Integrated Condition-Based Maintenance Using Unsupervised Health Indicators for Offshore Wind Turbines

Master's Thesis Report

MSc: Sustainable Energy Technology

Sebastiaan Geerts

Delft University of Technology

TU Delft

Integrated Condition-Based Maintenance Using Unsupervised Health Indicators for Offshore Wind Turbines

Master's Thesis Report

by

Sebastiaan Geerts

Delft, August 2025

To obtain the degree of Master of Science in Sustainable Energy Technology
at Delft University of Technology
To be defended publicly on 26 August, 2025

Research carried out at:
Wind Energy Group, Faculty of Aerospace Engineering, Delft University of Technology

Student number:	4923057
Project duration:	November 11, 2024 – August 26, 2025

Thesis committee

Dr. Ir. W. A. A. M. Bierbooms	Wind Energy, Aerospace Engineering, TU Delft
Dr. Ir. X. L. Jiang	Transport Engineering & Logistics, Mechanical Engineering, TU Delft
Dr. D. Zappalá	Wind Energy, Aerospace Engineering, TU Delft

Advisor

Dr. M. A. Mitici	Data Intensive Systems, Science, Utrecht University

An electronic version of this thesis is available at `http://repository.tudelft.nl`

TU Delft

Preface

This thesis represents the culmination of my Master's studies in Sustainable Energy Technology at Delft University of Technology. It explores how deep learning techniques can be integrated into a complete Prognostics and Health Management cycle, spanning both condition monitoring and maintenance scheduling. In particular, the thesis explores a new dataset and focusses on rotor bearing failures in this dataset.

This topic appealed to me because of its multidisciplinary nature. It is grounded in engineering, but enabled me to strengten my programming skills. Although I was initially nervous about my coding abilities, the greatest challenges actually lay in undertaking my first fully independent research project. Making sound academic choices required careful consideration, but through that process I learnt a lot. This research feels like a fitting conclusion to my time at Delft.

I owe my deepest gratitude to my supervisors, Donatella Zapalá and Mihaela Mitici. I could not have wished for better guidance. They pushed me to get the most out of the project and continued to encourage me when progress was slow. I want to thank them for their flexibility to explore the project's direction. Their combined expertise resulted in sharp insights and suggestions that greatly improved the quality of this research.

In addition, I would like to thank the Ph.D. candidates Ali Milani and Marco Borsotti for generously making time to meet with me and share their expertise, which helped me make some key decisions. In addition, I extend my gratitude to Christian Gück and Cyriana Roelofs from the CARE project for providing the dataset and patiently answering my questions. Finally, I feel truly blessed to have a great social network of my family, friends, roommates, and my girlfriend who supported me throughout the project.

This report is intended for researchers, students, and wind turbine operators interested in prognostics and health management. I wish you a pleasant reading experience.

Sebastiaan Geerts
Delft, August 2025

Abstract

Wind energy plays a pivotal role in the energy transition, however, a major obstacle for offshore wind farms remains the high Operations and Maintenance (O&M) costs, which can be up to 35% of the levelized cost of energy (LCOE). Traditional maintenance strategies often result in excessive downtime, increased costs, or premature component replacements. To counter these problems, condition-based maintenance (CBM) uses real-time data to estimate machine health and guide maintenance decisions. Although most CBM studies address either health assessment or maintenance optimisation in isolation, this thesis proposes an end-to-end CBM framework that derives health indicators (HIs) and integrates them into a multiple-threshold CBM framework.

The framework is validated on rotor bearing failures from the CARE data set, which supplies real-world Supervisory Control and Data Acquisition (SCADA) measurements. Rotor bearings, despite their relatively low failure frequencies, cause high downtime and exhibit a prolonged degradation pattern. HIs are extracted with a long- and short-term memory (LSTM) autoencoder (AE) trained solely on healthy turbines to learn normal behaviour. On avarage, the resulting HIs identify anomalies 195 days before failure.

These HIs are fed into a CBM strategy that prioritises timely, minor interventions over costly last-minute replacements. Compared to a purely corrective replacement policy, the strategy reduces annual maintenance expenditure on rotor bearings by an average of 62.5%, mainly by prolonging the useful life of the components and reducing downtime. This thesis therefore demonstrates the value of rotor-bearing HIs, derived from widely available SCADA data, for wind turbine maintenance.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
AE	Autoencoder
ANN	Artificial Neural Network
CBM	Condition-Based Maintenance
CMS	Condition Monitoring System
CNN	Convolutional Neural Network
CTV	Crew Transfer Vessels
HI	Health Indicator
LCOE	Levelized Cost of Energy
LSTM	Long Short-Term Memory
MILP	Mixed-Integer Linear Programming
MSE	Mean Squared Error
NN	Neural Network
O&M	Operation and Maintenance
PdM	Predictive Maintenance
PHM	Prognostics and Health Management
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
SCADA	Supervisory Control and Data Acquisition
SOV	Service Operation Vessel

Symbols

Symbol	Definition	Unit
T	Sliding-window length (time steps)	–
F	Number of SCADA features	–
N	Batch size during training	–
(H)	Hidden dimension from the LSTM layer	–
$\mathbf{x}_c^t$	Input vector at time t and case c	–
$\mathbf{x}_c^t$	Reconstructed vector at time t and case c	–
$\hat{\mathbf{X}}_\mathbf{c}$	Reconstructed window for case c $\{\mathbf{x}_c^1, \mathbf{x}_c^2, \ldots, \mathbf{x}_c^T\}$	–
$\mathbf{X}_\mathbf{c}$	Input window for case c $\{\widehat{\mathbf{x}}_c^1, \widehat{\mathbf{x}}_c^2, \ldots, \widehat{\mathbf{x}}_c^T\}$	–
$\vec{z}$	Latent (bottleneck) vector	–

Symbol	Definition	Unit
$\mathbf{Z_c}$	Latent space window for case c $\{\mathbf{z}_c^1, \mathbf{z}_c^2, \ldots, \mathbf{z}_c^T\}$	–
G	Dimension of the latent space ($\mathbf{Z}$)	–
$\mathbf{O}_c$	Operating-condition window for case c: $\{\mathbf{o}_c^1, \mathbf{o}_c^2, \ldots, \mathbf{o}_c^T\}$	–
$\mathbf{o}_c^t$	Operating conditons vector for time t and case d	–
O	Dimension of operating conditions ($\mathbf{O}$)	–
$\mathcal{L}$	Loss function (MSE)	–
$\mathcal{L}_d^{(c,s)}$	Mean reconstruction error of sensor s on day d (case c)	–
λ_d	Health indicator on day d	–
k	Threshold multiplier in Chebyshev's inequality	–
n	Number of consecutive days the threshold must be crossed in Chebyshev's inequality	–
$\mathcal{M}$	Monotonicity metric of the HI	–
$\mathcal{T}$	Trendability metric of the HI	–
T_{rotor1}	Rotor bearing temperature sensor 1	°C
T_{rotor2}	Rotor bearing temperature sensor 2	°C
P_{active}	Active power output of the turbine	kW
T_{amb}	Ambient temperature near the turbine	°C
v_{wind}	Wind speed measured at the nacelle	m/s
TH_{minor}	Threshold for minor repair	–
TH_{major}	Threshold for major repair	–
TH_{repl}	Threshold for preventive replacement	–
L	Lead time	days
a	Maintenance-action type (minor / major / prev / corr)	–
OC_i	Operational cost in cycle i	€
C_{tot}	Total cost of a maintenance action	€
C_{mat}	Material cost	€
C_{tech}	Technician (labour) cost	€
C_{dt}	Downtime cost	€
C_{trans}	Vessel / transport cost	€
C_{minor}	Material costs for minor repair	€
C_{major}	Material costs for major repair	€
C_{prev}	Material costs for preventive replacement	€
C_{corr}	Material costs for corrective replacement	€
N_{tech}	Number of technicians required	–
R_{hour}	Hourly wage per technician	€/h
T_{repair}	On-site repair duration	h
T_{maint}	Total maintenance time (incl. travel and jack-up and down)	h
T_{dt}	Total downtime duration	h
P_{avg}	Average turbine power output	MW
C_{el}	Electricity price	€/MWh
C_{mob}	Vessel mobilisation cost	€
R_{day}	Daily vessel charter rate	€/day

Symbol	Definition	Unit
D_{trav}	One-way travel distance to turbine	km
F	Fuel consumption per km	–
N_{days}	Number of operable days in planning horizon	days
$\mathcal{L}_0$	Lifetime at the moment of the first maintenance action	days
life_ext(a)	Fractional lifetime extension per maintenance action	–
ext_{max}	Maximum allowed total lifetime extension	–
N_{repairs}	Number of identical maintenance actions applied	–

1
Introduction

In order to reduce the pace of global warming, the world needs to transition away from relying on fossil fuels for energy production. Governments worldwide have committed to ambitious climate targets, such as the 2015 Paris Agreement's commitment to limit the increase in the global average temperature to below 2 degrees Celsius [1], the European Commission's REPowerEU plan to end reliance on Russian fossil fuels [2], and the Net Zero Strategy, where the governments worldwide has committed to achieving net zero carbon emissions by 2050 [3]. Renewable energy sources are becoming increasingly important not only for decarbonisation but also for energy independence, especially in Europe, as became evident after the energy crisis following the war in Ukraine [4]. Among various renewable energy sources, wind power stands out as one of the most significant due to its relatively high technological readiness level, abundant availability, and low environmental impact [5]. According to the Global Wind Energy Council Global Wind Report for 2023, the total worldwide wind energy capacity has reached 906 GW, and an additional 680 GW is expected to be installed by 2027 [6]. This capacity corresponds to over 300,000 wind turbines in operation worldwide, both onshore and offshore [7].

However, recent offshore wind developments have faced financial challenges, including the postponement of two Dutch tenders [8] and the cancellation of a 3 GW tender in Denmark [9]. Operation and maintenance (O&M) costs represent a significant portion of the total lifecycle expenses in wind energy, particularly for offshore wind farms. These costs are influenced not only by the frequency of failures, exacerbated by harsh environmental conditions, but also by the logistical challenges associated with accessing turbines in remote offshore locations. In some offshore developments, O&M costs have been reported to account for up to 35% of levelized cost of energy (LCOE) [10]. In absolute terms, the average O&M cost for fixed-bottom offshore wind projects has been reported at $30.3 per MWh, whereas those of onshore projects are approximately $12.1 per MWh [11]. In particular, maintenance costs tend to have a greater influence on controlling the levelised cost of energy than general operating costs [5]. As a result of the high O&M cost, wind farm operators are under growing pressure to reduce these expenses [12].

Currently, offshore wind O&M strategies are largely centred around corrective and periodic approaches, though some predictive tools are starting to see implementation [12], [13]. Corrective maintenance, which involves replacing components after they fail, can lead to long downtimes and high repair costs due to consequential damage. Periodic maintenance replaces or inspects parts at set intervals, potentially wasting useful life or failing to prevent unexpected failures. Condition-based maintenance (CBM) uses real-time data to estimate current and future machine health and integrates these insights

into maintenance planning to reduce the number of incisions and corrective maintenance actions needed [14]. Empirical studies confirm that CBM can significantly reduce downtime and costs while extending component lifetime [15]–[18].

Advancing offshore wind through smarter O&M also has a significant socio-economic impact, touching upon six UN Sustainable Development Goals (SDGs). The maintenance of offshore turbines exposes technicians to considerable health and safety risks, with a documented increase in serious injuries and fatalities in recent years [19]. CBM can significantly improve the health and safety of maintenance technicians by reducing the need for unplanned or high-risk interventions [20], directly impacting *SDG 3 (Good Health & Well-being)*. The CBM extends the useful life of the components through timely interventions and replaces only when their useful life is exhausted [16], supporting *SDG 12 (Responsible Consumption & Production)*. CBM also reduces unplanned downtime [21], which, together with the extension of the useful life of the component, further reduces the LCOE and therefore aligning with*SDG 7 (Affordable & Clean Energy)*. Cheaper and more competitive renewable energy accelerates the climate transition and promotes economic growth through cheap and abundant energy, reinforcing *SDG 8 (Decent Work & Economic Growth)* and *SDG 13 (Climate Action)*. CBM is part of the automated, data-driven future of energy, aligning with *SDG 9 (Industry, Innovation & Infrastructure)*.

