Ibrahim Abdullahi

A Distributed Digital Twin Framework for Effective Asset Management in Industry 4.0

Faculty of Engineering and Applied Sciences

PhD in Manufacturing Systems

PhD

Academic Year 2021 to 2024

Supervisor: Dr. Stefano Longo

Associate Supervisor: Dr. Mohammad Samie

02/2025

Submitted in partial fulfilment of the requirement for the degree of Doctor of Philosophy.

ACADEMIC INTEGRITY DECLARATION

I declare that:

- the thesis submitted has been written by me alone.
- the thesis submitted has not been previously submitted to this university or any other.
- that all content, including primary and/or secondary data, is true to the best of my knowledge.
- that all quotations and references have been duly acknowledged according to the requirements of academic research.

I understand that to knowingly submit work in violation of the above statement will be considered by examiners as academic misconduct.

ABSTRACT

Wind energy is crucial for achieving net-zero emissions by 2050. However, unexpected wind turbine failures, including fires and abrupt breakdowns, pose significant challenges to the reliability of wind energy. Predictive Maintenance (PdM) is an Asset Management (AM) strategy used to forecast equipment fault and resolve them before they occur, while Condition Monitoring (CM) is another asset management strategy used to monitor the condition of equipment at any given time. This work focuses on utilizing a technology called Digital Twin (DT) – a digital replica of wind turbines' physical components in the digital domain, to support asset (equipment) management using predictive maintenance. This DT is used to achieve condition monitoring of wind turbine components and Machine Learning (ML) is applied to achieve predictive maintenance (PdM). The DT framework achieved in this work shows the use of streaming sensor data from the Supervisory Control and Data Acquisition (SCADA) system of wind turbines to constantly monitor the asset condition and use ML to anticipate component failures well in advance so as to provide operational teams with a cost-effective solution to simulate configurations that can help in managing turbine fault. While there have been successes in using digital twins and predictive maintenance in manufacturing systems, DT itself is an emerging technology that is still being explored as a concept. The explorations of the application of DT in literature highlighted various misconceptions, limitations and challenges that were necessary to look at in this study before achieving a wholistic framework that proposes a valuable and cost-effective DT implementation for the wind turbine predictive maintenance case study. To achieve the DT framework, multiple enabling technologies such as Industrial Internet of Things (IIoT), cloud computing and machine learning were considered and broken down into layers and applications. To standardize the DT framework, the ISO 23247 DT standard was used to guide the implementation of multiple DT architectures while evaluating various technology options. The methodology brought together and evaluated different machine learning techniques, software and hardware configurations, data streaming and storage technologies as well as architectural design patterns towards clarifying and validating the proposed DT framework. Experiments and simulations with open data, and results from real operational wind turbine SCADA data provided by an industry partner helped in validation.

The PhD contribution is a functional architectural framework that is effective in the adoption of digital twins for predictive maintenance in wind turbines. Thus, the framework serves operational, resource utilization, scalability, standardization and cost benefits using a distributed digital twin approach for predictive maintenance. The work highlights and covers gaps from the associated challenges and limitations such as the lack of standardization found in earlier approaches in DT research within the domain of predictive maintenance in IIoT applications.

Keywords: Distributed digital twin, predictive maintenance, wind turbines, gearbox, generator, feedback, fog, cloud, edge.

ACKNOWLEDGEMENTS

I give my profound gratitude to Almighty Allah for blessing me with the opportunity to take on this adventure. Alhamdulillah. I heartily acknowledge my parents, Abdullahi Bala Adamu, and late Hauwa Musa, for instilling resilience and lots of love and prayer that nurtured, guided and protected me. I appreciate my lovely wife, Aliyatu, for her love and support throughout this journey, and my family and friends for their kind words of encouragement.

I wish to also appreciate the support and guidance of my supervisory team, Dr Stefano Longo for his continuous belief, support and the unrelenting advice on my PhD. Additionally. I appreciate further advice from Dr. Muhammad Samie and Prof Suresh Perinpanayagam for their support and guidance which helped in my PhD journey.

I am proud to have been part of the Cranfield School of Aerospace, Transport and Manufacturing (SATM), where I have received support.

Finally, I wish to appreciate and acknowledge the Petroleum Technology Development Fund, Nigeria, for sponsoring my PhD.

Thank you.

Dedicated to my parents.

LIST OF PUBLICATIONS

The research papers contributed in this thesis are named papers 1 to 5, based on suggested reading order.

PAPER 1: *A Fog Computing Based Approach Towards Improving Asset Management and Performance of Wind Turbine Plants Using Digital Twins*

Abdullahi, I., Perinpanayagam, S., Hamidu, I., 2022. A Fog Computing Based Approach Towards Improving Asset Management and Performance of Wind Turbine Plants Using Digital Twins, in: 2022 27th International Conference on Automation and Computing (ICAC). pp. 1–5. https://doi.org/10.1109/ICAC55051.2022.9911170

PAPER 2: *Development of an Open Distributed Digital Twin Model for Effective Asset management in Industrial Internet of Things (IIOT)*

Abdullahi, I., Perinpanayagam, S., 2022. Development of an Open Distributed Digital Twin Model for Effective Asset Management in Industrial Internet of Things (IIOT). Annu. Conf. PHM Soc. 14.

PAPER 3: *Towards a Distributed Digital Twin Framework for Predictive Maintenance in Industrial Internet of Things (IIoT)*

Abdullahi, I., Longo, S., Samie, M., 2024. Towards a Distributed Digital Twin Framework for Predictive Maintenance in Industrial Internet of Things (IIoT). Sensors 24. https://doi.org/10.3390/s24082663

PAPER 4: *An Enabling Architecture for Computational Cost Efficiency in Predictive Maintenance Digital Twins*

Abdullahi, I., Longo, S., Samie, M., 2024. An Enabling Architecture for Computational Cost Efficiency in Predictive Maintenance Digital Twins, in: 2024 International Conference on Cyber-Physical Social Intelligence (ICCSI). pp. 1–6. https://doi.org/10.1109/ICCSI62669.2024.10799418

PAPER 5: *Integrating a Fog Based Distributed Digital Twin Framework with the Cloud for Improving Asset Management with Predictive Maintenance in Wind Farms* (available as a preprint, under peer review)

Abdullahi, I., Longo, S., Samie, M., Perinpanayagam, S., 2023. Integrating a Fog Based Distributed Digital Twin Framework with the Cloud for Improving Asset Management with Predictive Maintenance in Wind Farms. https://doi.org/10.2139/ssrn.5082896

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION 13
1.1 INTRODUCTION 13
1.2 BACKGROUND 15
1.2.1 Wind Turbines and SCADA Systems 15
1.2.2 Industry 4.0: Cloud, Fog, and Edge Computing 16
1.2.3 Digital Twins (DT) and Data Intelligence 17
1.2.4 Wind Turbine Asset Management 18
1.2.5 Machine Learning and Predictive Maintenance 18
1.3 RESEARCH MOTIVATION 19
1.3.1 Research Gaps 19
1.3.2 Research Aim and Objectives 21
1.3.3 Research Questions (RQs) 21
1.4 RESEARCH METHODOLOGY 24
1.5 CONTRIBUTIONS 25
1.6 THESIS STRUCTURE 27
CHAPTER 2: THEORETICAL BACKGROUND AND RELATED WORK 29
2.1 THEORETICAL BACKGROUND 29
2.1.1 Digital Twin Technology 29
2.1.2 Fog Computing in IIoT 30
2.1.3 Integration of Fog Computing with Digital Twins and CPS 32
2.2 RELATED WORK: LITERATURE REVIEW 34
2.2.1 Digital Twins for Predictive Maintenance 34
2.2.2 Machine Learning and Statistical Approaches for Predictive Maintenance 35
2.2.3 Data Requirements for Predictive Maintenance Digital Twins 39
2.2.4 Fog Computing for IIoT Applications 40
2.3 CHALLENGES, RESEARCH GAPS AND REQUIREMENTS: CONTRIBUTIONS FOR RQ1 42
2.3.1 Challenges and Research Gaps 42
2.3.2 DT Framework Requirements 42
2.3.3 Alignment with RQ1 and Thesis Contributions 43
2.4 CHAPTER SUMMARY: CONTRIBUTIONS ANSWERING RQ1 43

CHAPTER 3: THE METHODOLOGY FOR A DISTRIBUTED DIGITAL TWIN FRAMEWORK FOR DATA-DRIVEN PREDICTIVE MAINTENANCE OF WIND TURBINES 44
3.1 INTRODUCTION 44
3.2 THE FRAMEWORK 45
3.3 THE METHODOLOGY AND RESULTS 47
3.3.1 System Architecture and Experiment Setup 47
3.3.2 The Predictive Maintenance (PdM) Models and Algorithms 50
3.3.3 Evaluation and Performance Metrics 54
3.4 CHAPTER SUMMARY: CONTRIBUTIONS ADDRESSING RQ2 57
CHAPTER 4: FRAMEWORK VALIDATION 58
4.1 INTRODUCTION 58
4.2 FRAMEWORK VALIDATION: COMPUTATIONAL PERFORMANCE 59
4.2.1 Scenarios and Results for Validation 61
4.2.2 Framework Standardization with ISO23247 – Answering RQ2 66
4.3 FRAMEWORK VALIDATION: INTEGRATION WITH A CLOUD-BASED DATA INTELLIGENCE PLATFORM 67
4.3.1 Integration Architecture 67
4.3.2 Fog Enabled Security Mechanism 69
4.3.3 Validation Experiments 70
4.3.4 Framework Validation – Answering RQ3 74
4.4 CHAPTER SUMMARY: CONTRIBUTIONS ANSWERING RQ3 74
CHAPTER 5: INDUSTRY VALIDATION 76
5.1 INTRODUCTION 76
5.1.1 Aim and Objectives 76
5.1.2 Project Stakeholders 76
5.2 PROJECT METHODOLOGY 77
5.2.1 The SCADA Data and Failure Logs 77
5.3 VALIDATION RESULTS 79
5.3.1 Gearbox Bearing Temperature Prediction 79
5.3.2 Gearbox Failure Prediction Using SPC 80
5.4 CONTRIBUTIONS 80
5.5 DISCUSSION AND FUTURE PROSPECT 81
5.6 ACKNOWLEDGEMENT 82
CHAPTER 6: CONCLUSION, LIMITATIONS & FUTURE WORK 83
6.1 CONCLUSION 83

6.1.1 The Impact/Benefit: Justification of the Architectural Framework 83
6.1.2 Summary of Contributions 84
6.1.3 Challenges, Lessons, and Limitations 85
6.2 FUTURE RESEARCH 87
REFERENCES 89

LIST OF FIGURES

Figure 1: Enabling technologies for predictive maintenance digital twins with associated challenges. 14
Figure 2: Inside a wind turbine (adopted from energy.gov [11]) 15
Figure 3: Predictive Maintenance DT with some research gaps (from paper 3). 20
Figure 4: Research Question 1 (RQ1) with associated Hypotheses 22
Figure 5: Research Question 2 (RQ2) with associated Hypotheses 23
Figure 6: Research Question 3 (RQ3) and Hypothesis 23
Figure 7: Overview of research questions and gaps covered. 24
Figure 8: Overview of the research process. 25
Figure 9: Overview of contributions with metrics/outputs, research questions, and benefits. 26
Figure 10: Clarifying DT misconceptions 30
Figure 11: Proposed architecture: a fog-based distributed DT framework (paper 1) 31
Figure 12: Fog layer application with intelligent DT functionalities. 33
Figure 13: DT model hierarchy for wind turbine PHM 33
Figure 14: Distributed DT architecture 46
Figure 15: ISO23247 reference architecture mapping. 47
Figure 16: Data model for wind turbine DT based on the EDP dataset. 48
Figure 17: Implemented system architecture with middleware and models. 48
Figure 18: Experiment setup example. 50
Figure 19: Predictive Maintenance DT workflow. 51
Figure 20: Actual vs Predicted for gearbox bearing temperature. 52
Figure 21: Feature importance ranking for gearbox bearing temperature predictions. 52
Figure 22: Shewhart control chart for T06 Gearbox failure prediction 53
Figure 23: SPC operations using DT feedback mechanism—30% RPM reduction example. 54
Figure 24: T01 and T06—2016 vs. 2017 accuracy comparison by component. 56
Figure 25: Workflow for methodology/experiments 60
Figure 26: Architecture for Scenario A. 61
Figure 27a: Edge PC % CPU Utilization – Actual run 61
Figure 27b: Edge PC % CPU Utilization – Feedback run 62
Figure 28: Edge-PC Actual run vs Edge-Device feedback run CPU Latency 62
Figure 29: Architecture for Scenario B. 63
Figure 30: Cloud system % CPU Utilization – Actual run 63
Figure 31: Cloud-Actual vs Edge-Feedback system CPU Latency 64
Figure 32: Architecture for Scenario C. 64
Figure 33: Fog-PC % CPU Utilization – Actual run 65
Figure 34a: Fog-Container % CPU Utilization – Actual run 65
Figure 34b: Fog-Container % CPU Utilization – Actual run 65
Figure 35: Fog-Container CPU Latency – Actual vs feedback 66
Figure 36: Framework refinement based on experiment results. 66
Figure 37: Wind Turbines SCADA and Distributed DT Integration Architecture 67
Figure 38: Fog node placement strategy 69
Figure 39: Fog-enabled security mechanism. 69
Figure 40: Response Time – Fog vs Non-Fog Setup 71
Figure 41: Average Response Time across 3 Cases 72
Figure 42a: Security overhead response time comparison 72

Figure 42b: Security overhead response time comparison 73
Figure 43: Accuracy Comparison: Fog vs Cloud 73
Figure 44: Cost Comparison of running the framework. 74
Figure 45: Industry Validation Project Stages. 77
Figure 46: Maintenance/Failure logs from January 2021 to December 2024. 78
Figure 47a: Actual vs Predicted for T06 within failure timeframe. 79
Figure 47b: Fault prediction using SPC for T06 Gearbox fault. 80
Figure 48a: Sample predictive maintenance DT dashboard 81
Figure 48b: Sample predictive maintenance DT dashboard 81
Figure 49: NP's Detection and Active Response Curtailment (DARC) system [132] 82

LIST OF TABLES

Table 1: Selected algorithms for the PdM DT. 36
Table 2: Input vs Outputs. 37
Table 3: Summary of DT Challenges 42
Table 4: Failure log from EDP dataset 53
Table 5: Model Accuracy Results. 55
Table 6.1: Problems covered with evaluation metrics (chapter 3). 57
Table 6.2: Problems covered with evaluation metrics (chapter 4). 59
Table 7: Scenarios for computational performance experiments 60
Table 8: Dataset Scaling 71
Table 9: Validation project stakeholders 76
Table 10: NP-T06 Gearbox – Selected fault for validation. 78
Table 11: Accuracy Metrics for NP-T06 79

LIST OF ABBREVIATIONS

AAS – Asset Administration Shell

AI – Artificial Intelligence

AM – Asset Management

CM – Condition Monitoring

CPS – Cyber-Physical System

DL – Deep Learning

DT – Digital Twin

DTDL – Digital Twin Definition Language

IIoT – Industrial Internet of Things

ML – Machine Learning

NP – Natural Power Services Limited

PdM – Predictive Maintenance

PHM – Prognostics and Health Management

SCADA – Supervisory Control and Data Acquisition

CHAPTER 1: INTRODUCTION

This chapter sets the stage for the thesis by discussing it in the bigger picture of managing wind turbine assets and how important it is, with a focus on how digital twins can be used for predictive maintenance. It begins by exploring the enabling technologies—Industry 4.0, digital twins, and predictive maintenance—that address the challenges of implementing digital twin solutions for managing wind turbine assets. The chapter outlines the motivations behind this research, including gaps in existing approaches, and formulates the research questions aimed at addressing these gaps. It highlights challenges associated with integrating machine learning and IIoT for predictive maintenance using digital twins. The chapter further provides an overview of the adopted methodology to solve the identified challenges through developing a cost-effective digital twin framework. The chapter introduces the thesis contributions, with a focus on achieving standardization, scalability, and operational efficiency through DT implementation. The chapter concludes with a roadmap of the thesis, outlining how subsequent chapters build on this foundation to implement and validate the proposed framework.

1.1 INTRODUCTION

It is the year 2024, in the 4th industrial revolution (Industry 4.0) and there is a tight race to achieve zero carbon emissions by 2050 to save the environment and the world from the negative effects of climate change. One of the key factors contributing to emissions has been the need for energy in manufacturing, transportation, and all other key sectors of the world economy. Wind energy provides a viable source of renewable energy through the conversion of mechanical energy to electrical energy using a complex piece of engineering asset called a wind turbine. The International Energy Agency (IEA) [1], International Renewable Energy Agency (IRENA) [2] and the Global Wind Energy Council (GWEC) [3] suggest that by 2050, not less than a projected 35% of global energy generation will be from wind energy, requiring an estimated investment of up to 11 trillion US Dollars [4]. The GWEC emphasizes that the current pace of wind energy installation is insufficient and must be tripled to achieve NET Zero by 2050 [3]. This means that there will be a scaling in both onshore and offshore wind farms to reach this target. However, there is a problem with regards to managing wind turbine asset and this can hinder achieving this target. Wind turbines worldwide are experiencing abrupt failures, including instances of fire, despite current maintenance strategies aimed at managing these costly assets. Notable failures have occurred in wind farms such as in Portland, Oregon [5] , Scroby Sands in the UK [6] , and Bjerringbro in western Denmark [7] , among others. A decade-long study by NREL (2003–2012) on wind turbine component reliability identified the gearbox, generator, electrical system, rotor, hydraulics, and main shaft/bearing as the top six components causing downtime [8]. Despite technological advancements, the gearbox consistently remains the most failure-prone component, as confirmed by recent failure logs and case examples [5], [6], [7]. These failures, particularly those involving fires, show the critical importance of temperature monitoring as a metric for effective condition monitoring of wind turbines [9].

Nonetheless, to effectively monitor and manage wind turbine assets, the typical architecture and technology network requires an infrastructure that supports data gathering and energy production in remote wind farms. This is expensive, as wind farms are often placed in difficult terrains where wind energy is sufficient. This challenge necessitates the implementation of robust systems called Supervisory Control and Data Acquisition Systems (SCADA). The

Industrial Internet of Things (IIoT) is an extension of device-to-device communication to industrial systems, which plays a vital role in implementing modern SCADA systems. By enhancing sensors to gather and transmit data with enhanced precision and efficiency, IIoT has revolutionised SCADA systems. This presents a unique opportunity to leverage other industry 4.0 technologies such as edge computing, cloud computing, machine learning and digital twins (DT) to achieve better asset condition monitoring and utilization driven by data intelligence. Digital Twin (DT) is considered an Industry 4.0 technology that creates the replication of physical assets in a digital space [10]. In manufacturing systems, asset management refers to a systematic approach to operating, maintaining, and optimizing physical assets to maximize performance. Therefore, to improve asset management of industrial assets such as wind turbines with improved, rapid, and effective condition monitoring, DTs offer an advantage to traditional SCADA condition monitoring because of applications such as condition monitoring, predictive maintenance and simulations that can be achieved in a digital twin environment. Predictive Maintenance (PdM) facilitates forecasting component conditions in the future before failures occur, rather than preventive maintenance done with a schedule, or reactive maintenance done when a failure occurs. The application of DT for predictive maintenance further supports operational/asset management teams with improved and timely insights. However, to achieve a full-fledged digital twin that can seamlessly support intelligent asset management, studies must clear up misconceptions about DT and adopt standardized approaches to show the value of DT application in asset management for industrial systems. To achieve this, a framework is required that integrates relevant enabling technologies while addressing key challenges, to demonstrate the value of a novel approach to utilizing digital twins for enhanced asset management. Figure 1 below gives an overview of bringing these enabling technologies together, along with factors that highlight potential challenges in implementing a predictive maintenance digital twin.

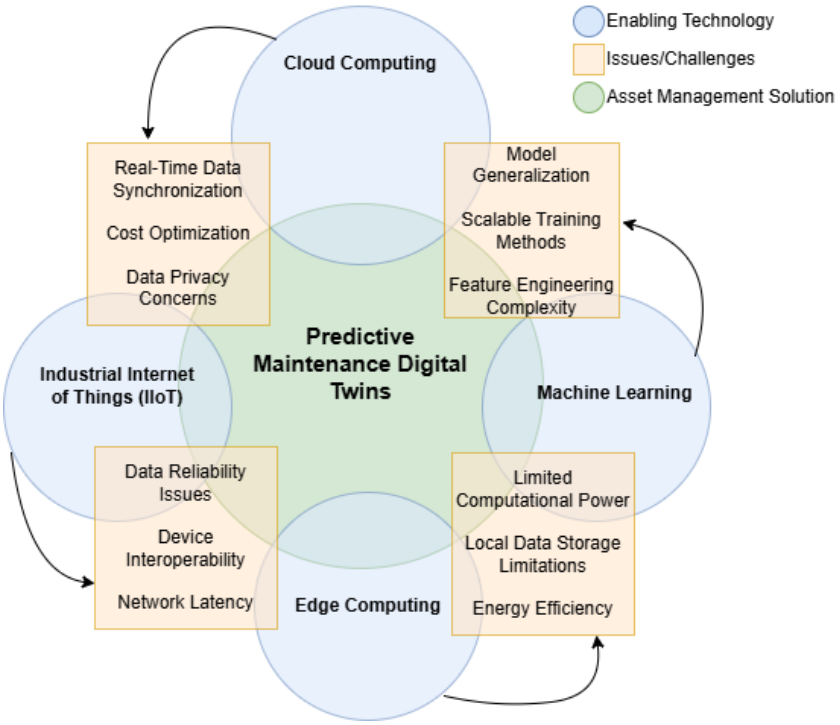


Figure 1: Enabling technologies for predictive maintenance digital twins with associated challenges.

1.2 BACKGROUND

This section gives a background on the key concepts around the case study and the enabling technologies for the proposed solution, as described in figure 1 above.

1.2.1 Wind Turbines and SCADA Systems

Wind Turbines

A wind turbine is a machine that converts kinetic energy from wind into electrical energy by using several components. As wind propels the turbine's blades, a rotation is caused which turns a rotor that is connected to a main shaft, and a gearbox which spins a generator to produce electricity [11]. Multiple other important components work together to make this power generation feasible. Figure 2 below from the US Department of Energy [11] describes a visual description of a wind turbine and some of its components.

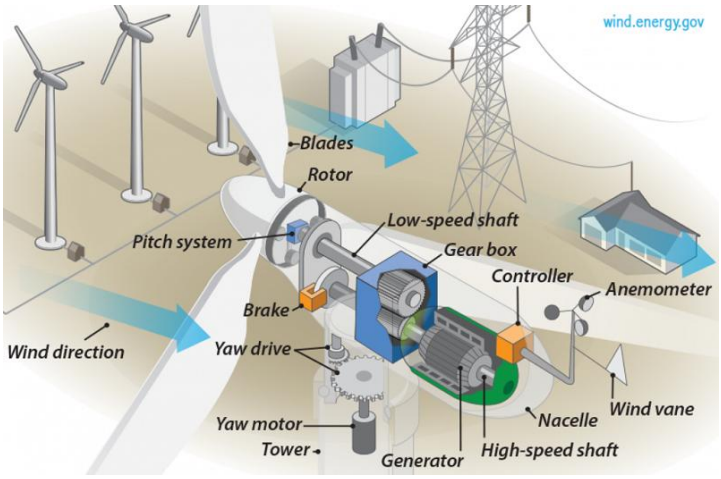


Figure 2: Inside a wind turbine (adopted from energy.gov [11])

The internal workings of a wind turbine involve mechanical systems that generate significant heat. Heat generated by the gearbox and generator subcomponents makes temperature monitoring crucial for assessing the health of these key components. Sites known as wind farms typically install a collection of wind turbines. Wind farms are typically classified into two categories: onshore and offshore wind farms. Onshore wind farms are situated on land, while offshore wind farms are situated on the sea. Depending on wind farm size, wind farms can be classified into small, medium, and large based on the number of turbines and total capacity, as shown below.

- **Small Wind Farms:** Wind farms consisting of 1 to 15 wind turbines with a total capacity of up to 20 megawatts (MW). Example: Bear Creek Wind Farm [12] (on-shore) with 12 wind turbines and 24MW capacity, located in Pennsylvania, USA.
- **Medium Wind Farms:** Wind farms consisting of 10 to 50 wind turbines with a total capacity of up to 100 megawatts. Example: Scroby Sands [13] (off-shore) with 30 wind turbines and 60MW, located off the coast of Great Yarmouth, Norfolk, UK.
- **Large Wind Farms:** Wind farms consisting of more than 50 wind turbines with a total capacity of more than 100 megawatts. Example: Whitelee Wind Farm [14] (on-shore) with 215 wind turbines and 539MW capacity, located in East Renfrewshire, Scotland, UK.

One of the largest wind farms in the world is the Gansu Wind farm [15] with 7000 wind turbines, a current capacity of 10,000 MW, and a planned capacity of 20,000 MW.

SCADA Systems

SCADA is short for Supervisory Control and Data Acquisition (SCADA), which is described as "systems used for controlling, monitoring, and analysing industrial devices and processes, consisting of both software and hardware components that enable the remote and on-site gathering of data from industrial equipment" [16]. SCADA allows operators to " remotely manage industrial sites such as wind farms, by accessing turbine data and control them without being on-site" [16]. This gives a good description of why SCADA is an important aspect of asset management in wind turbines. SCADA systems serve as a foundation for the Information and Communications Technology (ICT) infrastructure necessary for enabling data collection, monitoring, and control of wind turbine components. Furthermore, SCADA's integration with the Internet of Things (IoT) serves as the basis for the concept called the Industrial Internet of Things (IIoT).

1.2.2 Industry 4.0: Cloud, Fog, and Edge Computing

Industry 4.0 stands for the fourth industrial revolution. 'Industry 4.0' is a term coined in Germany in 2011 as part of the government's efforts to digitise manufacturing, thus symbolizing the advent of a new era in industrial production [17]. Industry 4.0 is a broad term comprising of technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), Machine Learning (ML), robotics, and cloud computing that have advanced manufacturing and other sectors. Therefore, in this work, key aspects of industry 4.0 that are important to the case study are focused on. The following sections further elaborate on the key enabling technologies for the digital twin framework in a background.

Industrial Internet of Things (IIoT)

Internet of Things (IoT) is the concept of device-to-device communication between physical objects which enables the collection and sharing of data from embedded sensors [18]. The extension of this concept to manufacturing and industrial systems, often regarded as Cyber-Physical Systems (CPS), is termed the Industrial Internet of Things (IIoT). In essence, the application of IoT in SCADA based systems such as wind turbines can be described as Industrial Internet of Things (IIoT). The SCADA system in wind turbines include embedded sensors that measure telemetry such as temperature, vibration, rotation etc. from specific sub-components such as gearbox bearings and transmit them through Programming Logic Controllers (PLCs) and other sub-systems for processing and control. Due to the nature of industrial assets and the data generated, local computing systems (servers) often support the data collection. However, for more advanced analytics and intelligence, the cloud is usually employed for scalability.

Real-Time Data Transfer and Control

'Real-time' in the context of this research denotes considering specific time constraints whether in nanoseconds, milliseconds, or seconds as required by the sensors and actuators of an IIoT case study.

Cloud Computing

Cloud computing delivers computing services including servers, storage, databases, and networking over the internet [19]. This computing paradigm allows computing operations and advanced analytics to be performed thousands of miles away from where the data is generated therefore allowing robust, managed, and efficient computing at scale. Cloud computing has further enabled IIoT by providing scalable resources for data storage, processing, and analytics. However, due to the nature of cloud architectures, an integration with SCADA-based systems like remotely located wind turbines require navigating challenges such as real-time control security, data privacy, and cost [20], [21].

Fog Computing

Fog computing is a term introduced by Cisco in 2014 [22], as a paradigm that extends cloud computing capabilities to the edge of the network closer to where data is generated [23]. This architecture was designed to complement traditional cloud-based models. To further promote its adoption, the OpenFog consortium was founded in 2015 by ARM, Cisco, Dell, Intel, Microsoft, and Princeton University. The OpenFog consortium provided a reference architecture for the design and development of fog-based systems to support industries from transportation, agriculture, and manufacturing, to energy (such as wind farms) where the IoT domain is specifically required [24]. Researchers have since then explored and adopted fog computing and its potential to improve advanced analytics in IIoT [25]. As research on the potentials of fog computing furthered, it was necessary to explore how the benefits of fog computing can be leveraged for improving applications such as in Prognostics and Health Management as in this study.

Edge Computing

Edge computing is similar to fog computing. However, edge computing enables IoT devices to process data directly by eliminating the need to rely on the fog or cloud infrastructure [23]. This feature supports low-latency applications with the capacity to handle instant processing and decision-making. Most recent advances in computing hardware such as NVIDIA's Jetson for edge computing [26], provide the ability to deploy AI on IoT devices. While this stands the chance to accelerate decision-making in most IoT applications, SCADA-based systems such as wind turbine control are currently limited in this type of integration. Therefore, traditional SCADA systems that support wind farms require an integration with other external systems for advanced analytics. In this work, edge computing is limited to the IoT devices and PLCs embedded within turbine components for data collection and control.

1.2.3 Digital Twins (DT) and Data Intelligence

The concept of digital twins was first presented in 2002 by Micheal Grieves at the University of Michigan, as a concept ideal for Product Life Cycle Management (PLM) [27]. In the presentation, Grieves described the concept of a real space and a virtual space with a two-way data and information flow between them [27]. The National Aeronautics and Space Administration (NASA) further adopted the concept and re-introduced its application in which they defined it as "an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin" [28]. As the digital twin concept became more popular in both industry and academia, multiple definitions and applications of it became

more apparent. However, misconceptions remained in both its definition and application. Work by [29] did well to clear these misconceptions by distinguishing a digital twin from a digital shadow or digital model based on the two-way automatic data flow that synchronizes the physical and digital states of an asset, rather than a manual data flow in a digital model or a semi-automatic data flow in a digital shadow. This aligns with both Greive's and NASA's definition of digital twins. Furthermore, DT applications reviewed in [30] [31] suggested that over 85% of DT research is in maintenance and Prognostics and Health Management (PHM) focusing on manufacturing assets. The adoption, implementation, and application of DT to improve manufacturing processes shows challenges and limitations with regards to standardization and design considerations that can support its effective utilization throughout the lifecycle of an asset. This PhD work explores DT in the context of maintenance in manufacturing to cover some of the identified research issues further elaborated in chapter 2 (literature review). A core aspect of realizing valuable digital twins is data intelligence. Data intelligence deals with the process of collecting, analysing and utilizing large amounts of data to enhance decision-making [32] [33] [34]. As digital twins provide realistic replicas of physical asset based on IIoT, data-driven solutions requiring data management, computation and storage are essential for achieving optimized operations [33]. Realizing data intelligence with digital twins in the context of IIoT further presents challenges as highlighted on a high level in figure 1 of this work.

It is based on this background that this PhD project focused on developing a novel distributed digital twin framework for predictive maintenance in wind turbines while considering the research gaps, challenges, and limitations in the adoption of digital twins within the PHM domain.

1.2.4 Wind Turbine Asset Management

Asset Management in this context deals with optimizing the performance of wind turbines. Wind turbines, as described in section 1.2.1, are complex pieces of engineering equipment that require adequate maintenance to achieve a lifetime of 20 to 30 years [35]. While the maintenance schedule varies from 6 months to a year, some components require more adequate maintenance due to rapid wear and tear or failure [36]. The gearbox and generator are such components, and this is why a timely condition monitoring solution for such assets is necessary to optimize operations and alert asset managers of potential issues ahead of time. The National Renewable Energy Laboratory (NREL) has reported on the evolution of gearbox failure as a component more prone to failure [8]. Predicting asset conditions in the future enables prognostics, planning, and handling of issues through the aid of predictive maintenance. This background motivated this study to investigate predictive maintenance digital twins as a case study for developing a digital twin framework.

1.2.5 Machine Learning and Predictive Maintenance

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that offers a variety of techniques for data analytics, enabling systems to learn from data by identifying patterns to support decision-making [37]. Both Machine Learning and Deep Learning (ML/DL) approaches have been used with a high level of success in implementing predictive maintenance solutions [38]. ML/DL approaches can be either ***supervised learning*** such as regression where the output is predicted based on certain input, or ***unsupervised learning*** such as clustering where the output is known by identifying patterns/behaviour from associations in

the input. In terms of the type of model used for predictive maintenance, it can be classified into either of the following [39].

- **Data-driven model:** a data-driven model in which the output is solely determined from the data (SCADA measurements from components).
- **Physics-driven model:** a physics-driven model in which the output is solely derived based on laws of physics associated with how a component works.
- **Hybrid model:** A hybrid model combines both data and physics-driven approaches.

This work relies on SCADA data from IoT sensors and hence, adopts a data-driven approach. The underlying requirement is the provision of an effective digital twin that can support the implementation of a prognostic solution leveraging digital twins and predictive maintenance models [39] [40].

Predictive maintenance is a maintenance strategy that informs asset management teams of potential equipment failure ahead of time. This is in contrast to scheduled (preventive) maintenance done in regular intervals or reactive maintenance done when an equipment failure occurs [41]. Predictive maintenance in this work's case study involves the prediction of a future state, such as component measurements from IoT sensors of an asset and then using condition monitoring or prognostic approaches to ascertain the likelihood of a failure occurring. Statistical and process control approaches such as Shewhart Process Control (SPC) have been successfully used for decades to identify deviations from normal operation in manufacturing systems. [42], [43]. Therefore, the adoption of data-driven ML/DL techniques for SCADA data prediction and statistical process control for condition monitoring gives a sound methodology for component fault prediction. This methodology [44] was adopted in the wind turbine predictive maintenance case study of this research work.

1.3 RESEARCH MOTIVATION

Digital Twins (DTs) are positioned within the industry 4.0 context as a technology that is built on other key technologies for enhancing the visibility of industrial assets and processes. This makes DTs suitable for prognostics and health management as hardware, data and software enable the implementation of predictive maintenance algorithms that can help improve asset management of CPS such as wind turbines. However, to fully utilize digital twins, their implementation and deployment methodologies need refinement for a more optimal performance that effectively improves predictive maintenance as an asset management strategy. The refinement of digital twin technology with standardization using a case study is the major motivation of this study. To achieve this, research trends, challenges and limitations were studied, and findings further validate the need for a refined digital twin framework with architectural considerations that cover research gaps and answers to research questions for improving DT effectiveness in supporting asset management in an IIoT setting.

1.3.1 Research Gaps

Despite DTs gaining popularity over the last decade, a standard definition of the concept is usually ambiguous as researchers adopt it based on their case study. However, it can be unanimously considered as the convergence of physical entities and their corresponding digital replica connected with a two-way (bi-directional) automatic update on status, condition, and configuration, which enables the fusion of data and intelligence that can facilitate decision support in a Cyber-Physical System (CPS). This is just an example of one way of seeing what

a DT is and could be. In essence, there exist misconceptions about what a DT is, as shown in the review by [29], where many applications presented in literature merely describe a digital shadow or a digital model instead of a digital twin. These works [29], [30] showed that more than 40% of potential use cases of DT is in Manufacturing. The relevance of DT in manufacturing has also been highlighted in other reviewed literature in this study. However, some gaps exist in DT research, being an emerging technology. A high level of these gaps in the literature on DTs [38] is in the implementation of predictive analytics, whether using data-driven, physics-driven or hybrid models for manufacturing systems using digital twins [39]. Figure 3 below gives this high-level view of some key research gaps.

