communicates directly with Azure IoT Hub service via this MQTT protocol. Here, “Azure SDK for C” Arduino library was used. When the ESP32 restarts, it establishes an IoT Hub client and an MQTT client after connecting to WiFi. If the WiFi connection is lost, this MQTT connection will also be dropped. If the MQTT connection is lost without a WiFi disruption, the ESP32 will attempt to reconnect in 10-second intervals and if this fails for 1 minute, the ESP32 will restart and establish all connections from the beginning. If the WiFi and MQTT connections are successful, the MQTT client sends the created data string to the Azure IoT Hub at 30-second intervals.

Data string buffer

A buffer is used to store data strings in case of WiFi or MQTT connection failure. This buffer can store up to 50 data strings. If more data strings must be stored, old data strings will be discarded to store the new data strings. A data string consists of 30 parameters of the battery status.

3.3.3 Machine Learning Model Development

3.3.3.1 Remaining Useful Life Prediction

Overview

RUL prediction for a battery system is the process of estimating how much longer a battery can be expected to function effectively before needing replacement. RUL prediction involves monitoring battery parameters like voltage, current, temperature, and capacity, and estimate the remaining lifespan using machine learning algorithms. Our model follows a data-driven approach that leverages historical data on battery health and performance.

Dataset

A summary of the dataset from Severson et al. [41] used for the prediction of RUL is as follows. The dataset consists of 124 Lithium Iron Phosphate (LFP) / graphite cells (A123 APR18650M1A) cycled in three “batches”. These cells were split into a training set (41 cells), primary test set (43 cells) and secondary test set (40 cells). Each cell has a nominal capacity of 1.1 Ah and nominal voltage of 3.3V. These cells have been cycled with one of 72 different fast charging protocols, but all cells were identically discharged at 4C (C refers to the nominal rate required to discharge a cell in 1 h). The dataset consists of standard electrochemical data like voltage, capacity, cycle number, internal resistance and surface temperature of each cell measured from a thermocouple mounted on the cells.

Objective

Objective of this approach was to predict the log-transformed cycle life using data from only the first 100 cycles of a cell. Cycle life is defined as the number of cycles before the capacity falls below 80% of the nominal capacity (80% of 1.1 Ah or 0.88Ah). The dataset contains cycle lives in a range from 300 to 2237.

The main predictive feature of the model is derived from the voltage vs discharge capacity curves. The feature summarizes the difference between the voltage vs discharge capacity curves of the 100th and 10th cycles, denoted $\Delta Q_{100-10}(V)$. Our model uses the $\log_{10}(\text{variance}(\Delta Q_{100-10}(V)))$ to predict the RUL of a cell.

Model Implementation

In principle, the relationship between voltage and capacity is continuous. But, in practice this relationship is sampled at a finite rate after data acquisition. The discharge voltage curve is fit with a smoothing spline and then sampled at 1000 evenly spaced points in voltage. The

operating window spans 1.6 V (3.6- 2 V), thus distance between each capacity sample is 1.6 mV. Then we fit the capacity as a function of voltage since voltage window is constant with cycle number and capacity changes with cycling.

In conclusion, each discharge curve will have 1000 capacity points at 1000 pre-defined voltage points. For the first 100 cycles, each cell will have 100,000 voltage-capacity data points.

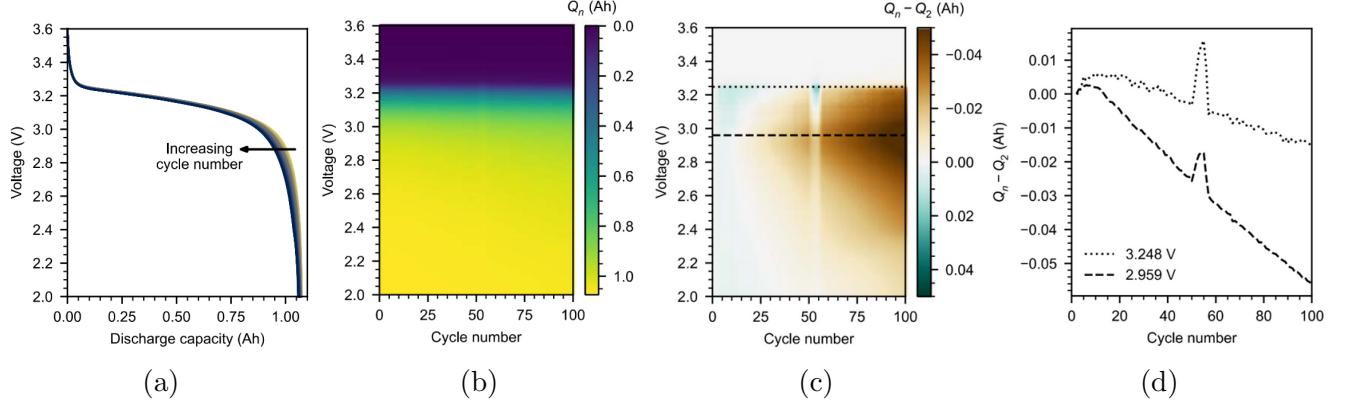


Figure 3.34: RUL prediction model implementation

Figure 3.34a shows the voltage vs discharge capacity curves for the first 100 cycles of a specific battery. Note that curves shift slightly to the left with increasing cycle number.

Figure 3.34b below shows the visualization of 100,000 voltage-capacity data points termed the “capacity matrix”.

Figure 3.34c shows the “baseline-subtracted capacity matrix” denoted $Q_n - Q_2$, which is the matrix in Figure 3.34b subtracted by the cycle 2 discharge capacity.

Figure 3.34d shows $Q_n - Q_2$ at two selected voltages, two rows of the “baseline-subtracted capacity matrix”. We use cycle 10 as a baseline cycle for this model since this cycle number avoids the initial rise in capacity.

It must be noted that the sum of all elements in a column of the baseline-subtracted capacity matrix is proportional to the change in discharge energy between the cycle of interest and the baseline cycle at a given voltage. Different summary statistics and transformations were applied on $\Delta Q_{100-10}(V)$. The highest performing univariate model was found to be with the feature $\log_{10}(\text{variance}(\Delta Q_{100-10}(V)))$.

Model Evaluation

The dataset was evaluated on the following models; Ridge regression, Elastic Net Regression, Principal Component Regression, Partial Least Squares Regression, Random Forests, Support Vector Regression.

Figure 3.35 shows the Root Mean Squared Error (RMSE) in terms of cycles for the training, primary test and secondary test sets.

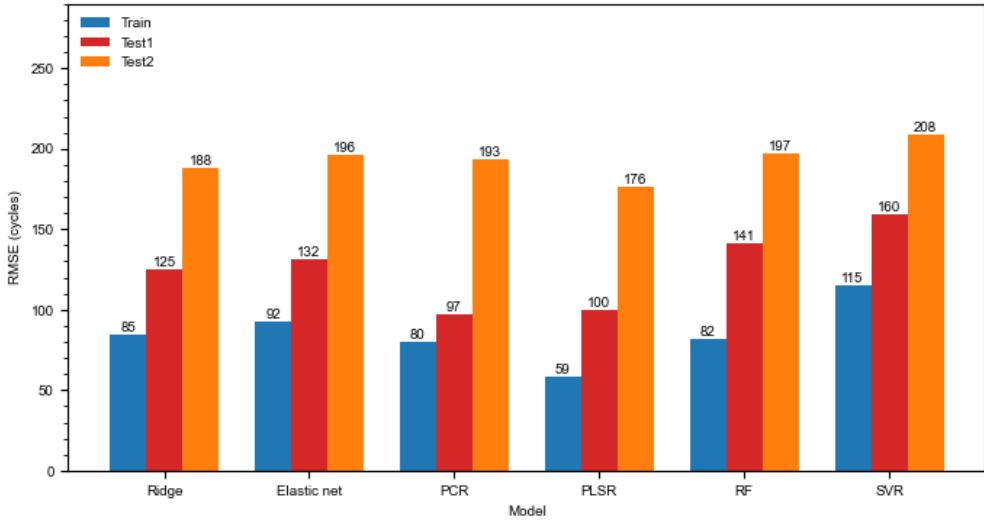


Figure 3.35: RMSE (cycles) for the training, primary and secondary test sets

The best results on both primary test and secondary test sets were obtained from the Partial Least Squares Regression (PLSR) model. The optimal number of PLS components that give the least RMSE is nine. Optimization was done by looping through components one to twenty, performing cross-validation on the training set, and then selecting the component that gives the least RMSE.

3.3.3.2 State of Health Prediction

Overview

SoH is a battery health measurement that is based on the battery's current performance. Generally, SoH is defined as the ratio of the current capacity and the initial capacity of the battery and expressed as a percentage. The initial SoH value would be 100%, while it decrease over time and usage. The degradation pattern will depend on a variety of elements, including driver patterns, climate, thermal dynamics, infrastructure, etc. This model was to predict 30 SoH values of the battery for the next 30 cycles.