1.1. Background and motivation

CBM has been the subject of extensive research, but despite advances in academic research, CBM is still far from being fully implemented in the wind industry [12]. A possible barrier is the lack of integrated, practical CBM studies. Most CBM research only considers diagnostics or prognostics and without linking predictions with maintenance decisions [22]–[30], and does not quantify their findings or neglect practical difficulties with their prediction. Or, research on maintenance typically assumes overly simple degradation paths [15], [16], [31]–[33], despite the fact that such precise degradation profiles are rarely available in practice. Evidence from the aerospace sector shows that fully integrated, data-driven approaches can unlock substantial cost savings [34]–[37].

However, such integrated CBM approaches remain rare in the wind sector, as discussed in Subsection 2.4.1. These papers typically rely on synthetic or simulated condition monitoring system (CMS) data, such as high-frequency vibration signals. In recent years, there has been a notable shift in the wind energy sector from reliance on vibration data to a increased use of supervisory control and data acquisition (SCADA) data for condition monitoring [38], [39]. Most large-scale wind turbines already provide SCADA data as standard [40]. Although SCADA was not specifically designed for condition monitoring, it records low-frequency operation parameters that can support this analysis [39], [41]. Because operators can use the existing SCADA infrastructure, extracting relevant information promises rapid deployment and modest set-up costs [41].

The shift to SCADA is largely driven by the use of machine learning techniques, which enable advanced fault detection and prognostics [38]. The sheer volume of data from SCADA systems is particularly useful for machine learning due to its efficiency and practicality in the modelling of complex nonlinear systems [42]. In particular, deep learning techniques have outperformed traditional techniques in recent years, mainly due to their superior performance [38]. Advance deep learning models such as long- and short-term memory (LSTM) are also well suited to process varying-length time series [43], such as in SCADA data from wind turbines. However, the success of deep learning models depends on the availability of substantial run-to-failure datasets [11]. Acquiring data sets is a key

challenge in the wind industry due to the limited public availability of real-world data, especially those that include detailed information about anomalies or component faults [44]. This scarcity is mainly due to concerns about data security and privacy in the wind industry [38].

The recently published Coverage, Accuracy, Reliability and Earliness (CARE) to Compare dataset [44] is a notable attempt to close this gap. This dataset is a comprehensive and anonymised collection of SCADA time series and failure logs from three offshore wind farms. From this dataset, this study investigates the rotor bearing failures. Mechanical components are responsible for the highest downtime in wind turbines, despite representing a relatively small number of failures [45]–[47]. Among these, bearings are one of the most critical subcomponents [48]. Rotor bearings are part of the low speed system and are known for their low rotational speed and high loads make them less sensitive to vibrations caused by defects or damage [49]. As a result, many techniques that perform well in high-speed and moderately loaded conditions have proven unreliable for rotor bearing fault detection [49], making them a suited component for research with SCADA data such as shown by [41].

In real-world datasets, such as CARE, run-to-failure instances on the same component are scarce. Components typically have a long lifetime or are replaced preventively, and although failures can still occur (both in older and newer turbines), data from these events is often unavailable or inaccessible. Limited run-to-failure cases make training supervised models requiring labelled data difficult. To address the lack of available data on failures, researchers apply unsupervised learning methods, particularly autoencoders (AEs), which are commonly used for condition motoring [27]–[30]. An AE encodes the original sensor inputs into a compressed latent space and then reconstructs them. Unsupervised AE typically training exclusively on healthy data, which is data recorded while the system operates normally. This enables a model to learn normal behaviour. By learning the normal behaviour, the model can estimate a current state in the form of a health indicator (HI). For example, using the reconstruction error between the predicted and real value [28] or the latent space [50]. Since the model relies only on healthy data, no run-to-failure cases are required for training, only for validation.

Unsupervised HI methods have the additional benefit of not assuming degradation patterns. Different failure modes are expected to show a number of possible degradation patterns [41]. Supervised HI models often rely on a predefined degradation shape, such as a linear decline [51] or a quadratic curve [52]. These assumptions oversimplify the problem and do not capture the stochastic nature intrinsic to most complex degradation processes [27]. In contrast, unsupervised HI models can adapt to the underlying failure trajectory without requiring any prior assumptions about its form. This allows for a more realistic and generalisable representation of varying degradation behaviours across asset condition monitoring. Despite these advantages, unsupervised deep learning approaches are a scarce domain in bearing HI construction [27], and are rarely included in integrated wind turbine frameworks.

In summary, the literature lacks integrated CBM frameworks that show the impact of advanced unsupervised rotor bearing HI models from real-world SCADA data. This thesis aims to address this gap by developing an end-to-end framework for CBM of offshore wind turbine rotor bearings, where an HI is developed and integrated into offshore multiple-threshold maintenance. The HI are made using a sefisticated LSTM AE that learns normal behaviour from healthy SCADA data, and uses operating conditions informatively in the LSTM. The HIs are fed a multithreshold maintenance model that favours early, low-impact interventions over costly preventive replacements. Because training is required only healthy data from standard deployed SCADA systems, the method is inexpensive to deploy and broadly applicable, even when failure records are sparse.

1.2. Research questions

The main research question of this thesis is as follows:

How can rotor bearing health indicators for wind turbines be developed from SCADA data and integrated into a condition-based maintenance strategy?

This question leads to the following subquestions:

1. *Which method can effectively derive rotor bearing health indicators and detect anomalies from SCADA data when run-to-failure cases are scarce?*
2. *What evaluation metrics are appropriate for the health indicators, and how well do the derived health indicators perform on the CARE dataset?*
3. *How can a practical condition-based maintenance framework for offshore wind turbines be developed based on the health indicators?*
4. *How does the proposed framework compare to corrective replacement, preventive replacement and perfectly timed maintenance strategies?*

1.3. Report outline

The remainder of the report is structured as follows. First a review of the relevant literature supporting the development of CBM is presented in Chapter 2. Chapter 3 introduces the theoretical foundations on deep learning needed to develop the method. The progressed methodology is presented in Chapter 4. Chapter 5 introduces the CARE dataset used in this thesis. The corresponding results and discussion of the CARE dataset and the method are presented in Chapter 6. Finally, Chapter 7 concludes the report and provides recommendations for future research.

The roadmap for the CBM framework in this thesis, from raw SCADA data to maintenance scheduling results, is shown in Figure 1.1. An in-depth flow diagram of the methodology used in this thesis can be found in Figure 4.1

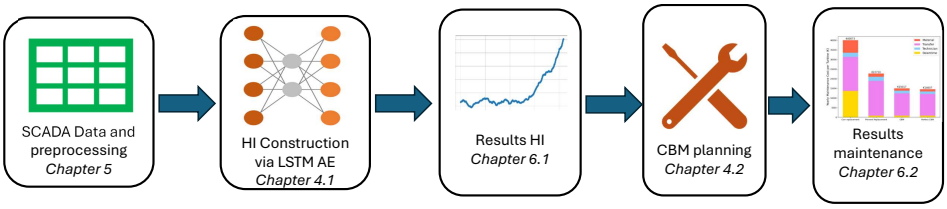


Figure 1.1: the roadmap for CBM, from raw SCADA data to maintenance schedule

2
Literature Review

This chapter presents a review of the relevant literature supporting the development of CBM. CBM is part of Prognostics and Health Management (PHM), a comprehensive management cycle that aims to ensure reliable operation and minimise unplanned downtime [53]. PHM consists of three main steps: observation, analysis, and action [48], [54]. For wind turbines, these steps are equivalent to condition monitoring, fault detection, or prognostics, and maintenance actions. The purpose of this review of the literature is to understand how the three axes are addressed in current state-of-the-art research and, more specifically, what papers provide an end-to-end CBM framework.

Multiple literature reviews have explored CBM in the wind energy sector, focusing either on prognostic and condition motoring [42], [55]–[58] or on maintenance strategies [5], [12], [59]–[61]. Fox et al. [11] considers both aspects of CBM in their review, but do not consider studies that integrate both aspects in one study. The literature review of this thesis is different from the existing literature reviews, as it focusses on literature that integrates both aspects of CBM into their research.

This chapter is built up as follows. First, condition monitoring is introduced in Section 2.1, along with the identification of critical subcomponents of wind turbines and a specific focus on the modes of failure of the rotor bearing. In Section 2.2 the different maintenance strategies are discussed. In Section 2.3 condition monitoring is considered, in particular the different methods used. Integrated approaches within the wind industry and the aircraft industry are examined in Section 2.4.

2.1. Condition monitoring and critical components

Condition monitoring provides the data stream that forms the basis for a CBM strategy. Its purpose is to track the evolving degradation of critical turbine subsystems. This section establishes the fundamentals of condition monitoring and identifies critical components and failure modes of the rotor bearing. Subsection 2.1.1 contrasts the two main data sources available in wind energy: dedicated CMS and SCADA systems. Subsection 2.1.2 then reviews published failure statistics to assess the criticality of individual subcomponents, and delves into the fialure modes for rotor bearings.

2.1.1. Monitoring systems

Condition monitoring refers to the process of continuously evaluating the operational state of wind turbine components, starting with the implementation of a monitoring system. Generally, two types of monitoring systems are used for the diagnosis and prognosis of wind turbines: CMS and Performance Monitoring Systems, in the form of SCADA systems [48].

CMSs are installed as dedicated accessories designed by a number of manufacturers to record indicators such as vehicle vibration, oil quality, and component temperatures [40]. These indicators can be processed and analysed to derive meaningful degradation information. However, these systems come at a significant cost of typically exceeding €12,000 per turbine, resulting in multimillion dollar investments for large wind farms [21].

In contrast, SCADA systems are standard in large-scale wind turbines and primarily designed for operational control. These systems use hundreds of sensors that typically record at 1 Hz, and process this data into statistical summaries (e.g. mean, standard deviation, minimum, and maximum) at 10-minute intervals [57]. The monitored parameters commonly include wind speed, power output, rotor speed, and the temperature of critical components.

In recent years, there has been a noticeable shift in the wind energy sector from reliance on vibration data to increased use of SCADA data for condition monitoring. This shift is largely driven by the integration of SCADA data with machine learning techniques, which enable advanced fault detection and prognostics [38].

The strengths and limitations of using CMS and PMS data sources for PHM are summarised in Table 2.1.

Table 2.1: Advantages and disadvantages of SCADA and CMS [48], [62]

Monitoring system	Strengths	Weaknesses
SCADA	+ Readily accessible data + Can detect abnormal trends	- Usually records 10 minute average data, so spectral analysis cannot be applied to interpreting them - Not able to identify and isolate specific faults - Faults are harder to detect and prone to false alarms due to reliance on (fluctuating) operational conditions
CMS	+ Able to identify and isolate specific faults + Captures high-frequency data, allowing for spectral analysis	- Costly due to instrumentation

2.1.2. Reliability of critical components

Ensuring that the CMS can reliably detect and identify a failure is a critical prerequisite for developing a prognostic model [55]. To understand the common types of wind turbine failures, several surveys [45]–[47], have been carried out. The challenges in comparing databases are due to the lack of standardised methodologies for data collection on failures within the wind industry [46]. Furthermore, variations in the characteristics of the data set, such as the volume of data, turbine power ratings, location, weather conditions, and accessibility, contribute to inconsistencies [45]. However, critical sub-assemblies remain quite consistent across different data sources [45].

For example, Figure 2.1 compares three key European databases for onshore wind turbines: WMEP, LWK, and Elforsk/Vindstat (Swedish Survey). Although these datasets are dated (1993–2006), they are among the most comprehensive publicly available and align closely with more recent surveys [46]. A common characteristic is that the highest failure rate subassemblies in onshore turbines do not necessarily result in the most downtime. Specifically, mechanical subassemblies, such as gearboxes, have relatively low failure rates, but cause significant downtime due to complex logistic and technical repairs [46]. In contrast, electrical components fail more frequently, but are repaired more quickly.