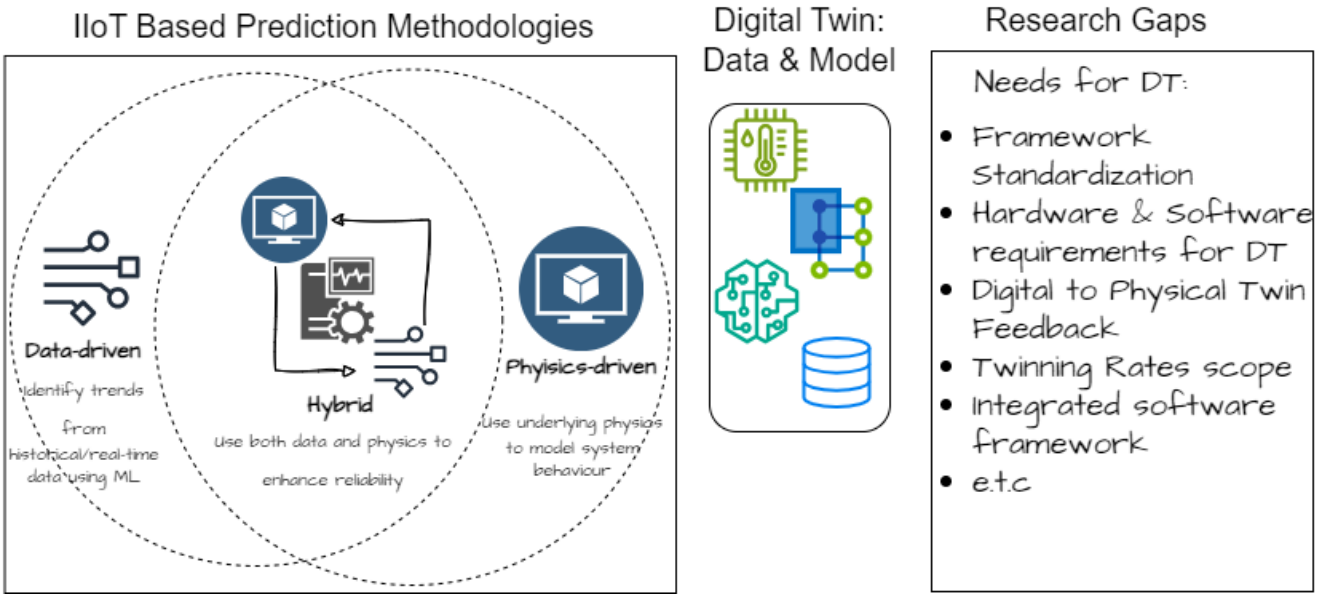


Figure 3: Predictive Maintenance DT with some research gaps (from paper 3).

One of the challenges identified with current trends in DT is the lack of standard frameworks that effectively manage the features, characteristics, requirements, and components in the development of DTs. A standard called ISO23247 [45] has been identified as a recent development towards the standardization of DT in manufacturing. However, works in literature have not showcased its utilization as detailed in systematic literature reviews [38], [46]. In addition to this, research in this domain has shown that the potential benefits of using digital twins are in condition monitoring and predictive maintenance of IoT assets, and this adds to the requirement of handling data and analytics with high velocity and volume at scale in DT architectures.

In consideration of the above-mentioned challenges, the need arises for a suitable architecture for deploying DTs, with a model that supports an architecture that will improve the performance of the system by gathering, processing, and analysing data for decision support using DTs. Based on some existing theoretical and empirical concepts [47] and recently introduced [48], a decentralized approach to DTs can foster lower latency and lower computation cost, thereby indicating the potential of distributed fog computing which can optimize intelligent DTs by decomposing at the edge, fog and the cloud. This brings an opportunity to cover some of the requirements that highlight gaps in implementing predictive maintenance DTs. This can be achieved through the design and implementation of an architecture that considers standards such as ISO23247 [45] which the model will adopt. This

should ideally cover the components of the model and the metrics that will justify the performance and validation of the proposed model.

1.3.2 Research Aim and Objectives

Aim

The aim of this PhD work is to develop a distributed digital twin framework suitable for asset management using predictive maintenance in Industrial IoT Systems with the following key objectives.

Objectives

1. To apply the proposed DT framework in supporting decentralized and intelligent PHM (with improved feedback mechanism).
2. To develop the distributed digital twin framework based on standards.
3. To use fog computing as a tool for improving efficiency (and effectiveness) of digital twin solutions, while ensuring performance based on metrics (communication latency, computation cost, cloud services cost, and handling security).

To achieve the above stated objectives, the PhD research asked some questions that cover the objectives in more detail. Objective 1 is elaborated and covered by research question 1, objective 2 is elaborated and covered by research question 2 and objective 3 is elaborated and covered by both research questions 1 and 3. The following section discusses these research questions.

1.3.3 Research Questions (RQs)

Formulating the research questions helped in adequately unlocking the potential of the proposed digital twin framework. The research questions were designed to untie and address some of the limitations in the integration of predictive maintenance and digital twins, as concepts from IIoT, cloud computing, edge computing, and machine/deep learning are essential to achieve an effective predictive maintenance solution for wind turbines. A total of three research questions were formulated around the introduction of **fog computing** as a concept that enables distributed computing by enabling compute resources closer to the data sources. Each research question has its own set of hypotheses evaluated towards achieving the research objective. This section summarises the questions and the hypotheses assessed in the study.

RQ1: ***What is the most suitable architecture that can handle data at the right synchronization rate in a DT through all layers and components of an IIoT system, considering efficiency in terms of accuracy, computing resource utilization, and enabling feedback from the digital twin to the physical twin (the IIoT Asset)?.***

RQ1 is aimed at finding an appropriate implementation architecture that positions the proposed digital twin framework as a solution to established challenges such as achieving the right synchronization rate (in real-time or near real-time: milliseconds or seconds – based on case study requirements), and computational requirements of a predictive maintenance solution using DTs. This research question builds on the application and potential of fog computing as a paradigm that complements centralized cloud computing approaches in deploying digital twins. The conceptual framework towards achieving the predictive maintenance digital twin for wind turbines was introduced based on the need to answer this question. Building on related works [25], [47], [48], fog computing was reviewed for its significance in developing the

proposed framework. The hypotheses further break down elements of the research question to inform the study whether fog computing qualifies to be considered in the next stages of the research. Hypothesis 1 (H-1) was used as a benchmark to study fog computing, its applications from literature, and its potential to improve some performance metrics such as communication latency (response time), compute resources and support for modular/distributed DTs that can enable feedback. Hypothesis 2 (H-2) builds on H-1 by conducting a qualitative assessment of how fog computing enables decentralization of the unit (sub-components e.g. rotor - RPM), system (gearbox and generator) and system of systems (whole turbine or wind farm) level of abstraction in a CPS, and the potential of such distribution in supporting two-way feedback in DTs. Hypothesis 3 (H-3) connects H-1 and H-2 to provide a fact of the bigger picture on how a fog-based architecture can support PHM by enabling a feedback mechanism. Figure 4 summarizes RQ1 along with its hypotheses.

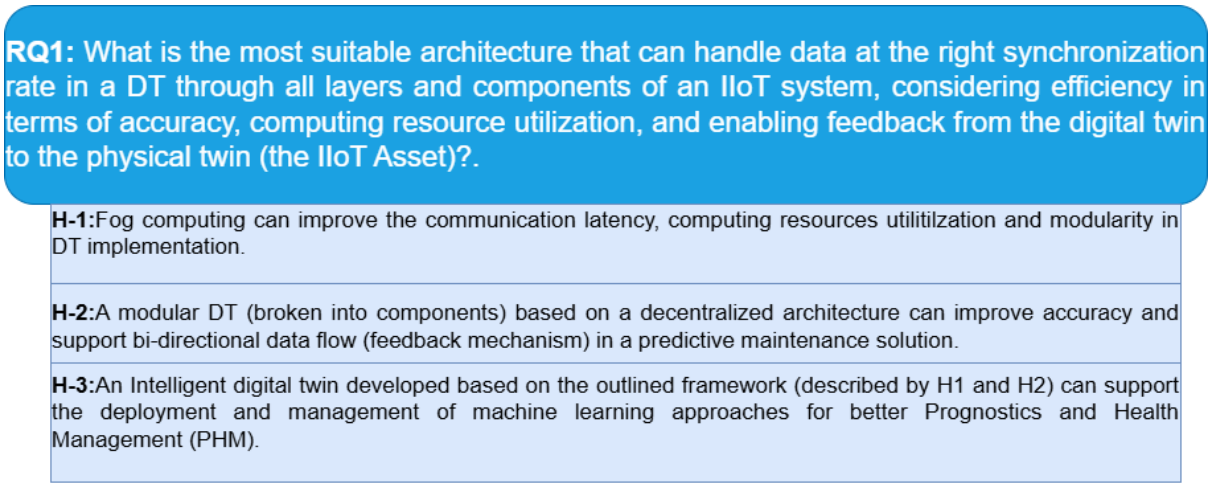


Figure 4: Research Question 1 (RQ1) with associated hypotheses.

RQ2: ***What are the requirements for building a functional DT as in RQ1 that uses a standard framework for DTs?***

Having established a conceptual framework and metrics in RQ1, RQ2 investigates the gap in adopting standardization in digital twin development. This research question is aimed at satisfying the requirement to standardize the proposed framework with accepted standard frameworks that when adopted can provide seamless adoption in other use cases. As DTs are still maturing, it was necessary to check for completeness of the proposed digital framework and how it fits within the broader context of DT standardization. Beyond the wind turbine predictive maintenance case study, answering this question is meant to showcase the feasibility of such a framework in aligning with upcoming standards, particularly the ISO23247 – digital twin framework for manufacturing [45]. To satisfy this question, requirements of a digital twin framework using the wind turbine PdM case study as an example are established. To further this, hypothesis 4 (H-4) allowed the consideration of DT development as an IoT development using Software Development Lifecycle (SDLC) principles to quantify the value of adopting a standardized framework to the product (PdM DT). Hypothesis 5 (H-5) checks the developed DT for an improvement to the PHM of wind turbines, having adopted a standardized framework to achieve it. Figure 5 below describes RQ2 and its hypotheses.

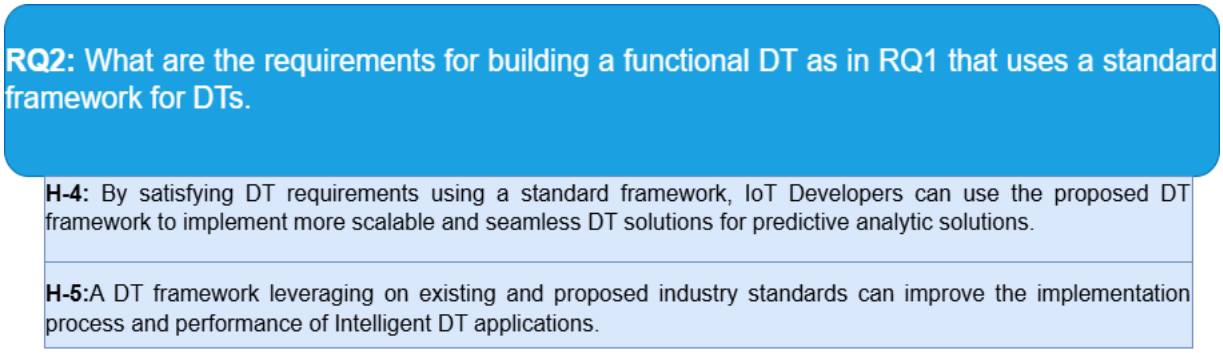


Figure 5: Research Question 2 (RQ2) with associated hypotheses.

RQ3: W*hat is the impact of the proposed framework on existing approaches in SCADA-based wind turbine predictive maintenance?*

Once the predictive maintenance digital twin for wind turbines was developed to answer questions 1 and 2, there was a need to quantify its value in terms of cost effectives, scalability, and the potential to handle security. It is an industry-standard requirement to ensure the security and data privacy of any technology. These requirements necessitated asking the question of how introducing the additional fog layer improves rather than impair cost and security to the impact of the digital twin framework in supporting wind turbine predictive maintenance. Based on this, a validation was designed to ascertain how integrating the framework with a real-world cloud-based data intelligence platform beyond the initial experiment rig could highlight the benefits of the framework in a real-world deployment. Hypothesis 6 allowed the evaluation of where the fog nodes can be placed, while hypothesis 7 helped discern how the fog nodes can improve security and scalability without negatively impacting the performance of the overall DT framework.

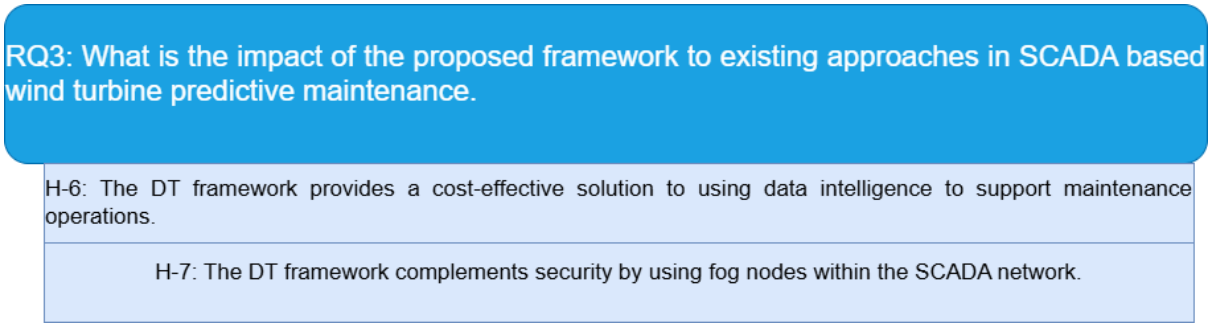


Figure 6: Research Question 3 (RQ3) with associated hypotheses.

In Summary, figure 7 below gives an overview of how the research questions cover associated challenges and research gaps identified in the literature, towards developing a predictive maintenance digital twin such as the case study in this work.

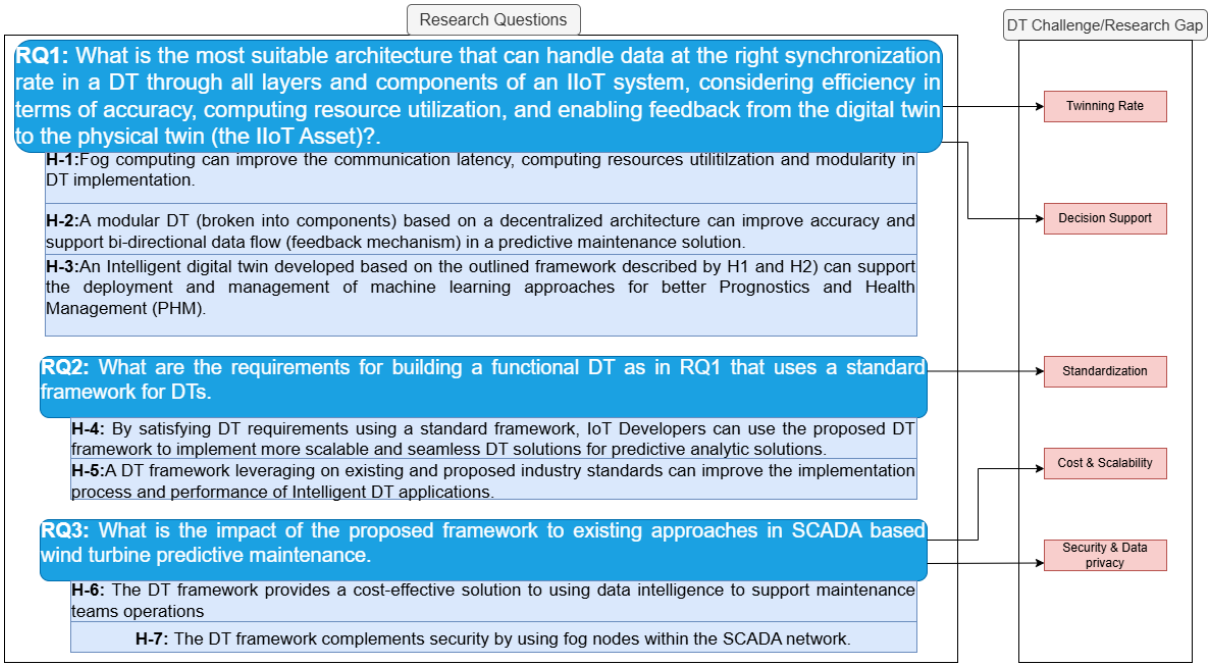


Figure 7: Overview of research questions and gaps covered.

1.4 RESEARCH METHODOLOGY

The research process adopted for this work was a robust methodology that used a hybrid approach. The methodology combines elements of Design Science Research [49] and Experimental Computer Science and Engineering (ECSE) [50]. DSR is a research strategy that uses a problem-solving approach to evaluate artefacts that address identified problems, enabling innovation that advance both theoretical knowledge and practical applications systematically [49]. ECSE on the other hand, focuses on the empirical investigation of computational artefacts—such as hardware systems, software systems, algorithms, and processes—through observation, experimentation, and analysis[50]. Below is a description of how this work adopts these strategies.

- **Design Science Research (DSR):** At its core, the research methods in this work adopt a DSR in that it approaches the problem by building a theoretical foundation using a literature review to identify gaps and formulate research questions and hypotheses, then goes further to design and implement a solution that shows practical application benefits. Using an iterative approach, this practical application is reviewed using experiments leading to a more refined solution with research outputs at each stage. The experiment aspect is where the ECSE methodology comes in.
- **Experimental Computer Science and Engineering (ECSE):** ECSE as a methodology, enabled the research to evaluate and validate the digital twin framework based on an empirical evaluation of the performance in terms of metrics such as accuracy of the PdM algorithms, computational resource utilization and communication latency (response time) in different architectural scenarios, and the overall effectiveness of the digital twin. In an iterative approach, this helped validate the predictive maintenance digital twin in experiments set up in real-world scenarios.

Overall, the adoption of a multimethodology approach by combining qualitative and quantitative techniques [51] enabled the achievement of the novel distributed digital twin framework in this PhD study. Figure 8 below captures a high-level view of the research process.

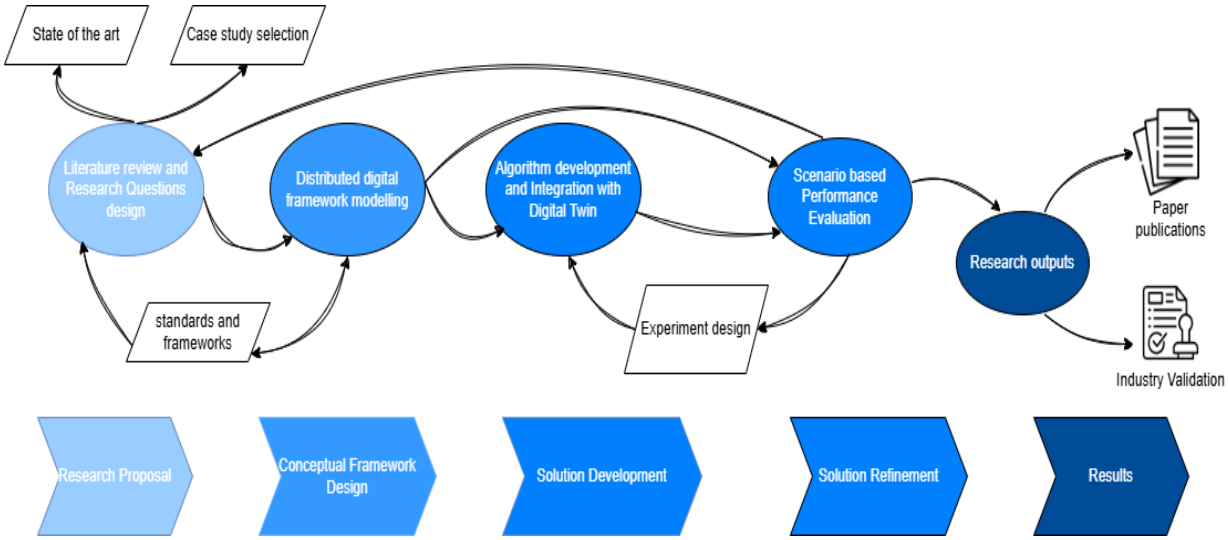


Figure 8: Overview of the research process.

1.5 CONTRIBUTIONS

In simple terms, this PhD work contributes to theory and practice. Theoretically, this thesis contributes a digital twin framework that improves the implementation of predictive maintenance digital twins using fog computing to support modularity and decentralization. The practical benefits of this contribution are in the following key areas.

1. **Scalable DT Architecture for Predictive Maintenance:** The provision of a guiding architecture for implementing DTs that enhance effective decision support using machine learning and statistical techniques. This contribution covers the feedback from the digital to physical twin research gap. Furthermore, this contribution shows how the framework improves performance metrics such as response time between physical and digital twins (i.e. improving the twinning rate) and compute utilization for ML workloads. This practical improvement shows benefits of less infrastructure cost and scalability.
2. **DT Standardization Benefit:** Improving the adoption of a digital twin standard by utilizing the ISO23247 and showing its application across the enabling technologies of the solution.
3. **Security Integration Benefit:** This PhD work shows the contribution of practical benefits of the proposed architecture in terms of how it supports additional security mechanisms that can be implemented to secure predictive maintenance solutions.

At a high-level, this section breaks down the key contributions of this doctoral thesis by highlighting how the research findings answer the research questions outlined in section 1.3.3. Overall, the contributions of this PhD study can be described in the following form.

- **Paper publications:** This forms the core of the thesis, as the published research papers serve as research outputs that cover specific aspects of the contribution of this thesis. These are the papers named paper 1, 2, 3, 4 and 5, which will be discussed and

referenced throughout this thesis. Figure 9 below gives a summarized view of these papers with the metrics evaluated, and the individual contributions towards answering the stated research questions.

- **Industry Validation:** Interest from industry was picked up based on the output of the published paper 3 of this PhD work. This led to an industry validation project with a partnership between Cranfield University and a renewable energy consulting firm called Natural Power Services Limited. Natural Power manages wind turbine assets across Europe, UK, and the US. Natural Power provided real operational wind turbine SCADA data which helped in the validation of the predictive maintenance solution developed in this study. This industry validation is discussed in chapter 5.

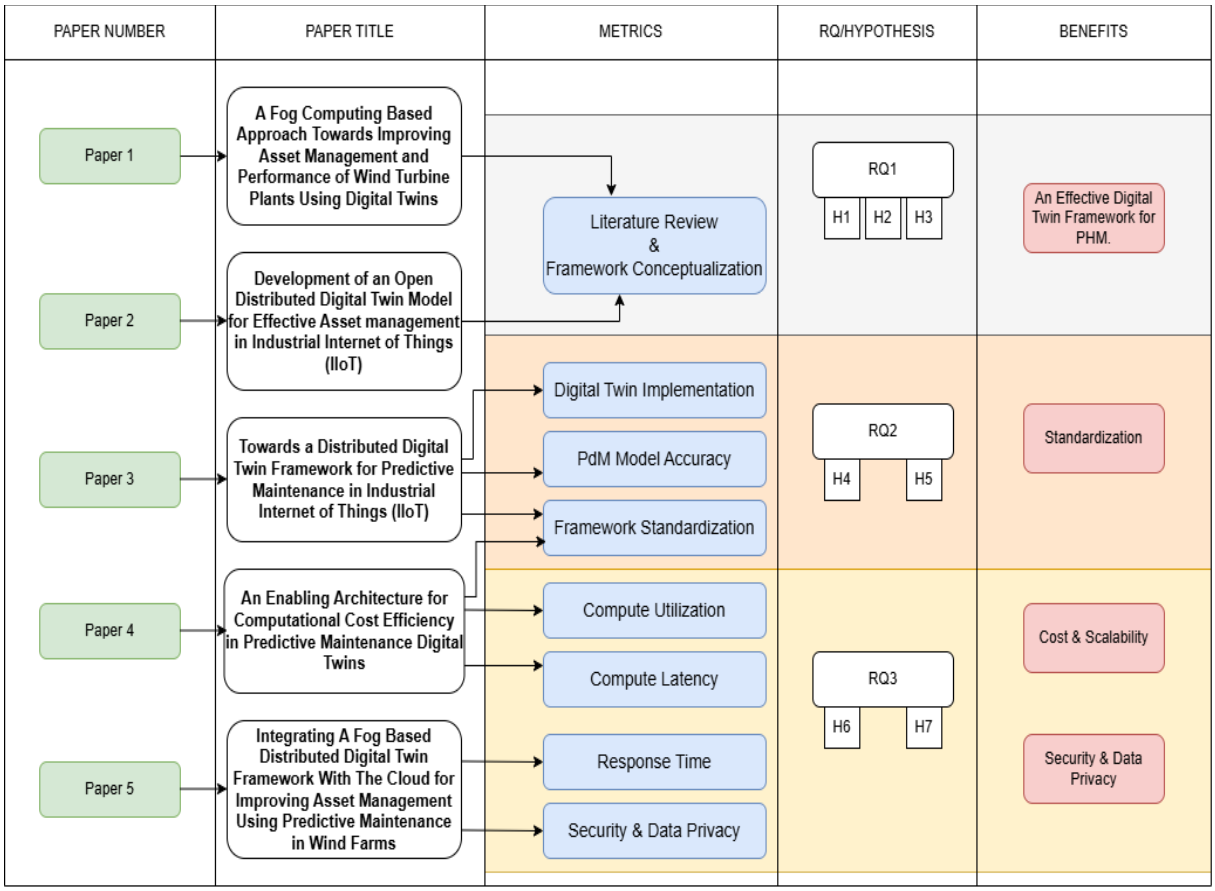


Figure 9: Overview of contributions with metrics/outputs, research questions, and benefits.

Papers 1 [52] **and 2** [53] jointly lay the foundation of the work with an exploration and literature review that provided a conceptual distributed digital twin framework architecture for predictive maintenance in wind turbines. Specifically, these papers review related works, discuss enabling technologies, highlight research gaps and present the fog computing-based architecture of the proposed solution.

Paper 3 [54] details the implementation of the distributed digital twin framework and the methodology used to achieve it. It further discusses the overall framework in terms of data, algorithms, metrics (model accuracy), experiments and results. Additionally, paper 3 introduces the two-way automatic digital twin-to-physical twin feedback simulation with practical results. **Paper 4** [55] further analyses the performance of the PdM models and fog-based DT architecture using three architectural scenarios in typical IIoT systems. Paper 4 focuses on

computational efficiency (processing time and compute utilization), thereby enabling a refinement of the architecture to align with a standard framework – ISO23247 [45].

Paper 5 [56] contributes to the integration of the fog-based architecture in a real-world setting, by using an industry-recognised cloud-based data intelligence platform to implement the proposed framework. Paper 5 further evaluates performance metrics in terms of communication latency (physical-to-digital twin response time), the cost of running the framework and its scalability. A fog-based security mechanism is also introduced in paper 5.

1.6 THESIS STRUCTURE

As summarised in figure 9, this thesis is structured based on the overall contributions (published papers) of this doctoral thesis. Each chapter linked to a published paper is aligned with the metrics – what was checked, the research questions – what was answered and the overall impact/benefit – research gap covered. A summary of the thesis organisation is as follows.

Chapter 1 introduces the thesis by highlighting the research area and the problem. It goes further to discuss the background on key enabling technologies and details the motivation behind the study. This chapter presents the research gap and breaks down the research questions, then summarises the methodology and contributions of the research.

Chapter 2 discusses the theoretical foundations pertinent to the development of a digital twin framework utilizing fog computing and other enabling technologies, towards addressing research question 1 (RQ1). It begins with an exploration of key enabling technologies, including digital twin technology, fog computing in IIoT, and the use of machine/deep learning for predictive maintenance. The chapter further reviews related work by covering digital twins for predictive maintenance, machine learning approaches for predictive maintenance, and the application of fog computing in IIoT systems. The discussion identifies existing gaps and challenges, such as the absence of standard frameworks, timely feedback limitations, data synchronization issues, and lack of standardization. The chapter concludes by summarizing these research gaps and aligning them with RQ1, discussing how papers 1 and 2 contribute to addressing these gaps and laying the groundwork for the subsequent sections of the thesis.

Chapter 3 summarises the contribution of paper 3. The chapter focuses on the methodology and implementation of the proposed distributed digital twin (DT) framework for predictive maintenance, addressing research question 2 (RQ2). It outlines the adoption of the Software Development Life Cycle (SDLC) and Experimental Computer Science and Engineering (ECSE) methodologies to design, implement, and validate the framework. The chapter details the integration of edge, fog, and cloud computing to enable condition monitoring and predictive maintenance of wind turbines at the right time. The chapter discusses the implementation of the predictive maintenance DT, with an experiment set-up highlighting key components, such as edge systems serving as SCADA, data pipelines, DT Model, and machine learning algorithms (MLR, XGBoost, LSTM). Components of the proposed architecture that support achieving accuracy, efficiency and feedback mechanisms are highlighted. The chapter also demonstrates the alignment of the framework with ISO 23247, being a standard for digital twin development in IIoT. These contributions are published in paper 3, which discuss how the framework supports predictive analytics and enables seamless synchronization between digital and physical twins in detail.

Chapter 4 is a discussion of the contributions of papers 4 and 5. The chapter validates the proposed digital twin framework and its standardization by addressing research question 3 (RQ3). The chapter begins with the contributions of paper 4, which evaluates the computational cost efficiency of the framework, comparing compute utilization and compute latency across three architectural scenarios (edge-only, edge-cloud, and edge-fog-cloud). These validations show results that suggest the framework's ability to optimize predictive maintenance performance. The chapter highlights the adoption of ISO23247 in paper 4, to standardize the framework. Next, paper 5 discusses the contribution of integrating the framework with a cloud-based data intelligence platform. The experiment set-up in paper 5 simulates a real-world industrial setting to assess response time, scalability, security, and cost-effectiveness. Additionally, a fog-based security mechanism is introduced and evaluated for its impact on the framework's performance. Contributions from papers 4 and 5 highlight the framework's significance in enhancing SCADA-based systems with the cloud for the wind turbine predictive maintenance case study.

Chapter 5 presents a report on an industry validation project with a renewable energy asset management company in the UK. The industry partner provided recent data which was used to further validate the predictive maintenance models. As the output of this PhD work, a DT platform was provided for the company's use in its asset management operations for wind turbines. This chapter highlights the outcomes of the project, highlighting with results, the validation of the framework in industry.

Chapter 6 concludes the thesis by highlighting the key contributions that justify the framework, summarizes the contributions, and discusses the challenges, limitations and lessons, as well as future research directions of the PhD research study.

CHAPTER 2: THEORETICAL BACKGROUND AND RELATED WORK

This chapter provides a more detailed discussion on the theoretical background of the key enabling technology framework that forms the conceptual digital twin framework based on fog computing. The discussions focus on the gaps and challenges that necessitate developing the digital twin framework in this study. The chapter aligns with papers 1 and 2 as it elaborates on the contributions towards answering RQ1. Overall, this chapter adopts the elements of Design Science Research (DSR) to build a theoretical foundation, and a literature review that discusses related work and presents the research gaps identified by the study. This DSR approach helped in identifying the requirements for the proposed solution.

2.1 THEORETICAL BACKGROUND

In chapter 1, the key concepts and technologies that enable the proposed framework were introduced at a high level. This theoretical background further discusses these concepts in the context of building the conceptual framework of the distributed digital twin architecture along with the methodologies that support its functionality.

2.1.1 Digital Twin Technology

Digital twins serve more as a concept than a technology when reviewed critically. This is because it is the utilization of data and software to create virtual representations of objects. The concept of Object-Oriented Programming (OOP) where software design is organised around objects that represent real-world entities having attributes (data) and behaviour (methods) [57] is relevant in understanding digital twins as a software system. Therefore, as a technology, digital twins rely on pre-existing constructs such as OOP to model the 'physical twins' whose attributes and behaviour can be represented with the right synchronization rate as they change instantaneously over time. However, the misconceptions of digital twins are caused by the flexibility of utilizing its enabling technologies to achieve digital replicas in diverse levels of abstraction or detail. To better understand this, work by [29] helped in the clarification of the concept along with Grieve's [27] initial definition. This clarification is that to qualify as a digital twin, a model must have a physical object, a digital object, and a two-way automatic dataflow between them. As such, other simulations that do not satisfy this basic requirement can be considered as something else. Below is a clarification of these concepts from [29].

- **Digital Model:** Models where the physical object is replicated with a one-way manual data flow. As an example, this can be the digital models of a physical structure designed by an expert, such as an architect.
- **Digital Shadow:** This model shadows a physical object in a digital space with a one-way automatic dataflow. An example can be a software such as typical IoT systems that present status information of assets using sensors. This model is mostly where misconceptions occur in describing digital twins as it does replicate the physical assets behaviours digitally but does not facilitate a two-way automatic data flow.
- **Digital Twin:** A digital twin is distinguished by its ability to provide replication through a physical to digital automatic and bi-directional dataflow. This physical-to-digital-to-physical makes it a two-way bi-directional automatic dataflow. This forms the foundation of the proposed framework used to implement a feedback mechanism when the ML-powered digital twin predicts a fault.

Figure 10 below is extended from [29] , and shows a visual distinction between digital twin misconceptions.

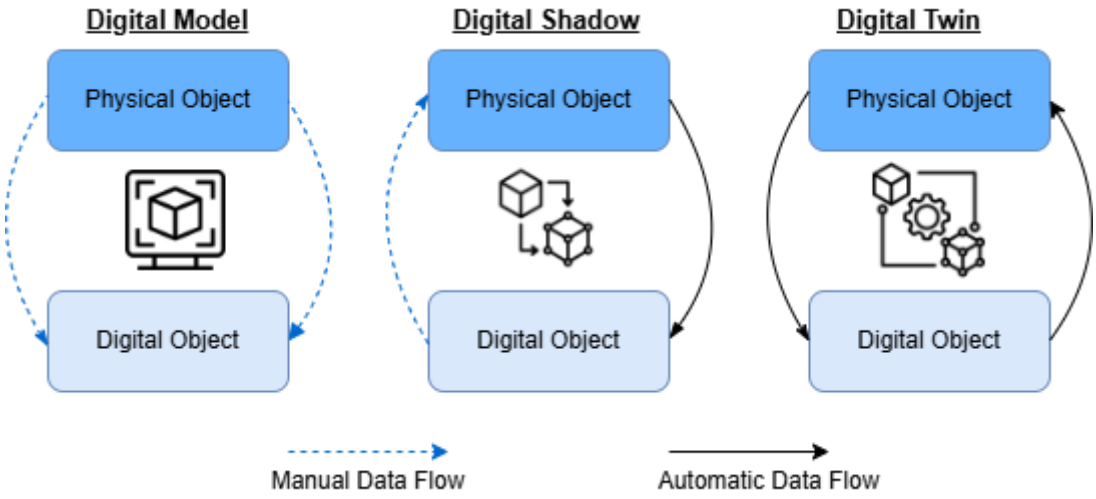


Figure 10: Clarifying DT misconceptions

A digital twin provides features such as data, software and hardware that can utilize a bi-directional automatic dataflow to support predictive maintenance. The ability to utilize this data to identify trends and predict future states of a physical asset, simulate **what-if scenarios**, as well as automate feedback to the physical asset to manage it, positions DT as a key enabler for predictive maintenance. The term '**what-if-scenarios'** is used to describe events where certain conditions can be assessed without affecting the normal operation of an asset.

2.1.2 Fog Computing in IIoT

As cloud computing emerged, providing a cost-effective solution to building secure and reliable solutions, companies conveniently deploy software, data, and applications thousands of miles away from where the data is generated. Cloud service providers manage data centres that allow the provisioning of infrastructure, platforms, and software, as services. Cloud computing has revolutionised computing by adopting this approach. However, with the emergence of Internet of Things (IoT) , where data emitting devices are embedded into real world assets such as home appliances, vehicles and industrial systems such as wind turbines, cloud computing introduced its own challenges concerning the latency of transferring such vast amount of data from IoT devices to far away data centres for processing. This necessitated the introduction of fog computing [23]. Fog computing positions infrastructure capable of handling IoT data for immediate processing and storage and only sends latency-insensitive data to the cloud. The OpenFog consortium [24] explains fog computing as an extension of cloud-based computing that 'selectively' moves computing, storage and communication closer to the data sources [24]. OpenFog consortium gives the definition of fog computing as 'a horizontal, system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum.' [24] .The reference architecture overview with a series of case study examples [24], breaks down the application of fog computing architectures considering factors such as N-tier design, security, networking, as well as data, analytics and control. In terms of IIoT systems, applications in factories, energy grids and other manufacturing sites like oil and gas rigs, require low-latency processing to handle the vast amount of data to be utilized for actionable insights. This presents fog computing as an

architecture worthy of consideration in the proposed framework. While challenges are present in its adoption, utilizing its benefits to solve other identified challenges especially in DT implementation presents an opportunity to cover research gaps.

This work, therefore, uses wind turbines as a case study to build on the foundation of fog computing within the domain of IIoT, and specifically predictive maintenance digital twins deployment using this paradigm. Figure 11 below shows the conceptual architecture of fog computing integration with an IIoT application such as wind turbines, introduced in paper 1.