Dataset

In this project, CS2 battery dataset from CALCE (The Center for Advanced Life Cycle Engineering) of University of Maryland was chosen to develop this SoH prediction model. This dataset consists of four 1.1 Ah LiCoO₂ batteries labeled CS2-35, CS2-36, CS2-37 and CS2-38. These batteries have been cycled at constant current of 1C (C refers to the nominal rate required to discharge a battery in 1 h). Fig.3.36 shows the degradation curves of the four batteries.

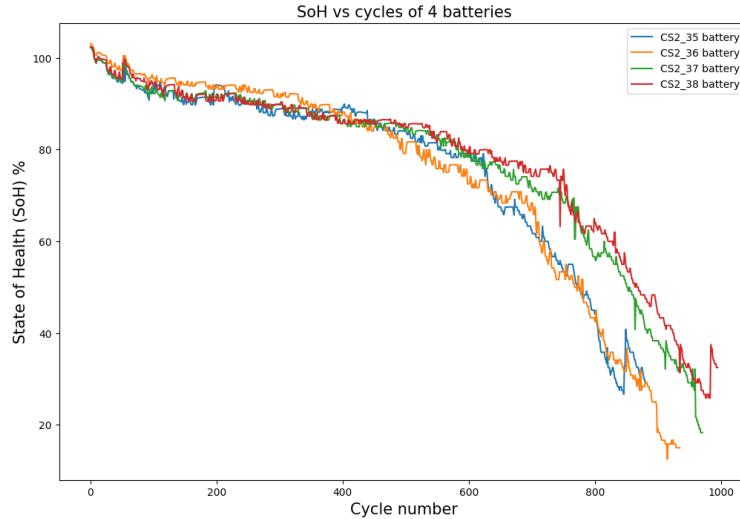


Figure 3.36: SoH degradation of CS2 batteries

Model Implementation

We developed four types of machine learning models: Support Vector Regression (SVR), Random Forest (RF), Multi Layer Perceptron (MLP) and LSTM. These models were trained and tested similarly to figure out the best model.

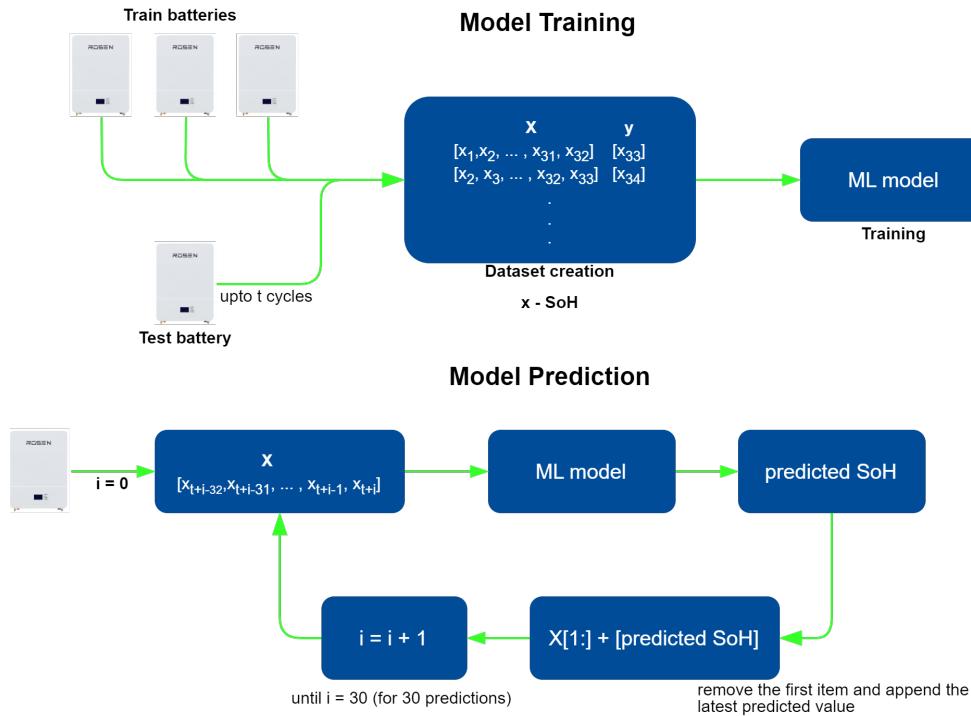


Figure 3.37: SoH prediction model architecture

The model architecture is shown in the Fig.3.37. SoH values were used as model input using a window technique with 32 SoH values per window. Models were trained using data from three batteries at a time, and then tested for the remaining battery. The SoH values for the next 30 cycles of the test battery would be predicted at a certain cycle, and data up to that cycle of the test battery was also used in the training set. This training and testing process was done for all models while the test battery was changed.

3.4 System Integration

3.4.1 Azure Integration

In this PdM framework, Microsoft Azure is used as the cloud platform. It provides a vast array of services, including computing, storage, databases, networking, machine learning, IoT, and more. In this project it also acts as the backend for the digital twin. Fig.3.38 shows the Azure services used in this framework. These services are listed below.

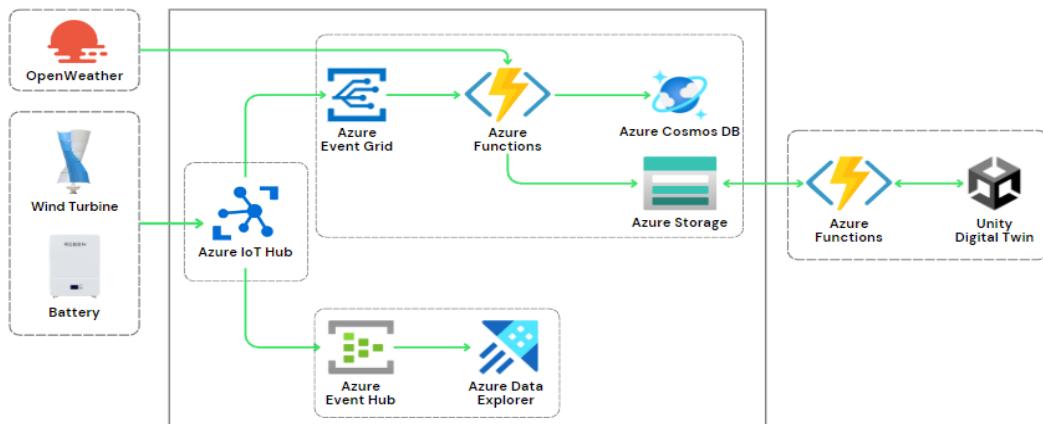


Figure 3.38: Azure services integration

Azure IoT Hub

Azure IoT Hub communicates with ESP32 devices and collects telemetry data. Multiple devices can be created under this service. In this project, three devices were created to collect data from the wind turbine and two batteries. This IoT Hub is then linked to an Azure Event Grid.

Azure Event Grid

This service is used to receive data from the IoT Hub and create multiple events based on the device using filters as new data arrives. Then those events trigger the Azure Functions.

Azure Functions

Azure functions are utilised as endpoints that respond to events. Based on the requirements, different Azure Functions services were utilised in this project, each with its own triggering type.

- Data storing

For this purpose, an Azure Functions service was initialized with an Event Grid trigger to trigger when new data arrives. Multiple functions were configured to receive events (with data) from the wind turbine and batteries separately. After receiving data, it is converted to a JSON object and stored in a Cosmos database. In wind turbine function, weather data are also collected by calling a OpenWeather API.

- Data requesting

The digital twin updates its visualizing data at defined intervals. A separate Azure Functions service with an HTTP trigger is used to get data at these intervals. The digital twin calls this functions via HTTP requests, then these functions get data from Azure Blob Storage and provide it to the digital twin.

Azure Cosmos DB

Azure Cosmos DB service is used to store data of the wind turbine and batteries. This is a NoSQL type database and there are many storage types in Cosmos DB. Since key-value pairs are used in this instance, Cosmos Table Storage is used. Two separate tables are used to store data of wind turbine and batteries.

Azure Storage

Azure Blob Storage is used here in this Azure Storage service. This blob storage stores the last data arrived from the wind turbine and batteries. This data is then provided to the digital twin to visualize by the Azure Function.

Azure Event Hub

Azure Event Hubs is a service that lets you ingest millions of real-time events per second from any source. It acts as a hub for data streaming, enabling further processing and analysis within Azure. Azure Event Hub is connected to Azure IoT Hub. Our architecture consists of several Event Hub instances for the two battery devices and wind turbine device.

Azure Data Explorer

Azure Data Explorer is a cloud-based service from Microsoft designed for fast and scalable analysis of large volumes of data streams. It excels at ingesting structured, semi-structured, and unstructured data, making it ideal for log analytics, time series analysis, and IoT data exploration. Azure Data Explorer leverages a proprietary query language, named Kusto Query Language (KQL) for efficient data retrieval and analysis, allowing users to gain insights

from their data in near real-time. Figure 3.39 shows a near real-time dashboard of data from both the batteries in the battery system.