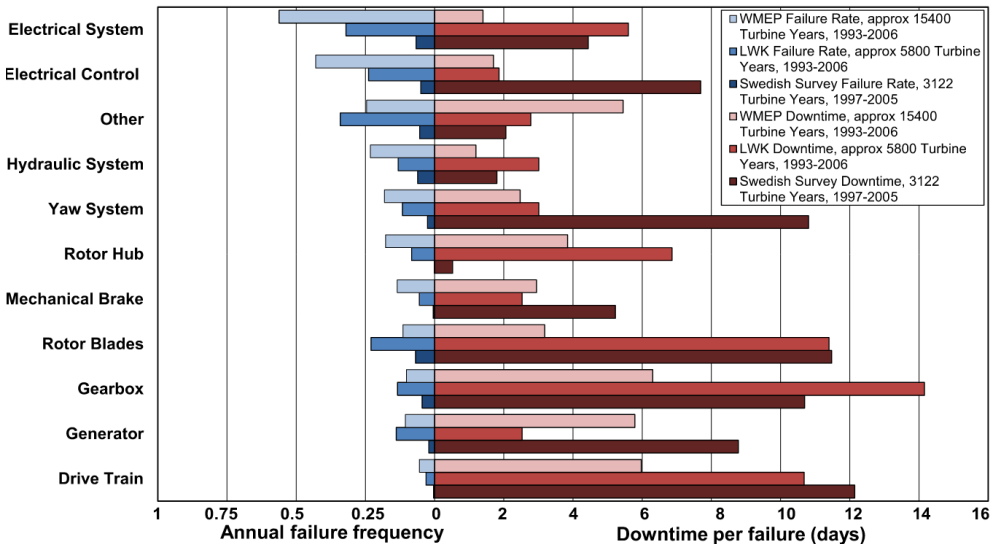


Figure 2.1: Failure rates and downtime for onshore wind turbines [46]

This trend differs slightly for offshore turbines due to harsher operating conditions and the logistical challenges of performing even simple repairs. Figure 2.2 compares the failure rates and downtime distributions of offshore and onshore wind turbine. Onshore turbines experience the highest failure rates in electrical and control systems, blades and hub, pitch systems, and generators. For offshore wind turbine this is similar, but the pitch system emerges as the most critical subassembly in terms of failure rates. Higher failure rates for gearboxes, generators, and drivetrains in offshore turbines may be due to limited databases (only four), as they include the Round 1 UK database, which reports unusually high gearbox failure rates and related issues [46]. With regard to downtime, gearboxes, generators, and blades/hub contribute the most for both onshore and offshore turbines.

In absolute terms, offshore failure rates are higher than onshore, partly explained by harsher offshore

operating conditions, including stronger winds and corrosive saltwater exposure [45]. Furthermore, the downtime per stop is approximately double offshore compared to onshore installations [45], most likely a result of complex offshore maintenance [46]. The average stop rate is slightly higher offshore [45].

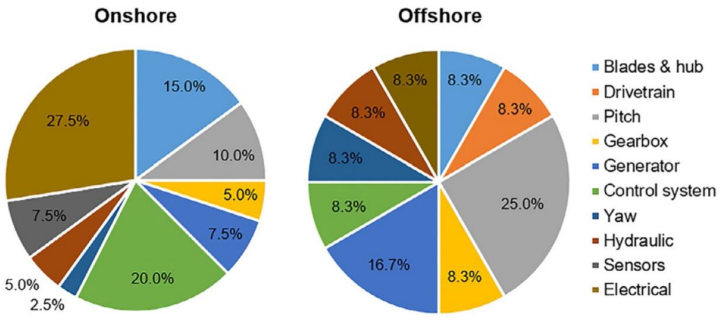


Figure 2.2: Comparison of onshore and offshore failure rates among wind turbine components [45]

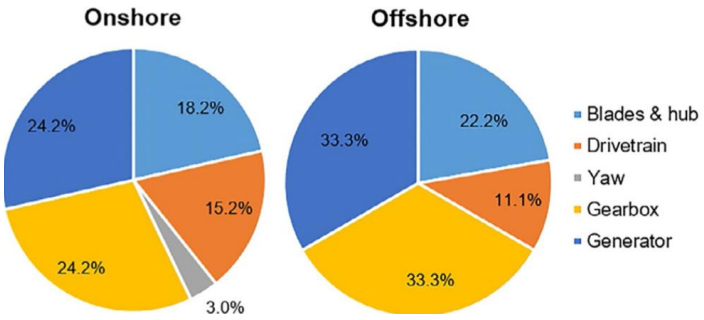
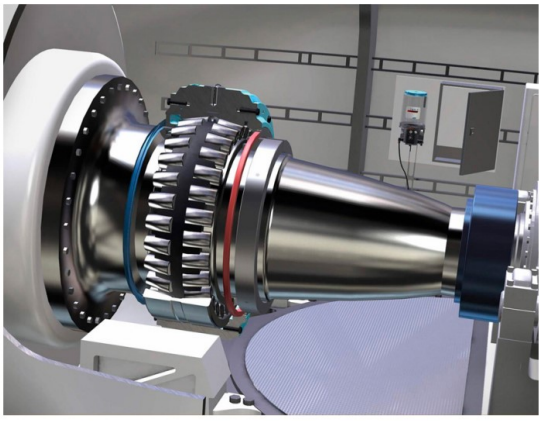


Figure 2.3: Comparison of onshore and offshore downtime among wind turbine components [45]

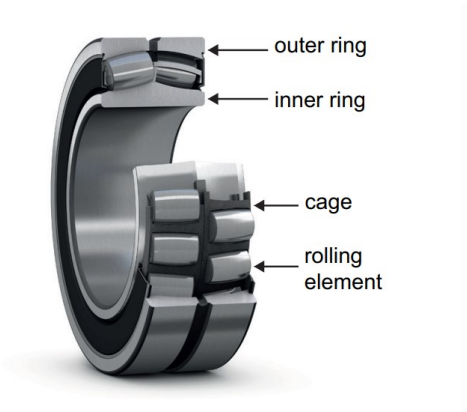
Main bearing faults

The main (rotor) bearing supports the low-speed main shaft and is shown in Figure 2.4. Figure 2.4a locates the bearing within the nacelle and presents a cutaway view of its housing, while Figure 2.4b schematically identifies the key elements: outer ring, inner ring, cage, and rolling elements [41]. Although the figure shows a spherical roller design, alternative drivetrain configurations also exist, including single and double bearings, integrated gearbox bearings, and floating arrangements in geared turbines [49]. The component can be damaged in a variety of ways, which will be discussed here. The objective of discussing this is to show that there are different patterns to predict a fault in this component, but rather a large number of possible patterns.

The bearing company SKF classified failure modes according to the ISO 15243 standard into the following failure modes: (i) fatigue, (ii) wear, (iii) corrosion, (iv) electrical erosion, (v) plastic deformation and (vi) fracture and cracking [63]. Understanding these failure modes is essential for effective condition monitoring, as in many cases SCADA-based temperature data provide the earliest signature of damage, while vibration is more effective for others. Below each of the failure modes are discussed [41]:



(a) Main bearing located inside a wind turbine



(b) Schematic of main roller bearing, with the different components

Figure 2.4: Spherical main bearing, from the company SKF [41]

(i) Fatigue occurs in two forms: subsurface initiated and surface initiated. Subsurface fatigue develops beneath the rolling elements as a result of repeated rolling contact stresses, forming cracks that propagate to the surface. Surface fatigue arises from inadequate lubrication, which damages surface asperities and initiates micro-pitting.

(ii) Wear accelerates the degradation initiated by fatigue. Abrasive wear is caused by hard particles, such as contaminants or spall debris, scratching the raceway, while adhesive wear results from metal transfer during transient sliding conditions. Wear is often lubricant-related and typically accompanied by frictional heat.

(iii) Corrosion presents in three forms: moisture corrosion, fretting, and false brinelling. Moisture corrosion results from water ingress, creating dull patches on the raceway. Fretting occurs when there is relative micro-motion at the bearing-housing interface, while false brinelling appears as wear marks during standstill under vibration.

(iv) Electrical erosion occurs when current travels through the rolling path. High current densities arc-weld the contact, leaving molten craters, while low leakage currents create clusters of shallow pits.

(v) Plastic deformation results from shock loads or debris dents that distort the raceway and increase local stresses. It can be caused by overleading, shock loads or improper handling. The debris or overlead can be an affect of wear or damage to an different component, such as a gearbox.

(vi) Fracture or cracking ends service life. Fractures are either forced fracture, fatigue fracture, or termal cracking. A forced fracture develops when local stresses exceed the material's ultimate tensile strength. A fatigue fracture, by contrast, emerges under repeated bending once the fatigue limit is surpassed. Finally, thermal cracking occurs when sliding contact generates intense frictional heat.

2.2. Maintenance strategies

The short-term maintenance strategy refers to the planning and scheduling of maintenance activities over a relatively limited horizon, typically from a few days to several months. This time frame aligns with prediction horizon from prognostics or diagnostics and allows timely interventions. Unlike long-term strategies that focus on lifecycle management, short-term approaches emphasise operational efficiency and responsiveness to near-term risks [5].

This section examines various maintenance strategies applicable to short-term operations and maintenance (O&M) planning. The focus is on how these strategies interact with the prediction horizons and support the effective scheduling of maintenance actions. An effective short-term maintenance strategy seeks to improve economic efficiency, extend the lifetime of components, minimise emergency repairs, reduce overtime labour costs, and mitigate the impact of unexpected equipment failures [5].

The international standard EN,13306 [64] distinguishes two fundamental forms of maintenance, namely Preventive Maintenance (PM) and Corrective Maintenane (CM), as is distingists in Figure 2.5. Preventive strategies are intended to avoid any failure, while corrective measures are the activities that occur after a component failure [65]. Preventive strategies can be further sub-devided in predetermined and condition-based settings, while corrective strategies can be sub-devided in immediate and delayed maintenance. Concepts such as predictive or opportunistic maintenance [5], are also briefly explained within the context of the EN,13306 classes. The section starts with explaining CM in Subsection 2.2.1, and then covers PM in Subsection 2.2.2.

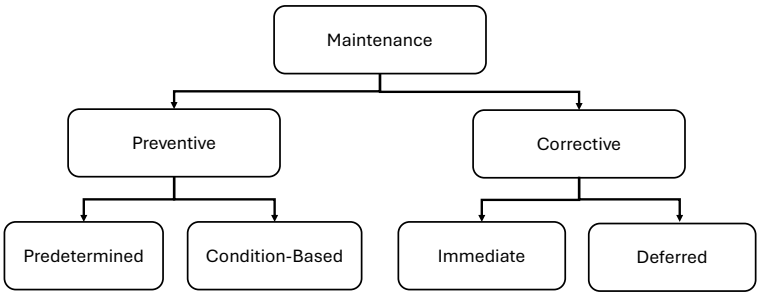


Figure 2.5: Maintenance strategies classifications. Adapted from: [64]

2.2.1. Corrective maintenance

CM, also called reactive maintenance, is a failure-based approach in which repairs or replacements are performed only after a component has failed. This reactive strategy is simple, as it relies solely on failure events without the need for condition monitoring, and maximises component utilisation [11]. However, it can result in unexpected downtime, high repair costs, and operational disruptions. It is not the most cost-effective approach, particularly for offshore wind turbines where access is challenging [12].

EN,13306 divides CM into two sub-classes [64], according to how quickly the repair is initiated. Immediate CM is begun without deliberate delay once the need is recognised, usually when the fault poses safety risks or may lead to severe secondary damage if operation continues. That means that the turbine cannot operate until the fault is repaired, leading to high production costs. In contrast, CM deferred replacements are postponed to a more convenient time because the fault has limited impact on production or safety.

2.2.2. Preventive maintenance

PM comprises actions performed before functional failure occurs, with the goal of reducing the probability of failure or the rate of degradation [65]. In offshore wind O&M it mitigates the high cost and weather-related risk of unscheduled access.

Predetermined maintenance, also referred to as periodic maintenance, takes a proactive, time-based approach, involving scheduled inspections and replacements at regular intervals. By addressing potential issues before they cause failures, this strategy reduces the likelihood of failures. However, periodic maintenance often leads to unnecessary waste of useful life of components and can still lead to failure if a component wears more than anticipated [11].