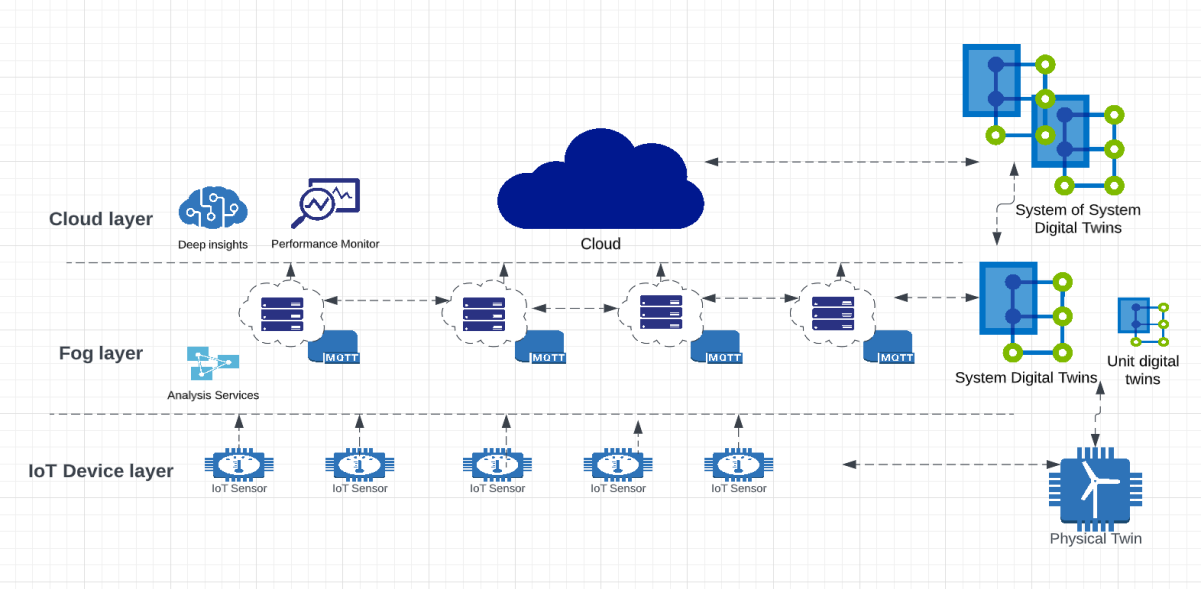


Figure 11: Proposed architecture: a fog-based distributed DT framework (paper 1)

The overall architecture presented in figure 11 is envisioned to support the achievement of a digital twin solution for intelligent PHM in wind turbines. Paper 2 [53] further discusses the components of the architecture needed to achieve this. In summary, below is an excerpt from paper 2, on the components of the layered architecture reviewed and implemented in chapter 3. “

- **Computational:** The relevant hardware and software as a requirement for the implementation of the DT.
- **Software:** This includes all relevant data extraction, data wrapping middleware, data intelligence algorithms, as well as the actual DT software for users.
- **Data Platform:** this deals with data storage, and data management of historical and real-time data for relevant predictive analytics models.
- **Functional:** This will be the application of the services of the DT – the predictive maintenance models.
- **Connectivity:** This will handle the communication of physical to digital and the middleware for integration of components to the open DT architecture.”

Fog computing as an extension of the cloud computing model in a distributed manner, was evaluated and recommended by industry experts forming the OpenFog consortium [24]. Works in literature [25], [47], [58] also found the paradigm to have capabilities that enable the above mentioned components. Ticking this checklist in formulating a digital twin framework further motivated the research to go further into adopting the fog-based architectural paradigm.

2.1.3 Integration of Fog Computing with Digital Twins and CPS

Fog computing supports a modular design of networked systems in terms of both physical topology and software architectural design [24]. In systems engineering and Cyber-Physical System (CPS) design, there are methodologies, standards, and reference architectures such as ISO/IEC 15288: Systems and Software Engineering — System Life Cycle Processes [59], Industrial Internet Reference Architecture (IIRA) [60], and RAMI 4.0 (Reference Architectural Model for Industry 4.0) [61] that adopt a modular component-based design using object-oriented design for manufacturing to breakdown an interconnected system into unit, system and system of systems. This paradigm is rooted in industry 4.0 principles where the hierarchical model shows how the complete system works, with some levels of interdependency [62]. This approach is also in line with Software Development Life Cycle (SDLC) principles and DT standards such as [45] and [63]. Work by [62] explores this methodology within the realm of developing digital twins. Therefore, to develop digital twins as software modules that replicate CPS, the adoption of such design principle is necessary.

From figure 11, the concept of the fog-based architecture is introduced to adopt the above-mentioned standards and align with the object-oriented design for manufacturing in which a wind turbine digital twin is broken down into the following layers.

- **Unit level:** These are smaller individual components that have both physical functionality and cyber capabilities such as data collection and communication. Examples can include temperature sensors (on the gearbox, nacelle, etc.) and actuators from the control pitch, yaw, and braking systems. Others in this category can include gearbox bearings, generator bearings and rotor.
- **System Level:** These are a collection of units that work together to perform a specific operational function. In this work's wind turbine case study, this level consists of components like the gearbox and the generator. While each are standalone systems that have sub-components, the rotation amplified from the gearbox is used by the generator to generate electricity, for instance. This shows the interdependency between the systems.
- **System of Systems:** This can be the Nacelle Control System that houses the gearbox and generator, which integrates data from various sensors and manages actuators to optimize turbine performance. However, the system of systems can also be considered as the whole wind turbine itself, or a network of turbines within a wind farm that work together through a centralized control system to optimize the farm's overall performance.

Regardless of the level of abstraction or classification used in the modular design, this functional decomposition, while not explicit, represents the basic building blocks of a digital twin model. It is based on this background that fog computing was found suitable in decomposing the digital twin to align with the distributed architecture as depicted in figure 11, where the unit-level digital twins represent the sub-components (IoT devices at the edge), the system-level digital twins represents complete components deployed in the fog nodes, and the whole system of systems is the wind turbine and the whole wind farm. The proposed DT architecture is designed to place fog nodes strategically to support Prognostics and Health Management by enabling hardware and software infrastructure for feedback control from the

unit and system levels, and this is why in figure 12, the unit and system level are aligned with a fog node.

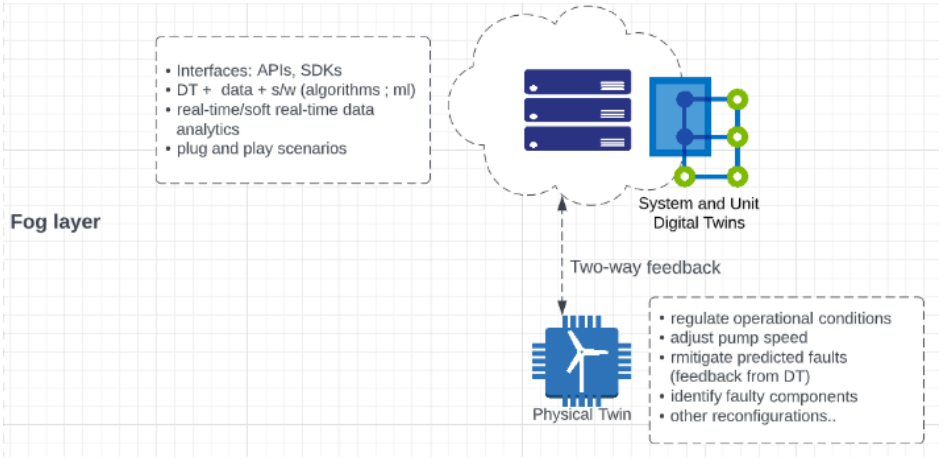


Figure 12: Fog layer application with intelligent DT functionalities.

Furthermore, to position the fog-based digital twin within the context of wind turbine predictive maintenance, the digital twin model needs to translate how the decomposed systems form part of a PHM solution. To lay the foundation for this, an object-oriented methodology [62] is adopted for the selected use case. Figure 13 (paper 3 [54]) depicts a hierarchical representation of the DT solution based on input, output and an aggregation to higher systems that facilitate achieving the PdM PHM DT solution.

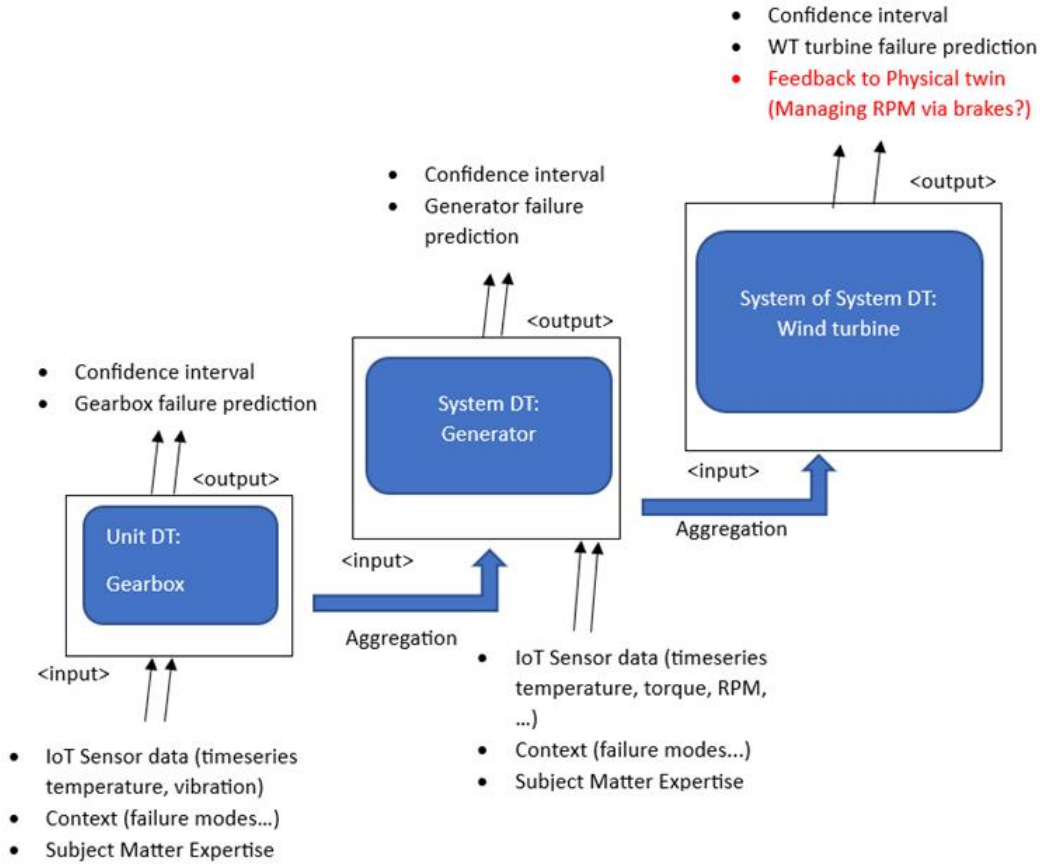


Figure 13: DT model hierarchy for wind turbine PHM

2.2 RELATED WORK: LITERATURE REVIEW

This section reviews literature relevant to the research area introduced in this study. The review focuses on related work on using digital twins for predictive maintenance and studies that explore the benefits as well as the challenges of integrating some of the key enabling technologies as is the focus of this study. At the core of the framework is the predictive maintenance solution enabled by machine learning. Some of the concepts in machine learning are discussed concerning the requirements, integration and evaluation of techniques used in predictive maintenance. This review further highlights research trends, challenges, and gaps that this PhD study covers. A research gap analysis was presented in paper 2 to establish the contribution envisioned by this study.

2.2.1 Digital Twins for Predictive Maintenance

Digital twins are most applicable in the manufacturing industry with up to 85% of its application in that domain [30], [64]. Predictive maintenance is also stated to be the use case for digital twins that attracts the most interest from both academia and industry [38], [65]. Predictive maintenance digital twins have been implemented with varying levels of detail adopting several methodologies [66], [67]. Work by [68] discusses the characteristics of predictive maintenance digital twins to have real-time perception with intelligent decision-making ability based on fault prediction and remaining useful life estimations (RUL). This study [68] further highlights the requirement for a data-aware feedback control functionality in a PdM DT architecture. This paper [69] showcased the use of a hybrid – physics and data-driven model for RUL estimation DT as a predictive maintenance strategy. Authors in [69] further emphasize the need for a reliable data acquisition system to feed the physics-based simulation. Looking at the requirements of a digital twin, work by [70] evaluated the fundamentals and state-of-the-art for predictive maintenance DTs and proposed a definition similar to NASA's [28], but suggests further that a DT should encapsulate "the full lifecycle data to facilitate knowledge sharing and integration". This further highlights the requirement for a long-term data storage layer in a digital twin framework. A study by [71] on a holistic predictive maintenance strategy using DTs discusses five steps necessary for integrating predictive maintenance DTs into existing processes. These steps by [71], are listed below.

1. **Objectives Definition and State Analysis:** Identifying components and what approach is feasible such as RUL estimation or fault prediction.
2. **Availability of data, information, and knowledge:** Using approaches such as Flexible Manufacturing System (FEM) to understand the components of the manufacturing system concerning production efficiency.
3. **Data-Driven Modelling:** How data from components will be collected and used in the DT.
4. **Physics-Based modelling:** Simulating the physical object using CAD tools and Failure Modes and Effects Analysis (FMEA).
5. **Hybrid modelling:** both physics and data-driven modelling integrated to create high fidelity DTs, with improved prediction model accuracy.

The steps above give a summary of a strategy for developing a PdM DT, and it aligns with many works in the literature that discuss PdM DTs based on a hybrid approach [72] [73]. While this PhD study does not explicitly include a physics-based model, its relevance is acknowledged, and requirements are considered in the proposed framework. In terms of

methodology, the Software Development Lifecycle (SDLC) is also relevant as a structured methodology for PdM DT development. Work by [62] introduced an SDLC-based methodology in which the DT development lifecycle includes five stages, namely; the envision, design, develop, verify and validate stages. The stages are divided into two phases: offline (data at rest) and online (data in motion). These phases when evaluated, correspond to the physics-driven (offline) and data-driven (online) DT modelling of a typical manufacturing system. While no explicit physics-driven model was implemented in this work, the SDLC-based methodology [62] is adopted. As such, the offline data relates to information from subject matter experts, included in the envisioning stage, on relevant sub-components that impact the PHM of a component (e.g. gearbox), and the online data relates to the data-driven modelling of the DT with sensor data streaming from sub-components (e.g. gearbox bearing temperature) based on the EDP open dataset [74].

The requirement for predictive maintenance digital twins to be designed in a modular manner is identified from its relevance in PHM applications and both IIRA and RAMI 4.0 reference architectures. For instance, figure 13 above describes this in the DT hierarchy. However, PHM frameworks such as [75] have discussed the impact of distributed condition monitoring using the OSA-CBM specification that promotes interoperability, reuse and reconfiguration. Researchers in [38] conducted an extensive review of 42 primary studies, addressing 10 research questions to explore current approaches, applications, and challenges in using DT for predictive maintenance. Key findings from [76] include the dominance of component-level DTs, the prevalent use of platforms like OpenModelica [77] and Simulink [78], and the use of vibration and temperature sensor data for monitoring. This study [76] highlighted gaps, such as the need for bi-directional DTs with decision support, a computational burden in handling data volume and velocity at the edge, and a lack of reference models, standards, or frameworks guiding predictive maintenance DT development to address these gaps. Furthermore, another extensive review of unit-level digital twin (DT) applications in manufacturing by [40] systematically analysed 1409 publications, identifying 96 studies that met inclusion criteria and analysed DTs by reference models (e.g. hierarchical, lifecycle), services (e.g., monitoring and optimization), DT content (models and data), and DT deployment (hardware and software). This work [40] identified key gaps, which include the lack of standardized models, mechanisms for digital-to-physical feedback, and robust hardware/software systems for real-time services. These findings directly inform the architectural considerations in this PhD work's digital twin framework, and the integration of intelligent, hybrid, and fog-based approaches.

2.2.2 Machine Learning and Statistical Approaches for Predictive Maintenance

Machine Learning Techniques for PDM.

Predictive Maintenance (PdM) leverages machine learning (ML) paradigms including supervised, unsupervised, and deep learning to optimize maintenance schedules and prevent equipment failures. These ML approaches are reviewed below.

- **Supervised learning** techniques, such as regression and classification models, require labelled datasets and are effective in predicting faults or estimating remaining useful life (RUL) [64], [79]. Supervised models like XGBoost are considered for robustness and interpretability in PdM tasks, particularly with structured datasets [44]. Work by [80] utilized XGBoost for predictive analytics (oil and gas asset maintenance), which demonstrates its capability in handling

structured data, outperforming random forest in another study as well [81]. XGBoost provides high accuracy in failure prediction. In other works, Support Vector Machine (SVM) has been reviewed and shown to have high accuracy in fault classification but deep learning such as Long Short Term Memory (LSTM) and Artificial Neural Networks (ANNs) outperforms SVM in terms of large datasets with complex non-linear relationships [82].

- **Unsupervised learning** methods, including clustering and anomaly detection, analyse unlabelled data to identify deviations from normal patterns, and this is often useful in early fault detection [83]. Clustering can also be used in detecting subtle changes in operating conditions without prior fault data [83]. Studies discuss that unsupervised learning can also aid in fault detection, particularly in scenarios where labelled data is scarce [82].
- **Deep learning** approaches, such as Long Short-Term Memory networks (LSTMs) have been suggested as effective tools for sequential data, enabling the modelling of complex temporal dependencies critical in PdM applications [84] [44]. Furthermore, while deep learning is data-intensive, the approach excels in extracting hierarchical features from raw data, as demonstrated by the superior performance of LSTMs in handling time-series SCADA data for wind turbines [44].

As this work on developing a predictive maintenance DT involves evaluating the architectural performance of deploying a PdM, it was necessary to select algorithms that will be reviewed based on complexity and accuracy, from simple to complex, as the framework is envisioned to support different ML techniques for PdM. Work by [44] showed the performance of three algorithms – Multiple Linear Regression (MLR), Gradient Boosting (XGBoost) and Long Short Term Memory (LSTM) with high accuracy in prediction of gearbox and generator failure based on SCADA data.

PdM Models: LSTM, XGBoost, and MLR

The outcome of the literature review further motivated the adoption of the three ML algorithms described in table 1 below. This satisfies the requirement of implementing the PdM algorithms in alignment with the research gaps the framework intends to cover.

Table 1: Selected algorithms for the PdM DT.

Algorithm	**Complexity**	**Description.**
MLR	Low	Linear model, no iterative learning, few hyperparameters [44]
XGBoost	Medium	Iterative boosting, non-linear relationships, multiple hyperparameters [44], [80], [81].
LSTM	High	Sequential data, deep learning architecture, computationally intensive [44].

Based on the review of previous studies, methodologies and approaches, the following justifications support the selection of LSTM, XGBoost, and MLR for the implementation of the PdM algorithms in this study. Below is a description of this justification.

- **MLR:** As a baseline model, MLR provides interpretable insights into linear relationships between features and target variables, aiding in understanding fundamental trends [79], [86].
- **XGBoost:** This gradient-boosted framework offers high accuracy and interpretability, in handling complex, high-dimensional SCADA datasets [44], [80], [81], [85].
- **LSTM:** As a deep learning model, its ability to capture long-term dependencies makes it ideal for time-series data with temporal correlations, as seen in wind turbine gearbox fault prediction work [44], [85].

Model Features

Machine learning models require certain inputs to predict an output, based on the following general formula as a relationship between inputs and outputs.

$$y = f(X) + \epsilon$$

Where:

- y: The output (target variable, dependent variable).
- X: The input (features, independent variables, predictors).
- $f(X)$: The function or model that maps inputs X to the predicted output $y^{\wedge}$
- ϵ: The error term (captures noise or variability not explained by the model).

These general principles are extended depending on the algorithm used. Therefore, using the selected wind turbine components case study, table 2 gives a summary of the inputs and outputs used in the PdM models developed in this work, identified from studies and subject matter experts.

Table 2: Input vs Outputs.

Component (System Level DT)	Input (Unit Level DT)	Output (Unit Level DT)	References
Gearbox	Nacelle temperature	Gearbox bearing temperature	[44], [87], [88], [89], [90]
	Rotor speed		
	Active power		
	Outdoor temperature		
	Gearbox oil temperature		
Generator	Nacelle temperature	Generator bearing temperature	[44], [87], [90]
	Active power		
	Generator speed		
	Generator stator temp		

Based on work done in literature on wind turbine PHM, reports such as NREL's gearbox reliability update [8] that give a background of relevant components most prone to failure over the past decade, and the methodology such as in [44], [90], table 2 provides the sub-components (unit level DT) serving as input and output for predicting the health of components (System Level DT) – gearbox and generator. The outputs indicate critical sub-components whose condition is predicted, monitored, and flagged when a fault is detected.

PdM Model Evaluation Metrics

The use of both machine learning and deep learning techniques and algorithms to implement predictive maintenance requires a comprehensive evaluation of these models to ensure their efficacy and accuracy. Work by [91] provide a thorough review of various deep learning models applied to predictive maintenance, highlighting the advantages and challenges of these approaches. One key aspect of evaluating predictive maintenance models is the selection of appropriate performance metrics. Depending on the predictive maintenance approach, a variety of metrics may be relevant, such as accuracy, precision, recall, F1-score, RMSE, MAE, R-squared, and MAPE [44], [83]. Additionally, metrics that capture the cost-effectiveness and business impact of the predictive maintenance solution, such as mean time to failure and operational availability are considered. In the predictive maintenance context, where the goal is to identify potential failures before they occur, the ability for a model to predict equipment behaviour with high accuracy can make already existing or new condition monitoring strategies more robust and effective. Work by [44], [92] showed how the use of metrics such as RMSE, MAE, and R-squared are impactful in validating a PdM model accuracy. The evaluation method using RMSE, MAE, and R-squared was adopted to evaluate accuracy of the models.

Sensitivity Analysis and Feature Importance

Sensitivity analysis and feature importance help in understanding the impact of different variables on model predictions. XGBoost specifically provides feature importance scores, and it can be deduced for other models. Studies suggest that LSTMs benefit from preprocessing techniques like feature scaling to enhance temporal pattern recognition [84]. Machine learning approaches generally support the identification of weights in models which can be used to determine parameters or features with most impact on the model as this PhD work presented in paper 3 [54]. Work by [93] discussed the importance of explainability of PdM models, as it provides insights into data, model behaviour, outcomes, and end-user interactions, thereby enhancing the transparency and trustworthiness of PdM systems.

In the case study involving the variables from the sub-components listed in table 2, the feature importance ranking is used to identify which sub-component or a collection of sub-components can be regulated through the DT. Studies on rotor speed and gearbox performance [94], [95] have indicated the impact of the rotor in the performance of gearboxes. As the rotor provides the rotational force making the gearbox operate, it was justified to model the DT feedback mechanism using regulating the RPM as an example of the feedback mechanism.

Using Statistical Approaches for Failure Prediction

In terms of fault detection, several approaches have been considered in literature to identify fault in components of a wind turbine. Some methodologies use only ML techniques while others use a combination of ML and statistical techniques. Authors in [96] developed a predictive maintenance DT using the prophet algorithm, a statistical modelling approach with elements of machine learning. The methodology in [96] involved time-series forecasting of vibration and temperature data, focusing on trends, seasonality, and anomaly detection to predict bearing failures. This study [86] used clustering combined with statistical thresholding to identify abnormal patterns. A study by [79] implements Random Forest (RF) models to classify machine states based on features extracted from sensor data, enabling fault detection and predictive maintenance. However, as this work aligns more closely with a data-driven

approach, the methodology adopted is extended on the machine learning techniques that predict component measurements and potential failures are predicted using a statistical approach. Hence, the adopted methodology distinctively separates the prediction of unit-level components' SCADA measurements and fault detection using statistical techniques. Therefore, this study finds the methodology by [44] as a suitable benchmark for the digital twin framework, as it satisfies the requirement of applying both prediction of measurements using ML and condition monitoring using a statistical process control approach. The work in [44] utilizes control limits derived from SCADA data to identify deviations indicative of anomalies leading to failure.

Statistical Process Control (SPC) is a method used to monitor and control a process to ensure that it operates at its full potential, by identifying deviations from normal operation [43], [97]. A fundamental tool in SPC is the Shewhart control chart, developed by Walter A. Shewhart in the 1920s [97]. Shewhart control charts are graphical representations that plot process data over time, allowing for the detection of variations that may indicate underlying issues. Each chart typically includes the following [42], [97].

- **Central Line (CL):** This represents the average or mean value of the process.
- **Upper Control Limit (UCL)** and **Lower Control Limit (LCL):** These are statistically calculated boundaries that define the acceptable range of variation for the process.

Adopting the SPC with wind turbine SCADA is a relevant condition-monitoring tool for wind turbines [44]. This approach was found to be suitable in satisfying some of the functionalities of the fog node in the proposed framework. Figure 12 highlights some of the fog layer's functionalities as reconfiguration and regulating operational conditions. The SPC serves as a tool for utilizing the fog layer's functionality in the DT framework. This serves as a tool for implementing the feedback mechanism in testing what-if scenarios.

2.2.3 Data Requirements for Predictive Maintenance Digital Twins

As discussed in chapter 1, predictive maintenance DTs can either be data-driven, physics-driven or hybrid. Hybrid approaches are shown to be superior as they integrate SCADA data from sensors and subject matter expertise using mathematical modelling. While the distributed DT framework in this work is hybrid, it is more aligned with a data-driven approach. High-quality data is therefore particularly important for both the creation of the digital twin to reflect asset conditions and the ML-powered predictive maintenance algorithms to make the DT intelligent. The data processing requirements for a predictive maintenance DT can be broadly classified into four components: data aggregation, data quality, data analytics, and data visualization.

Data Aggregation - Data Collection and Integration

The first step is to aggregate data from the SCADA sensors, maintenance logs, and subject matter expertise (physics modelling) and fuse them to the digital twin platform. This requires capabilities such as real-time data streaming, connectivity middleware, protocol conversion, and data storage (paper 2) [53]. This requires a DT framework to aggregate data from IoT sensors, enterprise systems, and maintenance records to enable predictive maintenance [54], [98].

Data Quality - Pre-processing Requirement

The accuracy and reliability of the digital twin model and its predictive capabilities is heavily dependent on the quality of the data [54], [76]. Data quality issues can include sensor drift, missing data, outliers, and data inconsistencies. A DT framework for predictive maintenance requires data quality checks and cleansing routines to ensure the data is appropriate for analytics [54]. Handling noise and outliers can be achieved using ML techniques such as interpolation, filtering, and outlier detection. The cleansed and pre-processed data will then be ready for analytics.

Data Analytics - Modelling Requirement

Once data is cleansed it requires to be modelled such that it reflects typical asset operation. For instance in a wind turbine DT, the data used for analytics must reflect the turbine's actual feasible operation, e.g. when wind speed is above cut-in [44]. The DT platform should enable advanced analytics and machine learning models to predict asset health, remaining useful life or both. This can include condition monitoring, anomaly detection, failure mode analysis, and predictive modelling. Work in [92] highlighted how a DT can be used to simulate the asset's performance and predict when maintenance or repairs are needed. This uses physics-based and data-driven models which are continuously updated to reflect the actual asset conditions [92].

Data Visualization - Monitoring and Decision Support

As the DT acts as a virtual representation of the physical asset, a key requirement is to provide an intuitive visualization of the asset's health, performance, and maintenance needs. The DT platform should provide dashboards and analytical visualizations to enable operators and maintenance engineers to monitor the asset's condition, gain insights, and make informed decisions.

2.2.4 Fog Computing for IIoT Applications

Fog computing decentralizes the computing infrastructure which extends the capabilities of the cloud to the edge of the network, allowing for real-time data processing, analysis, and decision-making closer to the source of data generation [24]. This approach is beneficial for IIoT applications, as it reduces latency, improves quality of service, and provides superior performance by minimizing the need for constant communication with remote cloud infrastructure [99]. Work on a fog-based architecture for digital twins [47] highlights the benefits of processing data locally at the fog layer. The research paper in [47] discuss how fog computing can enable real-time updates of digital twins by allowing edge devices to perform preliminary processing, thereby reducing the bandwidth required for transmitting data to the cloud. This minimizes latency and accelerates the decision-making process for IIoT systems, which is crucial for applications such as predictive maintenance and performance monitoring of industrial systems in real-time. Other studies have explored similar concepts, discussing a fog computing-based architecture for smart manufacturing and digital twins. These works [100], [101], [102] demonstrate that fog computing enhances the responsiveness of manufacturing systems by incorporating real-time data analytics at the edge, thus reducing the reliance on centralized cloud systems and helps in minimizing communication delays. Utilizing this concept for machine learning applications have also been explored by researchers [103], [104] and found to improve model deployment and resource management. This work [105] reviewed some of the approaches and proposed a taxonomy showing the benefits of fog

computing in resource provisioning, task offloading and load balancing. Federated learning uses distributed computing to train a model with local data, and fog computing provides an architecture that supports this technique by providing computational resources closer to the data sources [106]. This study [107] showcased the implementation of on-device intelligence based on a federated learning approach with edge devices.

Challenges in Fog-based Architectures for IIoT

While integrating fog computing into Industrial Internet of Things (IIoT) environments offers numerous advantages, such as reduced latency and improved real-time data processing, there exist several challenges that must be addressed to fully realize these benefits. Studies in literature have discussed these challenges and offered potential solutions. Some of these challenges are summarised below, referencing works that discuss them.

- **Security and Privacy Concerns:** Integrating fog nodes can introduce vulnerabilities for unauthorized access and data breaches. Addressing this requires robust data encryption, access control and intrusion detection mechanisms [108], [109], [110], [111], [112]
- **Resource Management and Scheduling:** Improper resource management among fog nodes and their applications can lead to performance degradation and increased latency. Addressing this can be from refining architectures and workflows that adopt dynamic resource allocation, load balancing and predictive analytics techniques [113], [114], [115], [116], [117]
- **Scalability and Interoperability:** As fog nodes are integrated with SCADA systems, it can become a challenge to ensure seamless communication and data exchange. To address this, it is essential to design architectures that adopt a modular architecture, considering standardization, and reliable middleware systems [106], [118], [119].
- **Data Management and Storage:** As fog computing aligns with IoT, fog nodes will manage data at high velocity and volume. As such, efficient data storage mechanism must be employed. It is also critical to decide how and which data is processed locally and which is processed in the cloud [58], [120]. Addressing this concern requires designing an architecture and framework for efficient data management [121].

Considering the above-mentioned challenges, and reviews on digital twin implementation informed by the state-of-the-art literature [64], [76], this work emphasizes the requirements of handling the issues peculiar to the envisioned digital twin framework. As the proposed framework is envisioned to utilize the flexibility and scalability of fog computing in its integration with SCADA systems such as wind turbines, challenges related to the case study are reviewed and presented in table 3 in the next section. Table 3 in the upcoming section summarizes the challenges based on the literature review performed in this study. The upcoming chapter, chapter 3, discusses the contributions of paper 3 in implementing the solution to satisfy the requirements and research gaps identified.

2.3 CHALLENGES, RESEARCH GAPS AND REQUIREMENTS: CONTRIBUTIONS FOR RQ1

2.3.1 Challenges and Research Gaps

This section provides an overview of the challenges and gaps in digital twin (DT) implementation and the integration of fog computing for predictive maintenance, aligning them with RQ1. Some of the requirements identified with the implementation of digital twins for predictive maintenance include data synchronization, modularity, and bi-directional feedback. While typical DT implementation architectures are centralized, introducing distributed architectures also presents challenges that must be addressed to achieve the full potential of digital twins. Overall, table 3 below highlights a summary of the relevant challenges identified from the literature that are pertinent to the proposed framework. Therefore, this study focuses on addressing these challenges.

Table 3: Summary of DT Challenges

Item	DT Challenge	Description	Reference
1.	Decision Support	A reliable Digital Twin to Physical Twin feedback mechanism for optimal/autonomous decision-making	[52], [54], [122]
2.	Standardization	Ensuring standardization across enabling technologies such as in architecture, interoperability, reliability etc.	[46], [63]
3.	Cost and Scalability	Optimizing infrastructure and operational costs while enabling scalability	[55], [106], [118], [119]
4.	Twinning Rate	Ensuring the right synchronization rate between the physical and digital twins.	[29], [40], [47], [123], [124]
5.	Security and Privacy	Addressing data governance, privacy, and security concerns in distributed architectures	[108], [109], [110], [111], [112]

To address these challenges, this study designed a conceptual framework, introduced in paper 1 [52] and paper 2 [53]. The contribution of these papers are the architectural frameworks introduced in figure 11, and the identification and definition of the requirements pertaining to solving the problem with novelty. The next section describes the requirements.

2.3.2 DT Framework Requirements

Data Synchronization (Twinning Rate)

The **twinning rate** refers to the right synchronization rate of statuses between the digital and physical twins for a specified case study. Existing implementations often lack standardized frameworks to manage the volume and velocity of IIoT data, resulting in inefficiencies in predictive maintenance solutions. Papers 1 and 2 highlight these challenges, emphasizing the need for robust mechanisms to ensure seamless data flow and archiving. A **connectivity middleware** and a **data platform** were highlighted to enable this requirement in the architecture.

Computational Complexity

Paper 1 demonstrates how fog computing can reduce latency and resource demands by enabling local processing, but scalability and resource allocation remain areas requiring further exploration. A **computational platform** was highlighted as a requirement to fulfil this requirement in the proposed framework.

Modularity and Feedback Mechanisms

A modular DT architecture is essential for scalable and efficient implementations. Paper 2 outlines that breaking the DT into components supports bidirectional data flow that can enhance prognostics. However, the integration of these modular components requires further considerations related to standardization and interoperability. A **software platform** that supports modularity and interoperability, along with a **functional** interface to operational teams was identified to satisfy this requirement.

2.3.3 Alignment with RQ1 and Thesis Contributions

This chapter lays the groundwork for answering RQ1 by proposing the most suitable architecture for timely data processing in a DT framework. The major contribution is establishing that by addressing gaps in data synchronization, enabling local processing on fog nodes, and adopting modularity, comprehensive predictive maintenance DTs with improved performance than typical cloud-based architectures can be achieved. Papers 1 and 2 suggest refined approaches to modularity and feedback mechanisms that can enhance predictive maintenance capabilities. While the requirement for an open DT framework suggested in paper 2 has been removed as the research progressed, all other concepts have been implemented. These developments provide the foundation for the predictive maintenance framework implemented in chapter 3. Therefore, the proposed conceptual framework introduced and discussed in papers 1 and 2 jointly contribute to establishing that a fog-based distributed digital twin framework with elements that handle **connectivity middleware, software, compute (hardware), data and functional platforms** can enable "**an effective digital twin framework for Prognostics and Health Management (PHM)**" of wind turbines.

2.4 CHAPTER SUMMARY: CONTRIBUTIONS ANSWERING RQ1

This chapter explored the theoretical background and related work necessary for developing a fog-based distributed digital twin (DT) framework for predictive maintenance, aligning with RQ1. It defines the foundational technologies of digital twins and fog computing, highlighting their roles in enabling replication with the right synchronization rate, bi-directional feedback, and reduced latency in Industrial Internet of Things (IIoT) systems. The chapter reviewed approaches in predictive maintenance using machine learning models, such as LSTM, XGBoost, and MLR, among others. The applicability of the technologies for fault prediction and prognostics are discussed. Challenges identified in literature include data synchronization, modularity, security, cost, and scalability, which hinder the realization of standardized and effective DT frameworks. This PhD study proposes a conceptual framework that leverages fog computing for local processing, modular architectures for scalability and enhanced feedback mechanisms to support intelligent decision-making to address the identified gaps. Contributions from papers 1 and 2 helped in refining the requirements while removing the requirement for an open DT system as the research progressed. These contributions form the foundation for the predictive maintenance solutions further detailed in chapter 3.

CHAPTER 3: THE METHODOLOGY FOR A DISTRIBUTED DIGITAL TWIN FRAMEWORK FOR DATA-DRIVEN PREDICTIVE MAINTENANCE OF WIND TURBINES

As mentioned in chapter 1 of this thesis, this study adopts a multi-methodology with elements of Design Science Research (DSR) and Experimental Computer Science and Engineering (ECSE). The DSR methodology was discussed in chapter 2 to establish the theoretical background and review of related concepts to establish research gaps. In chapter 3, elements of the ECSE are employed in the design, development and evaluation of the predictive maintenance digital twin based on fog computing. This chapter is aligned with the contributions of paper 3 [54] towards answering RQ2, which is relevant in achieving standardization of the framework. In paper 3, the distributed predictive maintenance digital twin implementation is discussed in detail, along with the experiment set-up, the workflow, and the application of the machine learning algorithms – MLR, XGBoost, and LSTM. The two metrics evaluated in this chapter are (1) the accuracy of the ML algorithms, and (2) the application of condition monitoring based on statistical process control to enable the feedback mechanism. The remainder of this chapter discusses the contributions from paper 3, which serve as the foundation of the PdM DT.