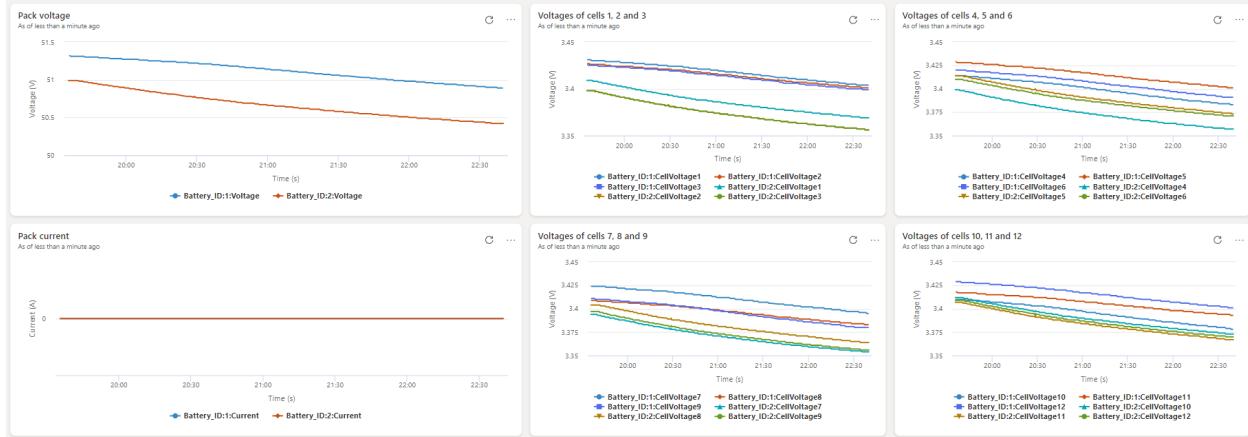


Figure 3.39: Battery dashboard in Azure Data Explorer

Digital Twin Integration

The DT platform is entirely designed with the Unity physics engine. This visualizes all of the data and results produced by the wind turbine, batteries, and Azure cloud. C# scripts are developed in Unity Editor to trigger Azure functions to acquire data from Azure blob storage periodically.

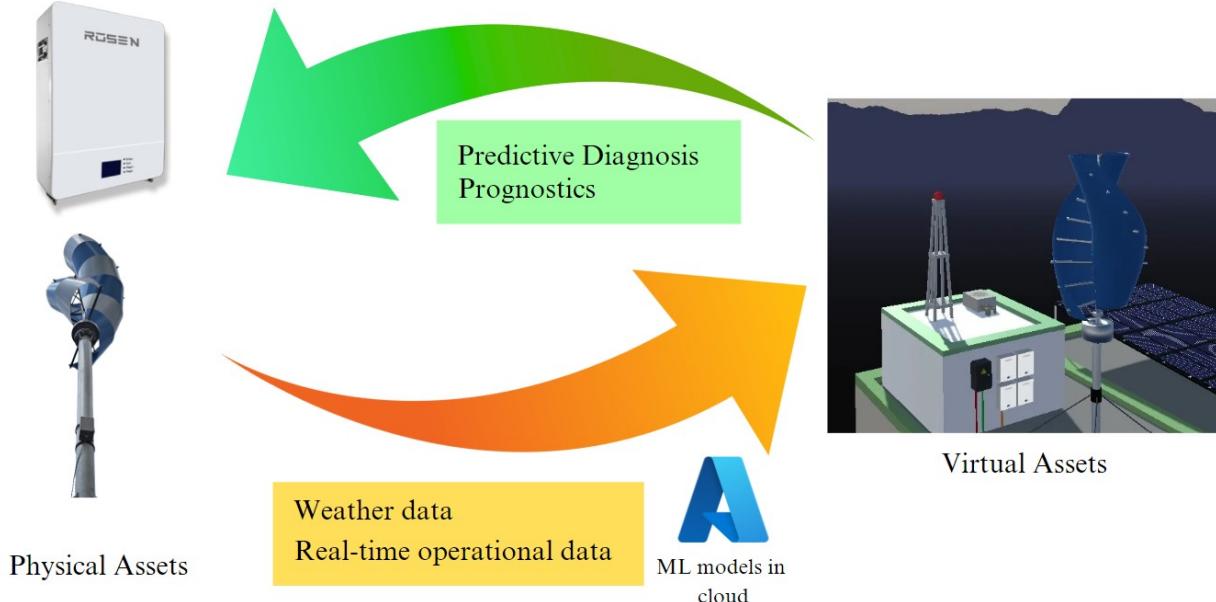


Figure 3.40: Digital twin for the VAWT and Battery

Chapter 4

RESULTS

4.1 Wind Energy System

4.1.1 Anomaly Detection

4.1.1.1 Anomaly Detection using Vibration Sensor Data

Our study has successfully demonstrated that using an LSTM autoencoder model can detect anomalies in three different datasets 2-3 days in advance. This predictive capability is crucial for implementing effective maintenance strategies.

Key Findings:

- Anomaly Detection: The LSTM autoencoder excels at processing time-series data, crucial for identifying early signs of system failures and uses only normal data for model training.
- Threshold Setting: For each dataset, a unique anomaly detection threshold was established based on the maximum reconstruction loss observed during the training on normal data. This approach ensures precise and tailored anomaly detection for each specific dataset.

Set No. 1:

- Recording Duration: October 22, 2003 12:06:24 to November 25, 2003 23:39:56
- Description: Inner race defect occurred in bearing 3 and roller element defect in bearing 4.

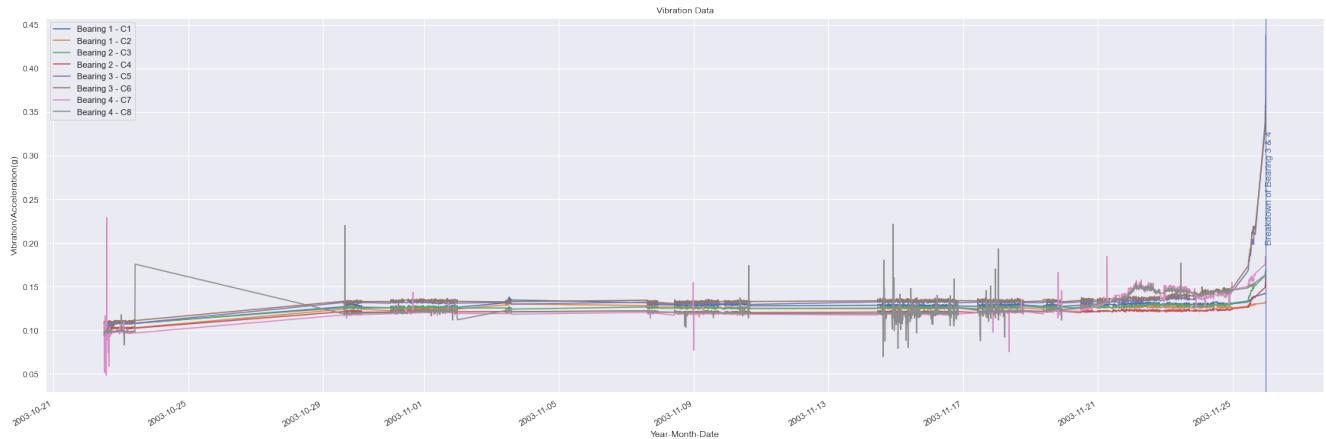


Figure 4.1: Vibration signals of set 01

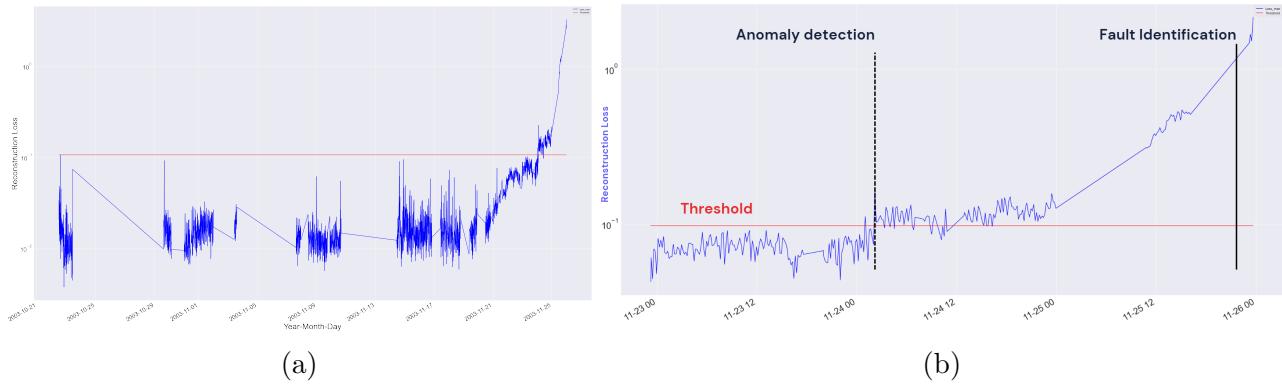


Figure 4.2: Reconstruction error vs Time

Result - Anomaly detection and fault prediction around 2 days in advance to the actual failure occurrence.

Set No. 2:

- Recording Duration: February 12, 2004 10:32:39 to February 19, 2004 06:22:39
- Description: Outer race failure occurred in bearing 1.