CBM focusses on real-time monitoring of equipment health to determine maintenance time based on a threshold. This approach uses measurements from specific sensors (from CMS or SCADA systems) to estimate the current state [66]. An important distinction in CBM is diagnosis and prognosis. Diagnosis is an indication of the current (and past) health of a system based, often expressed as HI, and prognosis is an assessment of future health, often expressed with an Remaining Useful Life (RUL) [65]. CBM in which prognostics is included is sometimes also refered to as Predictive Maintenance (PdM) [11].

The CBM strategy has been the subject of extensive research [12]. For example, Dao et al. [17] propose a CBM strategy that integrates real-time CM data with a physical degradation model based on Paris' Law. Maintenance decisions are based on an optimised threshold and a decision tree that determines whether to perform corrective maintenance, periodic maintenance, or CBM. Despite advances in academic research, CBM is still far from being fully implemented in the wind industry [12].

Finally, there is opportunistic maintenance, which is not a strategy on its own, but a combination of two strategies, aiming to use shutdown as an opportunity to perform PM on other subsystems [61], combining CM and PM. This strategy can also take advantage of favourable weather conditions or curtailment periods to schedule maintenance tasks during suitable periods or planned downtime. For example, Yang et al. [21] propose a framework that optimises maintenance by clustering tasks during such opportunities, or Donnelly en Carrol [67] considers planning maintenance during a curtailment periods. Studies can focus on opportunistic maintenance components within a single wind turbine such as Perez et al. [68], or expand the scope to include opportunities in multiple turbines or even entire wind parks such as in Bakir et al. [69].

2.3. Modelling condition and prognostics

Condition modelling or prognostics are central parts of CBM. Prognostics refers to the ability to reliably predict the future state or condition of health of a machine, expressed as an RUL [59]. In contrast, condition modelling aims to track the cumulative damage, often in the form of an HI. An HI provides the current condition of the component and helps to detect a pending failure early. Generally, the modelling methods for both RUL and HI fall under three main approaches: data-driven, physics-based, or hybrid models [11]. An overview of these approaches, along with example models, is provided in Figure 2.6. All three approaches can employ both SCADA systems or CMS.

This section covers the three approaches: first it describes the physics-based models in Subsection 2.3.1. Then the section examines data-driven techniques in Subsection 2.3.2 and hybrid strategies that fuse physical insight with operational data are addressed in Subsection 2.3.3. Particular attention is devoted to deep learning methods in Subsection 2.3.4, with an emphasis on unsupervised architectures to construct robust HI and translate them into RUL.

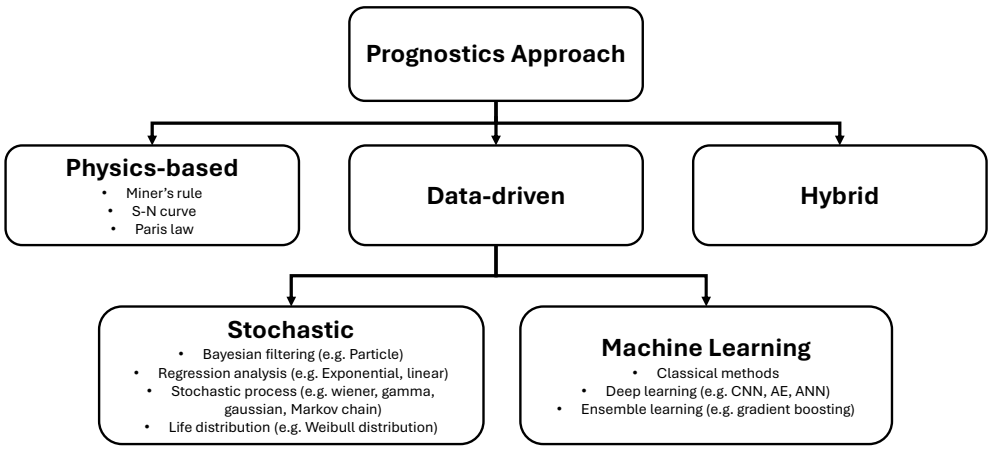


Figure 2.6: Overview of approaches to condition monitoring and prognostics. Adapted from: [42]

2.3.1. Physics-based techniques

Physics-based prognostic techniques rely on a deep understanding of the physical system or process and use scientifically established relationships to describe its behaviour [11]. These models employ first-principles approaches, such as materials mechanics, thermodynamics, and fluid dynamics, to simulate the degradation and failure mechanisms of components. For wind turbine condition monitoring, most physics-based models are based on fatigue lifetime prediction methodologies such as the S-N curve [11]. Other common approaches include using Paris' law to analyse crack propagation in materials or Palmgren–Miner's rule to estimate cumulative damage. For example, Florian et al. [22] developes a fracture mechanics model using Paris' law to estimate the RUL of wind turbine blades.

Physics-based models have the advantage of being grounded in physical laws, making their results more interpretable and particularly well suited for safety-critical applications [23]. They are effective for specific failure modes, providing precise predictions with fewer data needed compared to other techniques [48]. However, their effectiveness relies on high-fidelity modelling of specific defects. Furthermore, physical degradation processes are typically only well understood for relatively simple or critical components [23]. Consequently, the widespread use of physics-based models in real-world applications is limited.

2.3.2. Data-driven techniques

Data-driven methods use large volumes of data to assess the state of the wind turbine. These methods do not require prior knowledge of the system but rely on large datasets, which should include enough failure cases to learn from [11]. Data-driven approaches are subdivided into statistical and machine learning methods and are discussed separately below.

Statistical Models

Statistical models base their predictions on empirical data and statistical techniques. Life distribution, the simplest technique uses previous failure data to estimate model distributions (such as exponential, lognormal, Gaussian, and Weibull functions) to predict when failures are expected to occur [11]. In particular, the Weibull function is widely used to model the reliability of the components [70]. One of these Weibull models, the classical "bathtub curve", captures reliability over the lifetime of an component with three different Weibull distributions. Specifically, it assumes trends as decreasing failure rates during early life ($\beta < 1$), constant rates during normal operation ($\beta = 1$), and increasing rates as components wear out ($\beta > 1$) with β being the shape parameter [11].

Other key techniques include Hidden Markov Models [68], Particle Filters, autoregressive models, Wiener processes, Bayesian degradation models [69], [71], and Gamma processes [11]. In general, the advantage of statistical techniques lies in their ability to efficiently analyse historical data to provide reliable predictions with quantified uncertainty. However, their performance depends on the quality of the data and is less precise in predicting RUL of complex systems [58].

Machine learning models

Machine learning models can learn the underlying patterns within the data. The models can be dived into the classical, deep learning and ensemble learning methods [42]. For condition monitoring in the wind industry, the most widely used machine learning models are Neural networks (NN), support vector machines, and decision trees [55]. An important distinction in machine learning is supervised - unsupervised learning. Supervised learning predicts the results using labelled data, while unsupervised learning does this without labels. In PHM, labels typically indicate the operating state of the turbine [11].

Man et al. [25] investigate the performance of different machine learning, specificity models Classification And Regression Tree, support vector machines, NN, Deep NN, and Gradient Boosting. The study first applies the unsupervised learning technique K-means clustering to classify SCADA data, this classification is used as labels of the five algorithms. Among these models, Classification And Regression Tree and deep NNs demonstrated the best predictive performance.

The advantages of machine learning-based models lie in their efficiency and practicality in modelling complex nonlinear systems, and their relatively short development time compared to physics-based and hybrid approaches. However, their limitations include the need for large volumes of representative data for training (including failures), computational power, and low interpretability [42], [48].

2.3.3. Hybrid techniques

Hybrid models are the leading and most accurate approaches for wind turbine prognostics [48]. In hybrid models, two methodologies are combined to take advantage of the strengths of each model. For instance, Wang et al. [72] demonstrate a hybrid approach that integrates the Paris formula with a Particle Filter within a Bayesian framework for probabilistic RUL estimation of bearings. They present a combined diagnostic and prognostic approach, their diagnostics is achieved by extracting vibration characteristics from a wavelet transform-based method, which are then fused into a HI to represent the bearing condition. Hybrid models such as these enabled accurate forecasts with limited representativeness in the data set.

Another recent example is the work by Chao et al. [23], who combine thermodynamics and wear modelling with sensor data, processed using a deep NNs for RUL estimation of turbofan engines. Their study demonstrates that their hybrid framework outperformed purely data-driven approaches, extending the prediction horizon by almost 127 % , meaning it could anticipate failures more than twice as early and provide much longer lead times for maintenance planning. With this, they claimed superior performance over the best purely data-driven methods of the CMAPSS dataset. However, Chao et al.[23] noted that these hybrid models require careful design to avoid the combined drawbacks of the underlying methods, and they emphasise the need for adequate and accurate physics-based information to allow the model to function effectively.

2.3.4. Deep learning techniques

DL, a subset of machine learning, uses multiple nonlinear processing layers to automatically extract complex features from sensor data, making it very usefull for condition monitoring of complex systems [73]. Deep learning techniques are well suited for handling large volumes of noisy data extracted from wind farms and operate without requiring prior knowledge of the system [11]. However, these models require extensive run-to-failure data for training and often lack transparency, which can make them unsuitable for applications requiring interpretability [11].

Deep learning originated from Artificial Neural Networks (ANNs), which are NNs composed of multiple interconnected layers (input, hidden, and output) [73]. Different types of ANNs serve different purposes. Convolutional Neural Networks (CNNs) are a type of multilayer feedforward Artificial Neural Network (ANN) [73]. CNNs are particularly effective for tasks that involve spatial data and are widely used in applications such as image recognition and understanding [74]. This is achieved through convolutional layers, which capture local patterns and features from the input data by leveraging spatial hierarchies across multiple layers [74]. As a result, CNNs excel at analysing structured datasets, such as temperature patterns or other sensor data from wind turbines.

Recurrent Neural Networks (RNNs) are deep learning architectures that include feedback connections from hidden or output layers back to previous layers, allowing them to process sequential and dynamic information over time [73]. Among the RNN variants, Long Short-Term Memory (LSTM) networks are particularly popular due to their ability to retain information over extended time periods. This makes

them especially effective for capturing trends in system degradation [73].

Jankauskas et al. [24] uses the EDP SCADA dataset to evaluate the effectiveness of RNNs, comparing LSTM, gated recurrent unit, and biderectional LSTM to predict temperature anomalies for early fault detection. Their study focused on the gearbox, generator, generator bearing, transformer, and hydraulic group, with biderectional LSTM showing the best performance. This approach enabled the detection of faults up to 37 days before failure.

Similarly, Udo et al. [26] study the three models: Extreme Gradient Boosting, LSTM networks, and Statistical Process Control for fault detection in wind turbines using SCADA data. The study compares this on two datasets (EDP and ENGIE), with EDP having failure data and ENGIE not. This study focuses on monitoring gearbox and generator temperatures to identify faults. Among the models tested, LSTM demonstrates the highest effectiveness. The authors offered limited information on why LSTM outperformed the other methods. However, the paper presents a thorough methodology for setting up a deep learning model in wind turbines.

Unsupervised learning
Unsupervised learning methods do not need labelled data. In PHM, this means they can work without true RUL values. By not relying on labels, they can train on only healthy data to learn the normal behaviour of a machine. In this way, they are useful when failures are rare, for example when components are typically replaced before failure. These models can still detect changes in system behaviour and support health monitoring.

For example De Pater and Mitici [28] applie an AE with LSTM for turbofan engines and derive an RUL prognostic tool with fewer run-to-fault data compared to supervised method. They showe to make RUL predictions with as little as 2 run-to-failure cases. Their model is trained only on healthy data. An HI is then generated for the run-to-failure cases using reconstruction errors from the AE, based on the assumption that the reconstruction error increases as the system degrades. HI trends are evaluated using monotonicity, trendability, and prognosability metrics. RUL is then estimated using a similarity-based approach, comparing current HI trends with historical failure cases. They showe that the method can work with as few as two failure cases, although performance improves with more failures. Their results showes that this method outperforms traditional supervised approaches, reducing the prediction root mean squared error of RUL by 19% compared to 1D CNN and LSTM-based models.

By developing an HI and RUL, more information is provided. Based on the HI, an anomaly can be detected before failure, whereas the days to failure are given by the RUL. In general, the methods for the prediction of RUL based on HI can be categorised into three main groups: prognostic methods based on similarity, statistical methods based on models and data, and intelligent time series prediction methods [75].