3.1 INTRODUCTION

The Industrial Internet of Things (IIoT) has revolutionized the way industries approach asset management. Wind turbines are a suitable case study of Industrial Internet of Things based on SCADA infrastructure. These systems require robust predictive maintenance solutions to ensure continuous operation and optimal energy generation [54]. This chapter explores how a distributed digital twin framework that leverages the benefits of fog computing and machine learning enables predictive maintenance in wind turbine systems. This is in line with the contributions of paper 3 [54] that discusses the requirements of developing a predictive maintenance digital twin in the context of Industrial Internet of Things, and proposes an architecture that can be utilized to implement such a solution considering a selected standard - the ISO23247 [45].

In a high-level, the framework features three key components; (1) a distributed digital twin architecture that replicates the physical wind turbine system based on an aggregation of digital software artefacts (JSON based DTDL/AAS), from the edge, to the fog, to the cloud layers, (2) a data collection and processing pipeline that gathers and pre-processes sensor data from wind turbine components, and (3) a machine/deep learning-based predictive maintenance model that uses statistical process control techniques to forecast potential failures and allows for feedback simulations to handle faults effectively. To achieve the framework, in line with this PhD study's RQ2, paper 3 specifically answers the following questions. Excerpt from paper 3; "

- *How can digital twins support predictive maintenance at the right synchronization rate?*
- *What benefits will a digital twin framework implemented based on a standardized framework offer to a predictive maintenance solution in IIoT?*
- *How can this proposed digital twin framework be extended to act as an intelligent digital twin with a prediction feedback loop?* "

3.2 THE FRAMEWORK

Paper 3 discusses the requirements of a digital twin solution in the context of wind turbine predictive maintenance, and implements the solution based on the outlined framework. The contributions include a breakdown of the key components of the architecture and aligning it with the requirements of ISO23247, which standardizes digital twins in manufacturing. These components ensure the effective deployment of predictive maintenance in wind turbine systems. The framework functions of each of the components are described as follows.

Edge/IoT Device Layer

- This layer acquires data in specified intervals (seconds to minutes) from wind turbine components such as the gearbox, generator, and their sub-components.
- The layer performs initial data pre-processing, including noise reduction and cleaning, before transmitting the data.
- The layer utilizes lightweight communication protocols such as MQTT to enable efficient data transfer.

Fog Computing Layer

- The fog layer acts as an intermediary between the edge and cloud, enabling localized processing and short-term data storage to reduce latency.
- This layer hosts predictive maintenance algorithms that analyse timely data streams to detect early signs of failure.
- This layer implements microservices-based architecture for modular and scalable digital twin functionalities.
- This layer integrates a feedback loop that facilitates bidirectional communication between the digital and physical twins.

Cloud Computing Layer

- This layer executes high-computational machine learning models with historical data, including XGBoost, LSTM, and MLR, to predict failure trends.
- The cloud layer stores and manages historical turbine data for long-term analysis and model retraining.
- This layer provides visualization tools and interfaces for monitoring wind turbine health and maintenance planning across the whole wind farm.

Digital Twin Core Domain

- The core domain implements the digital representation of turbine components based on Digital Twin Definition Language (DTDL) and Asset Administration Shell (AAS).
- This domain allows the simulation of operational scenarios to assess potential faults and recommend mitigation strategies.
- This domain ensures synchronization at the right time between sensor data and digital models for up-to-date asset representation.

User Interaction and Feedback Mechanism

- This serves as the application layer supporting a user-friendly interface that enables maintenance engineers to interact with the digital twin for decision-making.
- This interface provides alerts and predictive insights through dashboards and reports.
- This interface facilitates operational adjustments by sending control commands from the digital twin to the physical assets.

These components collectively provide an architecture that enhances the application of predictive maintenance in wind turbines. Utilizing this solution will ensure efficient fault detection, reduced downtime, and optimized performance. The high-level architecture is presented in figure 14, as an excerpt from paper 3.

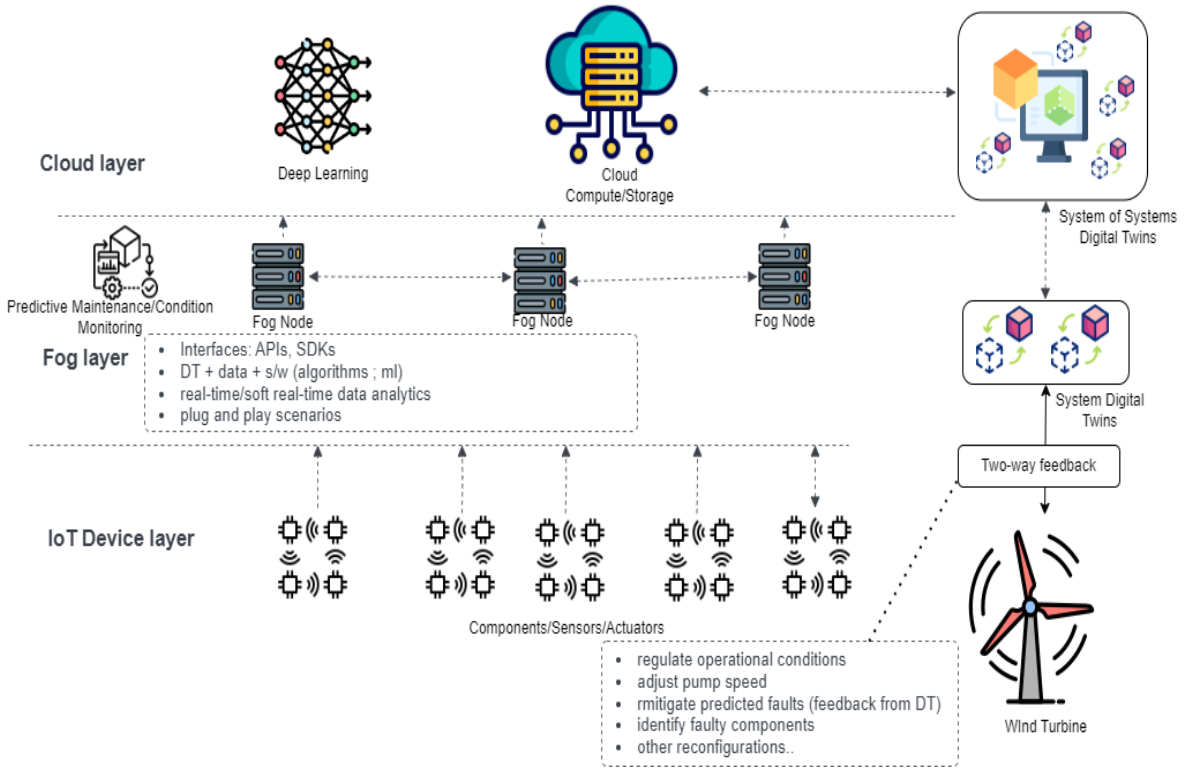


Figure 14: Distributed DT architecture; here the layered architecture which represent a novelty with respect to previous approaches can be observed.

The framework's adoption of ISO23247 to guide its development, as discussed in paper 3, is depicted in figure 15, satisfying the following key sections of the reference architecture.

1. **Observable Manufacturing Element (OME) Domain:** Context for the physical twin (each wind turbine component), which is the basis of the DT. This interacts with DT interfaces for data collection and device control–feedback mechanism.
2. **Data Collection and Device Control Domain:** This connects the physical twin (OME) to its unit DT through sensor data collection, synchronization, and actuating feedback to regulate operational conditions with decisions from the DT.
3. **Core Domain:** This domain handles all DT services from analytics and simulations to feedback and user interaction.
4. **User Domain:** This is the application layer through which users access the DT and see results through visualizations, alerts, and other functionalities.

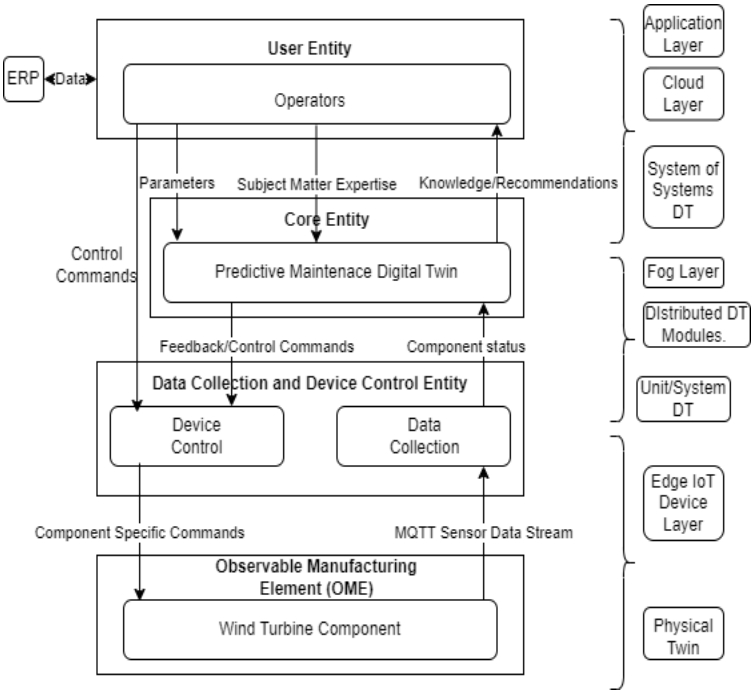


Figure 15: ISO23247 reference architecture mapping.

The methodology section will delve into the implementation details of each component, by discussing their role in achieving the goal of the framework in line with the requirements highlighted in chapter 2.

3.3 THE METHODOLOGY AND RESULTS

This section outlines the methodology adopted to design, develop, and implement the distributed digital twin framework for predictive maintenance, as presented in paper 3**.** The approach follows the Software Development Life Cycle (SDLC) for systematic development and Experimental Computer Science and Engineering (ECSE) for validation through controlled experiments, in iteration. The methodology is structured into key phases in alignment with the framework described in section 3.2 above, designed to answer RQ2.

3.3.1 System Architecture and Experiment Setup

System Architecture

The framework follows a three-layered distributed architecture—edge, fog, and cloud—to enable real-time condition monitoring, predictive analytics, and feedback control. Using an SDLC approach, the architecture is designed and implemented by executing the following stages:

1. **Planning:** This built on the requirements identified from chapter 2, the adoption of the conceptual distributed DT framework based on fog computing and the requirements from ISO23247 (figure 15) for the predictive maintenance case study.
2. **Design:** The DT framework was designed to align with the unit, system and system of systems levels of abstraction that relate to the edge-fog-cloud hierarchy. This design allowed for modularity, based on OOP concepts. Figure 16 below shows the software architecture modelled based on the EDP [74] dataset used in this work.

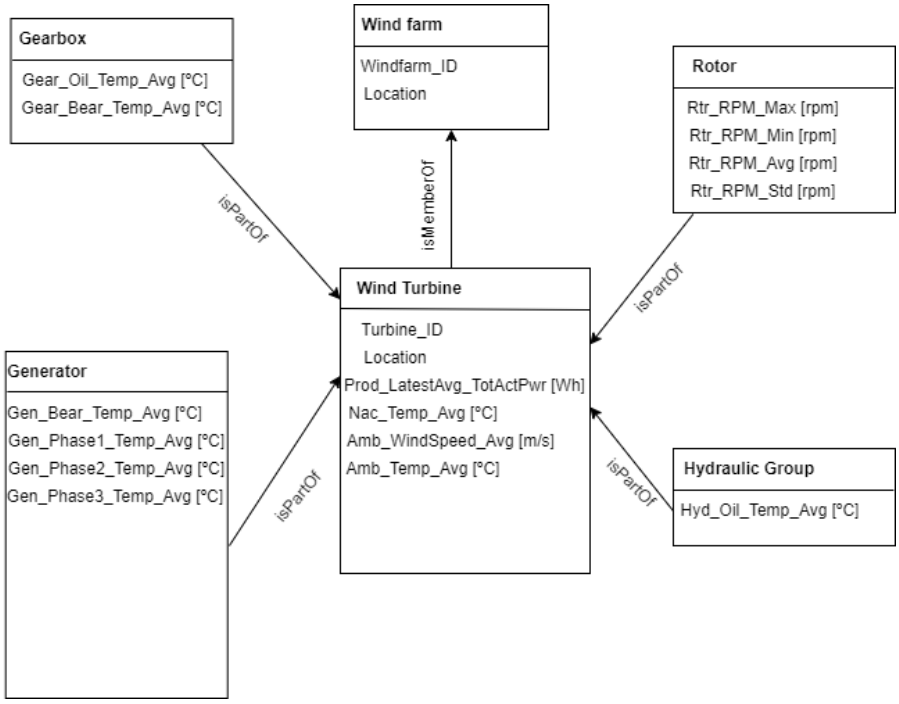


Figure 16: Data model for wind turbine DT based on the EDP dataset.

3. **Development:** The digital twin was implemented using Digital Twin Definition Language (DTDL) as a lightweight JSON-based language for translating physical component properties and their statuses. This configuration reflects the use of Asset Administration Shell (AAS). The connectivity between the data from sensors and the digital twin instance was implemented based on Message Queuing Telemetry Transport (MQTT) for publish/subscribe between the physical asset and the digital twin. InfluxDB with a Telegraf agent was used for real-time data storage. Additionally, the machine learning models were developed and tested individually before integration with the DT data pipeline depicted in figure 17.

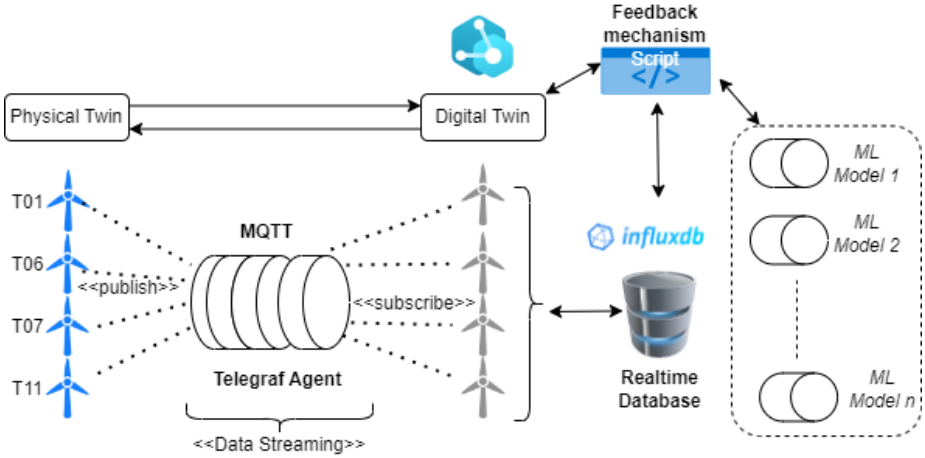


Figure 17: Implemented system architecture with middleware and models.

A data pipeline was designed to ensure a structured and scalable process that involves the following aspects.

a. **Acquisition:** Sensor data from wind turbines is collected via MQTT, simulated at 1-second intervals (depicting real-time).
b. **Pre-processing:** Data anomaly/outlier detection, noise filtering, and standardization are applied at the edge layer.
c. **Data Transmission**: Streaming of the EDP wind turbine SCADA data from independent SCADA sensors (raspberry pi device from rig) to fog and cloud layers using MQTT and telegraf.
d. **Storage:** Processed data is stored in InfluxDB for short term at the fog layer and for longer term in the cloud layer. Records are identifiable by timestamps, turbine ID and individual SCADA component measurements as depicted in the data model in figure 16.

The layered approach further ensures that real-time sensor readings are streamed efficiently and managed based on requirements of the DT, while historical data is accessible for model retraining.

4. **Testing:** In iteration, the ECSE methodology was employed in testing and validating the developed digital twin and the performance of the ML models. This iterative approach helped in improving model performance, validating system performance and failure detection using the SPC condition monitoring based on the EDP dataset and failure logs of 4 wind turbines.

Deployment/Experiment Setup

The deployment of the digital twin and the models was done in a controlled experiment setup. The predictive maintenance digital twin was hosted on edge devices, fog nodes, and cloud infrastructure, based on the distributed architectural framework earlier described in figure 14. The experiment setup adheres to ECSE principles, ensuring controlled data collection and evaluation based on the selected metrics. The contribution of paper 3 was on validating the accuracy of the predictive maintenance models and the digital-to-physical twin feedback mechanism using the developed DT. However, contributions from papers 4 and 5 in chapter 4 discuss further evaluation of other metrics. Specifically, paper 4 evaluates the computational latency, while response time, security, and cost of running the framework are discussed in paper 5. Figure 18 below depicts the example of an experiment setup in certain stages of the research, comprising the following components of the architecture.

- **Edge:** Raspberry Pi devices for real-time sensor data acquisition.
- **Fog:** Mini-PCs for local data processing fault detection and anomaly.
    - **Database:** InfluxDB for historical and real-time data storage.
    - **Containerization (Docker):** Simplifies cross-platform middleware deployment while reducing infrastructure overhead.
- **Cloud:** High-performance server for longer term data storage, model training and retraining.

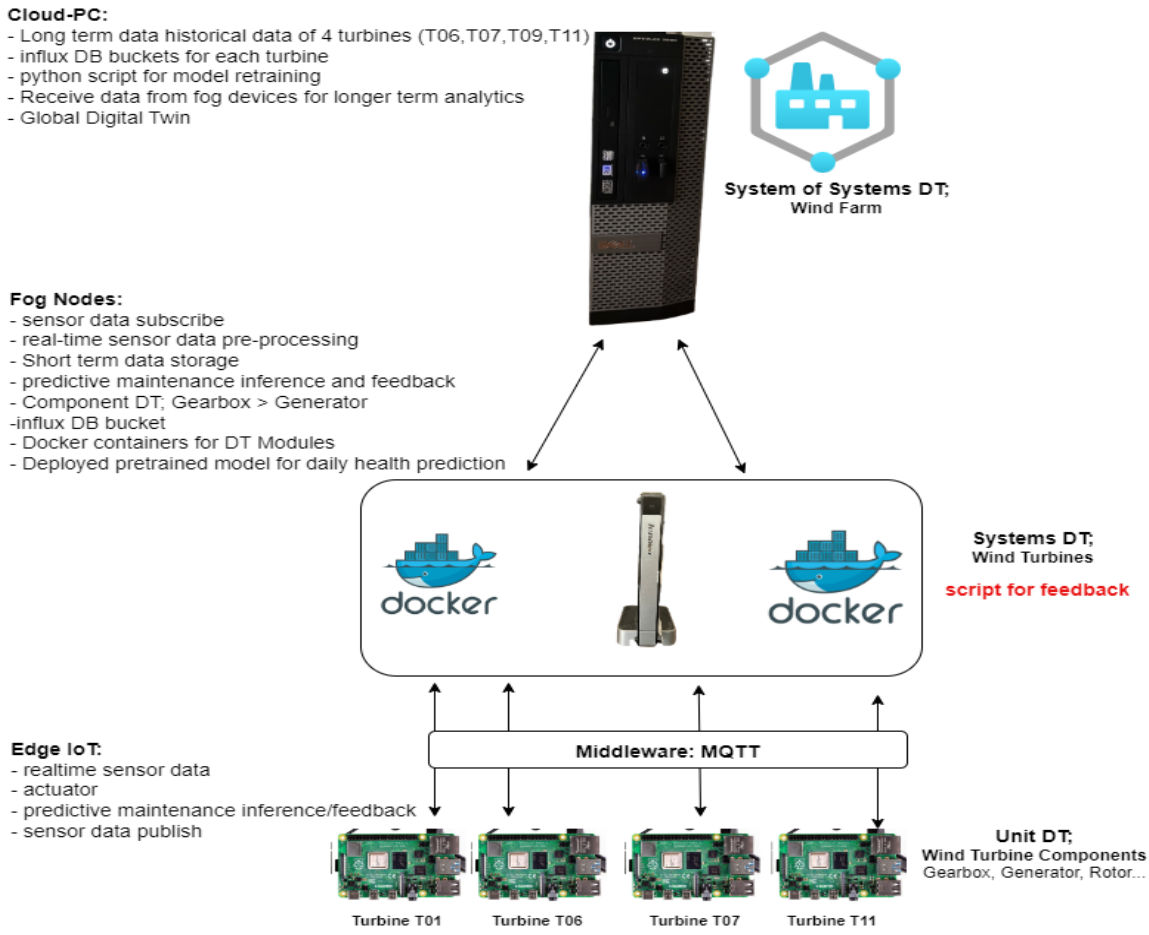


Figure 18: Experiment setup example.

3.3.2 The Predictive Maintenance (PdM) Models and Algorithms

The literature review of this research identified relevant methodologies and approaches of using machine learning and statistical techniques for predictive maintenance. The foundation was laid in chapter 2. In this section, an overview of paper 3's contributions are presented, along with the components that enable the integration of the PdM models with the distributed digital twin framework. The key components of the predictive maintenance DT contributed in paper 3 are mainly the following.

1. **ML/DL Algorithms:** The three machine learning algorithms (MLR, XGBoost, LSTM) used to predict components measurements based on the input and output variables from wind turbine SCADA presented in table 2 for this case study.
2. **Condition Monitoring/Anomaly Detection:** The statistical process control technique used for fault detection.
3. **Feedback Mechanism:** The simulation based digital-to-physical twin feedback that enables the predictive maintenance workflow to allow users run simulations of what-if scenarios.

Figure 19 below shows an overview of the predictive maintenance DT workflow based on four algorithms that cover the end-to-end data intelligence based on the developed digital twin and experiment setup described in sections above.

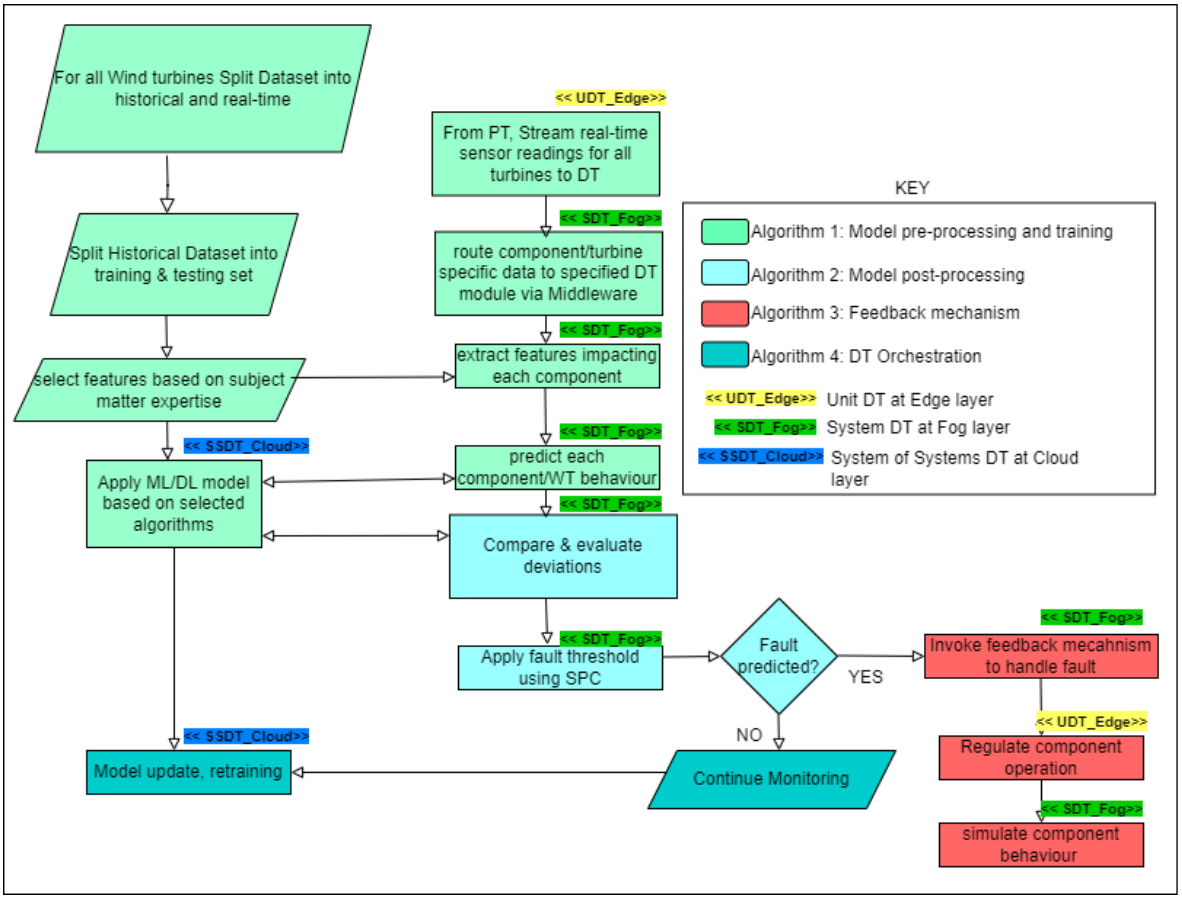


Figure 19: Predictive Maintenance DT workflow.

Algorithm 1: ML/DL Algorithms (Model Pre-processing & Training)

The first algorithm in the predictive maintenance DT workflow deals with the model pre-processing, initial model training and testing. In this stage the data is cleaned and adapted to reflect real-world turbine operation before the ML/DL models are applied.

Algorithm 1: Model Pre-processing and Training

Data: All turbines dataset, sensor data from components
Result: Predicting future trends
Start: Load 75% of each turbine's dataset as ***historical data*** to real-time DB
foreach x_i *captured by sensors* **do**
    if $x_i \in$ *timeout* **then**
    | *Delete*
    end
    if $x_i \in$ *loss data* **then**
    | $x_i = x_{i+1}$ *or* x_{i-1}
    end
    if $x_i < 0$ or x_i *is null* **then**
    | $x_i = 0$
    end
    if *turbine_power* $x_i <= 0$ *and wind_speed* $(w_i) >$ *cut-in-speed (3.5)* **then**
    | $x_i = 0$
    end
end

Based on the selected asset, the following steps are executed at this stage in the workflow of the DT.

- Identify variables of interest for each component and assign them as input and output variables.
- Split historical data into 80% train set and 20% test set.
- Forecast the future trend using MLR, XGBoost, and LSTM (then select best model).

Figure 20 below describes an example result from the XGBoost algorithm comparing the actual vs predicted for the gearbox bearing temperature as the output predicted based on the inputs - nacelle temperature, rotor speed, active power, ambient temperature, and gear oil temperature.

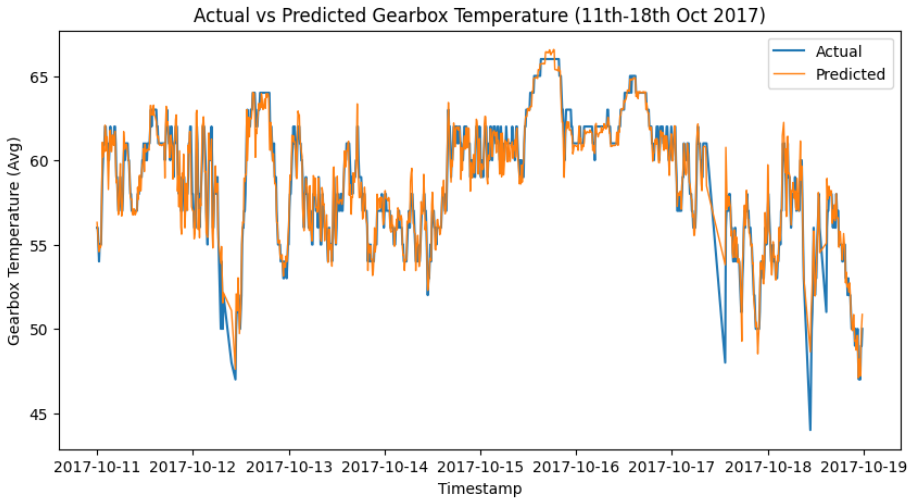


Figure 20: Actual vs Predicted for gearbox bearing temperature.

Based on the predictions, it was necessary to identify the variables (inputs) that have the most weight and therefore have the more impact to the behaviour of the asset's (gearbox) health. This is relevant in identifying components that will be regulated in algorithm 3 for the feedback mechanism feature. To achieve this, a feature importance ranking was used. Figure 21 shows an example of the feature ranking from the predictions shown above.

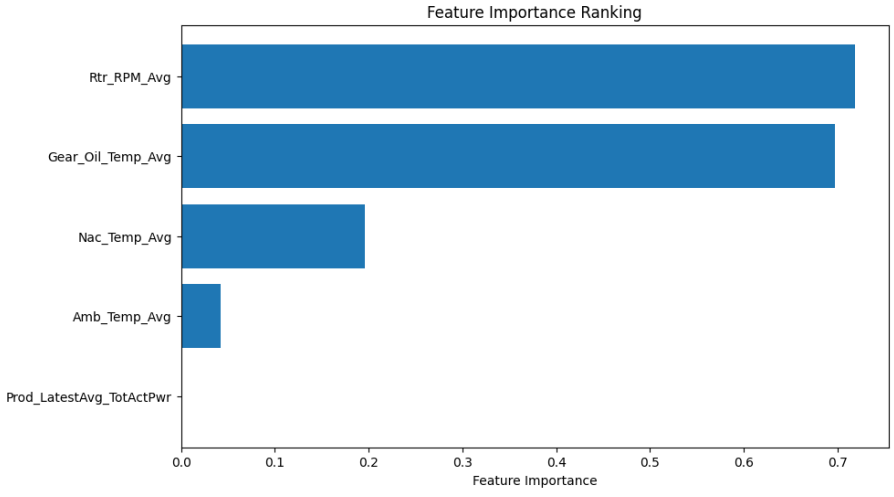


Figure 21: Feature importance ranking for gearbox bearing temperature predictions.

Algorithm 2: Condition Monitoring/Anomaly Detection (Model Post-processing)

Algorithm 2 uses predictions from the trained and tested models in algorithm 1 to apply condition monitoring using the Shewhart process control technique. Deviations above the upper control limit (UCL) indicate an upcoming failure of the specified asset. At this stage, failure records from the EDP dataset are used to validate the accuracy of the model. A gearbox failure was identified up to 3 weeks before it occurred, as detailed in paper 3. Below is Algorithm 2 in pseudocode.

Algorithm 2: Model post-processing

foreach x_i *in* ***test set and dataset*** **do**
    //calculate deviation:
    deviations = actual-predicted
end
$MovingRange = |deviations_i – deviations_{i-1}|$
Sigma = MovingRange/1.128
*UpperControlLimit = 3 * Sigma*
*LowerControlLimit = −3 * Sigma*
foreach *deviation* in *deviations* **do**
    if *deviation > UpperControlLimit* **do**
    | *AnomalyDetected = True*
    | *ActivateFeedbackMechanism()*
    else
    | *AnomalyDetected = False*
    | *ContinueMonitoring()*
End

Using the same gearbox from the EDP wind turbine – T06 as from the example in figures 20 and 21, the anomaly detection for the recorded failure described in table 4 below, is applied. The result is shown in figure 22 with a Shewhart control chart where the spikes above UCL can be seen about a week earlier from the actual failure occurrence. Paper 3 discusses this in more detail.

Table 4: Failure log from EDP dataset

Turbine	Component	Timestamp	Failure Type
T06	GEARBOX	2017-10-17T08:38	Gearbox bearings damaged
	HYDRAULIC_GROUP	2017-08-19T09:47	Oil leakage in hub

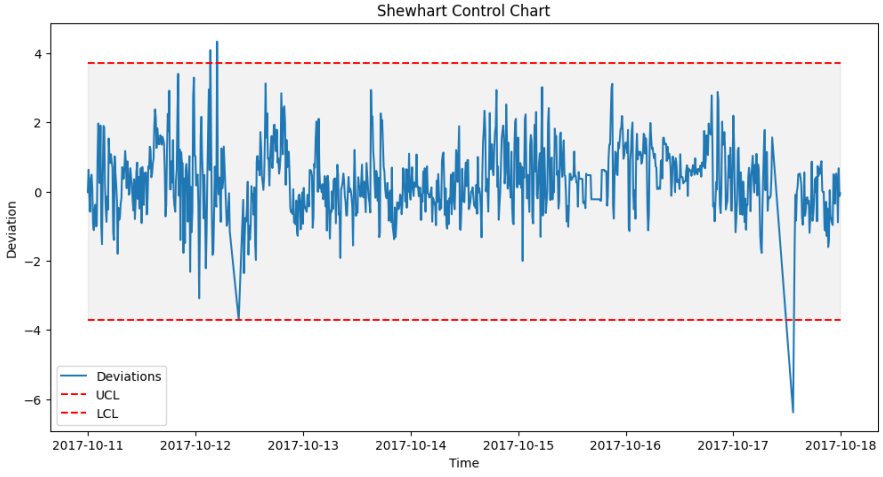


Figure 22: Shewhart control chart for T06 Gearbox failure prediction

Algorithm 3: Feedback Mechanism (Two-way feedback)

Algorithm 3 is the feature of the DT that facilitates the physical to digital to physical two-way feedback to satisfy the requirement of the DT.

Algorithm 3: Feedback Mechanism

```
Function ActivateFeedbackMechanism():
    foreach variable V in sensitive_variables do
        if V > UpperControlLimit do
        | Adjust V using actuators to bring it below UpperControlLimit
        else if V < LowerControlLimit do
        |  Adjust V using actuators to bring it above LowerControlLimit
        end if
    end for
End Function
```

By running variations of the model through reducing certain parameters such as RPM, operators can see results of the possible outcome to that effect. This is what is termed the **what if-scenario**, as introduced in chapter 2. Figure 23 below shows an example of the application of the feedback mechanism. From the point of predicted failure of the gearbox, the RPM is reduced by 30% and models run again as a simulation and the SPC is evaluated.

Algorithm 4 orchestrates the whole DT pipeline.

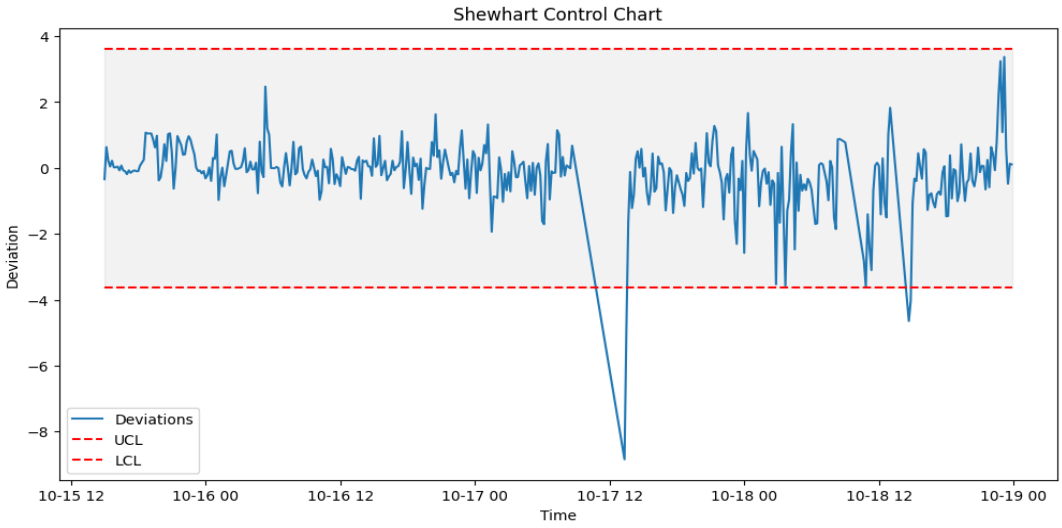


Figure 23: SPC operations using DT feedback mechanism—30% RPM reduction example.

3.3.3 Evaluation and Performance Metrics

At this stage of the experiments, it was essential to evaluate the performance of the predictive maintenance digital twin framework. Similar to the methodology described above with the implementation of algorithms 1, 2 and 3 for the gearbox, paper 3 discusses the same application for the generator component also based on the DT data model and the specific inputs/outputs presented in table 2. The EDP dataset for SCADA readings of the relevant components from 4 wind turbines (T01, T06, T07 and T11) was used to iterate the training and testing of the model for each individual turbine and component. The failure logs were then used to evaluate how well the models can predict the failure. Overall, in paper 3, the focus was on two metrics that

align with the objective of answering RQ2. The metrics used at this stage of the experiments are the model accuracy and the prediction feedback loop (feedback mechanism described in algorithm 3).

Model Accuracy

The use of RMSE, MAE, and R^2 for each ML model to evaluate how well the model is able to closely predict asset behaviour.