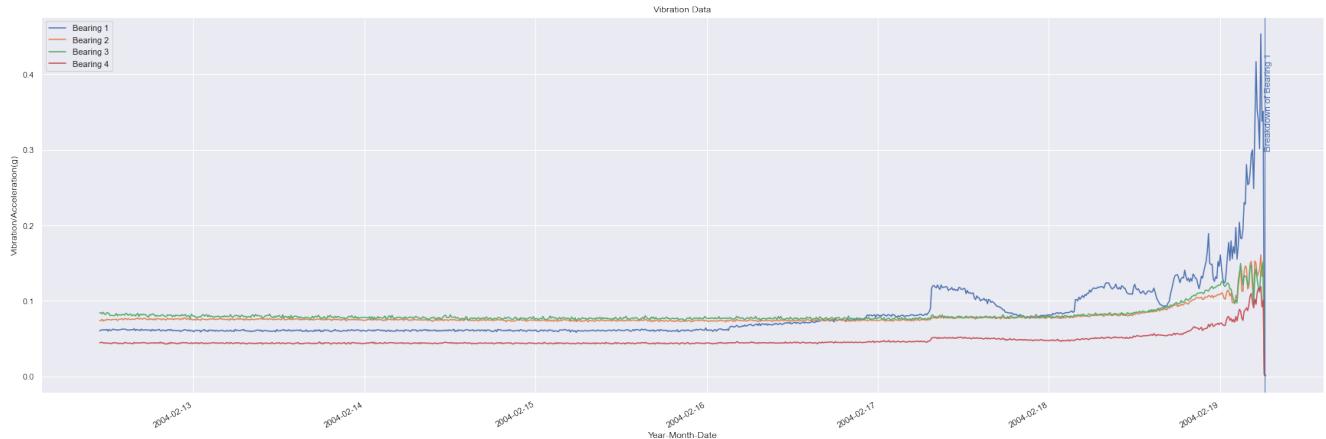


Figure 4.3: Vibration signals of set 02

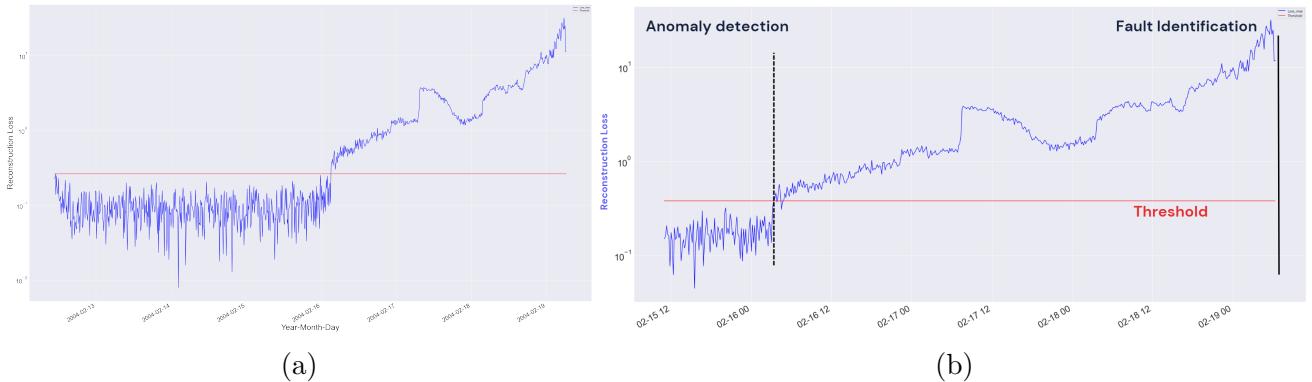


Figure 4.4: Reconstruction error vs Time

Result - Anomaly detection and fault prediction around 3 days in advance to the actual failure occurrence.

Set No. 3:

- Recording Duration: March 4, 2004 09:27:46 to April 4, 2004 19:01:57
- Description: Outer race failure occurred in bearing 3.

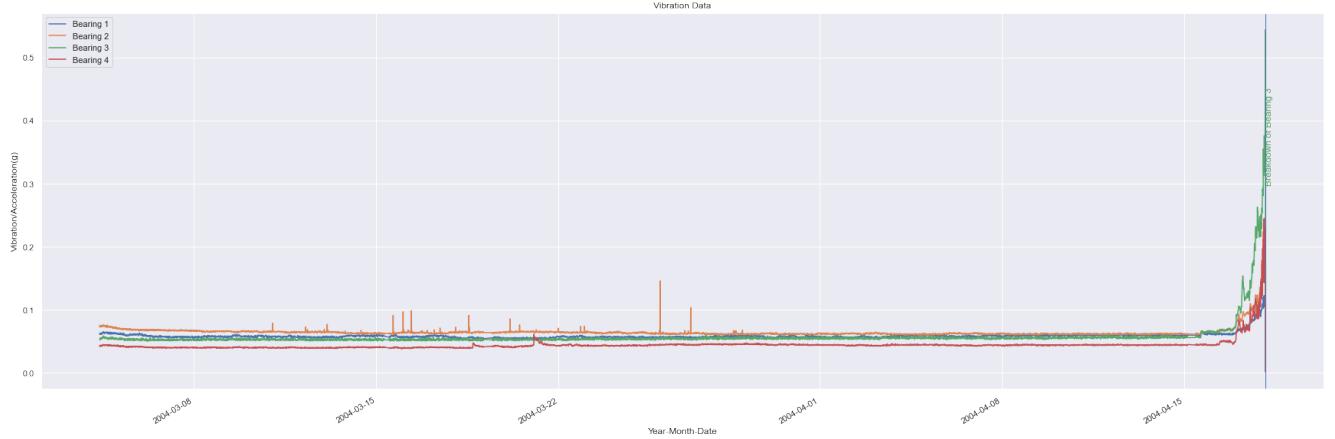


Figure 4.5: Vibration signals of set 03

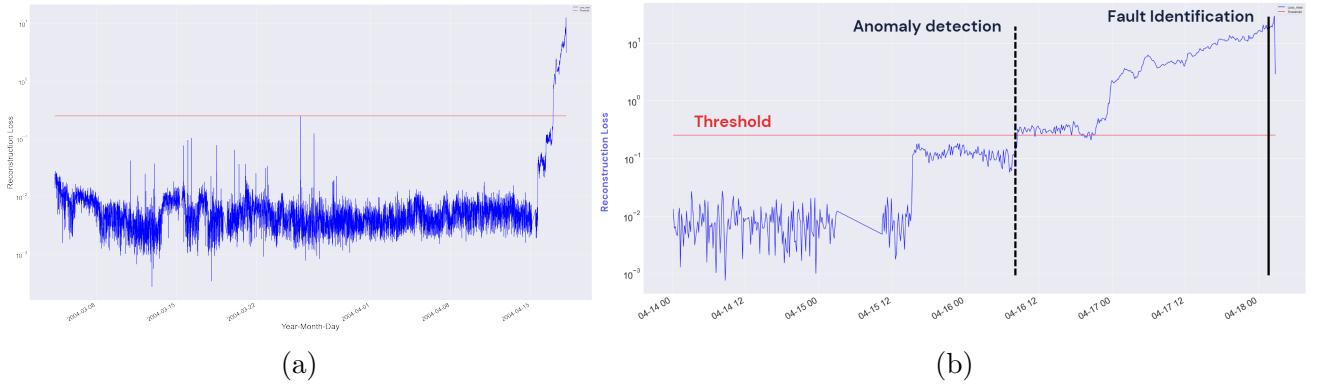


Figure 4.6: Reconstruction error vs Time

Result - Anomaly detection and fault prediction around 2 days in advance to the actual failure occurrence.

4.1.1.2 Anomaly Detection using SCADA parameters

As described in section 3.2.3.1 our anomaly detection model is able to detect failures on the EDP dataset prior to the fault timestamp given in the fault dataset. Following figure shows the training set for the anomaly detection model. Threshold is taken as the maximum training reconstruction error.

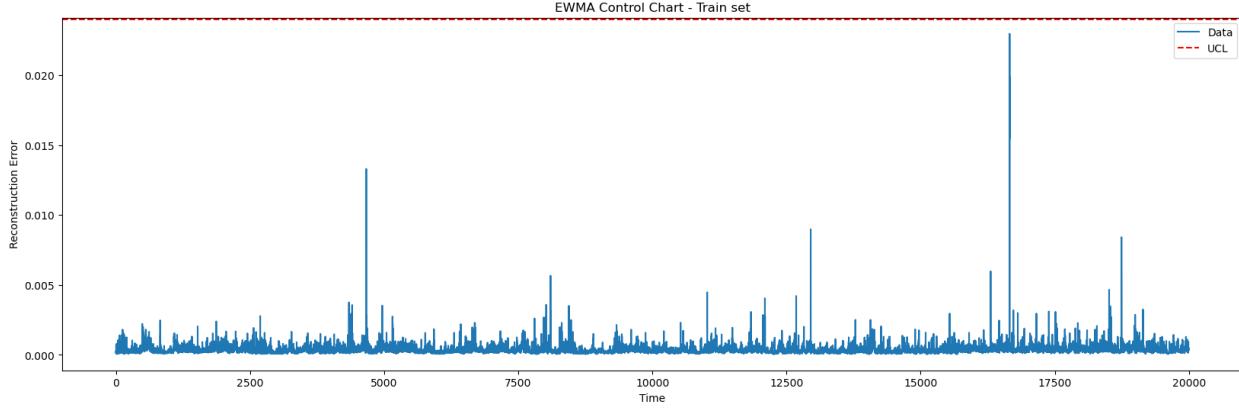


Figure 4.7: Training reconstruction loss

Figure 4.8 shows the test set for the generator bearing dataset. We can clearly see early warnings of anomaly is detected around 117 timestamps earlier which corresponds to around 19.5 hours early detection.