Han et al. [30] also propose an RUL prediction method using HI estimations based on an AE model. Their method is applied to high-frequency bearing data, which is described using features such as peaks, energy, and shape factor. Then four of these features are selected on the basis of a monotonicity equation. Their method first fuses multiple features into a single HI using an AE. Instead of using a similarity-based approach for RUL estimation, they train an LSTM model. The RUL is given on a relative scale from 0 to 1, which means that it lacks a direct time reference. This is not preferable in maintenance planning.

Yu et al. [29] applie similarity-based matching for the estimation of the RUL using an HI index similar as De Pater and Mitici [28]. They use an RNN AE, which does need run-to-failure training data to automatically fuse high-dimensional sensor data into a one-dimensional HI. The encoder's latent space representation served as the HI. The HI curve is stored offline and later matched with test instances using a similarity-based technique, where similarity is determined by the distance between the test HI curve and those in the library. The RUL for each match is estimated based on the length of the matched curve and the final RUL is obtained as a weighted average of the similarity scores and individual RUL values. To improve matching, a zero-centering rule is introduced, allowing curves to be shifted up or down for a better fit. However, this sometimes causes incorrect training instances to be considered similar, especially when many training samples are available. The method is applied to the C-MAPSS dataset, achieving competitive scores compared to state-of-the-art prognostic models. However, it require a large amount of run-to-failure data for effective performance, especially compared to the method of De Pater and Mitici [28].

Encalada-Dávila et al. [41] develops a purely SCADA-based HI for rotor bearings of wind turbines. Their unsupervised approach first trains a feed-forward ANN on one year of healthy data from twelve identical turbines to estimate the low-speed-shaft temperature. The residual absolute rotor temperature servs as a point-wise anomaly score, which they aggregat weekly into an HI normalised to $[0, 1]$ via a persistence criterion that counts samples whose residual exceeds a data-driven threshold, where the value of 1 is given if the threshold is croshed for a week. Only one rotor-bearing failure occurres in the data, which provides a 3.5-month early warning based on the HI. Importantly, the model does not require failure labels, no additional sensors, and provided an interpretable HI. However, in their HI, a value of 1 does not equal a failure as is desired. And their approach is validated on a single bearing fault, so wider generalisation remains to be demonstrated.

2.4. Integrated condition-based maintenance approaches

Integrated CBM approaches bring together the two essential steps: the prediction of component degradation and the use of this information in maintenance optimisation models. In the wind industry, such approaches are scarce and often dated. In Subsection 2.4.1 the literature that combines both aspects is discussed in the wind industry. Other industries have made significant strides in integrated CBM, most notably the aircraft maintenance industry, as highlighted by Borsotti et al. [15]. Both the wind turbine and the aircraft industry share key similarities, including high downtime costs and the technical complexity of the systems involved [15]. The relevant integrated CBM literature of the aircraft industry is discussed in Subsection 2.4.2

2.4.1. Application in the wind industry

Many studies in the wind industry focus on developing diagnostic or prognostic models without considering their practical implementations. Other studies focus on planning approaches that rely on simplified degradation assumptions, and only a limited number attempt to combine both. This sections gives an review of the papers that combine both aspects. An overview of the integrated approaches identified in this review is provided in Table 2.2 and Table 2.3. Table 2.2 summarises the prognostic and diagnostic aspects of the reviewed literature, while Table 2.3 focuses on the maintenance-related aspects.

Tian et al. [18] are among the first to combine prognostics and maintenance for the case of a wind turbine. They assume Weibull distributions as appropriate to describe the real failure times of the rotor, gearbox, and generator. This is a significant simplification over real word failure rates. An ANN is uses to predict the life percentages of these components from CMS data, assuming a standard deviation over the ANN life prediction. For maintenance planning, two thresholds are established to guide maintenance actions: one for CM and another lower threshold for PM. This way, when a CM activity is performed, it can be coupled with a PM activity. These thresholds are simpler than the dynamic optimisations seen in later research. The study relies on CMS data, as did all but one of the other articles reviewed.

Perez et al. [68] propose a Markov model based on wind speed data from a wind farm in Texas (USA) to synthetic model degradation and failure rates and proposed a maintenance model for a 100-turbine wind farm. Their framework considers gearboxes, generators, blades, and control systems, incorporating factors such as maintenance capacity, lead times, rescheduling due weather conditions, and separate costs for CM and PM. However, like Tian et al. [18], opportunistic maintenance activities are only considered in the same wind turbine, not considering activities between different wind farms. Additionally, the framework does not use dynamic optimisation but rather employs static CM and PM thresholds. Furthermore, the synthetically modelled degradation and failures do not represent real-world conditions.

Yildirim et al. [71] address simplifications by considering dependencies between turbines with a dynamic optimisation function. Their approach utilises a stochastic Bayesian degradation model to estimate current states of turbines, based on real-world vibration data. A dynamic cost function is used, which enables maintenance decisions to be more optimal compared to static thresholds. Although the method accounts for interdependencies between wind turbines within a wind farm, it does not explicitly consider the condition of individual components within each turbine.

Bakir et al. [69] uniquely consider the interaction of opportunistic maintenance actions between turbine components, individual turbines, and the entire wind farm. The paper proposes the self-named Multicomponent Condition-Based Opportunistic Maintenance framework, which combines probabilistic lifetime predictions of components with an opportunistic maintenance model applied across a wind farm. Prognostics have less emphasised in the paper and are achieved using Bayesian degradation models using real-time sensor data and parametric degradation models. Rather than field measurements, this study uses a laboratory-derived vibration-based degradation dataset from rotating machinery, together with KNMI North Sea wind records.

Zhou and Yin [70] leverage ANN to estimate the percentage of life of critical components of the wind turbine, including the rotor, bearings, gearbox, and generator. The ANN itself provides deterministic outputs derived from CMS data (e.g., vibration monitoring data, temperature monitoring data, oil quality monitoring data), and these outputs are assumed to have a normal deviation. Predictive analysis information is used to guide maintenance decisions for offshore wind turbine components. Their results demonstrate a significant reduction in costs compared to traditional preventive, corrective, or combined strategies.

Yang et al. [21] propose a comprehensive intelligent maintenance framework dependent on the age of the state for wind turbines. The framework employs Bayesian degradation modelling to predict the RUL of components while accounting for uncertainties, among dynamic wind condition forecasting using GRA. Although the inclusion of probabilistic degradation modelling and weather forecasting

is impressive, the optimisation approach is limited to a single turbine. The solution technique can therefore be a rule-based methodology implemented through a for loop, which evaluates possible maintenance actions. Furthermore, the framework also assumes robust sensor coverage for CMS measurements, and simulates this in their work, not based on real-world data.

Unlike the other studies, Bangalore and Patriksson [76] use real-world SCADA data from a Swedish wind farm to propose a maintenance management framework called the Self-Evolving Maintenance Scheduler. The framework is designed to integrate data from multiple sources and applies various maintenance strategies. First, a typical Weibull distribution is used for multiple components to model failure rates and to propose a time-based framework. For PdM, an ANN is trained to learn normal gearbox temperature patterns, allowing faults to be anticipated up to two months in advance. Probabilities are further refined using a gamma distribution. However, the fault detection approach provides less information for maintenance planning compared to an RUL estimate. In addition, its effectiveness is limited by the availability of only two recorded gearbox anomaly cases, highlighting a common challenge with appropriate run-to-failure data.

To summarise, the integrated approaches applied in the wind industry, as shown in Table 2.2 and Table 2.3, are most commonly based on CMS data, Bangalore and Patrikson [76] being the only application that uses real-world SCADA data. Despite the advantages of SCADA, such as continuous availability and wide coverage across turbines, and its growing dominance in current prognostic research [38], it remains under-represented in the integrated approaches reviewed. The studies most commonly use stochastic methods, with Bayesian degradation models being particularly, but also ANN's were popular. Most of them however are not truly integrated approaches as they do not use real sensor data. When it comes to maintenance planning, previous studies focus on static optimisation techniques, where opportunities for maintenance activities are treated as fixed. Later studies are solved under dynamic conditions using Mixed-Integer Linear Programming (MILP). All reviewed methods optimised for minimal maintenance cost over the lifetime.

2.4.2. Application in aircraft industry

Research on turbofan engines showcases advances in prognostic techniques, rolling horizon planning, and modelling under uncertainty. The availability of high-quality datasets, such as NASA's simulated jet engine dataset C-MAPSS, supports the development of PdM models. Widely used in studies such as [34]–[37], [56], it enables incremental improvements and facilitates meaningful model comparisons.

De Pater et al. [36] propose a dynamic maintenance framework for a fleet of aircrafts. Their CNN-based prognostic model demonstrates performance that is generally among the best existing CNN methods for the C-MAPSS dataset, although the LSTM networks achieved a lower root mean squared error. The paper employs a rolling maintenance schedule with a daily alarm threshold integrated with an integer linear programme. The total expected costs using imperfect Remaining of Usefull Life (RUL) prognostics are 24.3 % higher than those achievable with perfect RUL predictions. As the prognostics are point RUL, no insight is offered into the uncertainties of the estimations.

Building further on [36], Lee and Mitici [35] use a CNN model with Monte Carlo dropout for probolistic RUL prediction, which updates over time, and solve maintenance scheduling using a deep reinforcement learning approach. Where [36] uses a common 1D kernel for all features, this paper applies one 1D kernel per column of input, as it is shown to be effective for multivariate time series data, such as the C-MAPSS data set [35]. The Monte Carlo dropout introduces randomness with each forward pass

Table 2.2: An overview of the aspects related to prognostics or diagnostics of Integrated Approaches for Wind Turbines

Reference	Data Source	Components	Approach	Model
Tian et al. [18]	CMS	Rotor (including blades), main bearing, gearbox, generator	machine learning	ANN
Perez et al. [68]	Wind speed from a real wind farm, modelled degradation	Gearbox, generator, blade, control system	Stochastic	Markov model
Yildirim et al. [71]	CMS (vibration)	Blades, gearbox	Stochastic	Bayesian degradation model
Bakir et al. [69]	CMS (synthetic), and wind data	Gearbox, rotor, generator, bearing	Stochastic	Bayesian degradation model
Zhou and Yin [70]	CMS	Gearbox, rotor, generator, bearing	machine learning	ANN
Yang et al. [21]	CMS (vibration data)	Generator, gearbox, bearing, variable pitch	Stochastic	Bayesian degradation modelling
Bangalore and Patriksson [76]	SCADA	Gearbox	machine learning	ANN

Table 2.3: An overview of the aspects related to maintenance of Integrated Approaches for Wind Turbines

Reference	Location	Strategy	Level	Notes
Tian et al. [18]	Not specified	Opportunistic CBM	Component, Turbine	Static dual-threshold strategy; Opportunistic Maintenance only within WT
Perez et al. [68]	Onshore	Opportunistic CBM	Component, WT, WF	Opportunistic Maintenance only within WT
Yildirim et al. [71]	Not specified	Opportunistic CBM	WT, WF	Does not consider Opportunistic Maintenance within WT
Bakir et al. [69]	Not specified	Opportunistic CBM (PdM)	Component, WT, WF, Multi-WF	Optimizes across all levels, includes wind speeds
Zhou and Yin [70]	Offshore	Opportunistic CBM (PdM)	Component, WT	Considers dynamic lead time effects
Yang et al. [21]	Onshore	Opportunistic CBM (PdM)	Component, WT	Includes dynamic wind velocities
Bangalore and Patriksson [76]	Onshore	CBM	Component, WT	Considers both age-based (Weibull) and CBM strategies

that is used to provide confidence intervals for the RUL predictions, and the Monte Carlo dropout also prevents overfitting. The deep reinforcement learning framework determines the optimal moment for scheduling replacements, offering the advantage of a variable replacement threshold instead of fixed triggers. The method achieves a result close to the ideal maintenance strategy at true RUL. It is shown to significantly outperform approaches based on mean RUL estimates, though it results in a higher number of unscheduled replacements. The study focusses on a single component and simplifies assumptions about logistics and spare part availability.