1. **Root Mean Squared Error (RMSE):** RMSE was used to measure the average magnitude of prediction errors.
2. **Mean Absolute Error (MAE):** MAE calculates the average absolute differences between predicted and actual values. This provides a more interpretable and linear measure of error.
3. **R-Squared (R^2) Score:** R^2 represents the proportion of variance in the actual data explained by the model, ranging from 0 to 1 (or negative for poor models). A higher R^2 indicates better predictive power. This shows how well the model captures the relationships in turbine data for accurate forecasting.

Table 5 summarizes the model accuracy results for the models – MLR, XGBoost and LSTM, across the dataset used for the experiments in this work. The results have shown that a good prediction accuracy has been achieved. Looking at the R-squared in all cases (both gearbox and generator) which is from the range of 0 to 1 show that all the models performed well with varying levels of accuracy.

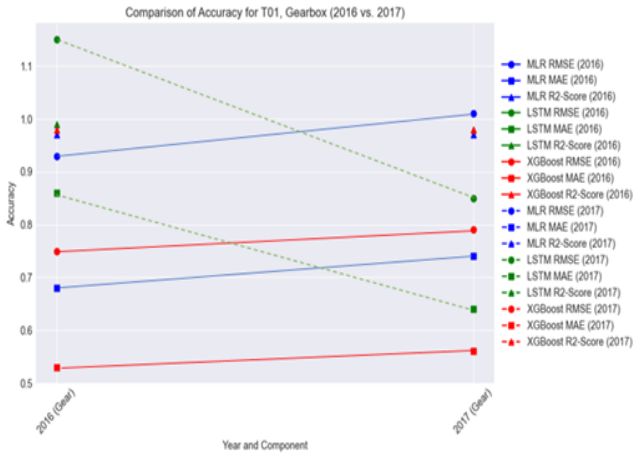
Table 5: Model Accuracy Results.

Turbine	Year	MLR			LSTM			XGBoost		
		RMSE	MAE	R^2-Score	RMSE	MAE	R^2-Score	RMSE	MAE	R^2-Score
T01	2016 (Gear)	0.93	0.68	0.97	1.15	0.86	0.99	0.75	0.53	0.98
	2016 (Gen)	3.91	3.25	0.90	3.86	3.10	0.90	3.98	3.17	0.90
	2017 (Gear)	1.01	0.74	0.97	0.85	0.64	0.98	0.79	0.56	0.98
	2017 (Gen)	4.28	3.17	0.89	4.04	2.94	0.90	3.83	2.77	0.91
T06	2016 (Gear)	0.96	0.72	0.97	1.21	0.91	0.99	0.78	0.56	0.98
	2016 (Gen)	2.99	2.01	0.92	3.19	2.26	0.91	3.36	2.44	0.90
	2017 (Gear)	1.07	0.81	0.97	0.84	0.61	0.98	0.86	0.61	0.98
	2017 (Gen)	3.63	2.26	0.89	3.89	2.38	0.85	3.63	2.2	0.87
T07	2016 (Gear)	1.04	0.74	0.96	1.15	0.86	0.99	0.81	0.57	0.98
	2016 (Gen)	2.55	2.04	0.95	2.49	1.96	0.95	2.60	2.05	0.95
	2017 (Gear)	1.03	0.76	0.96	0.86	0.65	0.97	0.79	0.56	0.98
	2017 (Gen)	3.59	3.11	0.92	4.87	4.42	0.80	3.83	2.77	0.91
T11	2016 (Gear)	1.16	0.87	0.96	1.12	0.80	0.99	0.86	0.62	0.98
	2016 (Gen)	3.30	2.61	0.89	3.45	2.68	0.88	3.31	2.62	0.89
	2017 (Gear)	1.22	0.92	0.96	1.02	0.76	0.97	0.87	0.64	0.98
	2017 (Gen)	3.75	2.95	0.90	3.43	2.82	0.91	3.46	2.80	0.90

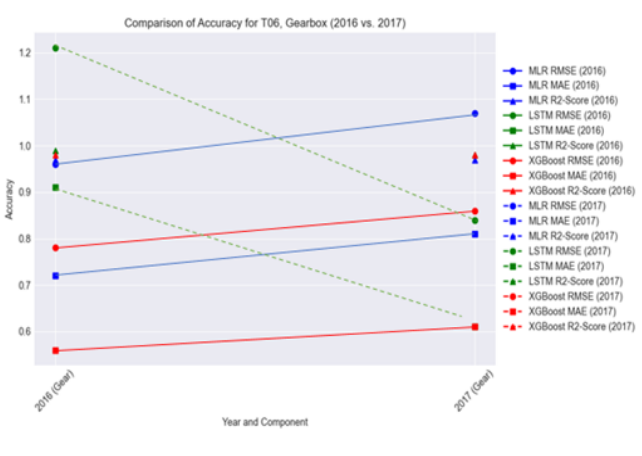
Further Evaluation – Answering RQ2

A further evaluation into the performance of the models was done in paper 3. In this evaluation, an insight found was from analysing the performance of turbines T01, T06, T07, and T11 across

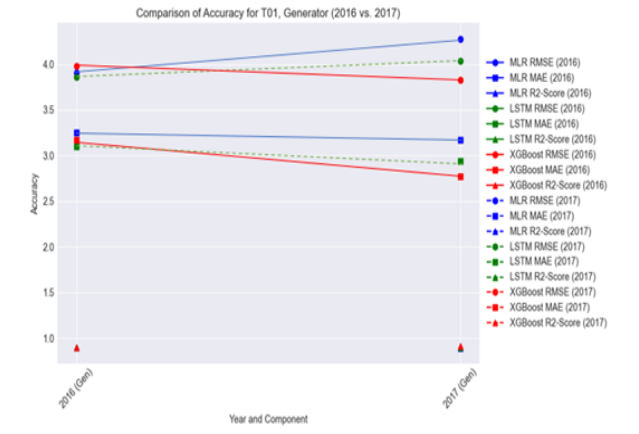
models and years. The study found that the gearbox is more affected by seasonal variations than the generator. This observation stems from the gearbox's proximity to input mechanical energy, due to its role in wind turbine operations. Accuracy metrics like RMSE and MSE confirm model reliability while R^2 values further validate model performance, with higher values indicating greater predictive accuracy. Therefore, the findings underline the significance of leveraging diverse algorithms for specific component behaviour, aligning with the need for a distributed DT framework as posited in H-4. As such, by accommodating experimentation and fine-tuning for Prognostics and Health Monitoring (PHM), the framework offers scalability and adaptability. Moreover, these results support H-5 by demonstrating how adhering to industry standards can enhance DT implementation, optimizing the process and performance of predictive analytics. This aligns with the modular architecture specifying what instances of the DT runs in what layer as presented in the predictive maintenance DT workflow (figure 19). Figures 24a – 24d show the variations in accuracy levels across year and component.



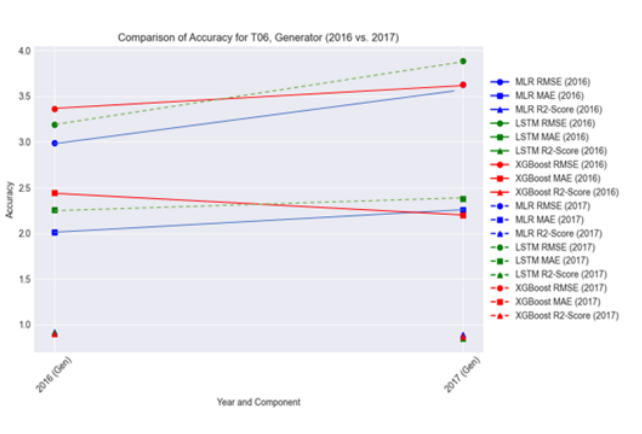
(**a**) T01—Gearbox



(**b**) T06—Gearbox



(**c**) T01—Generator



(**d**) T06—Generator

Figure 24: T01 and T06—2016 vs. 2017 accuracy comparison by component.

Prediction Feedback Loop

The feedback mechanism discussed and implemented in paper 3, as well as in the methodology section 3.3.2 (algorithm 3) showcases how this metric is satisfied as a tool that enables PHM of wind turbines based on the distributed DT framework. The fog layer plays a critical role in providing the compute resource for this.

3.4 CHAPTER SUMMARY: CONTRIBUTIONS ADDRESSING RQ2

In this chapter, paper 3's contribution was discussed as a methodology for the design, development, and implementation of a distributed digital twin framework for predictive maintenance. The chapter addressed RQ2 by emphasizing the importance of standardization and a scalable architecture for real-time analytics. The adoption of ISO23247 was discussed because it makes sure that the framework follows industry standards and gives digital twin development a structured and modular way to work. The main idea behind the DT implementation is that it uses a distributed edge-fog-cloud architecture to make monitoring, data management, and low-latency processing for predictive analytics possible in in a timely manner. The integration of machine learning models for predictive maintenance was also discussed. The ML models implemented include MLR, XGBoost, and LSTM, which allow the DT framework to predict faults and optimize operations using a feedback mechanism. The models were validated through high accuracy metrics from the training and testing with the EDP wind turbine dataset. Statistical process control (SPC) is used to find outliers, which makes it easier to find faults and provides a feedback loop for simulations that use the fog layer's computing power. The feedback mechanism allows simulation-based decision-making, which can help transform the digital twin into an intelligent system capable of operational adjustments and prognostics at the right time. Controlled experiments based on ECSE principles validated the methodology. The implemented architecture shows the framework's potential for scalability. This contribution highlights how it meets the requirements of RQ2 to address hypotheses H4 and H5. Chapter 3 discussed accuracy and feedback mechanisms as metrics used to evaluate the framework. In summary, table 6.1 is extended from table 3 to specifically show the problems addressed in chapter 3, using the evaluation metrics. Chapter 4 (papers 4 and 5) explores and evaluates additional metrics like latency, scalability, cost, and security. This study builds on the contributions of chapter 3 (paper 3) to establish an effective DT solution for predictive maintenance in IIoT systems.

Table 6.1: Problems covered with evaluation metrics (chapter 3).

Problem Criteria	**Metric**	**Justification**	**Paper**
Decision Support	• Model Accuracy • Fault Prediction and Feedback Mechanism	Enabling asset fault prediction and feedback mechanism for optimal decision-making	Paper 3
Standardization	• Use of ISO23247	Ensuring standardization across enabling technologies implementation/architecture	Paper 3

CHAPTER 4: FRAMEWORK VALIDATION

The previous chapter, chapter 3, discussed the methodology with details on the implementation of the predictive maintenance digital twin from paper 3's contribution. The metrics evaluated were the accuracy of the predictive maintenance ML algorithms and the fault prediction of gearbox and generator components using SPC. Furthermore, the implementation of the feedback mechanism concept with a predictive maintenance DT workflow involving three algorithms was discussed. The predictive maintenance DT workflow is the basis of the next stage of the study. This chapter talks about how the contributions of papers 4 and 5 validate the distributed DT framework by using additional metrics that address some of the research gaps mentioned in chapter 2. Paper 4 uses the ECSE methodology in applying the distributed DT framework to evaluate two metrics—computational cost (CPU utilization) and computational latency (processing times)—in three architectural scenarios (edge only, edge-cloud, and edge-fog-cloud). Analysing the performance of these three scenarios uncovered insights used to refine the framework. In paper 5, extra measurements are evaluated to confirm the effectiveness of the distributed DT framework. The experiment setup was changed to better reflect a real cloud-based data intelligence platform. This redesign shows the placement of fog nodes within a SCADA network. It also assesses how well this setup performs using response time, scalability, security, and operational costs as measures. The remainder of this chapter discusses the contributions of papers 4 and 5 towards answering research question 3 (RQ3) of this thesis.

4.1 INTRODUCTION

The rapid growth of the Industrial Internet of Things (IIoT) and big data has driven the need for innovative solutions in asset management of manufacturing systems. Novel approaches to implementing data intelligence solutions such as predictive maintenance digital twins [76], [96] offer a promising approach to improving asset management strategies. In earlier chapters, this study reviewed this concept and found gaps and challenges with regards to the combination of enabling technologies such as cloud computing, IIoT, and machine learning in implementing digital twin solutions. This study further developed a predictive maintenance digital twin solution that enables a prediction feedback loop for supporting PHM simulations. However, the developed framework requires further validation through experiments designed to evaluate metrics pertaining to the efficiency of certain aspects of the framework. As highlighted in table 3 of the literature review section, the key challenges relevant to this study are the twinning rate, decision support, cost and scalability, standardization, and security considerations in DT implementation. The implementation of the DT in chapter 3 discussed results that cover decision support with a framework standardization approach. Building on the developed DT, further work investigated the remaining challenges based on the metrics relevant to assessing the effectiveness of the framework. Based on this requirement, the contributions of papers 4 and 5 provide a validation of the distributed DT framework within the case study of enabling a cost-effective PHM for wind turbines. As a continuation of table 6.1, table 6.2 shows the challenges and metrics along with a justification based on the literature review (table 3 in chapter 2) and the paper that goes into more detail about the contribution.

Table 6.2: Problems covered with evaluation metrics (chapter 4).

Problem Criteria	Metric	Justification	Paper
Computational cost & Scalability	• Computational cost (CPU utilization) • computational latency (processing times)	evaluated towards validating infrastructure and operational costs to support scalability	Paper 4
Twinning Rate	• Response time (end-to-end data transfer latency)	Improving the synchronization time (based on requirement; milliseconds, seconds or minutes) between the physical and digital twin instances.	Paper 5
Security & Privacy	• Security and data privacy mechanisms	The impact of applying additional security and data privacy mechanisms to the metrics above	Paper 5

Paper 4 further re-enforces standardization by refining the application of the ISO23247 based on results, and paper 5 showcases the behaviour of the framework when the data set is scaled to a small to medium sized wind farm with 16 turbines.

The remainder of this chapter is divided into three parts. Section 4.2 covers the contributions of paper 4, section 4.3 covers the contributions of paper 5, and section 4.4 summarizes the chapter linking the contributions to answering research question 3 (RQ3).

4.2 FRAMEWORK VALIDATION: COMPUTATIONAL PERFORMANCE

The work in paper 4 focused on evaluating the computational performance of the distributed DT framework implemented in chapter 3 (paper 3). The key metrics evaluated are the computational cost (CPU utilization) and computational latency (processing times) of running the experiments based on three architectural scenarios discussed in the next sub-section, section 4.2.1. The contributions of paper 4 specifically tests Hypothesis 6 (H-6) relating to research question 3 (RQ3). Therefore, paper 4 asserts the following questions and presents a methodology used to answer them. “

- *How can a distributed digital twin framework improve computational efficiency and reduce infrastructure costs for a wind turbine predictive maintenance application?*
- *What are the enabling technologies and design principles to realize such an architecture that can improve the overall solution, from edge to cloud?”*

To answer the above questions, the methodology adopted is an ECSE approach, by running experiments using the predictive maintenance workflow described in Figure 19. However, the experiments focused on evaluating the computational cost (CPU utilization) and computational latency (processing times) in running all the three models – MLR, XGBoost and LSTM based

on an actual run and a feedback run. Below is a description of the actual and feedback runs in the context of the predictive maintenance DT workflow (Figure 19).

- **Actual Run:** The training and testing of all three models with historical data to detect failures using SPC. This is run every 24 hours or end of day.
- **Feedback Run:** The retraining and testing of simulation-based data slices, driven by faults identified from actual runs. This is used for simulating what-if-scenarios to handle certain faults.

The methodology used is depicted in figure 25 below, where the three layers are tested for hosting the DT workload by interchanging the arrangements across three scenarios described in table 7, below.

Table 7: Scenarios for computational performance experiments

Scenario	Architecture	Description
A	Edge only	All DT functionalities deployed on raspberry pi device and edge PC support (local server for data storage).
B	Edge-cloud	Global (system of systems) DT functionalities placed in cloud, while system and unit functionalities on edge device and edge PC.
C	Edge-fog-cloud	DT functionalities split into a three-layered architecture. (distributed)

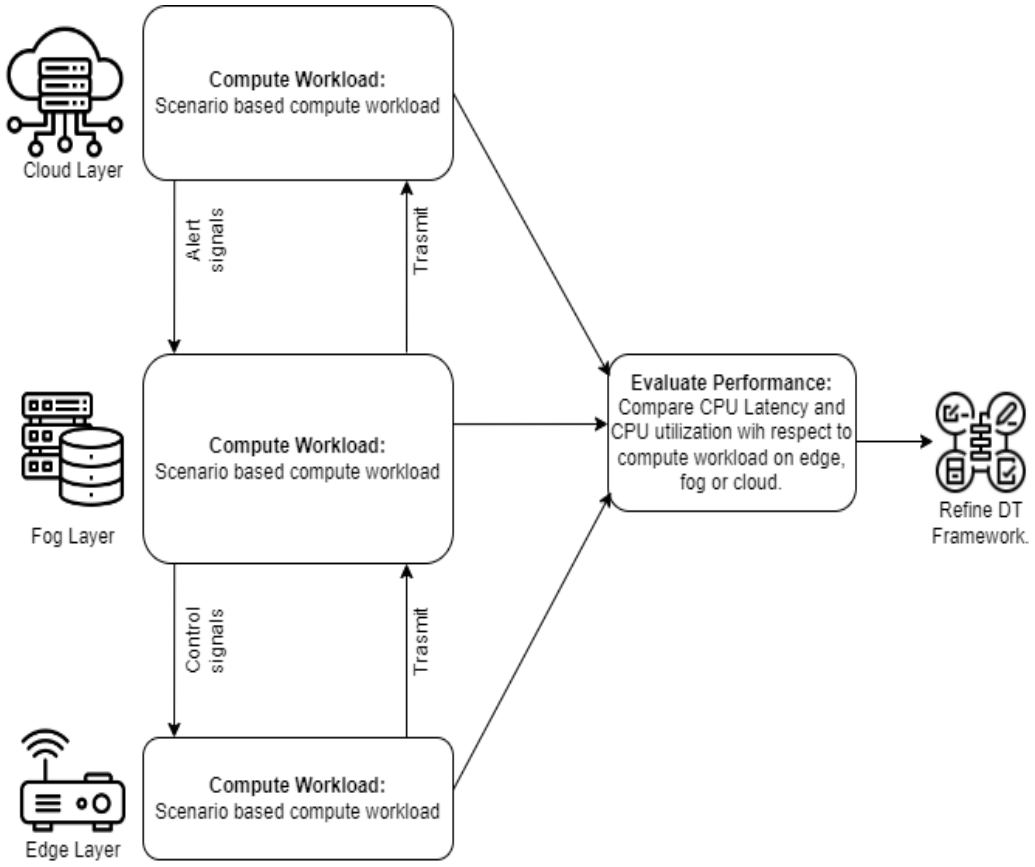


Figure 25: Workflow for methodology/experiments

The following metrics are the performance metrics recorded for each scenario.

- **CPU utilization (%)**: serves as resource usage efficiency measure for computational cost.
- **Computational Latency (Time (s)):** serves as processing time.

4.2.1 Scenarios and Results for Validation

Scenario A – Edge only deployment

In scenario A, all DT workload is placed in the edge layer which is the raspberry-pi device and edge PC acting as a server. Figure 26 shows the architecture of this set up.

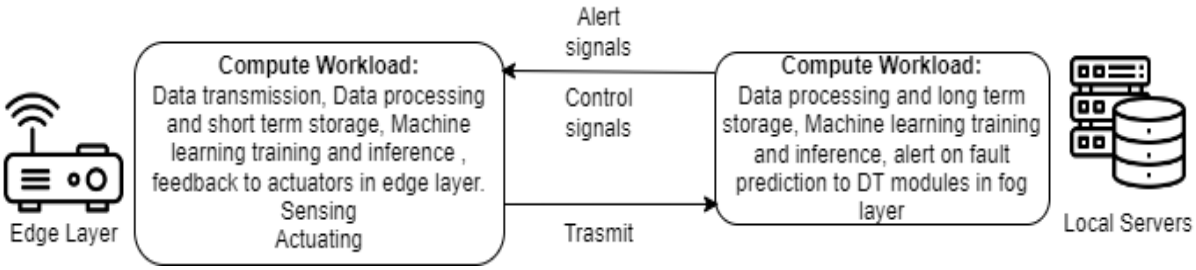


Figure 26: Architecture for Scenario A.

In this architecture, the compute workload assigned to the raspberry pi and the edge PC with 4 CPU cores (local server) is to fully accommodate the predictive maintenance workflow. This is in line with the algorithms and workflow implemented as shown in figure 19. As discussed in more detail in paper 4, the edge device (raspberry pi) with lower computational capacity performed poorly due to resources constraint when considered for the ML workload. However, the following results shows the results in instances where the edge PC handled the ML workload.

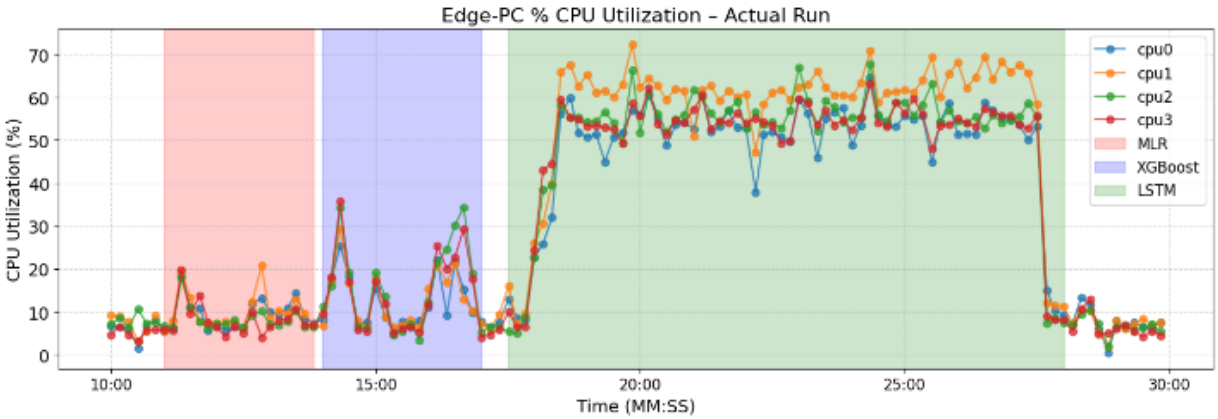


Figure 27a: Edge PC % CPU Utilization – Actual run

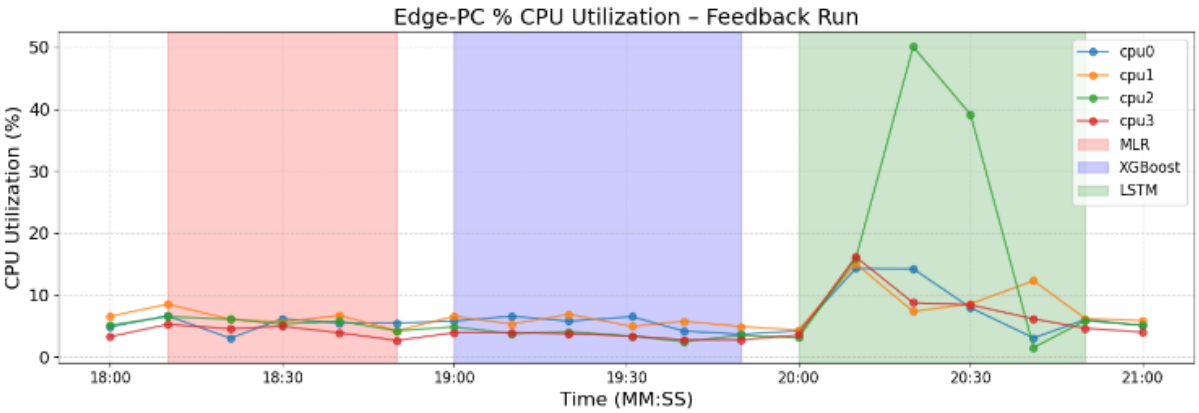


Figure 27b: Edge PC % CPU Utilization – Feedback run

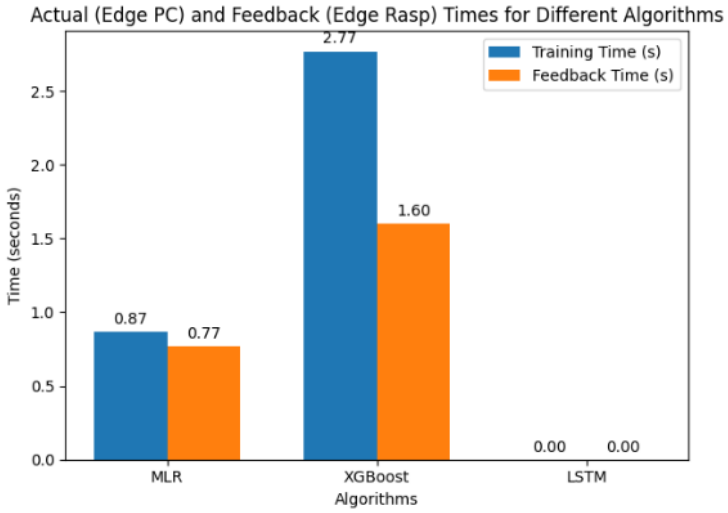


Figure 28: Edge-PC Actual run vs Edge-Device feedback run CPU Latency

What can be observed from the CPU utilization comparison (figures 27an and 27b) is that in terms of the actual run, the edge PC struggles as almost 70% of its CPU is occupied with the workload. Additionally, a point to note is that the LSTM algorithm being a deep learning algorithm could not be run successfully to provide results, by either of the hardware (edge pc or edge device) both in the actual and feedback run instances. In figure 28, the time it takes to run feedback on the edge device for the two algorithms that could run (MLR and XGBoost) is compared with an actual run on the edge PC.

Scenario B: Edge-cloud deployment

Scenario B is considered as the typical centralized cloud architecture where all ML DT workload for predictive maintenance from figure 19, are deployed in the cloud while light weight transmission is done at the edge layer. This is depicted in figure 29.

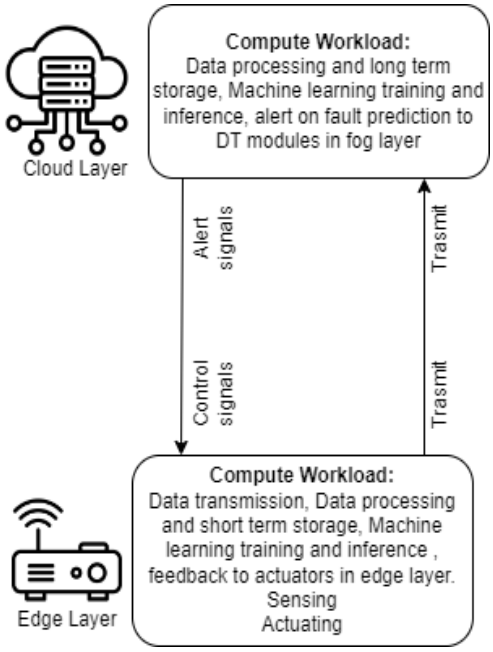


Figure 29: Architecture for Scenario B.

In the experiment setup for scenario B, a high-capacity computer with Graphics Processing Unit (GPU) was used as the cloud infrastructure. In this experiment, the communication latency (end-to-end response time) was not evaluated. The response time is covered in the next chapter. Figure 30 shows the performance behaviour for the CPU utilization metric of running the actual run in the cloud computer with 9 CPU cores. A feedback run was not run in the cloud as the distributed DT framework does not provide for this requirement.

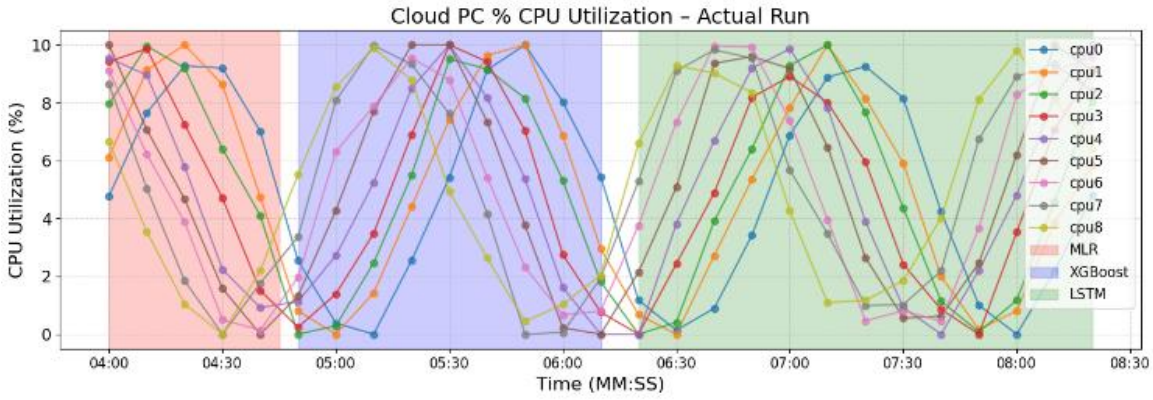


Figure 30: Cloud system % CPU Utilization – Actual run

It can be observed from figure 30 that in all cases, not up to 10% of the CPU is occupied by any of the models. Making it convenient to run all models simultaneously. Additionally, the cyclic and phased behaviour suggests how the system is not handling overlapping tasks but instead focuses resources on a single model at a time.

For contrast, figure 31 below shows the difference between the feedback run on edge device versus the actual run in the cloud computer. It can be observed that for MLR and XGBoost, the cloud was still faster in handling actual run than the edge device's feedback run. While the edge device cannot run the LSTM, the cloud system took 2.4 seconds to run.

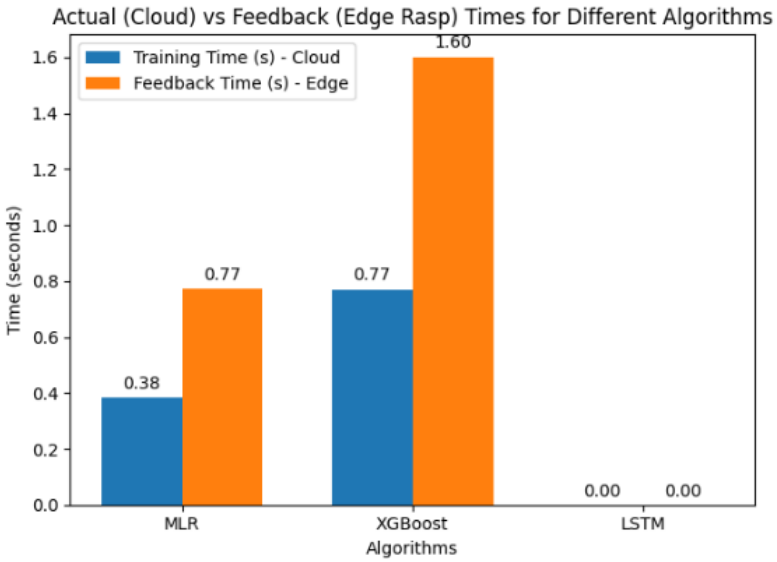


Figure 31: Cloud-Actual vs Edge-Feedback system CPU Latency

Scenario C: Edge-fog-cloud deployment (The Distributed DT)

Scenario C specifically evaluates the performance of the proposal in this PhD study, for a distributed digital twin development using fog computing. The predictive maintenance DT workflow (figure 19) was refined further based on the results of this experiment. Figure 32 describes the architecture for scenario C in which the fog layer sits in between the edge devices and the cloud layer to provide a balance of compute resources. Additionally, a microservices approach is used to complement the performance of the architecture using containerization which helps to further achieve modularization within the fog infrastructure.

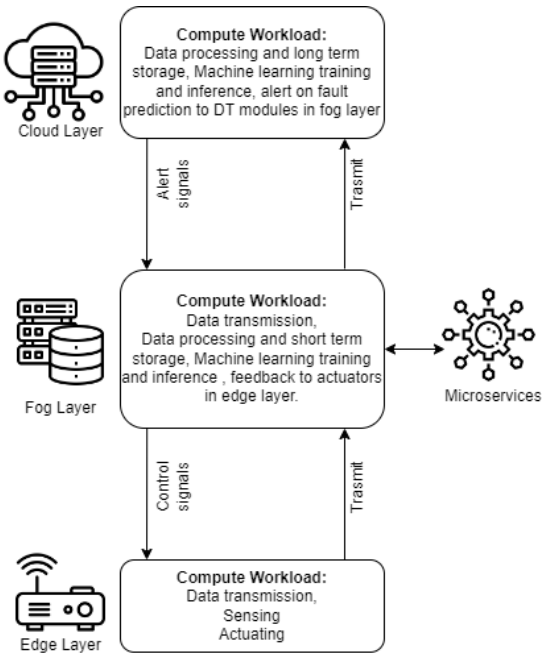


Figure 32: Architecture for Scenario C.

In the feedback run, the fog PC which has 10 CPU cores behaved similarly to the feedback run on the edge PC described in figure 27b. However, the fog PC was able to run LSTM with CPU

utilization of less than 50% and less than 15% for MLR and XGBoost in the feedback run. For the actual run, the fog PC – with no containerization, also performed well but the resource distribution was not optimized as depicted in the results from figure 33 below.

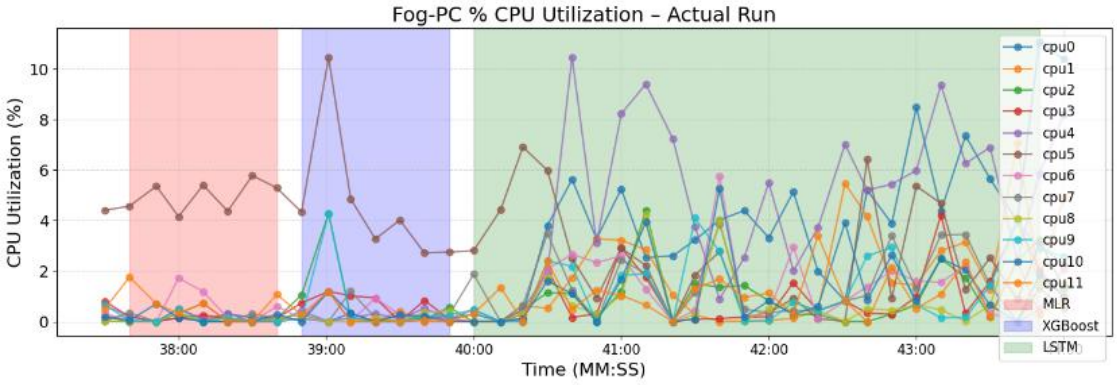


Figure 33: Fog-PC % CPU Utilization – Actual run

A refinement of the deployment involved using Docker [125] to containerize the deployment of the PdM DT workflow at the fog layer. This helped in achieving better utilization of the fog infrastructure, by managing the resources across all the CPU cores better. Figure 34a shows the actual run with containerization for the actual run while figure 34b shows the feedback run with containerization. In both cases, the CPU utilization was more streamlined, allowing the simulation of the feedback to run and provide results efficiently.

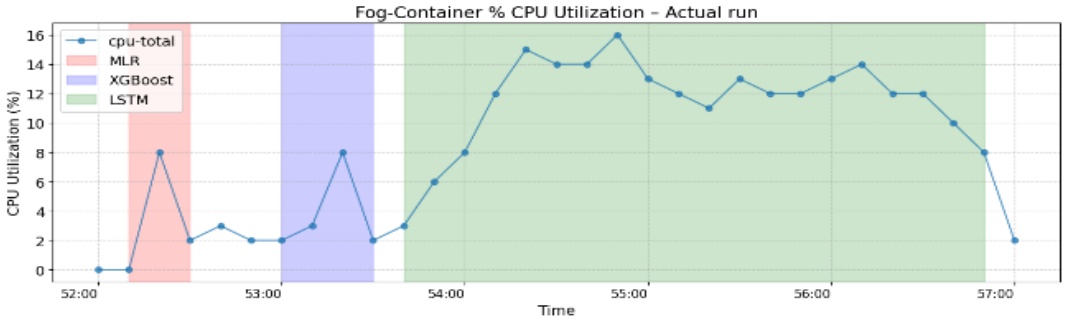


Figure 34a: Fog-Container % CPU Utilization – Actual run

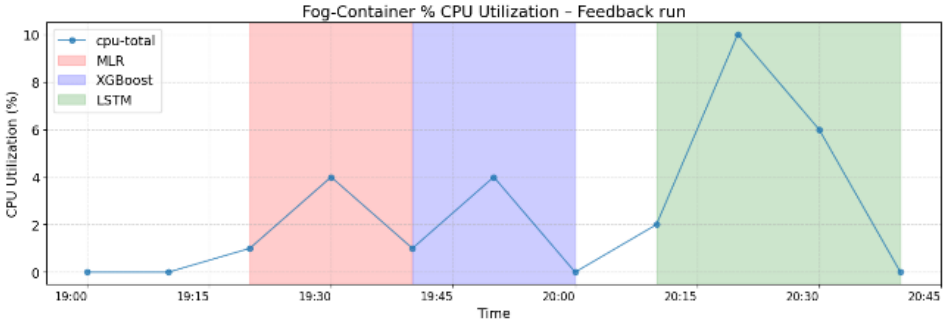


Figure 34b: Fog-Container % CPU Utilization – Actual run

In terms of CPU latency, it can be observed from figure 35, that the difference between actual and feedback run was negligible when running MLR and XGBoost compared to when running LSTM. Regardless of actual run, feedback run, or the model selected, the fog layer shows its ability to manage the workloads to satisfy the requirements of the PdM DT workflow.