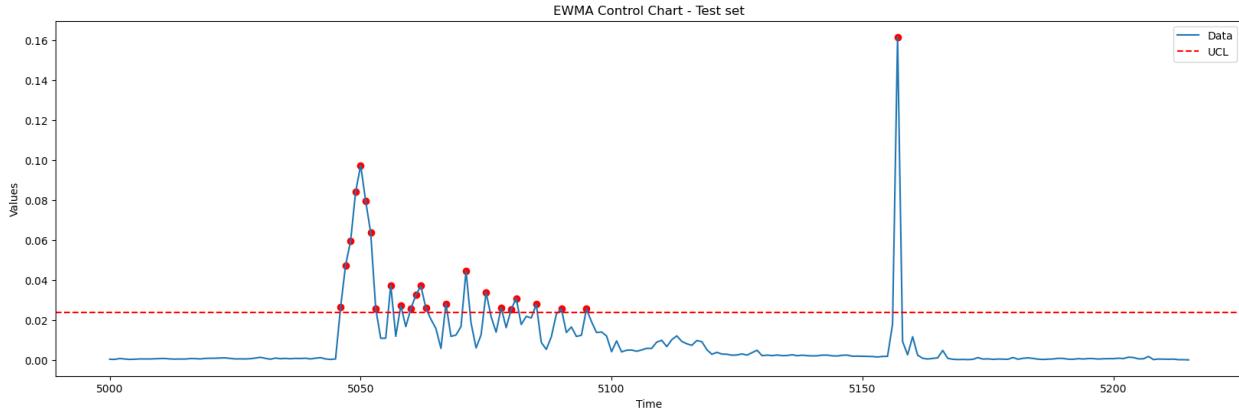


Figure 4.8: Bearing fault reconstruction loss

For the additional faults listed in the 2016 dataset, our model was able to detect early signs of anomalies earlier than the fault timestamp provided in the dataset except for the Transformer fault. Table 4.1 presents the results of our early detection analysis.

Component	Early detection ours (hrs)	Early detection [3](hrs)
Generator Bearing	19.5	11
Transformer	-	-
Generator	23	-
Hydraulic System	3.3	-
Gearbox	67	-

Table 4.1: Results of anomaly detection

4.1.2 Fault Classification

Below is the t-SNE plot for the embedding from the encoder of the Anomaly detection model in 3.2.3.1.

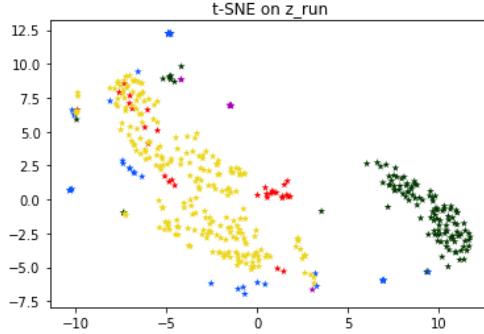


Figure 4.9: t-SNE plot

4.1.3 Remaining Useful Life Prediction

The following table shows the amount of TP, FN and FP as described in the section

Component	TP	FP	FN
Generator Bearing	2	24	0
Transformer	1.	32	0
Generator	1	29	0
Hydraulic System	4	29	1
Gearbox	1.	32	0

Table 4.2: Results of RUL prediction

Together with the results from above table and the equations in Figure 3.21, we were able to calculate the cost of maintenance for the wind turbine system. Using our model, the cost incurred will be 302 424 € which is 44% lesser than reactive maintenance cost of 540 000 € and 29% lesser than scheduled maintenance cost of 427 500 €. Hence we show that using a PdM model we can reduce the maintenance cost compared to traditional maintenance methods.

Component	Savings ours (€)	Savings [4] (€)
Generator Bearing	-90791	-10516
Transformer	-1500	-103000
Generator	-100000	-85000
Hydraulic System	-16800	-85386
Gearbox	-93333	-64712
Sum	-302 424	-348614

Table 4.3: Cost savings

4.2 Energy Storage System

4.2.1 Remaining Useful Life Prediction

Figure 4.10 shows the results of RUL prediction on the training, primary test and secondary test sets of the battery. It can be stated that predictions for batteries with high observed cycle life have higher deviations. This is mostly because the training dataset contains a low number of batteries with higher observed cycle life.

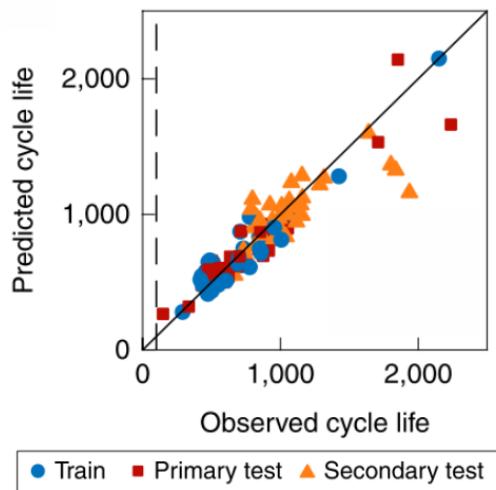


Figure 4.10: Observed cycle life vs Predicted cycle life

4.2.2 State of Health Prediction

The Fig.4.11 shows intermediate results of the models at 350^{th} , 425^{th} , 475^{th} and 550^{th} cycles when CS2-37 battery was tested and others were trained. The blue line represents the actual SoH degradation of the CS2-37 battery and the orange line represents the predictions.

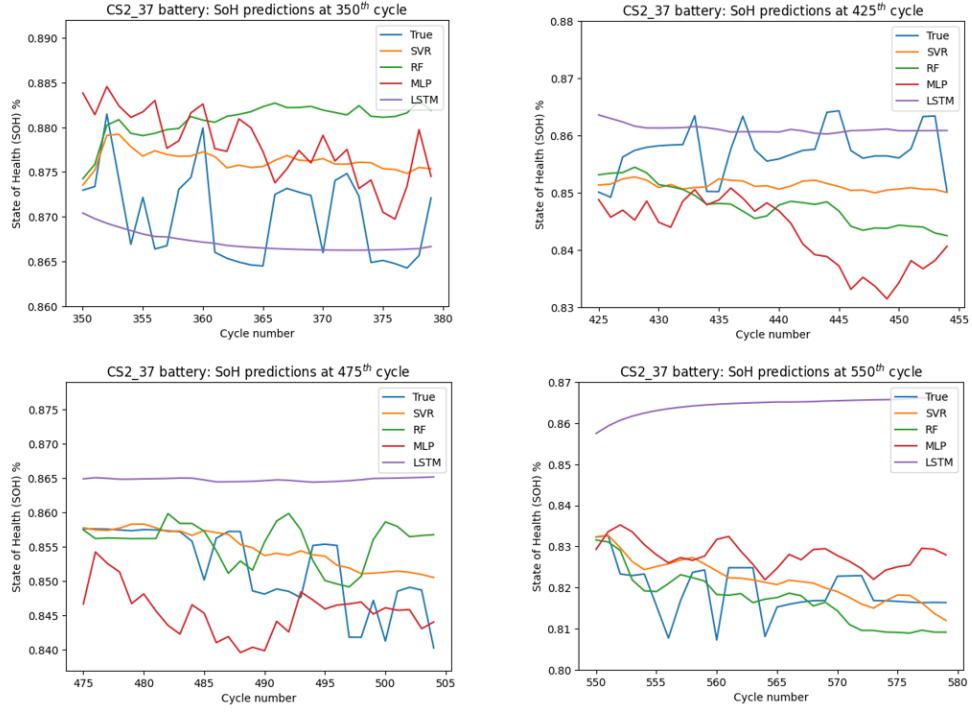


Figure 4.11: SoH predictions of models

		Average MSE Loss			
		SVR	RF	MLP	LSTM
Testing Battery	CS2_35	0.007836	0.00802	0.033613	0.014258
	CS2_36	0.010488	0.009332	0.014597	0.013845
	CS2_37	0.006543	0.006518	0.012048	0.013079
	CS2_38	0.00732	0.007428	0.012636	0.011366
Standard Deviation					
Testing Battery	CS2_35	0.004664	0.005672	0.110218	0.005589
	CS2_36	0.004116	0.00372	0.010942	0.008913
	CS2_37	0.0032	0.002122	0.013563	0.011442
	CS2_38	0.003501	0.003482	0.00888	0.007042

Table 4.4: SoH prediction - Model Evaluation

The results of the models are shown in Fig.4.4. The evaluation is based on average Mean Squared Error (MSE) loss and standard deviation of residuals. Based on the results, the SVR model had the least average MSE loss and standard deviation. Therefore, SVR model was chosen as the final model.