In the study by Nguyen and Medjaher [37], the authors propose a dynamic PdM methodology that uses an LSTM algorithm for prognostics. Instead of providing a precise RUL value, the approach predicts the probability of system failure in three future time windows, resulting in flexibility in maintenance planning and reducing the risk of incorrect early life decisions. However, this provides less detailed information. For maintenance planning and prediction cannot be compared with other papers using common regression evaluation criteria. The paper is also unique for its optimisation of maintenance decisions based on two cost rates ('replacement' and 'do nothing'), solved using the particle swarm optimisation method. In addition, they integrate inventory with out-of-stock penalties. The results demonstrate performance close to the ideal scenario with perfect RUL information, outperforming traditional methods.

Mitici et al. [34] propose a dynamic PdM framework designed for systems with multiple components, using CNNs with Monte Carlo dropout to obtain probabilistic RUL predictions, similar to the approach employed by Lee et al. [35]. Mitici et al. [34] differ from Lee et al. [35] by considering multiple components and integrating a renewal–reward process. For turbofan engines, the study shows that optimal replacement times can be aligned with the lower bound of the 99 % confidence interval of the RUL, achieving a significant (53 % cost reduction) compared to maintenance based on time while achieving a very low failure rate.

Zhuang et al. [56] attempt to address both epistemic (model) and aleatoric (data) uncertainties in RUL prediction with a Bayesian framework. Aleatoric uncertainty is modelled through a Bayesian framework, with the real RUL used to characterise the variance of the Gaussian noise term. Epistemic uncertainty is addressed by employing a Bayesian method within the biderectional LSTM architecture, where the model parameters are treated as probability distributions instead of fixed values. Their prognostic scores outperformed the other models compared in the C-MAPSS dataset.

In summary, the articles reviewed, as shown in Table 2.4, favour CNNs or bidirectional LSTM for prognostics, as CNNs excel at extracting features from multivariate time series data [74], while bidirectional LSTMs retain sequential information, making them well suited for capturing temporal dependencies in degradation signals [73]. Probabilities on the RUL are included using the Monte Carlo dropout in [34], [35], a Bayesian framework in [56]. With regard to maintenance, the most commonly used approach is cost optimisation solved by MILP. Except for Nguyen et al. [37] , who use Particle Swarm Optimisation for cost rate optimisation, and Lee et al. [35], who employ Deep Reinforcement Learning.

Table 2.4: Integrated Prognostics and Maintenance Approaches in the Aerospace Industry

Reference	Prognostics: Model	Maintenance: Objective Function, Solution Technique	Attributes
De Pater and Mitici [36]	CNN	Cost, Integer Linear Programming	Uses alarm-based scheduling
Lee and Mitici [35]	CNN with Monte Carlo dropout	Cost, Deep Reinforcement Learning	Adaptively schedules maintenance without fixed thresholds
Nguyen and Medjaher [37]	LSTM	Cost rate (for both replace and do-nothing), Particle Swarm Optimisation	Unique in its approach regarding RUL (predicting change of failure in a time period), maintenance cost rate and solving technique
Mitici et al. [34]	CNNs with Monte Carlo dropout	Cost, MILP	Provides a comprehensive complete framework for multiple components.
Zhuang et al. [56]	Bidirectional-LSTM with an bayesian framework	Cost, MILP	Addressed both epistemic and aleatoric uncertainty

Summary

CBM forms the execution layer of the broader PHM cycle, which progresses from *observation* through *analysis* to *action*. In wind turbines, observation is provided by monitoring systems. These CMS can either be dedicated CMS installations or the SCADA infrastructure already present on every utility-scale machine. CMS delivers high-frequency vibration and oil data that can localise individual defects, while SCADA supplies continuous, farm-wide coverage at little additional cost. Industry failure surveys consistently show that mechanical subsystems (e.g. generator, gearboxes, bearings) account for the bulk of downtime despite exhibiting comparatively low failure rates, underscoring the value of condition-based interventions focused on these critical components. Most of the rotor bearing failure mechanisms are expected to show an increase in (SCADA) temperature values, while others show an anitial response in vibration sensors.

Analysis transforms these CMS sensor streams into actionable health metrics. Condition monitoring delivers a HI that reflects the current state of the asset, while prognostics gives a forecast of the RUL forecast. Physics-based fatigue models yield more transparency for well-characterised damage modes but require detailed modelling and material and load data. Data-driven approaches extract degradation patterns directly from CMS or SCADA records and now dominate the literature, especially deep learning techniques. Unsupervised deep learning methods learn normal behaviour from healthy data alone, a decisive advantage when run-to-failure examples are scarce and do assume trends. Hybrid schemes fuse physical insight with these data-hungry models to regularise learning and quantify uncertainty more realistically, but are complex to design. Action is implemented through maintenance scheduling. The EN 13306 standard divides strategies into corrective and preventive classes, the latter spanning periodic to condition-based.

Studies that cover a full CBM cycle, integrated CBM frameworks, remain rare in the wind sector. Most prognostic papers end once a RUL curve is produced, and most scheduling models still insert that life curve through simplified Weibull or age-based assumptions. Only a small cluster of publications

links both to address the practical implications of the developed model. These paper typically use extensive vibration-based data, estimate degradation with Bayesian or other stochastic filters (sometimes wrapped in a shallow ANN), and then pass the resulting RUL distributions to a mixed-integer cost model. In aircraft maintenance, public benchmarks such as NASA's (simulated) C-MAPSS have enabled complex prognostics. CNNs or (bi-)LSTMs learn multivariate time series behaviour and then feed rolling-horizon or RL planners. With suitable data, the same frameworks could be transplanted to the wind turbine domain.

The literature highlights a clear research gap: the majority of studies either develop advanced prognostic models without translating the results into maintenance decision-making or formulate scheduling frameworks that rely on oversimplified lifetime assumptions. The integrated CBM studies for wind turbines are largely dated and do not take advantage of advanced deep learning techniques capable of harnessing the widely available SCADA data. This research will try to address this research gap, in the next chapter (Chapter 3) the theoretical background of developing such a deep learning model will be discussed.

3
Theoretical Background

Deep learning models are a powerful tool in condition monitoring because they can learn complex, non-linear relations directly from CMS data. The purpose of this chapter is to provide the theoretical background for the AE method introduced in Chapter 4, and this chapter builds on three concepts. First, ANNs (Section 3.1) are the foundational architecture of deep learning models as they map an input vector to an output through layers of weighted, and understanding them is the basis for understanding AE. Second, RNNs (Section 3.2) and in particular LSTM units add gated feedback connections that let the model retain information over many time-steps, crucial for tracking the slow evolution of mechanical fault. Third, an AE (Section 3.3) is a specific ANN architecture composed of an encoder that compresses each multivariate input window into a low-dimensional latent vector and a decoder that reconstructs the original signal.

3.1. Artificial neural networks

ANNs are computational models inspired by the structure of the human brain [77]. They are designed to approximate the relationships between inputs and outputs through one or more interconnected processing units, known as neurones or nodes [78]. The section explains these ANNs by first explaining a single perceptrons (Subsection 3.1.1), which is the single-neuron units that underpin all NNs. The section then shows how stacking these neurons into input, hidden, and output layers forms a multilayer perceptron (Subsection 3.1.2), extending a simple linear separator into a deep model. Next, the section examines activation functions (Subsection 3.1.3) and how their non-linearities steer gradient flow and learning speed, before outlining the training process (Subsection 3.1.4) as iterative weight updates. Overfitting of an ANN occurs when the model learns noise instead of general patterns. The section closes with overfitting countermeasures (Subsection 3.1.5), introducing early stopping and dropout.

3.1.1. Single perceptron

The perceptron is the simplest form of an ANN, consisting of a single neuron. The neural maps input features to an output through a weighted summation and a non-linear activation function, as shown in Figure 3.1. A single perceptron consists of a set of input nodes $\{x_1, x_2, \ldots, x_n\}$, each connected to an output node via a corresponding weights $\{w_1, w_2, \ldots, w_n\}$. These inputs are aggregated into a

linear combination, and a bias term b is added:

$$a = \sum_{i=1}^{n} w_i x_i + b \tag{3.1}$$

The inclusion of a bias term b allows the activation function to be shifted, improving the network's ability to model data not centred around the origin. The weighted sum a is then passed through an activation function f (more over f in Subsection 3.1.3) to produce the final output y:

$$y = f(a) \tag{3.2}$$

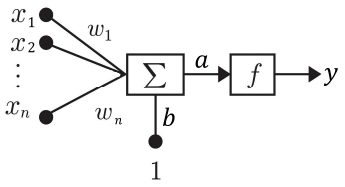


Figure 3.1: Single perceptron calculation process

3.1.2. Multi-layer perceptrons

While a single perceptron can only model simple linearly separable relationships, more complex patterns can be learnt by combining multiple perceptrons into a layered structure. This results in a feedforward ANNs, where neurones are organised in successive layers and data flows in one direction: from input to output. There are also variations in feedforward architectures. For example, with skip connections, branching structures, or residual layers. Such variants are not considered here, but later in Section 3.2 RNNs are introduced.

A typical multi-layer perceptron ANN architecture is shown in Figure 3.2. In the image three types of layers are shown [79]:

- **Input layer:** The input layer receives the raw data. Each neuron in this layer corresponds to one characteristic of the input vector $\mathbf{x}$. It does not perform any transformation and simply passes the values on to the next layer.
- **Hidden layers:** These intermediate layers perform the actual calculation. Each neuron in a hidden layer receives input from all neurones in the previous layer, applies a weighted summation and a non-linear activation function, and passes the result to the next layer. The number of hidden layers and neurons per layer are important hyperparameters that influence the model's capacity.
- **Output layer:** The final layer produces the network's prediction. It can be a single neuron, for example for regression tasks. But it is also possible to have multiple neurons, for example when trying to reconstruct the signal.

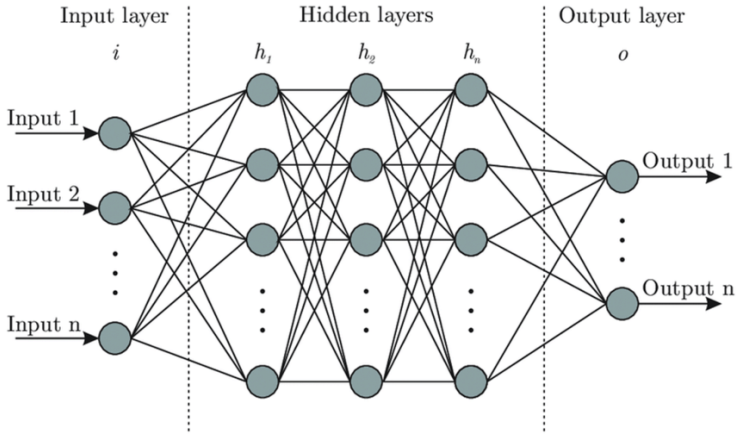


Figure 3.2: General ANNs architecture, shown the input, hidden and output layers [79]

Each neuron in a given layer functions simular to the single perceptron described previously: it computes a weighted sum of its inputs, adds a bias term, and applies a non-linear activation function to produce its output. Formally, for a neuron j in layer l, the computation proceeds as follows:

- Weighted summation with bias:

$$a_j^{(l)} = \sum_{i=1}^{n} x_i^{(l-1)} W_{i,j}^{(l)} + b_j^{(l)} \tag{3.3}$$

- Activation:

$$z_j^{(l)} = f(a_j^{(l)}) \tag{3.4}$$

Here, $x_i^{(l-1)}$ denotes the output from neuron i in the previous layer, $W_{i,j}^{(l)}$ is the weight connecting neuron i in layer $l-1$ to neuron j in layer l, and $b_j^{(l)}$ is the bias term associated with neuron j.

3.1.3. Activation Functions

The choice of activation function significantly impacts model behaviour. Some common options include [80]:

- **ReLU:** $f(a) = \max(0, a)$
- **Sigmoid:** $f(a) = \frac{1}{1+e^{-a}}$
- **Tanh:** $f(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}$

Each function has specific advantages depending on the data and architecture. For instance, ReLU is computationally efficient and often leads to faster convergence, while sigmoid and tanh are bounded, which can help stabilise training during early iterations [78].