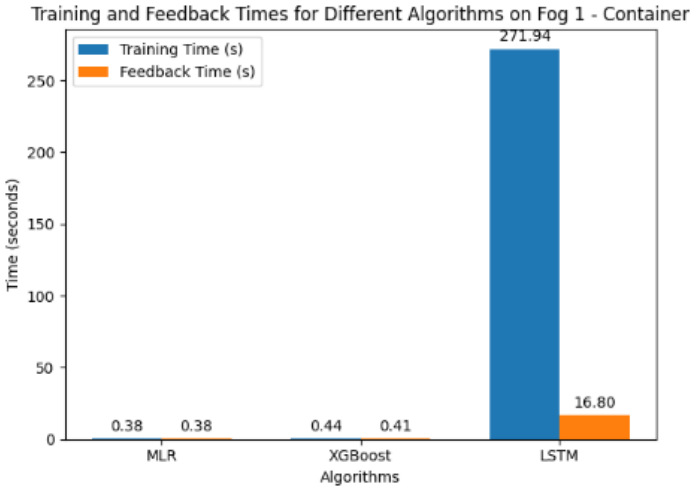


Figure 35: Fog-Container CPU Latency – Actual vs feedback

4.2.2 Framework Standardization with ISO23247 – Answering RQ2

The results from the contributions of paper 4 highlighted in section 4.2 above, helped in the refinement of the distributed digital twin framework by providing more clarity on the alignment of the layers of the architecture with segments of the ISO23247 framework. As introduced in chapter 3 (section 3.2, figure 15), this work (in paper 3) initially proposed the ISO23247 reference architecture mapping to the layers associated with unit, system, and system of systems DT. As the validation experiments described in section 4.2.1 were achieved, a further refinement based on the overall computational performance benefits of the fog layer provided more context which helped in refining the mapping with specific adjustments. This refinement is shown in figure 36 where the compute platform is highlighted, showing the actual runs are to run in the cloud while the feedback runs are to run in the fog layer. This insight helps answer hypothesis 6 of research question 3 by showing the specific compute platform placement pattern that performs better in implementing a predictive maintenance digital twin.

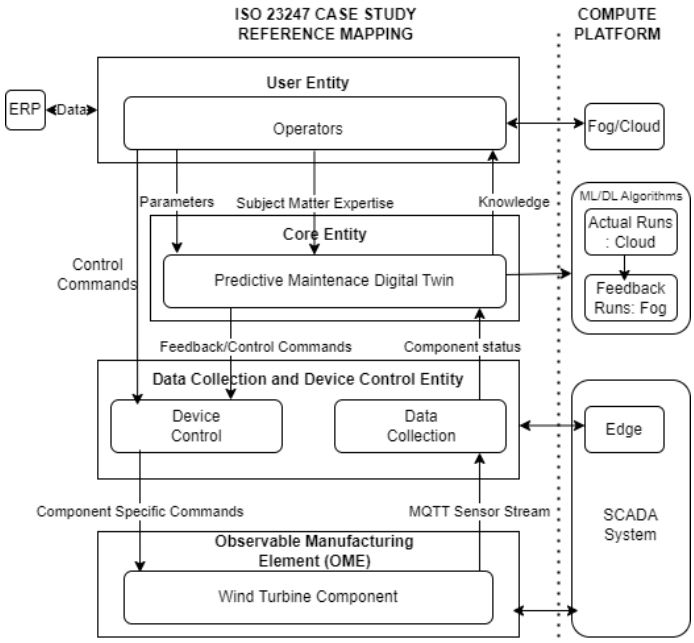


Figure 36: Framework refinement based on experiment results.

4.3 FRAMEWORK VALIDATION: INTEGRATION WITH A CLOUD-BASED DATA INTELLIGENCE PLATFORM

Once the computational performance metrics validation phase was achieved, it was necessary to further validate the framework in a context similar to a real-world deployment. Companies in the manufacturing industry often adopt cloud infrastructure to fully harness the benefits of IIoT technologies [126], [127]. To validate based on this concept, paper 5 explored the architectural consideration of integrating the framework with a real world, cloud-based data intelligence platform called Databricks [128]. Furthermore, paper 5 focused on redesigning an architecture that integrates the distributed DT with a typical SCADA network architecture. This helped in clarifying where fog nodes can be placed, with experiments for evaluation of scalability and metrics outlined in table 6.2 towards validation. The key metrics evaluated are the following.

- Response time (end-to-end data transfer latency between physical and digital twin).
- The impact of fog-based data security mechanisms to the response time metric.

4.3.1 Integration Architecture

The integration architecture introduced in paper 5 is an extension of the distributed DT architecture introduced in chapter 2 and implemented in chapter 3. Firstly, a review of the state of the art in SCADA architecture for wind turbines was conducted. Studies in [129], [130], [131] showcase the design consideration for an Information and Communications Technology (ICT) network that enables the implementation of SCADA for wind turbines. This was used to identify the layers of the SCADA network, and a mapping was then achieved on the extension of this ICT network to integrate the distributed DT framework. Figure 37 below shows this integration architecture.

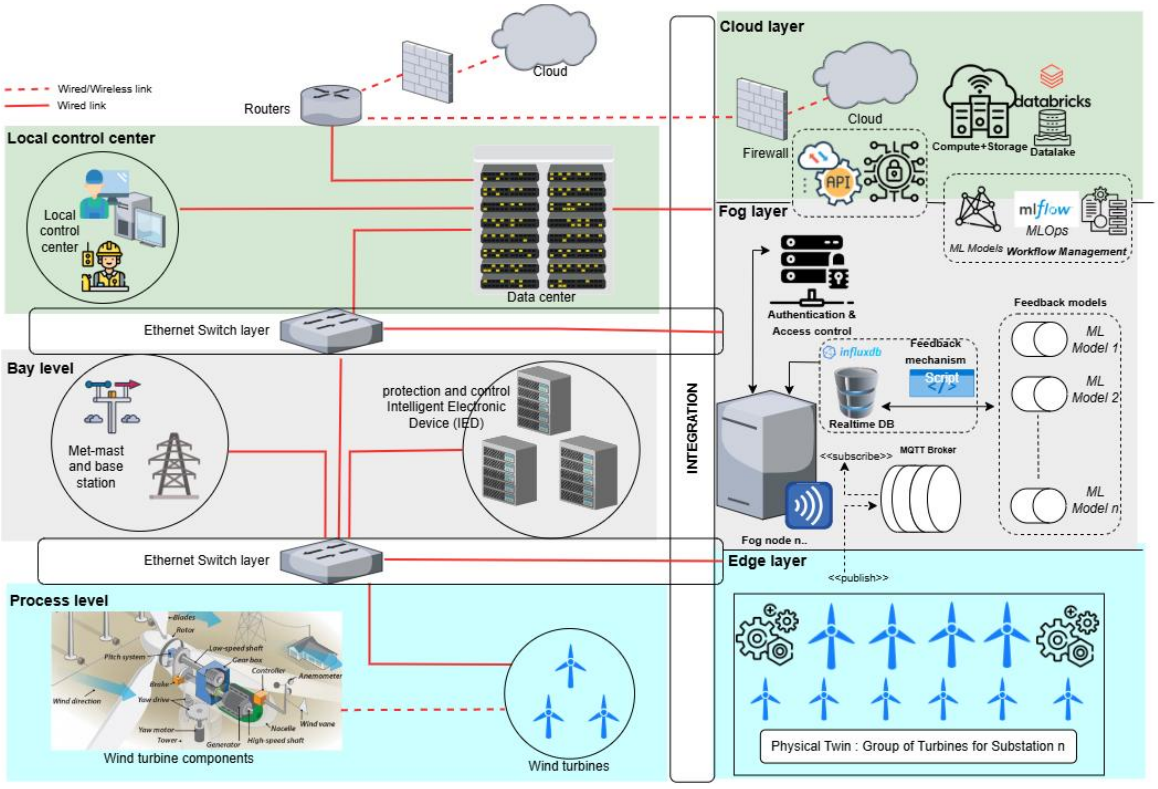


Figure 37: Wind Turbines SCADA and Distributed DT Integration Architecture

The architecture in figure 37 is not an explicit arrangement of the solution, rather a template highlighting the potential of the framework's integration. This clarification is necessary to review the following points that highlight the integration of the process, bay and control centre levels described in [130].

- **Process Level and Edge Layer:** The process layer comprises of the primary equipment involved in power generation. This includes sensors and actuators for timely data collection and control connected via wired and wireless networks. This process level aligns with the edge layer as the physical twin that emits sensor data and receive control instructions are in this layer.
- **Bay Level and Fog Layer:** The bay level manages outputs of a single or a group of turbines. Components of the SCADA infrastructure such as the switchgear, transformers and local controllers help in regulating electrical output. This level is connected to the local control centre using high-speed data links such as fibre optic and industrial ethernet. This aligns with the fog layer in the DT framework as it sits in the middle between the data collection systems and local control centre. Based on the proposed integration architecture, dedicated fog servers host the data streaming/storage, feedback models for running simulations and security mechanisms. This is connected to the cloud ML operations (MLOps) platform.
- **Control Centre and Cloud Layer:** The local control centre is the heart of the wind farm control. In this level, central servers, databases, and Human-Machine Interfaces (HMIs) enable monitoring, grid coordination and remote-control capabilities. This level is connected to external systems such as the national grid and weather forecasting systems via private Wide Area Networks (WANs). The control centre, aggregates data across the whole wind farm and can be considered as the system of systems level, where cloud services integration can be enabled. In the proposed distributed DT framework and experiments in paper 5, a connection of the edge and fog to the cloud is made via this layer. This can be directly from edge to cloud (centralized) or using the fog-based architecture (distributed).

The architecture in figure 37 describes the level at which the fog layer sits. However, this can be within the wind farm close to wind turbines or at the local control centre based on a group of turbines. For instance, an energy services company, Natural Power, designed and implemented a bat avoidance system integrated with the SCADA system using a hub that sits beneath individual turbines. Natural Power's system facilitates automated controls as an independent system outside of the wind turbine [132], [133]. This technology highlights the feasibility of the fog-based architecture in terms of integration. Figure 38 below, describes a fog placement concept introduced in paper 5 to depict the strategy for fog utilization in managing both onshore and offshore wind farms.

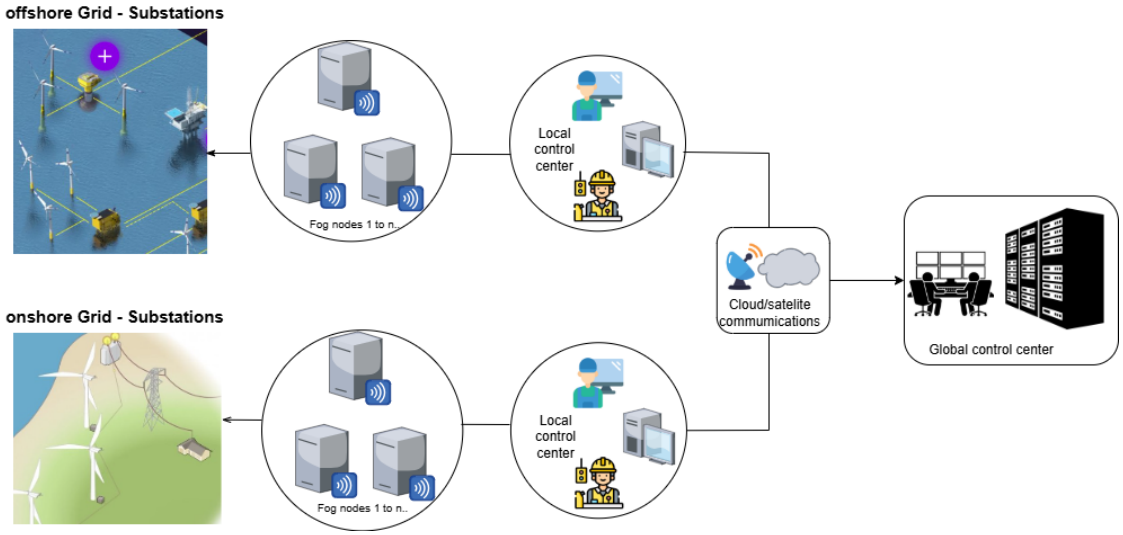


Figure 38: Fog node placement strategy

4.3.2 Fog Enabled Security Mechanism

As highlighted in the overview of problems covered with evaluation metrics in table 6.2, paper 5 introduced a fog-based security mechanism for additional layers of securing the integration of fog nodes as they handle data and control within the SCADA network. This methodology is extended from work by researchers that show the efficacy of utilizing fog nodes to complement security in IoT environments [112]. Figure 39 shows the security by design of this approach based on a predictive maintenance DT case study.

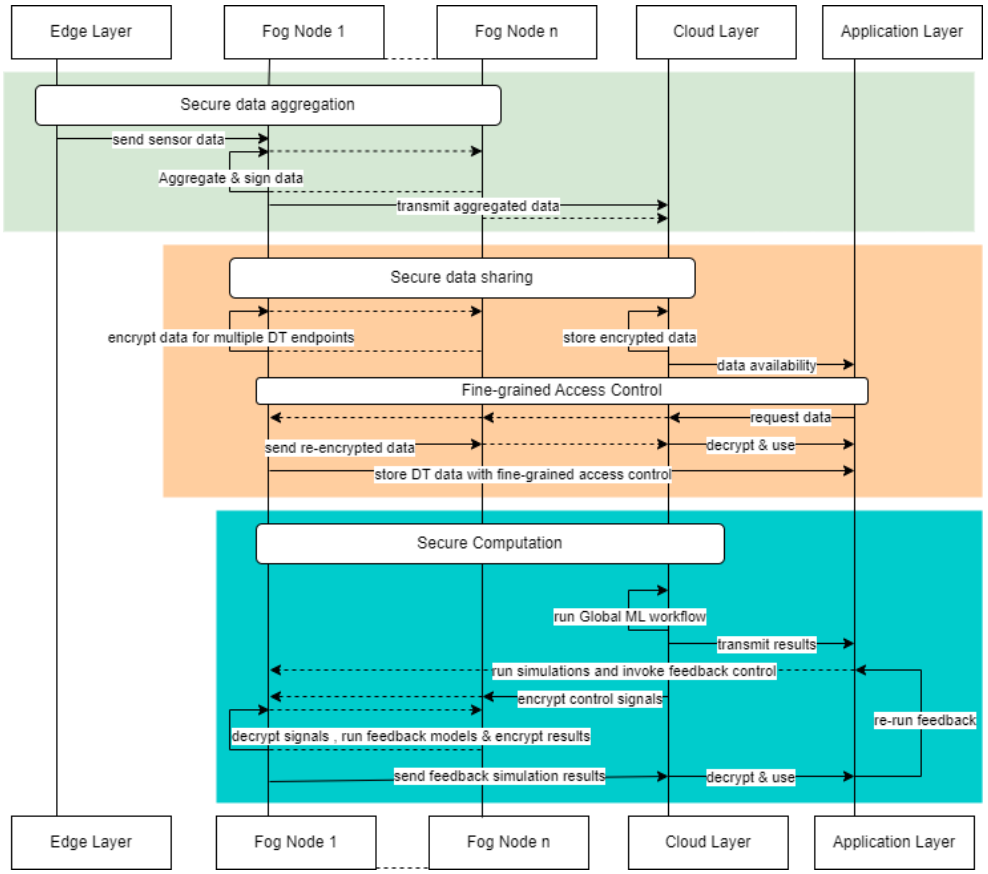


Figure 39: Fog-enabled security mechanism.

This fog-enabled security framework, once integrated within the SCADA, reduces the attack surface of introducing fog nodes, but ensures the secure integration of fog nodes in a predictive maintenance DT solution. The framework is divided into the following four main processes spanning multiple layers (edge, fog, cloud, and application layers).

1. **Secure Data Aggregation**
    - Sensor data is collected from wind turbines at the edge layer.
    - The fog nodes aggregate and sign the collected data.
    - The aggregated data is securely transmitted to the cloud for further processing.
2. **Secure Data Sharing & Fine-Grained Access Control**
    - The aggregated data is encrypted for multiple Digital Twin (DT) endpoints.
    - Access control mechanisms ensure fine-grained access, allowing authorized entities to request, decrypt, and use data securely. Data is re-encrypted before transmission to protect sensitive wind turbine operational data.
3. **Secure Computation for Predictive Maintenance**
    - The predictive maintenance DT workflow runs in the cloud and fog.
    - Fog nodes execute simulations invoking feedback control mechanisms. Control signals are encrypted before transmission to ensure security.
    - The system decrypts control signals, runs feedback models, and encrypts results. The feedback results are sent back securely for fine-tuning.
4. **Application Layer & Feedback Loop**
    - The application layer decrypts results and re-runs feedback models to continuously improve performance.
    - Secure prediction feedback loops ensure that predictive insights optimize wind turbine operations securely.

4.3.3 Validation Experiments

Once the security mechanism as described in the section above was implemented in the predictive maintenance DT workflow, the validation experiment was setup and executed to evaluate the behaviour of the framework with a connectivity to the cloud-based data intelligence platform (Databricks). Firstly, a data scaling strategy was done to increase the size of the dataset. The response time was further evaluated based on this scaling.

Dataset Scaling

In paper 5, the scaling strategy for the cloud integration experiments was discussed in detail. The work further utilized the same EDP [74] SCADA data from four operational wind turbines with failure logs, over two years. To test scalability, the data was synthesized using a scaling factor that expanded the simulation to a wind farm of 16 turbines. This scaling is equivalent to the Ray Wind Farm [134] which powers 30,000 homes annually with 54,400 kW capacity. This scaling helped in further assessing how effective the framework remains in real-world applications.

Table 8: Dataset Scaling

Items	Experiment Cases		
	Case A	**Case B**	**Case C**
Wind Turbines	1	4	16
SCADA Components	1 - Generator 1 - Gearbox	4 – Generator 4 - Gearbox	16 – Generator 16 - Gearbox
Features (Sub-components)	14	56	224
Fog Node(s)	1. 1 node serves a 1 turbine.	2. 1 node serves 2 turbines	8. 1 node serves 2 turbines

Metric Evaluation: Response Time.

The response time metric is selected to evaluate the framework based on the twinning rate research gap highlighted in the effectiveness of DT solutions [40], [76]. For the response time metric, this experiment phase considered real-time as 1-second interval for sensor readings and milliseconds to measure communication latency. The response time serves as the duration from when data is emitted by a SCADA sensor to when it is received for feedback processing at the cloud or fog node. A comparison was made for the difference between sending data directly from edge to the cloud versus from edge to the fog in the context of processing the feedback simulation runs. Figure 40 shows the results.

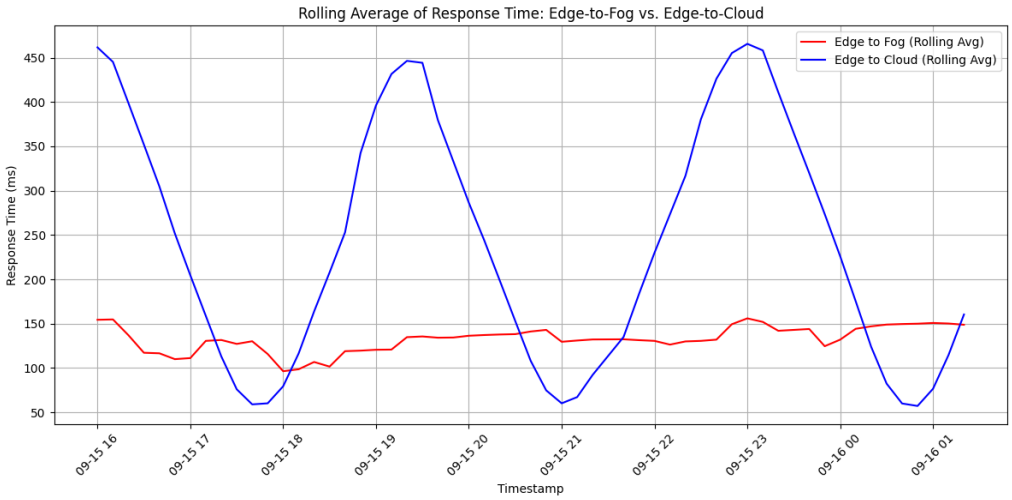


Figure 40: Response Time – Fog vs Non-Fog Setup

The result from figure 40 above, forms a key part of the contributions of paper 5. To satisfy the twinning rate requirement of DTs, the response time as described above is evaluated. As shown in figure 40, the edge-to-cloud (centralized) approach exhibits noticeable variations in response time, with spikes exceeding 450 milliseconds. In contrast, the edge-to-fog (distributed local processing) method maintains a stable response time, averaging between 100-150 milliseconds. This consistency validates the framework as it satisfies the low and predictable latency requirement for SCADA systems in supporting timely feedback and efficient control.

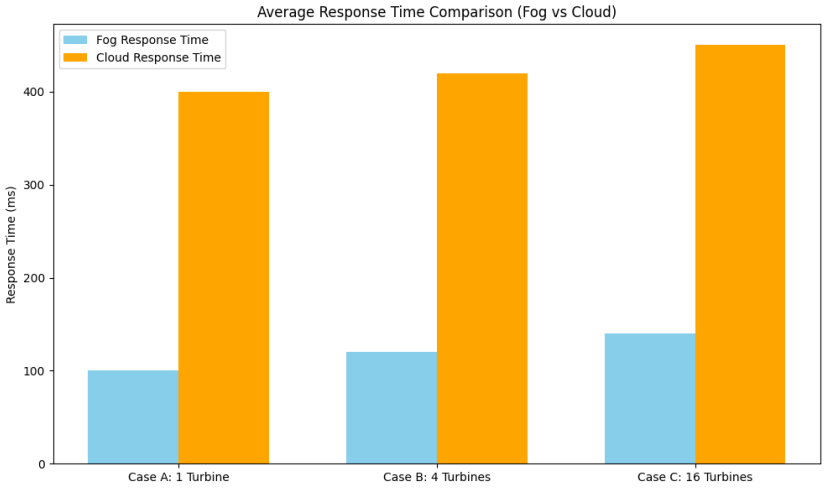


Figure 41: Average Response Time across 3 Cases

The response time was further compared in the following instances to check for the impact of scalability and the overhead of the proposed security framework.

- **Impact of scalability to response time:** Based on the scaling formula that synthesized the dataset to 16 wind turbines (table 8), a comparison on the effect of the size of wind turbine data and number of fog nodes was done. Figure 41 above shows these results.
- **Impact of security framework to response time:** In a similar approach, the overhead in processing time of applying the security framework was evaluated. A comparison is made in two scenarios, to assess the processing delay when the security framework is applied versus when it is not applied. Figures 42a and 42b shows this comparison.

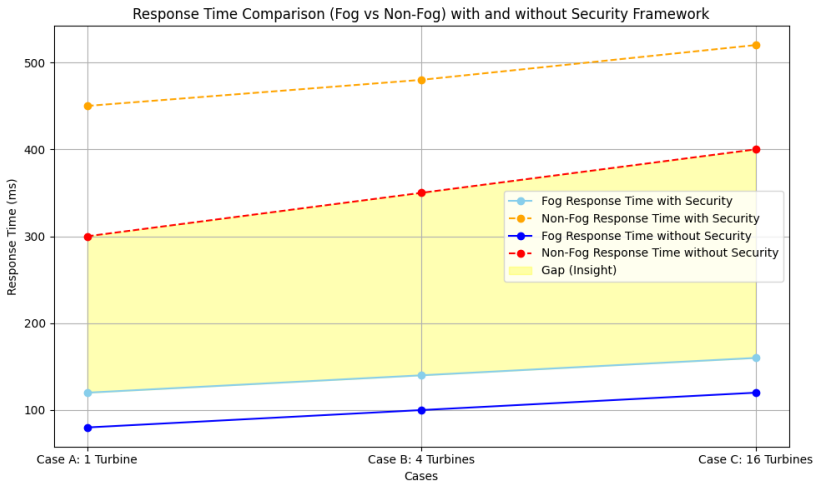


Figure 42a: Security overhead response time comparison

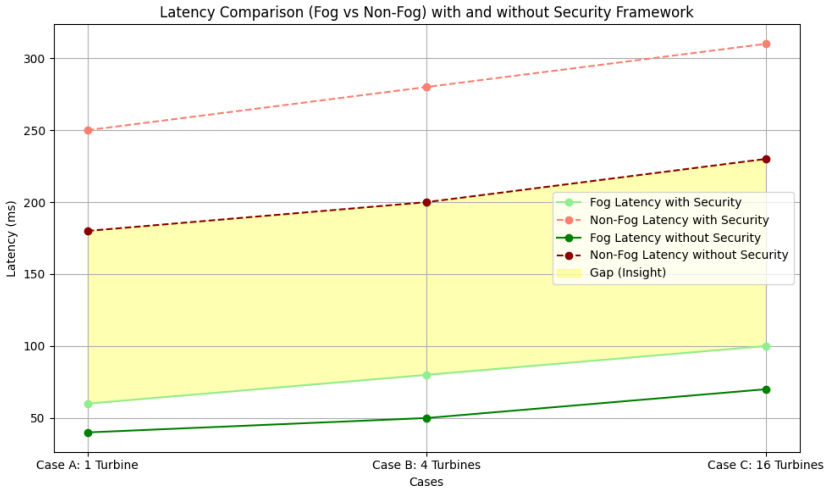


Figure 42b: Security overhead response time comparison

These evaluations assess the impact of scalability and security on response time. Figure 41 shows that distributing workloads across fog nodes reduces response times, even as turbine numbers increase. Figures 42a and 42b analyse the security framework's overhead and reveal that by offloading cryptographic tasks to fog nodes, the system can enhance security, without negatively impacting the response time for feedback. This confirms fog computing as an effective solution for scalable, secure, and real-time wind turbine control.

Metric Evaluation: Accuracy Validation

To further validate the framework with the cloud integration, the accuracy of running the feedback models (all three algorithms; MLR, XGBoost, LSTM) on the fog node was compared to running same models in the spark clusters on Databricks and the difference is negligible. This shows the capacity of the fog nodes to handle the models efficiently. Figure 43 shows the results.

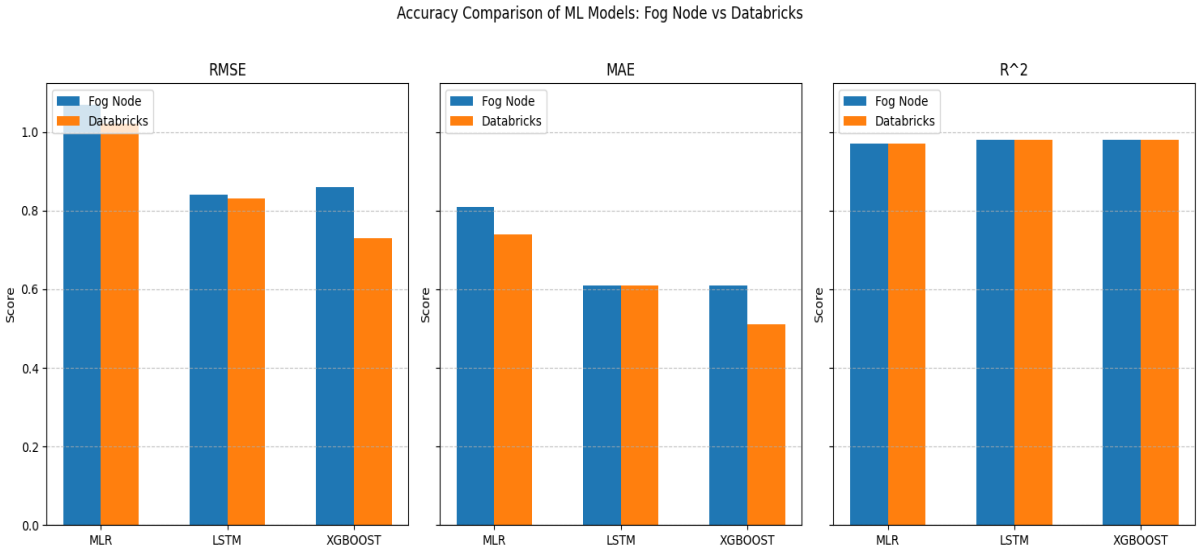


Figure 43: Accuracy Comparison: Fog vs Cloud

Metric Evaluation: Cost of running the framework

From a cost perspective, establishing a secure cloud connection was achieved using Virtual Networks (VNETs), load balancers, and data streaming services. In this experiment, the Azure Digital Twin (ADT) service was integrated with Databricks. The findings indicate that the non-fog setup incurs higher costs which can lead to a greater Total Cost of Ownership (TCO) over time. Notably, the expenses associated with maintaining a secure connection remain consistent across both use cases. Figure 44 illustrates this difference, where the first four days reflect the non-fog setup, while the last three days correspond to the fog setup.

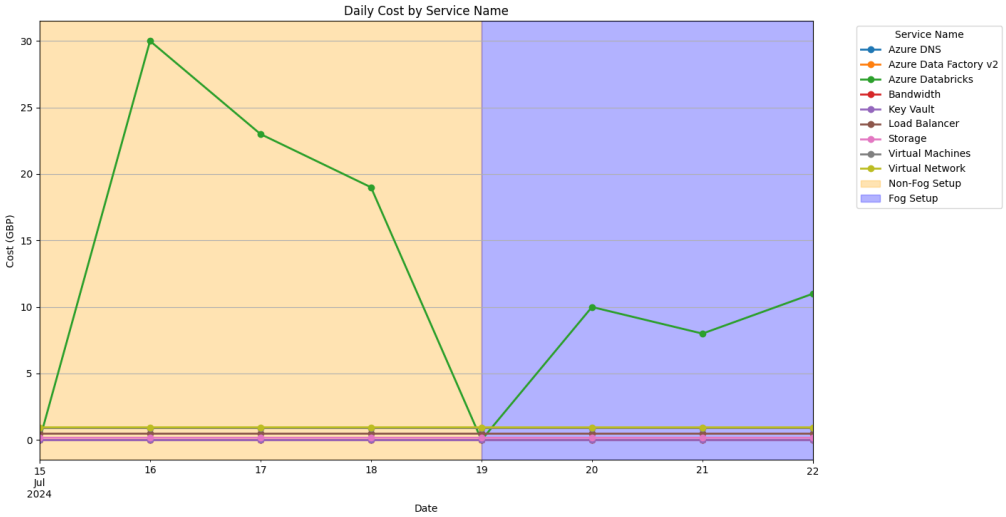


Figure 44: Cost Comparison of running the framework.

4.3.4 Framework Validation – Answering RQ3

The contributions in paper 5 collectively validate the framework based on covering the twinning rate and security requirements identified as gaps in DT development by the literature review in this study. An integration architecture of a SCADA system for wind turbines with a cloud-based data intelligence platform was designed, and experiments were conducted to evaluate the impact of the distributed DT framework. The impact of the proposed distributed DT framework was assessed in terms of the communication latency (response time) between the physical and digital twin. The response time metric was used as a benchmark to further evaluate the relevance of a fog-enabled security framework in improving security without negatively impacting performance. Additionally, a scalability validation methodology was used to compare the performance of the distributed DT framework in comparison to centralized approaches. Finally, the cost of running the framework with the cloud integration shows the commercial benefit in reducing the Total Cost of Ownership (TCO) of such a solution.

4.4 CHAPTER SUMMARY: CONTRIBUTIONS ANSWERING RQ3

This chapter provided a validation account of the distributed digital twin (DT) framework by evaluating its computational efficiency, security, scalability, and cost-effectiveness in SCADA-based wind turbine predictive maintenance. Building on chapter 3's predictive maintenance workflow, paper 4 examined three architectural scenarios—edge-only, edge-cloud, and edge-fog-cloud—assessing computational cost and latency. The results confirm that the edge-fog-cloud architecture optimally balances processing loads, which improves low-latency performance while reducing infrastructure costs. This finding directly supports hypothesis H-

6. Paper 5 extends the validation by providing an architecture that clarifies fog node placement, which discusses an integration of the distributed DT framework with a SCADA network and a cloud-based data intelligence platform. Paper 5 evaluates response time, the impact of a data security framework, and the impact of scalability on response time. The evaluation demonstrated that the fog layer enhances the synchronization rate between physical and digital twins while mitigating security risks through a fog-enabled security mechanism. The findings confirm that the distributed DT framework improves operational efficiency, supports secure predictive maintenance, and reduces the total cost of ownership. This proves hypothesis H-7 and answers RQ3 by showing how the suggested DT framework improves SCADA-based wind turbine management with secure, low-latency feedback loops. This validation makes sure that the solution for Prognostics and Health Management (PHM) is both affordable and scalable.

CHAPTER 5: INDUSTRY VALIDATION

An industry research partnership was initiated based on the contributions of this PhD work, with interest picked up by industry. This chapter presents some of the results obtained from applying the developed predictive maintenance solution in this PhD to further validate the framework, with real and current operational wind turbine SCADA data provided by a research partner, Natural Power Services (NP). NP is an independent consultant in the renewable energy industry. This chapter is structured with an introduction mentioning the project's objectives and stakeholders, a discussion on the methodology adopted, and a summary of validation results. A discussion on the contributions and future prospects of the project is also highlighted. The chapter concludes with an acknowledgment.

5.1 INTRODUCTION

The published research articles in this PhD work garnered industry interest. As a result, a research partnership was initiated, and a research project agreement was signed between Cranfield University and Natural Power Services Limited (NP). Natural Power is an independent consultant and service provider, working on green energy projects. Part of the services Natural Power offer is asset management of wind energy infrastructure across the globe. With 12 site offices across Europe and America, Natural Power manages wind farm assets for multiple clients in the renewable energy industry.

5.1.1 Aim and Objectives

The project is aimed at the training and testing of already developed predictive maintenance machine learning models (from this PhD work) for the prediction of selected component readings and the identification of fault from wind turbine Supervisory Control and Data Acquisition System (SCADA) data. The project has the following key objectives.

1. Use of wind turbine SCADA data to train, test and validate the accuracy of existing predictive maintenance models.
2. Review results and subject matter expert feedback to refine and re-validate models.
3. Use a mechanism for collaboration on validating the wind turbine maintenance prediction service through an accessible dashboard, with a framework integration scope for deployment of the framework contributed in this PhD.

5.1.2 Project Stakeholders

Table 9: Validation project stakeholders

No.	Stakeholder/Role	Responsibilities
1.	PhD Researcher (Ibrahim. A.)	• Access and manage anonymized data securely in local environments/cloud. • Perform experiments. • Provide regular updates, feedback, and results with Natural Power.
2.	PhD Supervision Team (S.L & M.S)	• Supervise, advise, and recommend best practices
3.	NP's Performance, Reliability and Optimization (PRO) team (A.I, J.G)	• Provide data and subject matter expertise for validation

The project involves stakeholders from Cranfield university and NP's PRO team. Table 9 above describes the collaboration in terms of roles and responsibilities of the stakeholders in the project.

5.2 PROJECT METHODOLOGY

The project is phased into the following stages described in figure 45 below, envisioned to cover an industry validation of the predictive maintenance models developed in this PhD work. Once the confidentiality agreement was signed, the first stage involves the initial data exploration meetings with the PRO team of Natural Power. A review of the data to refine the subject matter expert knowledge used in feature engineering was also done at this stage. In Stage 2, the provided data of a wind farm with 6 operational wind turbines was used to train and test the existing predictive maintenance models contributed in paper 3 of this PhD thesis. As the data provided is recent (2021 to 2024), stage 3 involved sessions for reviewing the results in terms of accuracy and evaluating the correctness of fault predictions. Finally, stage 4 of the project is on scoping the full implementation and deployment of the distributed DT framework within the context of NP's asset management operations.