4.3 Digital Twin

A DT model was created to represent components as they appear in the physical world. The purpose of this DT is to visualize the current operating condition of the components and to integrate predictions from the models deployed in the cloud. The following figures illustrate the user interface of the DT model. This interface provides a detailed visualization of the components, allowing users to monitor their status in real-time and access predictive insights.

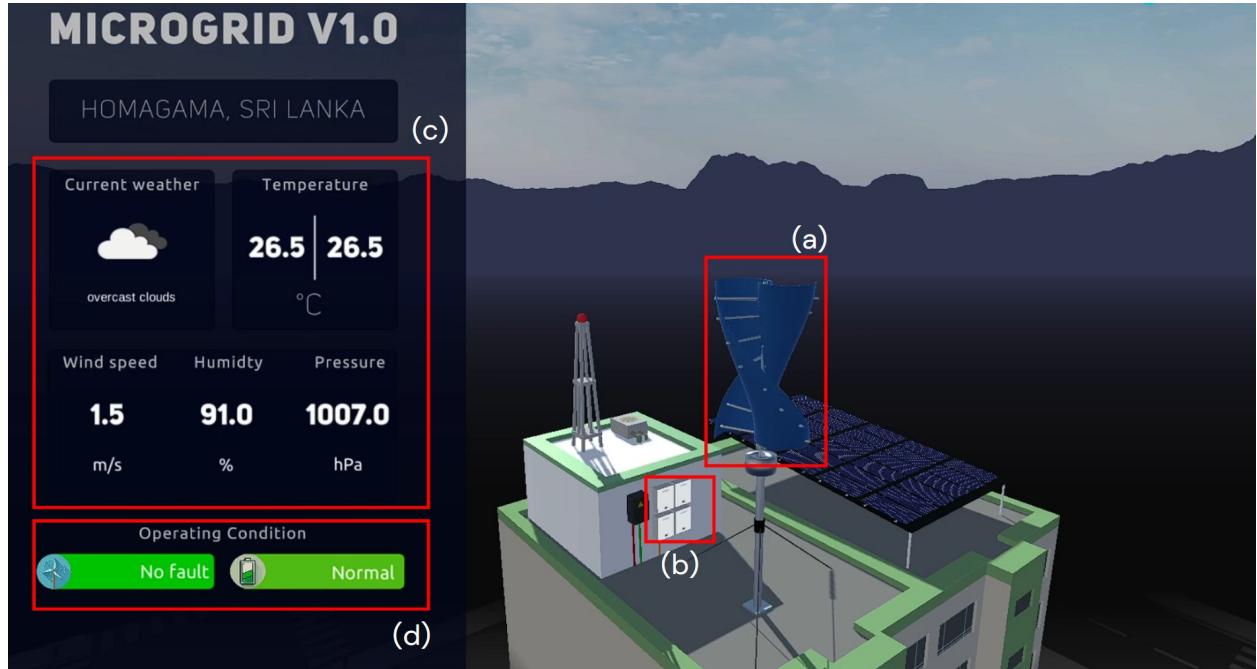


Figure 4.12: Digital Twin Main Screen

The main screen of the DT showcases 3D models of the MG components, including the VAWT (Figure 4.12(a)) and the Battery System (Figure 4.12(b)). On the left side of the screen, a panel provides an overview of the weather conditions at the location (Figure 4.12(c)). Weather data is retrieved through the OpenWeatherMap API at 10-minute intervals. The wind turbine's rotation speed is synchronized with the actual turbine, ensuring it reflects real-world conditions. However, the rate of change in rotation speed depends on how frequently the DT model updates its data. The panel below the weather panel (Figure 4.12(d)) provides a view of the operating conditions for the two components in the MG.

Each component in DT model is clickable, allowing users to view a brief overview of its parameters in a panel on the left side of the screen. For the VAWT, the displayed parameters include current, voltage, and power output as shown in Figure 4.13a. For the batteries, the SoC, SoH, Cycle count and Remaining capacity are shown as in Figure 4.13b.

A separate panel provides a detailed description of parameters and predictions from our machine learning models for each component. The wind turbine interface (Figure 4.14) displays the sensor readings, providing real-time information about the turbine's performance.

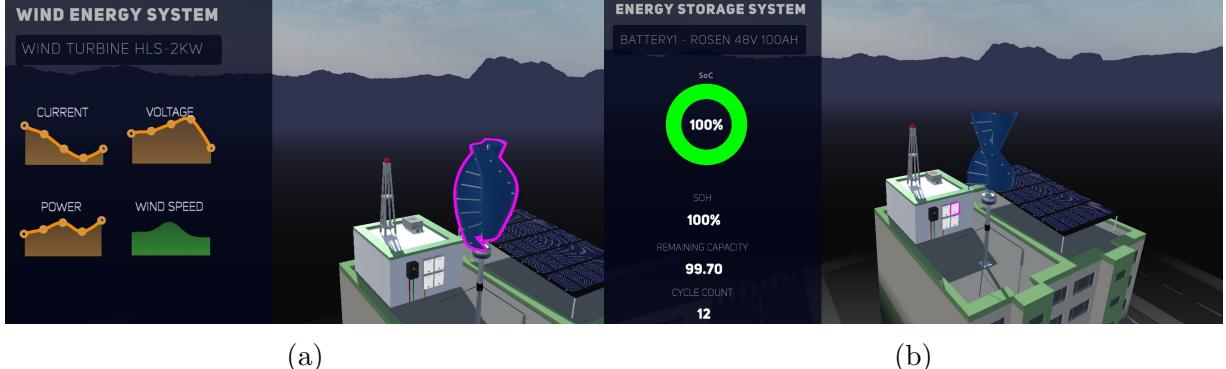


Figure 4.13: (a) Wind turbine selected (b) Battery system selected

For demonstration purposes, the public dataset serves as the input for the wind turbine model, offering a realistic data source to simulate operational conditions. The interface also shows the predictions from the fault classification model, allowing users to monitor potential issues.

The model described in Figure 3.16 is implemented in the DT. If the model detects an anomaly, the User Interface (UI) issues a warning, indicating that a potential fault has been identified. Along with the warning, the UI specifies which component is likely experiencing the anomaly, helping users to quickly locate and address the problem. (Figure 4.14b) This functionality enhances the ability to take proactive maintenance actions, contributing to improved wind turbine reliability and reduced downtime.



Figure 4.14: (a) Wind turbine interface when no fault (b) Wind turbine interface during a fault