3.1.4. Training

Training the network involves adjusting the weights $W^{(l)}$ and biases $b^{(l)}$ to minimise the discrepancy between the predicted output y and the true target value t. This discrepancy is quantified using a loss function, which provides a scalar value measuring prediction error. The weights are adjusted via backwards propagation, where the error gradient is passed backward through the network using the chain rule to compute how each weight contributes to the loss. These gradients are then used by an optimisation strategy to update the weights.

Loss function

Several loss functions are commonly used in regression tasks, each with different sensitivity to outliers and statistical assumptions [81].

- **Mean Absolute Error** measures the average magnitude of the absolute differences between predicted and true values. It corresponds to the L^1-norm and is robust to outliers. The mean absolute error is defined as:

$$\text{Mean Absolute Error} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - t_i\right| \tag{3.5}$$

- **Mean Squared Error (MSE)** penalises larger errors more heavily by squaring the deviations. It corresponds to the L^2-norm and assumes normally distributed errors. The MSE is defined as:

$$\text{Mean Squared Error} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - t_i\right)^2 \tag{3.6}$$

- **Root Mean Squared Error** is the square root of the MSE and has the same unit as the predicted variable, which often helps interpretability. It is defined as:

$$\text{Root Mean Squared Error} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - t_i\right)^2} \tag{3.7}$$

These loss functions are chosen on the basis of the statistical properties of the target data and the learning objectives. In particular, the mean absolute error is more suitable when robustness to noise and bounded generalisation are prioritised, while the MSE and the root mean squared error are preferred in settings where Gaussian noise dominates and larger deviations must be penalised more severely [82].

Optimisation strategy

The network first initialises its weights and biases randomly. Using these initial parameters, a forward pass is performed to generate predictions. Once the predictions are obtained, the backpropagation algorithm is applied [83]. This algorithm computes the gradient of the loss function with respect to each weight by propagating the error backwards from the output layer through the hidden layers. These gradients are then used in a gradient-based optimisation method to update the weights and minimise the loss.

To improve the accuracy and training performance of neural networks, various optimisation strategies have been developed. Commonly used methods include stochastic gradient descent, its improved variants with momentum and Nesterov acceleration, and adaptive techniques such as AdaGrad, RMSProp, Adadelta, and Adam [83]. These are all examples of so-called first-order methods, which means they use only the gradient (i.e. the slope) of the loss function to update the model's weights. Each method adds its own improvements. For example, momentum keeps track of previous weight updates to help the optimiser move more smoothly and quickly towards the minimum. Adam further improves on this by adjusting the learning rate for each parameter based on how often it changes and by correcting for any initial bias in the updates. These enhancements make Adam especially effective for training deep neural networks with complex data.

3.1.5. Overfitting

Overfitting occurs when a model performs well in training data, but does not generalise to unseen test data [84]. A common indicator of overfitting is an increasing gap between training and testing loss. This suggests that the model is memorising patterns in the training data rather than learning the underlying relationships. Several techniques have been developed to address this problem. For a broader discussion on overfitting mechanisms and mitigation strategies, readers are referred to the work of Shaeke Salman and Xiuwen Liu [84]. Two commonly used methods to reduce overfitting are early stopping and dropout, which are discussed below.

Early stopping

A clear indication of overfitting is when the training error keeps decreasing, while the test error does not. Early stopping is a technique that addresses this by stopping the training process when the performance of the model in the validation set no longer improves for a defined number of consecutive epochs. Once this condition is met, the training stops and the model reverts to the best-performing state recorded during the training process.

Dropout

Dropout adresses overfitting by randomly deactivating neurons from the neural network during training [85], as illustrated in Figure 3.3 were the crossed out neurons are due to the dropout and change each training epoch randomly. This prevents the network from becoming overly reliant on specific nodes and encourages the development of more robust features that are distributed across the network. During testing, all neurons are active and their outputs are scaled to account for the dropout applied during training.

3.2. Recurrent neural networks

Feedforward NNs are typically used for tasks such as classification or regression that rely on single input vectors [80]. For time series, RNNs offer a more natural way to handle sequential data by introducing recurrent connections within the architecture. These connections allow the network to maintain a hidden state that is updated at each time step, enabling the model to capture temporal dependencies across arbitrary intervals [86]. LSTMs (Subsection 3.2.1) are an advanced type of RNNs, are explained. To use LSTMs, sliding windows (Subsection 3.2.2) are often applied to segment the time series into overlapping fixed-length sequences.

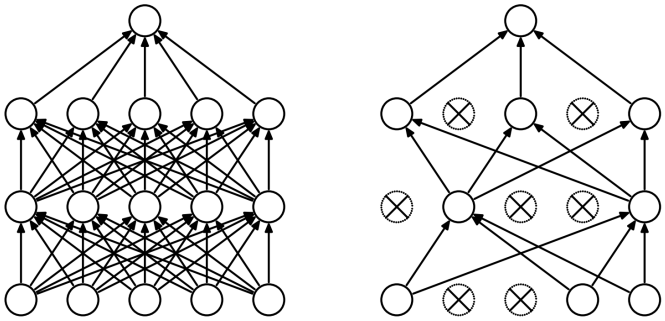


Figure 3.3: Dropout applied to an neural network model [85]

3.2.1. Long short-term memory

A major limitation of standard RNNs is their difficulty in learning long-term dependencies, particularly when there is a significant delay between an input and its corresponding output [80]. During training, the backpropagation of error through time can lead to vanishing or exploding gradients, depending on the recurrent weights. To overcome this, LSTM networks were introduced. LSTM networks are a specialised form of RNNs, designed by Hochreiter and Schmidhuber [87] to overcome the vanishing gradient problem inherent in traditional RNNs. LSTMs are capable of capturing both short- and long-term dependencies in sequential data.

Each cell in an LSTM network, as illustrated in Figure 3.4, contains three gates: the input gate, forget gate, and output gate [88]. These gates control the flow of information into and out of the memory cell. This structure allows LSTM networks to selectively retain or discard past information over multiple time steps.

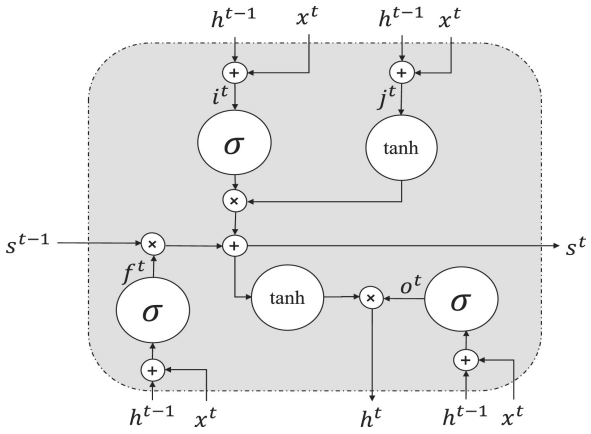


Figure 3.4: LSTM cell [88]

At time step t, given the input vector x_t, the previous hidden state h_{t-1}, and the previous cell state s_{t-1}, the LSTM updates the hidden layer h^t as follows [86]:

$$i^t = \sigma(W^i x^t + U^i h^{t-1} + b^i) \tag{3.8}$$
$$j^t = g(W^j x^t + U^j h^{t-1} + b^j) \tag{3.9}$$
$$f^t = \sigma(W^f x^t + U^f h^{t-1} + b^f) \tag{3.10}$$
$$s^t = f^t \otimes s^{t-1} + i^t \otimes j^t \tag{3.11}$$
$$o^t = \sigma(W^o x^t + U^o h^{t-1} + b^o) \tag{3.12}$$
$$h^t = o^t \otimes g(s^t) \tag{3.13}$$

where:

$\sigma(\cdot)$ is the element-wise sigmoid activation function,

$\tanh(\cdot)$ is the hyperbolic tangent activation,

$\odot$ denotes element-wise multiplication,

$\oplus$ denotes element-wise addition,

f_t forget gate vector, controlling memory retention,

i_t input gate vector, controlling new memory input,

$\tilde{c}_t$ candidate cell state,

c_t updated cell state,

o_t output gate vector, controlling hidden state exposure,

h_t hidden state at time t,

W_*, V_*, U_* trainable weight matrices mapping, respectively, the sensor inputs $\mathbf{x}_c^t$, operating conditions $\mathbf{o}_c^t$, and previous hidden state h_{t-1} to each gate,

b_* bias vectors specific to each gate.

Each gate equation combines the current sensor input, operating conditions, and past hidden state to compute its contribution to the memory update and output.

3.2.2. Sliding window

A sliding window, as shown in Figure 3.5, is a method of generating multiple subsequences from a time series which is particularly useful in LSTM models, as they require fixed-length inputs for training. The sliding window creates these sequences by taking a fixed length window n_{sw} of consecutive points, then sliding this window with a stride s along a time series [89]. The stride s controls how much the window shifts between each segment smaller values lead to more overlap and more samples. For LSTM models, the sliding window size will be the same as the LSTM sequience length.

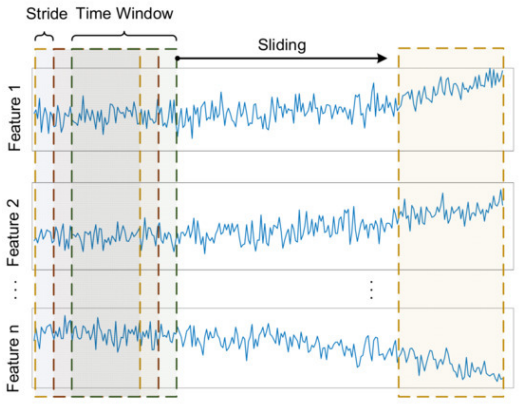


Figure 3.5: Sliding window [89]

3.3. Autoencoder

An AE is a ANN architecture that learns a compact latent representation of its input and then reconstructs the original signal from this lower-dimensional code [73]. AE are widely used for normal behaviour modeling of wind turbines. As illustrated in Figure 3.6, the model consists of two main components: an encoder and a decoder. The encoder maps the high-dimensional input signal $\vec{x} \in \mathbb{R}^F$ into a lower-dimensional latent vector $\vec{z} \in \mathbb{R}^G$, where $F >> G$ [27]. The reconstructed signal is the same dimension as the input signal, meaning $\hat{\vec{x}} \in \mathbb{R}^F$. The intermediate representation $\hat{\vec{z}}$ forces the network to compress and prioritize information and is often referred to as the bottleneck.

The decoder then attempts to reconstruct the original signal from this latent representation, producing an output $\hat{\vec{x}}$. The training objective is to minimise the reconstruction error between $\vec{x}$ and $\hat{\vec{x}}$, encouraging the model to preserve essential structure in the data.

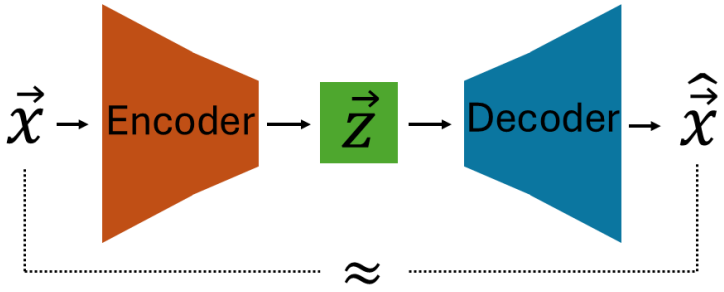


Figure 3.6: General architecture of an autoencoder

With the theoretical foundations for the LSTM-AE now established, the next chapter (Chapter 4) will describe the implementation of the concrete model and show how its reconstruction error is transformed into a HI. The chapter also demonstrates how this HI integrates into a maintenance decision-making framework.

4
Methodology

The methodology framework used to answer the research question consists of two interconnected parts: (i) the development of a HI (Section 4.1) and (ii) its application within a CBM scheme (Section 4.2).

For the development of a HI, an unsupervised LSTM AE is used that learns the normal behaviour solely on healthy data. An unsupervised model can be trained using few run-to-failure data and does not need any assumption on degradation patterns, which can deviate for different failure modes. By learning normal behaviour, the network reconstructs healthy signals accurately, whereas its reconstruction error increases as the components degrade. This error of specific sensors, from the basis for the reconstruction error HI, and after z-score normalisation, yields the HI. This part of the method will be covered in Section 4.1.