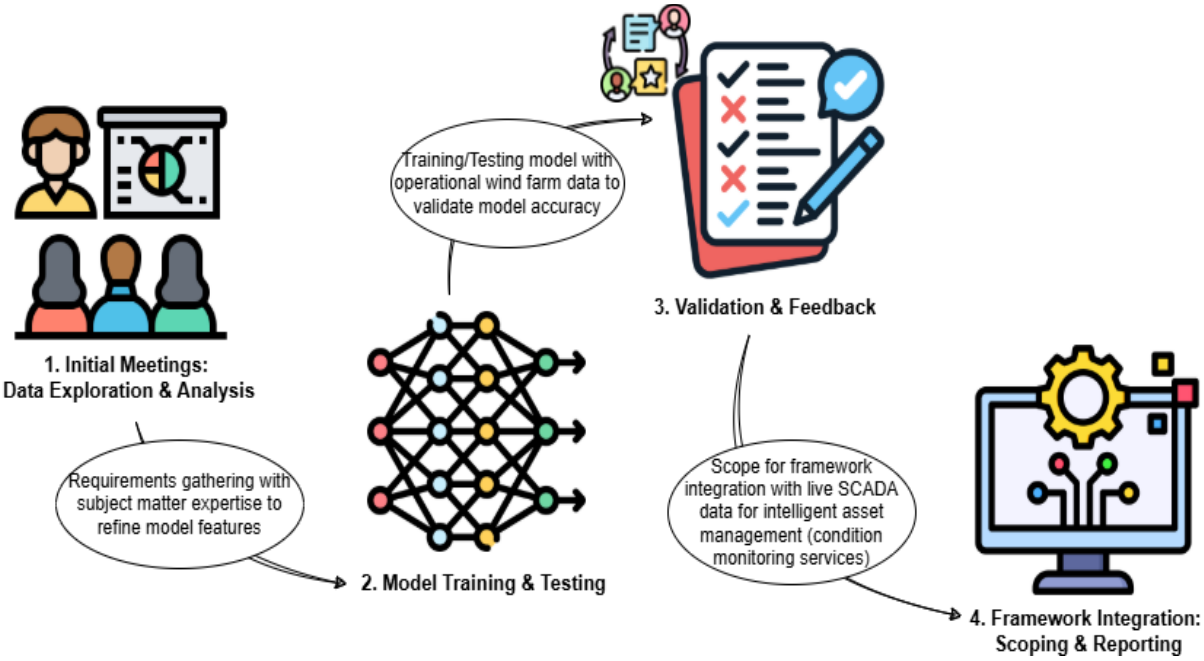


Figure 45: Industry Validation Project Stages.

5.2.1 The SCADA Data and Failure Logs

SCADA Data

The provided data is from 6 operational onshore wind turbines in a UK wind farm. The data is of 10-minute average like the EDP open dataset used in this PhD study. However, naming conventions of signals from the SCADA system is different and a mapping was required to identify the relevant variables of interest for the predictive maintenance models.

Fault/Maintenance Logs

The goal of the predictive maintenance DT is to provide asset managers with actionable predictions on asset health in order to respond swiftly. Therefore, the first step of the

exploration is to review the fault logs and confirm the fault trend of components from each of the wind turbines. Figure 46 below provides a high-level summary of the records of failure/maintenance events from all the wind turbines in the provided wind farm SCADA data from January 2021 to December 2024.

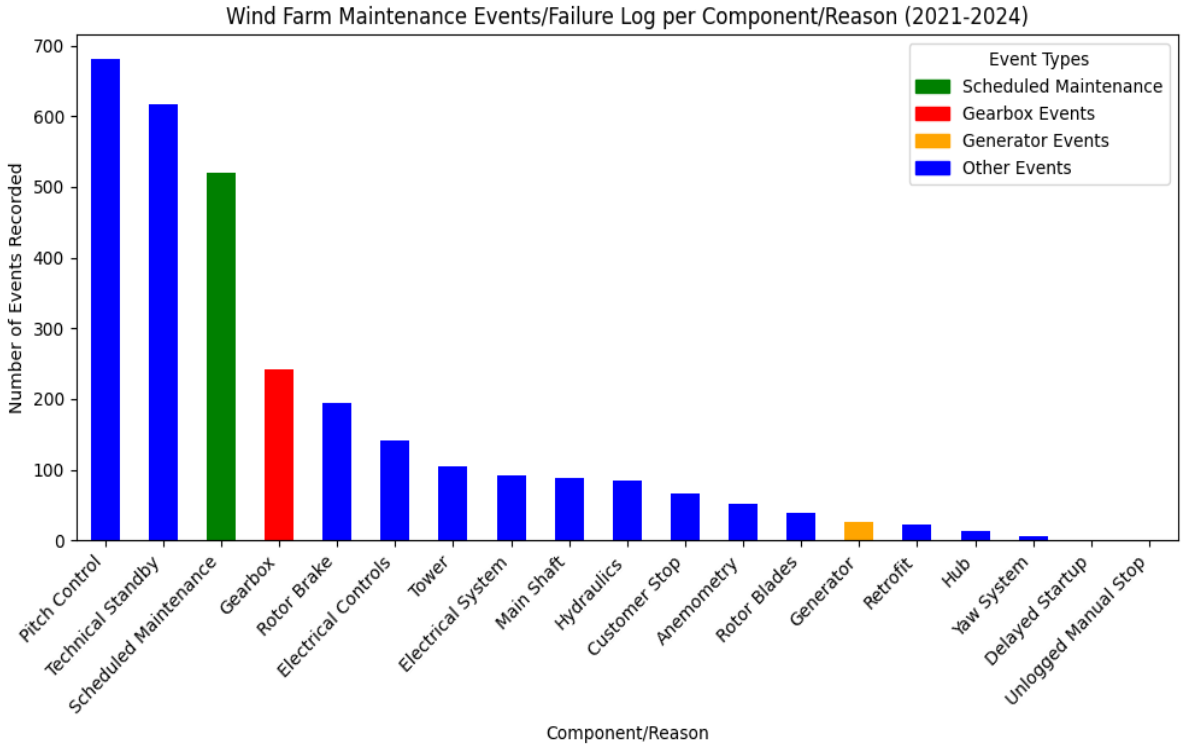


Figure 46: Maintenance/Failure logs from January 2021 to December 2024.

From figure 46, with the turbines stoppage event for scheduled maintenance being the third highest, it can be observed that scheduled maintenance is actively performed to manage and maintain the wind turbine assets. From the fault logs analysis, this maintenance strategy ensures less downtime as it is observable that the scheduled maintenance is done in a periodic manner. However, the failure trend also indicates that predictive maintenance can improve the asset management operations of NP. Additionally, the events of interest to this study are the gearbox highlighted in red, the generator highlighted in orange, and the scheduled maintenance highlighted in green. The logs validate this PhD study's selection of the gearbox as one of the key components causing wind turbine failure. A further analysis into the failure logs identified turbine NP-T06 had a failure or maintenance event of the gearbox 68 times over the 4 years (2021 to 2024). Selecting the fault logs of 2024, it was found that the gearbox of turbine NP-T06 recorded faults in July of 2024, described in table 10 below. This fault from table 10 was used to validate the models in this chapter.

Table 10: NP-T06 Gearbox – Selected fault for validation.

Turbine	Alert Start	Alert End	Alerts	Comment
T06	22/07/2024 06:41	23/07/2024 12:46	8	Low gearbox oil pressure fault requiring replacement of the oil pump coupling

5.3 VALIDATION RESULTS

For the validation experiment, the fault identified in table 10 is used. This section discusses the results from applying the predictive maintenance DT workflow in this study to confirm its accuracy in identifying the fault. After cleaning and pre-processing the data, it was split into training and testing sets. Next, the three models—MLR, XGBoost, and LSTM— are used for training and predictions. Finally, the SPC technique for fault prediction is used to identify faults, in line with the methodology discussed in chapter 3.

5.3.1 Gearbox Bearing Temperature Prediction

The same input and output variables used in the methodology of this PhD study were further used in training and testing the models. For the experiment shown here, the training split is from January to June (2024), while the testing split is from July to October (2024). Figure 47 shows the actual vs predicted results from the predictions with the XGBoost algorithm.

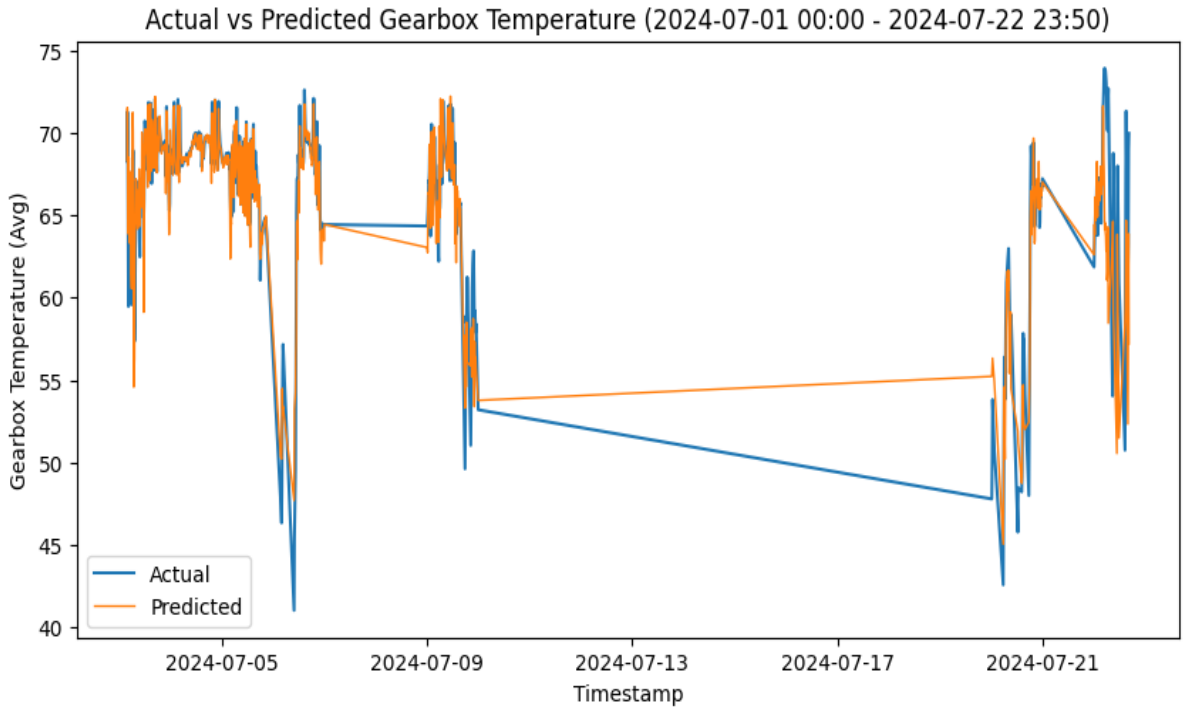


Figure 47a: Actual vs Predicted for T06 within failure timeframe.

From figure 47a, it can be observed that the high prediction accuracy is similar to the contributions in paper 3 showing the validity of the selected features from earlier in this research. However, it can also be observed that the model performs well in stable periods, but it struggles with sudden changes in operation fluctuations. This suggests that the model has room for improvement to capture short term variations. Table 11 below shows the accuracy metrics for the three models used in the study.

Table 11: Accuracy Metrics for NP-T06

Turbine	Year	MLR			LSTM			XGBoost		
		RMSE	MAE	R^2-Score	RMSE	MAE	R^2-Score	RMSE	MAE	R^2-Score
T06	2024 (Gear)	1.47	1.01	0.84	1.21	0.88	0.92	1.43	0.94	0.85

5.3.2 Gearbox Failure Prediction Using SPC

As discussed in chapter 3 of this thesis, the Shewhart process control technique is the methodology used in fault identification from the predictions of the ML/DL algorithms. Using the same strategy, the SPC is applied to evaluate the behaviour of the gearbox based on the predicted values of the gearbox bearing temperature. Figure 47b shows the SPC chart with the deviations that indicate an upcoming fault identified up to 3 weeks in advance, from the 2nd of July and again on the 10th of July 2024, while the actual fault occurred on the 22nd of July 2024 as indicated in the fault log (table 10).

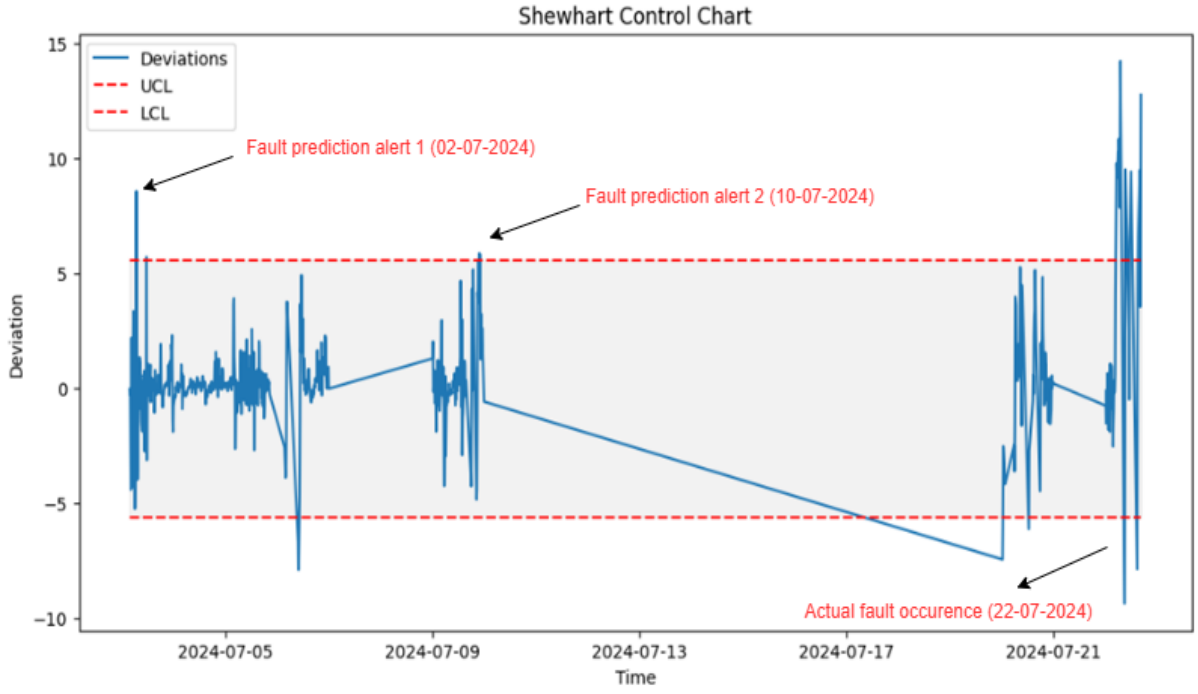


Figure 47b: Fault prediction using SPC for T06 Gearbox fault.

5.4 CONTRIBUTIONS

This validation project focused on validating the accuracy of the predictive maintenance models with fault identification. The metric used within the study is the model accuracy metric as evaluated in Chapter 3. Due the nature of the partnership, and time constraints, it was only feasible to show this aspect of the validation, and not the full DT framework implementation. Therefore, the major contributions of this industry validation project in line with the PhD work can be summarised in the points below.

- The use of real-life operational data to validate the accuracy in predicting gearbox component fault to enable asset managers receive alert up to 20 days before actual failure occurrence. However, further validation with NP is required due to the fault type.
- The provision of a 'Predictive Maintenance Digital Twin' dashboard that will be used by asset managers to continuously feed in live SCADA data for training/testing and predicting failures with visual analysis components. This is based on the output of this PhD study's methodology. Figure 48a and 48b below show a screenshot of the sample dashboard.

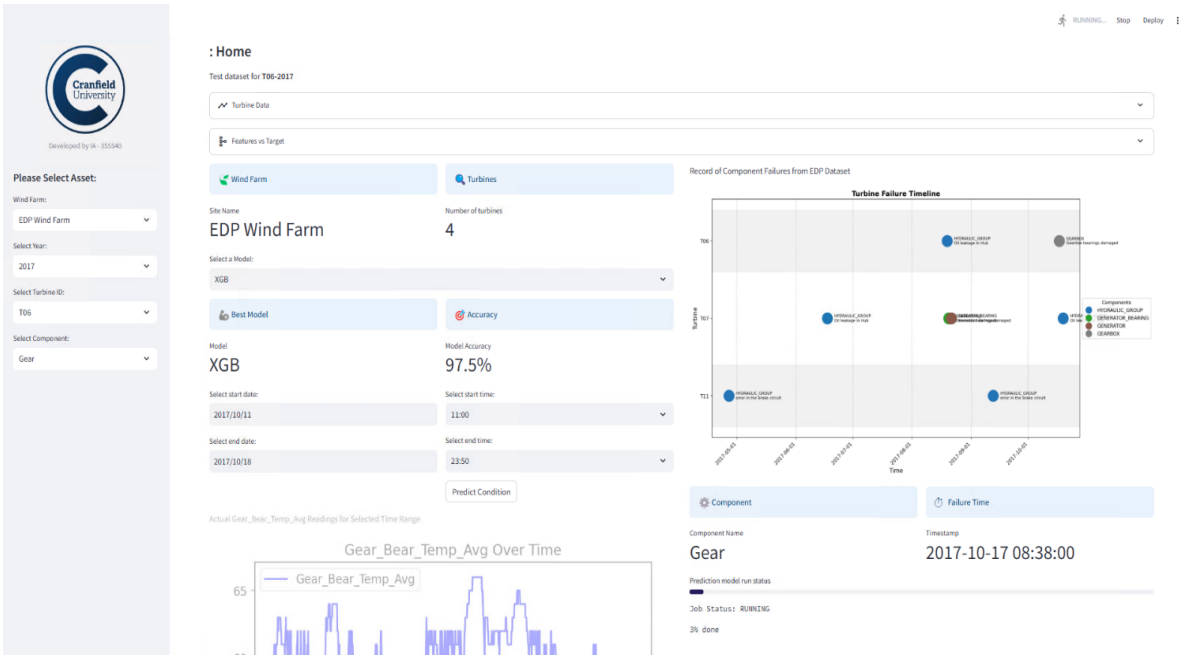


Figure 48a: Sample predictive maintenance DT dashboard: showing the user interface allowing users to select wind farm assets, components, and models for failure prediction.

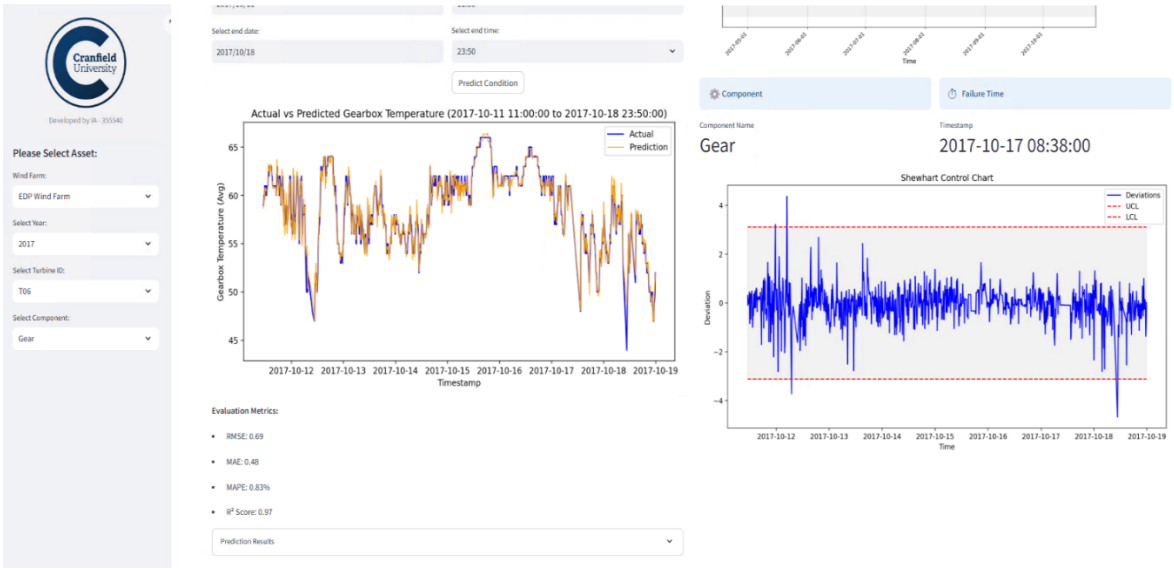


Figure 48b: Sample predictive maintenance DT dashboard: showing results from model runs based model configuration from selected components.

5.5 DISCUSSION AND FUTURE PROSPECT

The results presented from this validation project show the real-life impact of the developed solution in this PhD work. However, evident from the results, there is a room for improvement with subject matter expertise and operational knowledge that will fill in some of the gaps identified. For instance, where the models struggle to achieve full reflection of operational conditions when sudden drops happen. This project, therefore, presents a direct knowledge transfer from academia to industry with a potential for future improvement and extension into real-time prognostics and health management.

In chapter 3 of this thesis, an example was given to discuss a potential fog node placement strategy referencing a system called 'EchoSense. Natural Power invented EchoSense for bat curtailment while increasing energy production in wind farms [132], [133]. The system utilizes a Detection and Active Response Curtailment (DARC) system, as an integration of a dedicated server with the SCADA system to achieve real-time acoustic monitoring, which enables the bat curtailment strategy. Figure 49 below is an overview of the system developed by Natural Power[132], [133] .

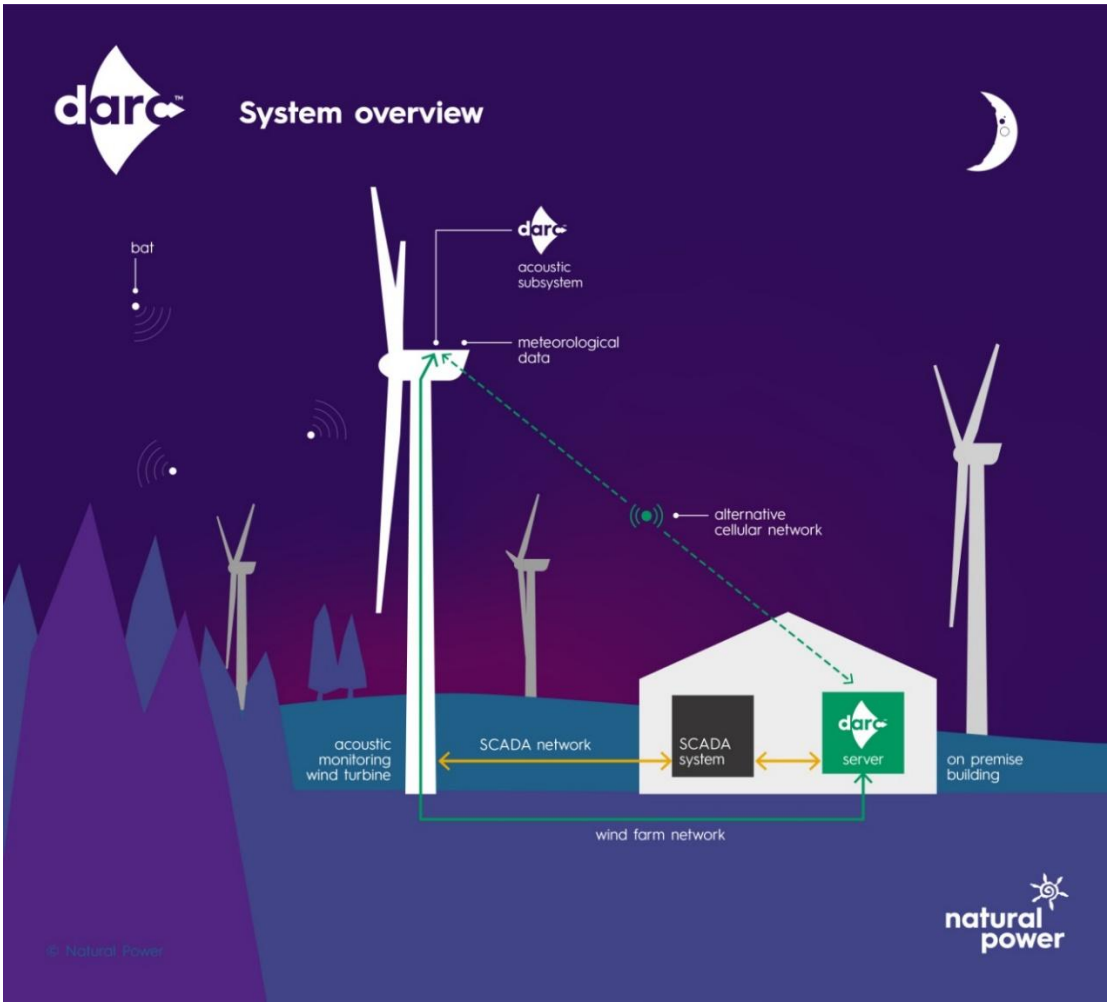


Figure 49: NP's Detection and Active Response Curtailment (DARC) system [132]

As a future prospect of this PhD study, a real-time condition monitoring solution based on fog computing (distributed DT framework) is achievable with AI/ML solutions to provide intelligent and autonomous PHM strategies, similar in deployment to the invention presented above by Natural Power.

5.6 ACKNOWLEDGEMENT

The author of this PhD thesis acknowledges Natural Power for supporting the research with real wind farm operational data. An appreciation is given to the Natural Power PRO team, Adam Ibrahim, PhD (Senior Systems Analyst) for finding the contributions from paper 3 insightful towards improving asset management at Natural Power and making contact, and Jacek Gibula (Head of PRO) for accepting the proposal. Additionally, the support from Cranfield's Research and Innovation office in the partnership is appreciated.

CHAPTER 6: CONCLUSION, LIMITATIONS & FUTURE WORK

This is the sixth and last chapter of this PhD thesis. From chapters 1 to 5, this thesis discussed the contributions of this PhD work from proposal to research design, to experiments and validation, aligning with research outputs in the form of paper publications and an industry validation project. This concluding chapter is in two sections. The first section concludes the thesis by highlighting the qualitative and quantitative benefits of the framework that justifies the framework architecture. A summary of contributions, challenges faced, lessons learned, and limitations are also provided in the first section. Finally, the second section of this chapter outlines future research opportunities building on the contributions and limitations of this PhD work.

6.1 CONCLUSION

Qualitatively, this PhD contributes an advancement to the theory of predictive maintenance digital twins by proposing a fog computing-based distributed DT framework that addresses key implementation gaps. The gaps covered are particularly in digital to physical twin feedback for decision support, standards adoption and performance of predictive maintenance solutions such as improving the synchronization rate and utilizing compute resources better. The practical contribution also demonstrates the potential of security integration for scalable and cost-effective deployment of digital twins in industrial settings such as wind turbines. Section 6.1.1 highlights the quantitative benefits that justify the architectural framework.

6.1.1 The Impact/Benefit: Justification of the Architectural Framework

From the contributions mentioned in chapter 1 of this thesis, three key areas were provided as the contributions of this PhD work. In this section, the three key areas of contribution are discussed with context on the quantitative benefits that justify the distributed DT framework.

1. **Scalable DT Architecture for Predictive Maintenance:** The metrics evaluated to validate the effectiveness of the distributed DT architecture, along with the benefits from experiment results are mentioned here.
    - **Prediction accuracy:** Overall, considering the R-squared accuracy metric, across all models (MLR, XGBoost and LSTM) and components, the XGBoost algorithm shows to be the most performing with up to 99% accuracy. However, each model presented its best performance for certain operational situations. Generally, whether running on the cloud or fog, the models achieved a minimum of 85% in feedback accuracy, but running the models on the edge layer had the least performance, with the LSTM not able to run at all on the edge. This quantitative result shows the benefit of the fog layer as it processes feedback close to the data sources without needing to send and wait on the cloud yet achieving the same accuracy. This enhances timely feedback control.
    - **Computational cost:** Similar to the above accuracy performance, across the deployment layers (edge, fog or cloud), the fog layer with microservices demonstrated the most scalable CPU performance, achieving up to 38% reduction in CPU utilization compared to the edge (figure 27b vs 34b). The edge layer had the least capacity for scalable operations, with CPU usage nearing maximum and longer feedback delays. The fog layer also maintained processing

times below 1 second across MLR and XGBoost, and less than 30 seconds for LSTM workloads, while the edge could not run LSTM (deep learning) at all. While the cloud offered high processing power, it incurred higher response times discussed in the next point. Therefore, these results confirm the benefit of the fog layer which delivers efficient, low-latency processing close to the source without overwhelming compute resources and enabling timely feedback control (from response time results). Furthermore, in terms of infrastructure running costs, the result show that the distributed DT framework outperforms the centralized architecture by achieving up to a 60% reduction in cloud service costs (from figure 44).

- **Response time:** For the response time metric showing the architecture with the best twinning rate, results from the cloud integration experiment discussed in paper 5 show that on average, the distributed DT framework based on fog computing achieved a 70% reduction in response time (from figure 40) for physical-to-digital twin communication. This further validates the benefits of the fog-based distributed architecture in improving the twinning rate.

2. **DT Standardization Benefits:** The quantitative results from (1) above helped in refining the architecture based on the ISO23247 standard. This contribution is mainly on showing an architectural benefit of using the standard with regards to how it can be used to cover a wholistic DT framework including aspects such as data storage for predictive maintenance which is a gap in literature. The implemented predictive maintenance digital twin solution in this work shows the adoption of ISO23247 to clarify where best the digital twin should be deployed for effective decision support in a predictive maintenance case study (from figure 36).
3. **Security Integration Benefit:** To show the potential benefits of the framework in improving security rather than impairing it, a security mechanism based on fog computing is introduced. Overall, security itself is not the focus of the work, but showing the frameworks potential in covering security concerns was necessary. The security mechanism is shown not to impair performance with respect to the response time (twinning rate). The limitation of the security aspect of the work is mentioned in section 6.1.3.

Enhancing SCADA Systems with Digital Twins

The contributions in this thesis justify that DTs add significant value over traditional SCADA systems by enabling real-time, bi-directional synchronization between physical wind turbine assets and their digital counterparts. Unlike SCADA, which primarily monitors and controls assets, DTs integrate machine learning to predict failures, simulate “what-if” scenarios, and can automate responses in terms of control. The inclusion of fog computing enhances this by enabling low-latency, decentralized data processing close to the turbines. This proximity and compute capacity at the edge reduces reliance on cloud infrastructure which is typically located in faraway data centres. The distributed architecture allows for predictive maintenance with faster decision-making. This benefit makes DT essential for effective and intelligent operations in industry 4.0 environments.

6.1.2 Summary of Contributions

This PhD study focused on developing a digital twin framework to improve predictive maintenance as an asset management strategy using the digital twin technology. Firstly, in

Chapter 1, the perceived issues from establishing the background on developing predictive maintenance digital twins involving the integration of key enabling technologies are highlighted. This initial survey helped in research question formulation, along with the research design using a multi-methodology approach involving elements of DSR and ECSE methodologies. In Chapter 2, a systematic literature review and framework conceptualization are contributed in published papers 1 and 2. Chapter 2 facilitated the establishment of the research gaps using the DSR methodology. The identified challenges and research gaps were used as concepts to explore in building the distributed digital twin framework using fog computing, which was considered an effective approach to digital twin development for predictive maintenance in IIoT. This approach helped in answering research question 1 (RQ1) of the thesis.

The PhD research took a phased approach, having identified key areas to cover in achieving the framework. To answer research question 2 (RQ2), Chapter 3 discussed the contributions of paper 3, which discusses the implementation of the distributed digital twin framework and evaluation of the accuracy of the machine learning and statistical techniques used for predictive maintenance. Chapter 3 also fills in a gap in the research on the feedback gap between the digital twin and the physical twin by introducing a prediction feedback loop.

Research question 3 (RQ3) was answered in chapter 4, which discussed the contributions of papers 4 and 5 that help in standardization. The contributions of papers 4 and 5 align with key metrics used in assessing the framework's impact. Specifically, the computational cost metric is covered in paper 4, while the response time, scalability, and security metrics are covered in paper 5. Paper 5 also evaluated the framework's running costs.

Finally, the contributions of this research show that an effective digital twin framework to improve prognostics and health management (PHM) is achievable with performance benefits, using a standardized approach based on the fog computing paradigm. However, there exist challenges and limitations. The next subsection below highlights the challenges faced, lessons learned, and limitations of this work in addressing the individual research gaps covered by this work.

6.1.3 Challenges, Lessons, and Limitations

As with any research work, this study faced and acknowledged challenges leading to limitations. However, the study provided some lessons that the researcher considers valuable and part of the contribution of the PhD. These challenges and lessons are mentioned below for each of the research gaps the framework covers.

Decision Support:

- **Challenge:** Establishing a reliable feedback mechanism between the Digital Twin and Physical Twin for optimal decision-making is complex due to the need for real-time data accuracy and integration.
- **Limitation:** The feedback mechanism was implemented as a concept, and systems control with actual hardware was not implemented due the limitation and scope of the experiment setup used.
- **Lesson:** A hybrid model with physics-based and hybrid-based model, informed by subject matter expertise is required to validate decision support digital twins. However, data quality checking is fundamental to a digital twin's accuracy.

Standardization:

- **Challenge:** Achieving standardization across various enabling technologies, with interoperability and reliability is difficult due to the diverse nature of industrial systems and components.
- **Limitation:** The framework's reliance on ISO23247 limits its applicability in environments where different standards are prevalent.
- **Lesson:** Developing a modular framework that can adapt to different standards and technologies can facilitate broader adoption and integration. A phased, case study approach is recommended.

Cost and Scalability:

- **Challenge:** Balancing infrastructure and operational costs while ensuring scalability in large-scale industrial applications must consider several factors. The best cost-effective solution may not be the most operationally viable solution.
- **Limitation:** Assumptions were made in the fog node placement design.
- **Lesson:** Leveraging fog computing and distributed architectures can optimize costs and improve scalability without compromising performance. However, the initial setup cost for fog computing infrastructure can be high, and scalability may be constrained by the reliance on the availability of fog nodes.

Twinning Rate:

- **Challenge:** Ensuring the right synchronization rate between the physical and digital twins is challenging due to latency and data processing delays. Isolating other background processes to optimize resources for the experiments was difficult.
- **Limitation:** The twinning rate in this work was considered based on simulations to measure the response time. The use of 1 second intervals for SCADA measurements and milliseconds for communication latency can be improved to find the right synchronization rate for the case study. Furthermore, highly dependent on network conditions and the computational power of edge devices. Multiple other factors were not considered, rather a reasonable experiment set up was used to achieve results. For instance, other communication technologies such as 5G/6G were not used.
- **Lesson:** Technology advances fast; a simple solution that works must consider room for improvement and integration of future technologies. An open architecture is beneficial for this.

Security and Privacy:

- **Challenge:** Addressing data governance, privacy, and security concerns in distributed architectures requires comprehensive strategies to protect sensitive information.
- **Limitation:** The security framework was not assessed for intrusion or attacks. Additionally, the framework did not consider existing security mechanisms used in typical SCADA infrastructure.
- **Lesson:** Incorporating encryption, access control, and secure communication protocols can mitigate security risks and enhance data privacy but implementation of additional security measures can introduce latency and increase computational overhead. There needs to be a balance.

The above-mentioned challenges, limitations, and lessons acknowledge the room for improvement in future studies pertaining to implementing predictive maintenance digital twins, considering both technical and operational aspects. However, this study's contributions provide a foundation for addressing these issues and advancing the field of digital twin technology in industrial applications.

6.2 FUTURE RESEARCH

The contributions of this PhD work have shown the potential of utilizing a distributed digital twin framework to improve prognostics and health management. The section above also acknowledges the challenges and limitations associated with the scope of this PhD research. However, it is important to mention some potential further research work that can build on the contributions and limitations of this PhD work. Below is a highlight of some of the identified future research opportunities with regards to the areas covered.

Decision Support: Intelligent and Autonomous Prognostics and Health Management Solutions.

Building on the feedback mechanism concept implemented in this work, a potential research area is in the use AI techniques to implement autonomous agents that can proactively control IIoT assets such as wind turbines on the prediction of faults. The distributed DT approach has laid a foundation of providing additional compute that enhances edge systems. This can be achieved in an approach like Natural Power's Detection and Active Response Curtailment (DARC) system [132] referenced in chapter 4 and 5.

Standardization: Development of a Universal Standards and Protocol Enhancing Interoperability

The digital twin technology remains an emerging technology that will benefit from further standardization considering applications in diverse industrial applications. While ISO 23247 presents a good starting point, further research can unlock opportunities that will promote interoperability. For instance, this study considered the Open DT concept to support plug and play scenarios in paper 2 but this was not explored further. As such there is a gap that can be covered in refining standards and protocols for integration using open-source technologies. This will promote the adoption of DT.