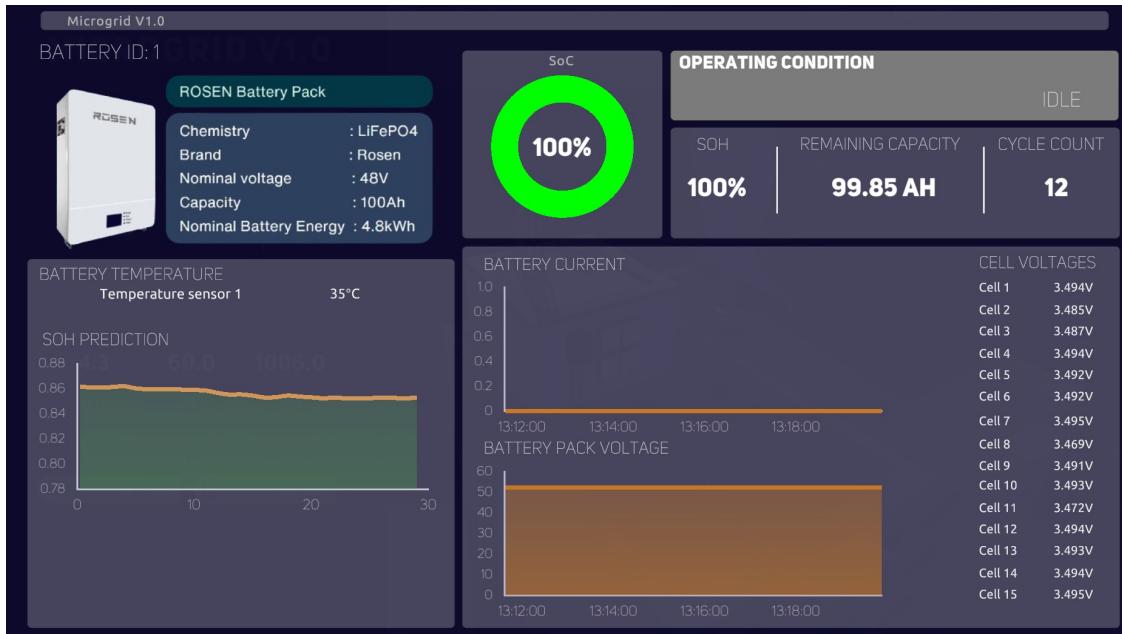


Figure 4.15: Battery Screen

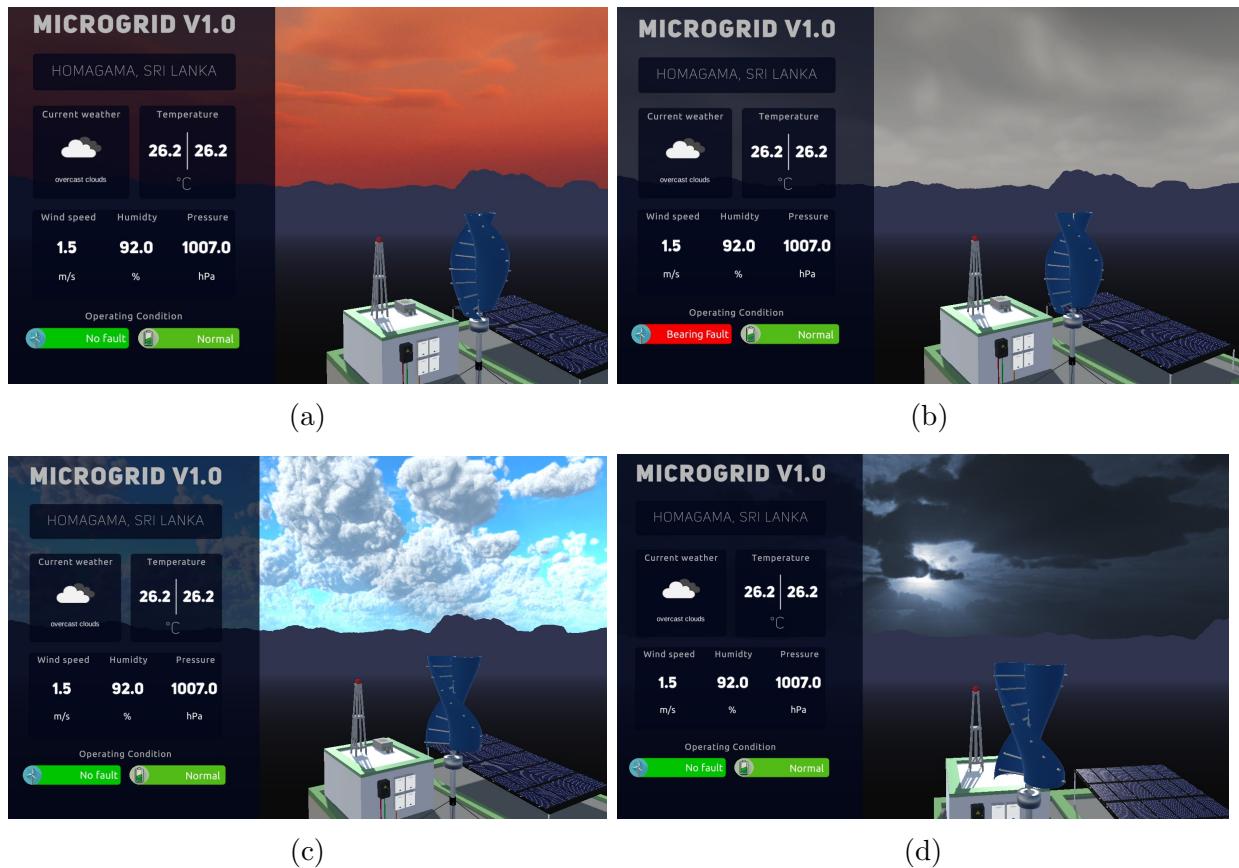


Figure 4.16: Dynamic weather in the DT

The battery interface displays real-time data from the BMS where our custom PCB was installed as shown in Figure 4.15. This interface allows users to monitor the battery's status and key metrics in real-time, providing insights into its performance and health. Additionally, the interface visualizes the predictions from the SoH Prediction model. This model indicates how the battery is expected to degrade over the next 30 cycles. It assesses the battery's condition and provides an indication of its RUL.

The integration of the OpenWeatherMap API in Unity allows for dynamic weather modeling in the DT. The skybox in the Unity environment changes according to the weather conditions provided by the API. Currently, four predefined skyboxes are implemented as shown in Figure 4.16.

Chapter 5

DISCUSSION & CONCLUSION

5.1 Discussion

5.1.1 Wind Energy System

5.1.1.1 Hardware Development

The hardware for the wind energy system uses ESP32 as the main micro-controller which is capable of WiFi and Bluetooth communications. Due to the advancements of IoT technologies, firmware for the hardware was developed to acquire data wirelessly. Our hardware collects 9 sensor data from three sensors and MQTT is used to transfer data to Azure Cloud Services. Due to the implemented method, data sampling frequency is limited to 30Hz. As the VAWT is a low RPM rotation machinery, we expect this sampling frequency to be sufficient in using the collected data for a PdM application, although a frequency domain analysis of vibration signals will always benefit from higher frequency of data collection.

The structure of the VAWT makes it difficult to design hardware for each of its components. Accessing the turbine was also a challenge to get accurate dimension measurements etc. at the initial design stage of the hardware. Hence it was decided to fix the hardware on the pole of the wind turbine to collect vibration data from the turbine. Together with the current and voltage data that will be collected from the inverter we expect the proposed PdM framework to be applicable for this VAWT.

5.1.1.2 Machine Learning Model Development

The successful implementation of anomaly detection techniques using the NASA vibration dataset and EDP wind dataset represents a critical baseline step in predictive maintenance strategies. By using these public datasets, our study demonstrates the capability to identify potential machine failures up to three days in advance, providing valuable lead time for proactive intervention.

This early detection of anomalies serves as a foundational pillar for predictive maintenance frameworks, enabling subsequent steps in fault detection and RUL prediction. By pinpointing deviations from normal operating conditions well in advance, maintenance teams can take preemptive actions to address underlying issues and prevent costly downtime. Mov-

ing forward, our research can extend beyond anomaly detection to incorporate fault detection methodologies, further refining the predictive maintenance process. Additionally, by integrating RUL prediction models, we can forecast the remaining operational lifespan of machinery components within the next 60 days, allowing for optimized maintenance scheduling and resource allocation.

However, the public version of the EDP wind dataset contains data only for four wind turbines, but the faults have occurred in five wind turbines. Turbine T09’s data is not available publicly. There are a good amount of faults that have occurred in 2016 in T09 turbine. Training without this data, makes some component’s RUL prediction models overfit to the training data, hence the prediction accuracy is lower on the 2017 test dataset. Also within the same component the type of fault that has occurred in 2016 and 2017 can be different. In such a case the pattern of anomalous data can be completely different even when the fault is occurring in the same component. Therefore, more data with various types of fault data will be needed to model highly generalized PdM applications.

5.1.2 Energy Storage System

5.1.2.1 Hardware Development

In this project, we designed a custom circuit board to address the challenge of acquiring individual battery data within the energy storage system. The “Battery PCB”, connects to each battery and collects real-time data on voltage, current, temperature, and other health readings from each cell. The PCB design prioritizes signal integrity and electrical safety, and passed various tests to ensure its functionality. The Battery PCB communicates with the BMS using an RS485 port and extracts data via an ESP32 micro-controller. Leveraging open-source firmware (esphome-pace-bms), the system transmits this data to the cloud using the MQTT protocol for further analysis and monitoring. The design also incorporates a data buffering mechanism to handle potential WiFi or MQTT connection issues.

5.1.2.2 Machine Learning Model Development

A data-driven machine learning model was developed to predict the RUL of Lithium Iron Phosphate (LFP) batteries within the energy storage system. The model leverages historical data on voltage, current, temperature, and capacity collected during battery cycling. The model utilizes data from the first 100 cycles of each battery, focusing on the difference between voltage and discharge capacity curves (ΔQ) between the 100th and 10th cycles. This difference is then summarized using the $\log_{10}(\text{variance}(\Delta Q))$ statistic as the key feature for RUL prediction. The model was evaluated using various machine learning algorithms, with Partial Least Squares Regression (PLSR) achieving the best performance on the test datasets.

The second machine learning model was developed to predict the SoH of the batteries over the next 30 cycles. SoH reflects the current health of the battery compared to its initial capacity. The model utilizes a dataset containing cycling data for four Lithium Cobalt Oxide (LiCoO₂) batteries. Four machine learning models (SVR, RF, MLP, and LSTM) were trained and tested, with the optimal model architecture to be determined based on the

results. The models use a windowing technique, feeding the model SoH data from previous cycles to predict future SoH values.

5.2 Conclusion

In this research we propose an end-to-end framework for PdM incorporating hardware development for acquiring data suitable for a PdM application. The developed hardware for the wind turbine will acquire vibration data for the operating VAWT. Together with the parameters that can be collected from the inverter which consists of fault codes as well, we will be developing a dataset which will be similar to the public dataset we have used for the model building. In the second part of the research, we propose a pipeline for PdM starting from anomaly detection, fault classification and going to the extent of RUL prediction. We show how an anomaly detection method can issue an early warning for the faults occurring in a wind turbine system then how such anomalies can be classified to identify where the fault is occurring. Finally we predict the faults occurring in a 60 day ahead time interval and by following the discussed maintenance scheme we show how maintenance cost can be reduced compared to traditional maintenance schemes.

The developed hardware for the energy storage system addresses the pain point of inability to monitor each battery in a MG system. The hardware has the capability of acquiring all the parameters of the battery BMS along with operating condition, warnings and fault codes. Then we show how availability of such data, facilitate PdM applications such as SoH prediction and RUL prediction for batteries.

Bibliography

- [1] O. Serradilla, E. Zugasti, and U. Zurutuza, “Deep learning models for predictive maintenance: A survey, comparison, challenges and prospect,” Oct 2020. [Online]. Available: <https://arxiv.org/abs/2010.03207>
- [2] O. San, A. Rasheed, and T. Kvamsdal, “Hybrid analysis and modeling, eclecticism, and multifidelity computing toward digital twin revolution,” *GAMM-Mitteilungen*, vol. 44, no. 2, p. e202100007, 2021.
- [3] K. Tidriri, A. Braydi, and H. Kazmi, “Data-driven decision-making methodology for prognostic and health management of wind turbines,” in *2021 Australian & New Zealand Control Conference (ANZCC)*. IEEE, 2021, pp. 104–109.
- [4] J. Eriksson, “Machine learning for predictive maintenance on wind turbines: Using scada data and the apache hadoop ecosystem,” 2020.
- [5] A. Chandra, G. K. Singh, and V. Pant, “Protection of ac microgrid integrated with renewable energy sources—a research review and future trends,” *Electric Power Systems Research*, vol. 193, p. 107036, 2021.
- [6] N. Bazmohammadi, A. Madary, J. C. Vasquez, H. B. Mohammadi, B. Khan, Y. Wu, and J. M. Guerrero, “Microgrid digital twins: Concepts, applications, and future trends,” *IEEE Access*, vol. 10, pp. 2284–2302, 2021.
- [7] C. Brosinsky, D. Westermann, and R. Krebs, “Recent and prospective developments in power system control centers: Adapting the digital twin technology for application in power system control centers,” in *2018 IEEE International Energy Conference (ENERGYCON)*. IEEE, 2018, pp. 1–6.
- [8] H. Mousavi, S. Mohammadi, A. Perina, R. Chellali, and V. Murino, “Analyzing tracklets for the detection of abnormal crowd behavior,” in *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2015, pp. 148–155.
- [9] W. Lu and A. A. Ghorbani, “Network anomaly detection based on wavelet analysis,” *EURASIP Journal on Advances in Signal processing*, vol. 2009, pp. 1–16, 2008.
- [10] A. Mahimkar, Z. Ge, J. Wang, J. Yates, Y. Zhang, J. Emmons, B. Huntley, and M. Stockert, “Rapid detection of maintenance induced changes in service performance,” in *Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies*, 2011, pp. 1–12.

- [11] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan, “Network anomography,” in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, 2005, pp. 30–30.
- [12] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang, “Multivariate time-series anomaly detection via graph attention network,” in *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2020, pp. 841–850.
- [13] N. Laptev, S. Amizadeh, and I. Flint, “Generic and scalable framework for automated time-series anomaly detection,” in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1939–1947.
- [14] N. Gugulothu, P. Malhotra, L. Vig, G. Shroff *et al.*, “Sparse neural networks for anomaly detection in high-dimensional time series,” in *AI4IOT workshop in conjunction with ICML, IJCAI and ECAI*, 2018, pp. 1551–3203.
- [15] M. Jankauskas, A. Serackis, M. Šapurov, R. Pomarnacki, A. Baskys, V. K. Hyunh, T. Vaimann, and J. Zakis, “Exploring the limits of early predictive maintenance in wind turbines applying an anomaly detection technique,” *Sensors*, vol. 23, no. 12, p. 5695, 2023.
- [16] “EDP Data.” [Online]. Available: <https://www.edp.com/en/innovation/open-data/data>
- [17] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, “Lstm-based encoder-decoder for multi-sensor anomaly detection,” *arXiv preprint arXiv:1607.00148*, 2016.
- [18] X. Jia, Y. Han, Y. Li, Y. Sang, and G. Zhang, “Condition monitoring and performance forecasting of wind turbines based on denoising autoencoder and novel convolutional neural networks,” *Energy Reports*, vol. 7, pp. 6354–6365, 2021.
- [19] S. K. J. Chaitali R. Patil, “Machine learning-based predictive maintenance of industrial machines.” [Online]. Available: <https://doi.org/10.14445/22312803/IJCTT-V71I3P108>
- [20] A. Betti, M. Lo Trovato, F. Leonardi, G. Leotta, F. Ruffini, and C. Lanzetta, “Predictive maintenance in photovoltaic plants with a big data approach,” *33rd European Photovoltaic Solar Energy Conference and Exhibition; 1895-1900*, 2017. [Online]. Available: <https://userarea.eupvsec.org/proceedings/EU-PVSEC-2017/6DP.2.4/>
- [21] Z. Kang, C. Catal, and B. Tekinerdogan, “Remaining useful life (rul) prediction of equipment in production lines using artificial neural networks,” *Sensors*, vol. 21, no. 3, 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/3/932>
- [22] Y. LI, T. KURFESS, and S. LIANG, “Stochastic prognostics for rolling element bearings,” *Mechanical Systems and Signal Processing*, vol. 14, no. 5, pp. 747–762, 2000. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0888327000913013>

- [23] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, “Remaining useful life estimation of engineered systems using vanilla lstm neural networks,” *Neurocomputing*, vol. 275, pp. 167–179, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231217309505>
- [24] X. et al., “Remaining useful life prediction of lithium-ion batteries with adaptive unscented kalman filter and optimized support vector regression,” *Neurocomputing*, vol. 376, pp. 95–102, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231219313426>
- [25] Z. et al., “An integrated unscented kalman filter and relevance vector regression approach for lithium-ion battery remaining useful life and short-term capacity prediction,” *Reliability Engineering System Safety*, vol. 144, pp. 74–82, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0951832015002148>
- [26] Şahan Yoruç Selçuk, P. Ünal, Ö. Albayrak, and M. Jomâa, “A workflow for synthetic data generation and predictive maintenance for vibration data,” *Inf.*, vol. 12, p. 386, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:240520076>
- [27] E. Söderäng, S. Hautala, M. Mikulski, X. Storm, and S. Niemi, “Development of a digital twin for real-time simulation of a combustion engine-based power plant with battery storage and grid coupling,” *Energy Conversion and Management*, vol. 266, p. 115793, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0196890422005891>
- [28] Raymon van Dinter and Bedir Tekinerdogan and Cagatay Catal, “Predictive maintenance using digital twins: A systematic literature review,” *Information and Software Technology*, vol. 151, p. 107008, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950584922001331>
- [29] A. Werner, N. Zimmermann, and J. Lentes, “Approach for a holistic predictive maintenance strategy by incorporating a digital twin,” *Procedia Manufacturing*, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:213446881>
- [30] S. et al., “Digital twins in wind energy: Emerging technologies and industry-informed future directions,” *IEEE Access*, 2023.
- [31] I. et al., “Digital twin of wind turbine based on microsoft® azure iot platform,” in *2023 IEEE Conference on Power Electronics and Renewable Energy (CPERE)*. IEEE, 2023, pp. 1–8.
- [32] H. et al., “Predictive digital twin for offshore wind farms,” *Energy Informatics*, vol. 6, no. 1, p. 1, 2023.
- [33] S. Mihai, W. Davis, D. Hung, R. Trestian, M. Karamanoglu, B. Barn, R. Prasad, H. Venkataraman, and H. Nguyen, “A digital twin framework for predictive maintenance in industry 4.0,” in *HPCS 2020: 18th Annual Meeting*. 80y5z, 2021.

- [34] M. Alhijazi, Q. Zeeshan, and H. Ghasemian, “Failure mode and effect analysis (fmea) of vertical axis wind turbines,” in *Industrial Engineering in the Digital Disruption Era: Selected papers from the Global Joint Conference on Industrial Engineering and Its Application Areas, GJCIE 2019, September 2-3, 2019, Gazimagusa, North Cyprus, Turkey*. Springer, 2020, pp. 51–62.
- [35] M. Shafiee and F. Dinmohammadi, “An fmea-based risk assessment approach for wind turbine systems: a comparative study of onshore and offshore,” *Energies*, vol. 7, no. 2, pp. 619–642, 2014.
- [36] C. Murphy, “Choosing the most suitable predictive maintenance sensor,” Sep 2020. [Online]. Available: <https://www.analog.com/en/technical-articles/choosing-the-most-suitable-predictive-maintenance-sensor.html>
- [37] “All about circuits.” [Online]. Available: <https://www.allaboutcircuits.com/industry-articles/choosing-the-most-suitable-mems-accelerometer-for-your-application-part-2/>
- [38] [Online]. Available: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/>
- [39] [Online]. Available: <https://www.edp.com/en/innovation/open-data/data>
- [40] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*. pmlr, 2015, pp. 448–456.
- [41] K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fragedakis *et al.*, “Data-driven prediction of battery cycle life before capacity degradation,” *Nature Energy*, vol. 4, no. 5, pp. 383–391, 2019.