Cost and Scalability: Refining DT Architectures for Scalability and Optimizing Cost.

This study was limited to highlighting the feedback mechanism with a simulation of regulating the RPM of the Rotor component. Further research into building models and deployment architectures that can support multiple simulations using the '*what-if-scenarios*' autonomously can be very impactful. For instance, building on the contributions of this PhD, a study on implementing AI/ML solutions that provide system control simulations with multiple signals instantaneously to find out what acceptable configuration control can extend remaining useful life (RUL) of the asset in real-time, will be powerful. For a better solution, this area can consider optimization of distributed nodes for resource and cost efficiency.

Twinning Rate: Digital Twins with Low-latency Real-time Synchronization Leveraging 5G/6G communications.

Within the DT hype in academia and industry, there has been some high-level research in the potential of DT in utilizing low-latency communications. This research has shown the background requirement for achieving the right synchronization rate covering the twinning rate requirement. However, It is important for further work to identify the best twinning rate such as nanoseconds, milliseconds, seconds or minutes that satisfy other case studies. Furthermore, this area can be further explored with advancements in AI in the next decade. This research area links back to the intelligent and autonomous PHM concept described above. AI agents can be developed for autonomous control of industrial assets when low latency, reliable communications is achieved.

Security and Privacy: Securing Industrial Internet of Things Infrastructure

Security, reliability and trust are fundamental requirements for building autonomous systems. If innovations in the field of using AI for intelligent PHM is to be achieved, research into securing such advancements is a huge area to be covered. This is because as advancement in AI influences better performing chips, cyber attackers can also leverage such systems to break into industrial systems. This, therefore, presents an opportunity for further research.

REFERENCES

[1] “Renewables 2024 - Analysis and forecast to 2030,” International Energy Agency, Oct. 2024. Accessed: Jan. 05, 2025. [Online]. Available: https://www.iea.org/reports/renewables-2024

[2] “Global landscape of renewable energy finance 2023,” International Renewable Energy Agency (IRENA), 978-92-9260-523–0, Feb. 2023. Accessed: Jan. 10, 2025. [Online]. Available: https://www.irena.org/Publications/2023/Feb/Global-landscape-of-renewable-energy-finance-2023

[3] Joyce Lee, Feng Zhao, “GWEC Global Wind Report 2021,” Global Wind Energy Council (GWEC), Mar. 2021. Accessed: Aug. 07, 2023. [Online]. Available: https://gwec.net/wp-content/uploads/2021/03/GWEC-Global-Wind-Report-2021.pdf

[4] Matthew Alabaster, Ajay Amin, and Naz Klendjian, “De-risking the energy transition in Europe: Investing in the path to net zero.” Accessed: Oct. 23, 2024. [Online]. Available: https://www.pwc.com/gx/en/issues/esg/the-energy-transition/de-risking-the-energy-transition-in-europe.html

[5] Ryan Beene and Josh Saul, “Wind Turbines Taller Than the Statue of Liberty Are Falling Over,” *Bloomberg*, Jan. 23, 2023. Accessed: Oct. 12, 2023. [Online]. Available: https://www.bloomberg.com/news/articles/2023-01-23/wind-turbine-collapses-punctuate-green-power-growing-pains

[6] Andy Trigg, “Scroby Sands: Investigation into wind turbine fire ‘may take weeks,’” *BBC News*, Aug. 16, 2023. Accessed: Oct. 13, 2023. [Online]. Available: https://www.bbc.co.uk/news/uk-england-norfolk-66525400

[7] “Vestas wind turbine catches fire in Denmark,” *Reuters*, Denmark, Oct. 16, 2024. Accessed: Nov. 02, 2024. [Online]. Available: https://www.reuters.com/world/europe/vestas-wind-turbine-catches-fire-denmark-2024-10-16/

[8] Shawn Sheng and Jon Keller, “Gearbox Reliability Collaborative Update,” presented at the Sandia Reliability Workshop, Alburquerque, New Mexico, Aug. 13, 2013. Accessed: Apr. 10, 2022. [Online]. Available: https://www.nrel.gov/docs/fy14osti/60141.pdf

[9] Raja V. Pulikollu and Brandon Fitchett, “Wind Turbine Gearbox Reliability Assessment,” Electric Power Research Institute (EPRI), California, USA, Dec. 2021. Accessed: Apr. 13, 2022. [Online]. Available: https://www.epri.com/research/products/000000003002021422

[10] D. Bamunuarachchi, D. Georgakopoulos, A. Banerjee, and P. P. Jayaraman, “Digital Twins Supporting Efficient Digital Industrial Transformation,” *Sens. - MDPI*, 2021.

[11] “Office of Energy Efficiency and Renewable Energy, "How Do Wind Turbines Work?," in *US Department of Energy, [Online*. [Online]. Available: https://www.energy.gov/eere/wind/how-do-wind-turbines-work.

[12] “Bear Creek Wind Farm,” West Penn Energy Fund, Jan. 2025. [Online]. Available: https://www.westpennenergyfund.org/project/bear-creek-wind-farm

[13] “Scroby Sands offshore wind farm,” RWE in the UK, 2025. [Online]. Available: https://uk.rwe.com/locations/scroby-sands-offshore-wind-farm/

[14] “Whitlee Wind Farm,” ScottishPower Renewables, 2025. [Online]. Available: https://www.whiteleewindfarm.co.uk/whitelee-windfarm-about-us

[15] “Power plant profile: Gansu Guazhou Wind Farm, China.” Accessed: Jan. 03, 2025. [Online]. Available: https://www.power-technology.com/data-insights/power-plant-profile-gansu-guazhou-wind-farm-china/

[16] “SCADA: Supervisory Control and Data Acquisition: What is SCADA, Who Uses it and How SCADA Has Evolved.” Accessed: Jul. 30, 2024. [Online]. Available: https://inductiveautomation.com/resources/article/what-is-scada
[17] “The history from Industry 1.0 to 4.0.” Accessed: Apr. 20, 2024. [Online]. Available: https://sitconsultores.com/en/blog/article/2021/06/12/the-history-from-industry-1.0-to-4.0
[18] “What is the Internet of Things (IoT)?,” IBM Think. [Online]. Available: https://www.ibm.com/think/topics/internet-of-things
[19] M. Azure, “What is Cloud Computing?,” *Microsoft*, 2023, [Online]. Available: https://azure.microsoft.com/en-au/resources/cloud-computing-dictionary/what-is-cloud-computing.
[20] Antonio Armenta, “Cloud Computing and the Industrial Internet of Things.” [Online]. Available: https://control.com/technical-articles/cloud-computing-and-the-industrial-internet-of-things/
[21] Hany F. Atlam, Ahmed Alenezi, Abdulrahman Alharthi, Robert J. Walters, and Gary B. Wills, “Integration of Cloud Computing with Internet of Things: Challenges and Open Issues,” presented at the 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK.: IEEE, Feb. 2018. doi: https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2017.105.
[22] “What Is Edge Computing?” Accessed: May 10, 2022. [Online]. Available: https://www.cisco.com/c/en/us/solutions/computing/what-is-edge-computing.html
[23] M. Azure, “Cloud computing vs. edge computing vs. fog computing,” *Microsoft*, 2023, [Online]. Available: https://azure.microsoft.com/en-gb/resources/cloud-computing-dictionary/what-is-edge-computing/.
[24] “OpenFog Reference Architecture for Fog Computing,” OpenFog Consortium, Reference Architecture, Feb. 2017. Accessed: Jun. 20, 2022. [Online]. Available: https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf
[25] G. Peralta, M. Iglesias-Urkia, M. Barcelo, R. Gomez, A. Moran, and J. Bilbao, “Fog computing based efficient IoT scheme for the Industry 4.0,” in *2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM*, Donostia, Spain, 2017.
[26] “Meet Jetson, the Platform for AI at the Edge.” Accessed: Dec. 20, 2024. [Online]. Available: https://developer.nvidia.com/embedded-computing
[27] M. Grieves and J. Vickers, “Origins of the digital twin concept,” *Fla. Inst. Technol.*, vol. 8, pp. 3–20, 2016.
[28] E. H. Glaessgen and D.S. Stargel, “The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles,” in *AIAA*, Honolulu, Hawaii: American Institute of Aeronautics and Astronautics, Apr. 2012. [Online]. Available: https://ntrs.nasa.gov/api/citations/20120008178/downloads/20120008178.pdf
[29] Aidan Fuller, Zhong Fan, Charles Day, and Chris Barlow, “Digital Twin: Enabling Technologies, Challenges and Open Research,” vol. 8, pp. 108952–108971, May 2020, doi: https://doi.org/10.1109/ACCESS.2020.2998358.
[30] Yuqian Lu, Chao Liu, Kevin I-Kai Wang, Huiyue Huang, and Xun Xu, “Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues,” vol. 61, Aug. 2019, doi: https://doi.org/10.1016/j.rcim.2019.101837.
[31] I. Errandonea, S. Beltrán, and S. Arrizabalaga, “Digital Twin for maintenance: A literature review,” *Comput. Ind.*, vol. 123, p. 103316, 2020, doi: https://doi.org/10.1016/j.compind.2020.103316.

[32] I. Abdullahi, H. Larijani, D. Liarokapis, J. Paterson, D. Jones, and S. Murray, “A Data-Intelligence-Driven Digital Twin Framework for Improving Sustainability in Logistics,” *Appl. Sci.*, vol. 15, no. 2, 2025, doi: 10.3390/app15020601.
[33] T. P. Raptis, A. Passarella, and M. Conti, “Data management in industry 4.0: State of the art and open challenges,” *IEEE Access*, vol. 7, pp. 97052–97093, 2019.
[34] H. M. Arruda, R. S. Bavaresco, R. Kunst, E. F. Bugs, G. C. Pesenti, and J. L. V. Barbosa, “Data Science Methods and Tools for Industry 4.0: A Systematic Literature Review and Taxonomy,” *Sensors*, vol. 23, no. 11, 2023, doi: 10.3390/s23115010.
[35] M. Shafiee, “Extending the Lifetime of Offshore Wind Turbines: Challenges and Opportunities,” *Energies*, vol. 17, no. 16, 2024, doi: 10.3390/en17164191.
[36] “Wind Turbine Maintenance: A Complete Guide,” BGB News & Articles. Accessed: Dec. 23, 2024. [Online]. Available: https://www.bgbinnovation.com/knowledge/news-and-articles/wind-turbine-maintenance-guide
[37] I. H. Sarker, “Machine Learning: Algorithms, Real-World Applications and Research Directions,” *SN Comput. Sci.*, vol. 2, no. 3, p. 160, Mar. 2021, doi: 10.1007/s42979-021-00592-x.
[38] Raymon van Dinter, Bedir Tekinerdogan, and Cagatay Catal, “Predictive maintenance using digital twins: A systematic literature review,” vol. 151, no. 107008, Jul. 2022, doi: https://doi.org/10.1016/j.infsof.2022.107008.
[39] J. Bofill, M. Abisado, J. Villaverde, and G. A. Sampedro, “Exploring Digital Twin-Based Fault Monitoring: Challenges and Opportunities,” *Sensors*, vol. 23, no. 16, 2023, doi: 10.3390/s23167087.
[40] T. Böttjer *et al.*, “A review of unit level digital twin applications in the manufacturing industry,” *CIRP J. Manuf. Sci. Technol.*, vol. 45, pp. 162-189, 2023.
[41] D. S. Satwaliya, H. P. Thethi, A. Dhyani, G. R. Kiran, M. Al-Taee, and M. B. Alazzam, “Predictive Maintenance using Machine Learning: A Case Study in Manufacturing Management,” in *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2023, pp. 872–876. doi: 10.1109/ICACITE57410.2023.10183012.
[42] Prof. David Hardt and Prof. Duane Boning, “Shewhart SPC & Process Capability,” presented at the Control of Manufacturing Processes, MIT, Feb. 28, 2008. [Online]. Available: https://ocw.mit.edu/courses/2-830j-control-of-manufacturing-processes-sma-6303-spring-2008/resources/lecture7/
[43] O. A. Vanli and E. D. Castillo, “Statistical Process Control in Manufacturing,” in *Encyclopedia of Systems and Control*, J. Baillieul and T. Samad, Eds., Cham: Springer International Publishing, 2021, pp. 2142–2150. doi: 10.1007/978-3-030-44184-5_258.
[44] W. Udo and Y. Muhammad, “Data-Driven Predictive Maintenance of Wind Turbine Based on SCADA Data,” *IEEE Access*, vol. 9, pp. 162370–162388, 2021, doi: 10.1109/ACCESS.2021.3132684.
[45] “ISO 23247 - Digital twin framework for manufacturing,” in *ISO (International Organization for Standardization*, Geneva, 2021.
[46] E. Ferko, A. Bucaioni, P. Pelliccione, and M. Behnam, “Standardisation in Digital Twin Architectures in Manufacturing,” in *2023 IEEE 20th International Conference on Software Architecture (ICSA)*, 2023, pp. 70–81. doi: 10.1109/ICSA56044.2023.00015.
[47] F. P. Knebel, J. A. Wickboldt, and E. P. Freitas, “A Cloud-Fog Computing Architecture for Real-Time Digital Twins,” *J. Internet Serv. Appl.*, 2021.
[48] A. Costantini *et al.*, “IoTwins: Toward Implementation of Distributed Digital Twins in Industry 4.0 Settings,” *Computers*, vol. 11, no. 5, 2022, doi: 10.3390/computers11050067.

[49] A. van der Merwe, A. Gerber, and H. Smuts, “Guidelines for Conducting Design Science Research in Information Systems,” in *ICT Education*, B. Tait, J. Kroeze, and S. Gruner, Eds., Cham: Springer International Publishing, 2020, pp. 163–178.
[50] National Academies of Sciences Engineering and Medicine, *Academic Careers for Experimental Computer Scientists and Engineers*. Washington, DC: The National Academies Press, 1994. doi: 10.17226/2236.
[51] J. Mingers and J. Brocklesby, “Multimethodology: Towards a framework for mixing methodologies,” *Omega*, vol. 25, no. 5, pp. 489–509, 1997, doi: https://doi.org/10.1016/S0305-0483(97)00018-2.
[52] I. Abdullahi, S. Perinpanayagam, and I. Hamidu, “A Fog Computing Based Approach Towards Improving Asset Management and Performance of Wind Turbine Plants Using Digital Twins,” in *2022 27th International Conference on Automation and Computing (ICAC)*, 2022, pp. 1–5. doi: 10.1109/ICAC55051.2022.9911170.
[53] I. Abdullahi and S. Perinpanayagam, “Development of an Open Distributed Digital Twin Model for Effective Asset Management in Industrial Internet of Things (IIOT),” *Annu. Conf. PHM Soc.*, vol. 14, no. 1, 2022, [Online]. Available: https://www.papers.phmsociety.org/index.php/phmconf/article/view/3411
[54] I. Abdullahi, S. Longo, and M. Samie, “Towards a Distributed Digital Twin Framework for Predictive Maintenance in Industrial Internet of Things (IIoT),” *Sensors*, vol. 24, no. 8, 2024, doi: 10.3390/s24082663.
[55] I. Abdullahi, S. Longo, and M. Samie, “An Enabling Architecture for Computational Cost Efficiency in Predictive Maintenance Digital Twins,” in *2024 International Conference on Cyber-Physical Social Intelligence (ICCSI)*, 2024, pp. 1–6. doi: 10.1109/ICCSI62669.2024.10799418.
[56] I. Abdullahi, S. Longo, M. Samie, and S. Perinpanayagam, “Integrating a Fog Based Distributed Digital Twin Framework with the Cloud for Improving Asset Management with Predictive Maintenance in Wind Farms.” 2023. doi: 10.2139/ssrn.5082896.
[57] N. Singh, S. S. Chouhan, and K. Verma, “Object Oriented Programming: Concepts, Limitations and Application Trends,” in *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, 2021, pp. 1–4. doi: 10.1109/ISCON52037.2021.9702463.
[58] A. L. Yacelga, N. B. Arevalo, and L. A. Zambrano, “Fog Computing in the Industrial Internet of Things: Challenges, Trends, and Strategies,” *Fusion Pract. Appl. FPA*, vol. 13, no. 02, pp. 91–105, 2023, doi: 10.54216/FPA.130208.
[59] *ISO/IEC/IEEE 15288:2023 Systems and software engineering — System life cycle processes*, ISO/IEC/IEEE 15288:2023, 2023. doi: https://www.iso.org/standard/81702.html.
[60] “The Industrial Internet Reference Architecture (IIRA).” Industry IoT Consortium, Nov. 07, 2022. Accessed: Apr. 12, 2023. [Online]. Available: https://www.iiconsortium.org/wp-content/uploads/sites/2/2022/11/IIRA-v1.10.pdf
[61] “RAMI4.0 – a reference framework for digitalisation,” presented at the RAMI4.0 – a reference framework for digitalisation, Aug. 09, 2018. Accessed: May 23, 2023. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.pdf?__blob=publicationFile&v=1
[62] Y. Qamsane *et al.*, “A Methodology to Develop and Implement Digital Twin Solutions for Manufacturing Systems,” *IEEE Access*, vol. 9, pp. 44247–44265, 2021, doi: 10.1109/ACCESS.2021.3065971.
[63] “Digital twin reference model and standardization to realize a sustainable industry.” Industrie 4.0, Apr. 15, 2024. Accessed: Oct. 17, 2024. [Online]. Available:

https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/202404_Digital_twin_sustainable_industry.html
[64] A. Thelen *et al.*, "A comprehensive review of digital twin — part 1: modeling and twinning enabling technologies," *Struct. Multidiscip. Optim.*, vol. 65, no. 12, p. 354, Nov. 2022, doi: 10.1007/s00158-022-03425-4.
[65] I. Errandonea, S. Beltran, and S. Arrizabalaga, "Digital Twin for maintenance: A literature review," *Comput. Ind.*, 2020.
[66] E. Cinar, S. Kalay, and I. Saricicek, "A Predictive Maintenance System Design and Implementation for Intelligent Manufacturing," *Mach. - MDPI*, 2022.
[67] S. Cavalieri and M. G. Salafia, "A Model for Predictive Maintenance Based on Asset Administration Shell," *Sens. MDPI*, 2020.
[68] D. Zhong, Z. Xia, Y. Zhu, and J. Duan, "Overview of predictive maintenance based on digital twin technology," *Heliyon*, vol. 9, no. 4, Apr. 2023, doi: 10.1016/j.heliyon.2023.e14534.
[69] P. Aivaliotis, K. Georgoulias, and G. Chryssolouris, "The use of Digital Twin for predictive maintenance in manufacturing," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 11, pp. 1067–1080, Nov. 2019, doi: 10.1080/0951192X.2019.1686173.
[70] G. Falekas and A. Karlis, "Digital Twin in Electrical Machine Control and Predictive Maintenance: State-of-the-Art and Future Prospects," *Energies*, vol. 14, no. 18, 2021, doi: 10.3390/en14185933.
[71] A. Werner, N. Zimmermann, and J. Lentes, "Approach for a Holistic Predictive Maintenance Strategy by Incorporating a Digital Twin," *25th Int. Conf. Prod. Res. Manuf. Innov. Cyber Phys. Manuf. August 9-14 2019 Chic. Ill. USA*, vol. 39, pp. 1743–1751, Jan. 2019, doi: 10.1016/j.promfg.2020.01.265.
[72] W. Luo, T. Hu, Y. Ye, C. Zhang, and Y. Wei, "A hybrid predictive maintenance approach for CNC machine tool driven by Digital Twin," *Robot. Comput.-Integr. Manuf.*, vol. 65, p. 101974, Oct. 2020, doi: 10.1016/j.rcim.2020.101974.
[73] S. Mihai *et al.*, "A digital twin framework for predictive maintenance in industry 4.0," in *Conference Proceedings*, IEEE, 2021. [Online]. Available: https://repository.mdx.ac.uk/item/893yy
[74] E. D. P. Group, "EDP Open Data." 2022.
[75] T. Sreenuch, A. Tsourdos, and I. K. Jennions, "Distributed embedded condition monitoring systems based on OSA-CBM standard," *Comput. Stand. Interfaces*, vol. 35, no. 2, pp. 238–246, Feb. 2013, doi: 10.1016/j.csi.2012.10.002.
[76] R. Dinter, B. Tekinerdogan, and C. Catal, *Predictive maintenance using digital twins: A systematic literature review*. Information and Software Technology, 2022.
[77] *OPENMODELICA*. OpenModelica. Accessed: Feb. 12, 2023. [Online]. Available: https://openmodelica.org/
[78] *Simulink*. Mathworks. Accessed: Feb. 22, 2022. [Online]. Available: https://www.mathworks.com/products/simulink.html
[79] M. Paolanti, L. Romeo, A. Felicetti, A. Mancini, E. Frontoni, and J. Loncarski, "Machine Learning approach for Predictive Maintenance in Industry 4.0," in *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, Jul. 2018, pp. 1–6. doi: 10.1109/MESA.2018.8449150.
[80] N. Aziz, M. H. A. Abdullah, N. A. Osman, M. N. Musa, and E. A. P. Akhir, "Predictive Analytics for Oil and Gas Asset Maintenance Using XGBoost Algorithm," in *Proceedings of the 2nd International Conference on Emerging Technologies and Intelligent Systems*, M. A. Al-Sharafi, M. Al-Emran, M. N. Al-Kabi, and K. Shaalan, Eds., Cham: Springer International Publishing, 2023, pp. 108–117.

[81] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si, “A Data-Driven Design for Fault Detection of Wind Turbines Using Random Forests and XGboost,” *IEEE Access*, vol. 6, pp. 21020–21031, 2018, doi: 10.1109/ACCESS.2018.2818678.
[82] N. L. Vithi and C. Chibaya, “Advancements in Predictive Maintenance: A Bibliometric Review of Diagnostic Models Using Machine Learning Techniques,” *Analytics*, vol. 3, no. 4, pp. 493–507, 2024, doi: 10.3390/analytics3040028.
[83] Patrick Jahnke, “Machine Learning Approaches for Failure Type Detection and Predictive Maintenance,” Msc, Technische Universitat Darmstadt, Germany, 2015. Accessed: Aug. 16, 2023. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=3e2d7b31f691f42f141e9f9ecd455c4879292be9
[84] L. P. Silvestrin, M. Hoogendoorn, and G. Koole, “A Comparative Study of State-of-the-Art Machine Learning Algorithms for Predictive Maintenance,” in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2019, pp. 1812–1819. doi: 10.1109/SSCI44817.2019.9002780.
[85] S. Ayvaz and K. Alpay, “Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time,” *Expert Syst. Appl.*, vol. 173, p. 114598, Jul. 2021, doi: 10.1016/j.eswa.2021.114598.
[86] S. Arena, E. Florian, I. Zennaro, P. F. Orrù, and F. Sgarbossa, “A novel decision support system for managing predictive maintenance strategies based on machine learning approaches,” *Saf. Sci.*, vol. 146, p. 105529, Feb. 2022, doi: 10.1016/j.ssci.2021.105529.
[87] Lindy Williams, Arch Desai, Yi Guo, Shawn Sheng, and Caleb Phillips, “Machine Learning for Gearbox Fault Prediction by Using Both Scada and Modeled Data,” presented at the Drivetrain Reliability Collaborative Workshop, Feb. 16, 2017. Accessed: Oct. 24, 2024. [Online]. Available: https://www.nrel.gov/docs/fy21osti/79167.pdf
[88] B. Puruncajas, F. Castellani, Y. Vidal, and C. Tutivén, “Early Detection of Gearbox Failures in Wind Turbines Using Artificial Neural Networks and SCADA Data,” in *Advances in Italian Mechanism Science*, G. Quaglia, G. Boschetti, and G. Carbone, Eds., Cham: Springer Nature Switzerland, 2024, pp. 337–350.
[89] M. Bessaoudi, H. Habbouche, T. Benkedjouh, and A. Mesloub, “A hybrid approach for gearbox fault diagnosis based on deep learning techniques,” *Int. J. Adv. Manuf. Technol.*, vol. 133, no. 5, pp. 2861–2874, Jul. 2024, doi: 10.1007/s00170-024-13873-5.
[90] W. Zhang, J. Vatn, and A. Rasheed, “Gearbox pump failure prognostics in offshore wind turbine by an integrated data-driven model,” *Appl. Energy*, vol. 380, p. 124829, Feb. 2025, doi: 10.1016/j.apenergy.2024.124829.
[91] O. Serradilla, E. Zugasti, J. Rodriguez, and U. Zurutuza, “Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects,” *Appl. Intell.*, vol. 52, no. 10, pp. 10934–10964, Aug. 2022, doi: 10.1007/s10489-021-03004-y.
[92] A. Sharida, N. F. Kamal, H. Alnuweiri, S. Bayhan, and H. Abu-Rub, “Digital-Twin-Based Diagnosis and Tolerant Control of T-Type Three-Level Rectifiers,” *IEEE Open J. Ind. Electron. Soc.*, vol. 4, pp. 230–241, 2023, doi: 10.1109/OJIES.2023.3290169.
[93] M. Kisten, A. E. -S. Ezugwu, and M. O. Olusanya, “Explainable Artificial Intelligence Model for Predictive Maintenance in Smart Agricultural Facilities,” *IEEE Access*, vol. 12, pp. 24348–24367, 2024, doi: 10.1109/ACCESS.2024.3365586.
[94] M. A. Stevens, D. G. Lewicki, and R. F. Handschuh, “Concepts for Multi-Speed Rotorcraft Drive System - Status of Design and Testing at NASA GRC,” in *Proceedings of the American Helicopter Society 71st Annual Forum*, American Helicopter Society, 2015. [Online]. Available: https://ntrs.nasa.gov/api/citations/20150010716/downloads/20150010716.pdf

[95] R. J. Drago, “The Effect of Start-Up Load Conditions on Gearbox Performance and Life Failure Analysis, with Supporting Case Study,” *Gear Technol.*, 2009, [Online]. Available: https://www.geartechnology.com/ext/resources/issues/0609x/drago.pdf
[96] A. Haghshenas, A. Hasan, O. Osen, and E. T. Mikalsen, “Predictive digital twin for offshore wind farms,” *Energy Inform.*, vol. 6, no. 1, p. 1, Jan. 2023, doi: 10.1186/s42162-023-00257-4.
[97] A. Mitra, “STATISTICAL PROCESS CONTROL USING CONTROL CHARTS,” in *Encyclopedia of Production and Manufacturing Management*, P. M. Swamidass, Ed., New York, NY: Springer US, 2000, pp. 720–736. doi: 10.1007/1-4020-0612-8_919.
[98] M. Torzoni, M. Tezzele, S. Mariani, A. Manzoni, and K. E. Willcox, “A digital twin framework for civil engineering structures,” *Comput. Methods Appl. Mech. Eng.*, vol. 418, p. 116584, Jan. 2024, doi: 10.1016/j.cma.2023.116584.
[99] N. Tariq *et al.*, “The Security of Big Data in Fog-Enabled IoT Applications Including Blockchain: A Survey,” *Sensors*, vol. 19, no. 8, 2019, doi: 10.3390/s19081788.
[100] R. Basir *et al.*, “Fog Computing Enabling Industrial Internet of Things: State-of-the-Art and Research Challenges,” *Sensors*, vol. 19, no. 21, 2019, doi: 10.3390/s19214807.
[101] J. Pushpa and S. A. Kalyani, “Chapter Three - Using fog computing/edge computing to leverage Digital Twin,” in *Advances in Computers*, vol. 117, 1 vols., P. Raj and P. Evangeline, Eds., Elsevier, 2020, pp. 51–77. doi: 10.1016/bs.adcom.2019.09.003.
[102] Q. Qi and F. Tao, “A Smart Manufacturing Service System Based on Edge Computing, Fog Computing, and Cloud Computing,” *IEEE Access*, vol. 7, pp. 86769–86777, 2019, doi: 10.1109/ACCESS.2019.2923610.
[103] M. Aqib, D. Kumar, and S. Tripathi, “Machine Learning for Fog Computing: Review, Opportunities and a Fog Application Classifier and Scheduler,” *Wirel. Pers. Commun.*, vol. 129, no. 2, pp. 853–880, Mar. 2023, doi: 10.1007/s11277-022-10160-y.
[104] G. K. Walia, M. Kumar, and S. S. Gill, “AI-Empowered Fog/Edge Resource Management for IoT Applications: A Comprehensive Review, Research Challenges, and Future Perspectives,” *IEEE Commun. Surv. Tutor.*, vol. 26, no. 1, pp. 619–669, Firstquarter 2024, doi: 10.1109/COMST.2023.3338015.
[105] S. Iftikhar *et al.*, “AI-based fog and edge computing: A systematic review, taxonomy and future directions,” *Internet Things*, vol. 21, p. 100674, Apr. 2023, doi: 10.1016/j.iot.2022.100674.
[106] E. Huaranga-Junco, S. González-Gerpe, M. Castillo-Cara, A. Cimmino, and R. García-Castro, “From cloud and fog computing to federated-fog computing: A comparative analysis of computational resources in real-time IoT applications based on semantic interoperability,” *Future Gener. Comput. Syst.*, vol. 159, pp. 134–150, Oct. 2024, doi: 10.1016/j.future.2024.05.001.
[107] J. Park *et al.*, “Distilling on-device intelligence at the network edge,” *ArXiv Prepr. ArXiv190805895*, 2019.
[108] Y. I. Alzoubi, V. H. Osmanaj, A. Jaradat, and A. Al-Ahmad, “Fog computing security and privacy for the Internet of Thing applications: State-of-the-art,” *Secur. Priv.*, vol. 4, no. 2, p. e145, 2021.
[109] J. Ni, K. Zhang, X. Lin, and X. Shen, “Securing Fog Computing for Internet of Things Applications: Challenges and Solutions,” *IEEE Commun. Surv. Tutor.*, vol. 20, no. 1, pp. 601–628, Firstquarter 2018, doi: 10.1109/COMST.2017.2762345.
[110] M. Mukherjee *et al.*, “Security and Privacy in Fog Computing: Challenges,” *IEEE Access*, vol. 5, pp. 19293–19304, 2017, doi: 10.1109/ACCESS.2017.2749422.
[111] A. M. Alwakeel, “An Overview of Fog Computing and Edge Computing Security and Privacy Issues,” *Sensors*, vol. 21, no. 24, 2021, doi: 10.3390/s21248226.

[112] J. Sengupta, S. Ruj, and S. D. Bit, “A Secure Fog-Based Architecture for Industrial Internet of Things and Industry 4.0,” *IEEE Trans. Ind. Inform.*, vol. 17, no. 4, pp. 2316–2324, Apr. 2021, doi: 10.1109/TII.2020.2998105.
[113] W. Chia Chuan, S. Ul Arfeen Laghari, S. Manickam, E. Ashraf, and S. Karuppayah, “Challenges and Opportunities in Fog Computing Scheduling: A Literature Review,” *IEEE Access*, vol. 13, pp. 14702–14726, 2025, doi: 10.1109/ACCESS.2024.3525261.
[114] D. Alsadie, “A Comprehensive Review of AI Techniques for Resource Management in Fog Computing: Trends, Challenges, and Future Directions,” *IEEE Access*, vol. 12, pp. 118007–118059, 2024, doi: 10.1109/ACCESS.2024.3447097.
[115] I. Martinez, A. S. Hafid, and A. Jarray, “Design, Resource Management, and Evaluation of Fog Computing Systems: A Survey,” *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2494–2516, Feb. 2021, doi: 10.1109/JIOT.2020.3022699.
[116] A. A. Mutlag *et al.*, “A new fog computing resource management (FRM) model based on hybrid load balancing and scheduling for critical healthcare applications,” *Phys. Commun.*, vol. 59, p. 102109, Aug. 2023, doi: 10.1016/j.phycom.2023.102109.
[117] A. Rahimikhanghah, M. Tajkey, B. Rezazadeh, and A. M. Rahmani, “Resource scheduling methods in cloud and fog computing environments: a systematic literature review,” *Clust. Comput.*, vol. 25, no. 2, pp. 911–945, Apr. 2022, doi: 10.1007/s10586-021-03467-1.
[118] S. Verma, P. Joshi, M. S. Pandey, D. Baloni, A. Malav, and S. K. Mahariya, “Fog and edge computing: State-of-art, application and challenges in IoT,” *AIP Conf. Proc.*, vol. 2771, no. 1, p. 020049, Sep. 2023, doi: 10.1063/5.0152441.
[119] R. Mahmud, F. L. Koch, and R. Buyya, “Cloud-Fog Interoperability in IoT-enabled Healthcare Solutions,” in *Proceedings of the 19th International Conference on Distributed Computing and Networking*, in ICDCN ’18. New York, NY, USA: Association for Computing Machinery, 2018. doi: 10.1145/3154273.3154347.
[120] V. Moysiadis, P. Sarigiannidis, and I. Moscholios, “Towards Distributed Data Management in Fog Computing,” *Wirel. Commun. Mob. Comput.*, vol. 2018, no. 1, p. 7597686, Jan. 2018, doi: 10.1155/2018/7597686.
[121] A. A. Sadri, A. M. Rahmani, M. Saberikamarposhti, and M. Hosseinzadeh, “Fog data management: A vision, challenges, and future directions,” *J. Netw. Comput. Appl.*, vol. 174, p. 102882, Jan. 2021, doi: 10.1016/j.jnca.2020.102882.
[122] C. M. Rebello, J. Jäschkea, and I. B. Nogueira, “Digital Twin Framework for Optimal and Autonomous Decision-Making in Cyber-Physical Systems: Enhancing Reliability and Adaptability in the Oil and Gas Industry,” *ArXiv Prepr. ArXiv231112755*, 2023.
[123] I. Ud Din, K. A. Awan, A. Almogren, and J. J. P. C. Rodrigues, “Resilient Production Control Using Digital Twins in the Industrial Internet of Things,” *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3204–3211, Feb. 2024, doi: 10.1109/TCE.2023.3325209.
[124] A. Sharma, E. Kosasih, J. Zhang, A. Brintrup, and A. Calinescu, “Digital Twins: State of the art theory and practice, challenges, and open research questions,” *J. Ind. Inf. Integr.*, vol. 30, p. 100383, Nov. 2022, doi: 10.1016/j.jii.2022.100383.
[125] Docker, “Docker.” Accessed: Apr. 20, 2023. [Online]. Available: https://www.docker.com/
[126] S. Ravuru and Y. Gudeta, “Accelerating Innovation at JetBlue Using Databricks,” *Databricks Blog*, 2023, [Online]. Available: https://www.databricks.com/blog/accelerating-innovation-jetblue-using-databricks
[127] S. Steffen, “Driving into the Future of Electric Transportation,” *Databricks Blog*, 2024, [Online]. Available: https://www.databricks.com/blog/driving-future-electric-transportation

[128] Databricks, “Your Data. Your AI: Own Them All on the New Data Intelligence Platform.” 2024. [Online]. Available: https://www.databricks.com/
[129] M. A. Ahmed and Y. C. Kim, “Hierarchical Communication Network Architectures for Offshore Wind Power Farms,” in *2014 International Symposium on Computer, Consumer and Control*, Jun. 2014, pp. 299–303. doi: 10.1109/IS3C.2014.85.
[130] M. A. Ahmed, Yong Cheol Kang, and Young-Chon Kim, “Modeling and simulation of ICT network architecture for cyber-physical wind energy system,” in *2015 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, Aug. 2015, pp. 1–6. doi: 10.1109/SEGE.2015.7324601.
[131] M. A. Ahmed and C.-H. Kim, “Communication Architecture for Grid Integration of Cyber Physical Wind Energy Systems,” *Appl. Sci.*, vol. 7, no. 10, 2017, doi: 10.3390/app7101034.
[132] U.S. Department of Energy, “New Tool Protects Bats While Increasing Energy Production,” *Wind Energy Technol. Off.*, 2021, [Online]. Available: https://www.energy.gov/eere/wind/articles/new-tool-protects-bats-while-increasing-energy-production
[133] R. Rodriguez, G. Vallejo, D. Saywers, J. Quillen, and K. Denman, “Balancing Bat Preservation and Wind Energy Production with EchoSense®,” *Nat. Power Publ.*, 2022, [Online]. Available: https://www.naturalpower.com/uk/insight/balancing-bat-preservation-and-wind-energy-production-with-echosense
[134] Vattenfall, “Ray Wind Farm.” 2024. [Online]. Available: https://group.vattenfall.com/uk/what-we-do/our-projects/ray-wind-farm