Section 4.2 provides the second part of the method by translating the HI into an operational maintenance policy. The HI is divided by fixed thresholds into degradation zones that trigger minor repair, major repair, or preventive replacement. These zones will be separated on the basis of an assumed distribution from a reference turbine. These zone-based decisions are embedded in a life-cycle cost model that factors in technician hours, vessel class, weather windows, and mobilisation lead times.

In Figure 4.1 a full research flow diagram of the method used in this thesis is provided. Module 1 contains the data preparation (Chapter 5), Module 2 covers the LSTM AE for construction of the HI (Section 4.1), and Module 3 describes the CBM logic (Section 4.2). The condition-monitoring scheme is highlighted in red and comprises Modules 1 and 2, whereas the maintenance scheme is highlighted in blue and comprises Module 3. The flowchart starts with the acquisition of SCADA data in the top left and ends with the condition monitoring and maintenance results in the bottom right. All important individual steps in each module are shown, and the arrows indicate data flow: green arrows represent healthy cases, orange arrows represent faulty cases, blue arrows correspond to maintenance, and black arrows indicate general flow.

4.1. Health indicator

The building blocks on deep learning this thesis are introduced in Chapter 3, this section shows these building blocks are integrated into the LSTM AE used in this thesis. The architecture follows the framework of de Pater and Mitici [28]. Subsection 4.1.1 outlines the AE architecture and its key elements. Subsection 4.1.2 describes how the reconstruction-error HI is extracted from the

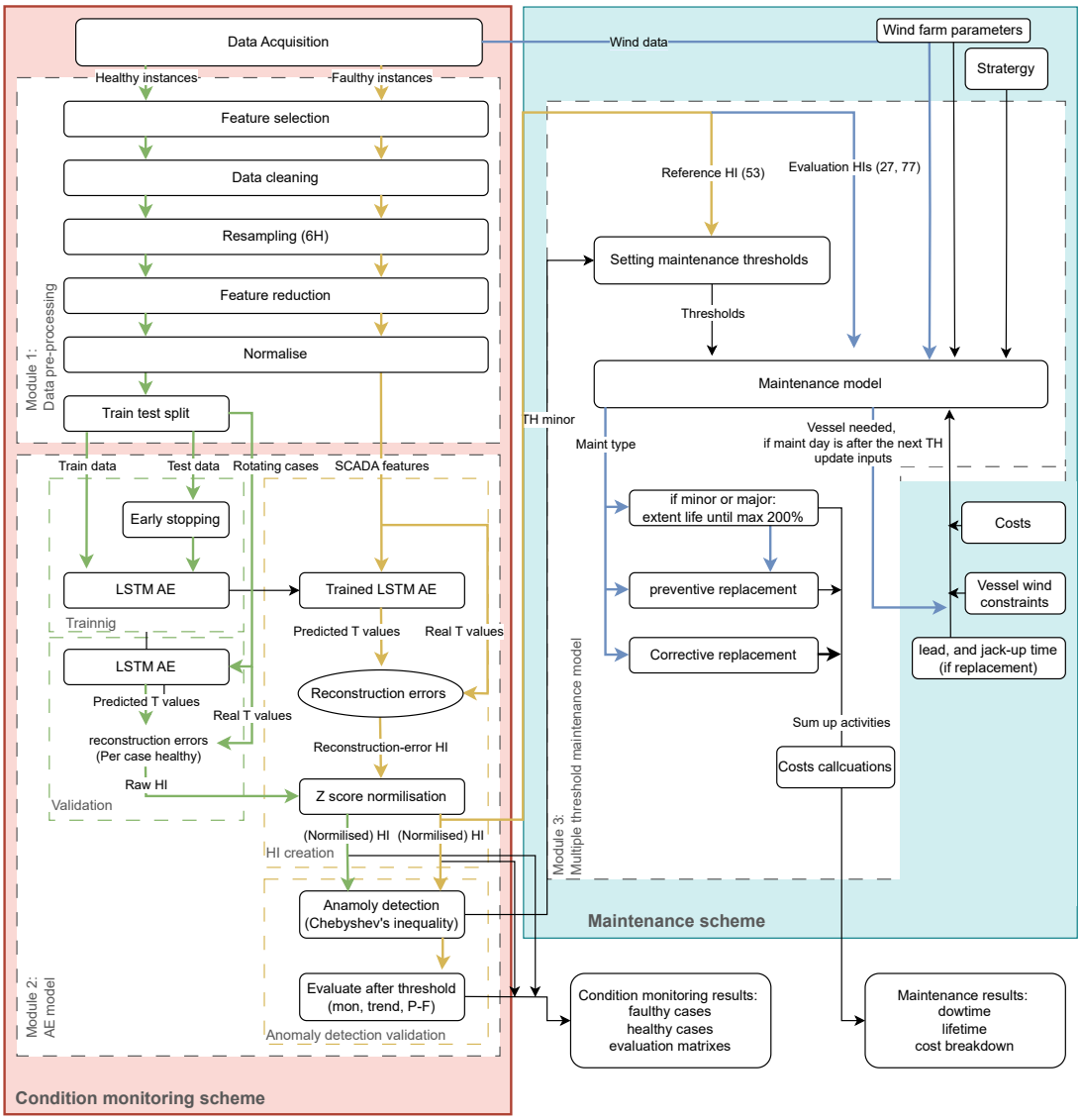


Figure 4.1: Flowchart condition monitoring and maintenance scheme of this thesis

reconstruction error, after which Subsection 4.1.3 applies Z-score normalisation. Fault detection relies on Chebyshev's inequality, detailed in Subsection 4.1.4. Once the threshold is crossed, the performance of the HI is assessed using the metrics given in Subsection 4.1.5.

4.1.1. Autoencoder

The encoder extracts temporal patterns from each input window and compresses them into a low-dimensional latent vector, the decoder then reconstructs the original sequence from this representation. For a more detailed background of AEs, see Section 3.3. The remainder of this subsection describes the specific implementation adopted in this thesis, starting with the operating condition-informed LSTM cells. Then the decoder and encoder are discussed, followed by the loss function used.

LSTM cells with informative operating conditions

In Subsection 3.2.1, the structure of a standard LSTM cell is described and illustrated in Figure 3.4. In this thesis, the LSTM architecture is adapted to include the operating conditions in an informative manner, as shown in Figure 4.2. In this image the operating conditions for case c ad time t are indicated by $\mathbf{o}_c^t \in \mathbb{R}^O$. This means that operating conditions are not treated as part of the output to be reconstructed, as replicating weather patterns is not the objective. Instead, these parameters are provided as additional input features during both encoding and decoding, alongside the primary input parameters $\mathbf{x}_c^t$. This approach follows the method proposed by De Pater and Mitici [28], who demonstrated that integrating operating conditions in this way results in more robust HIs under varying environmental conditions, which is particularly relevant for wind turbine SCADA data.

Let $\mathbf{x}_c^t$ represent the input vector for condition monitoring at time step t during case c, and let $\mathbf{o}_c^t$ denote the corresponding operating condition vector. These inputs are processed independently through their own trainable input weights for each gate in the LSTM. The LSTM equations at time step t are expressed as follows:

$$g_t = \sigma\left(W_g \mathbf{x}_c^t + V_g \mathbf{o}_c^t + U_g h_{t-1} + b_g\right), \tag{4.1}$$
$$i_t = \sigma\left(W_i \mathbf{x}_c^t + V_i \mathbf{o}_c^t + U_i h_{t-1} + b_i\right), \tag{4.2}$$
$$c_t^{\text{can}} = \tanh\left(W_c \mathbf{x}_c^t + V_c \mathbf{o}_c^t + U_c h_{t-1} + b_c\right), \tag{4.3}$$
$$c_t = (c_{t-1} \odot g_t) \oplus (i_t \odot c_t^{\text{can}}), \tag{4.4}$$
$$p_t = \sigma\left(W_p \mathbf{x}_c^t + V_p \mathbf{o}_c^t + U_p h_{t-1} + b_p\right), \tag{4.5}$$
$$h_t = p_t \odot \tanh(c_t), \tag{4.6}$$

where

$\mathbf{x}_c^t$ isthe input vector for condition monitoring at time step t during case c

$\sigma(\cdot)$ is the element-wise sigmoid function,

$\tanh(\cdot)$ is the hyperbolic tangent,

$\odot$ denotes element-wise multiplication,

$\oplus$ denotes element-wise addition,

f_t forget gate vector,

i_t input gate vector,

$\tilde{c}_t$ candidate cell state,

c_t cell state,

o_t output gate vector,

h_t hidden state,

W_*, V_*, U_* trainable weight matrices mapping, respectively, sensor inputs $\mathbf{x}_c^t$, operating conditions $\mathbf{o}_c^t$ and previous hidden state h_{t-1} to each gate,

b_* bias vectors for the corresponding gates.

In this formula $W_f \mathbf{x}_c^t$ represents the direct contribution of the current sensor measurements, while the second term $V_f \mathbf{o}_c^t$ captures the effect of operating conditions.

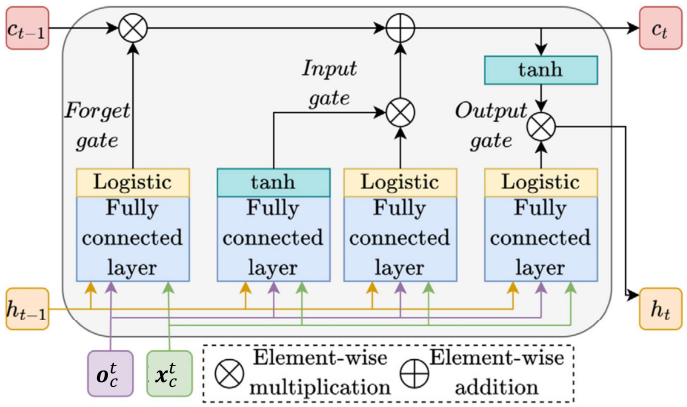


Figure 4.2: LSTM cell with informative operating conditions [28]

Encoder

Each input sample is a multivariate time series window of shape $[T, F]$, where T is the number of time steps defined by the size of the sliding window, which is designed to be the size of LSTM sequence as explained in Subsection 3.2.2, and F is the number of features (sensors signals). A complete batch is structured as $[N, T, F]$, where N is the number of sequences in the batch, T is the sequence length of the LSTM and F is the number of features per time step. Each input sequence from case c can be denoted as:

$$\mathbf{X}_c = \{\mathbf{x}_c^1, \mathbf{x}_c^2, \ldots, \mathbf{x}_c^T\}, \quad \mathbf{x}_c^t \in \mathbb{R}^F$$

where each row vector $\mathbf{x}_c^t$ contains the feature values at time step t.

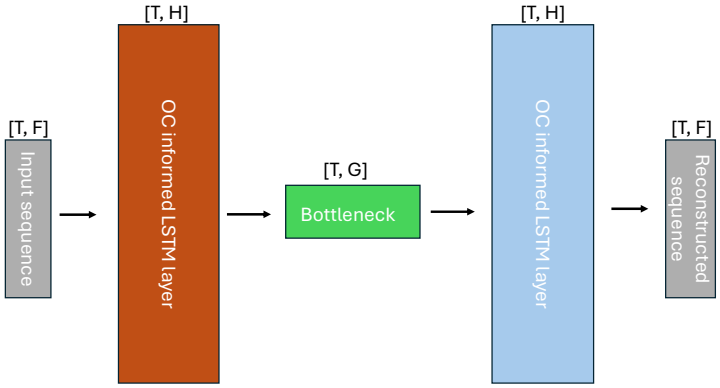


Figure 4.3: Schematic of the *operating-condition (OC) informed LSTM AE* employed in this thesis, with the tensor shape shown above the layer.

The encoder processes each input sequence $\mathbf{X}_c$, together with the corresponding operating condition $\mathbf{O}_c = \{\mathbf{o}_c^1, \mathbf{o}_c^2, \ldots, \mathbf{o}_c^T\}$ where each $\mathbf{o}_c^t \in \mathbb{R}^O$ which are used informatively in each LSTM module. The encoder takes the input dimension and projects it onto an hidden layer with a higher-order hidden dimension H. Then this hidden layer projects the hidden representation into a lower-dimensional latent space of size G by an linear layer. The result is a sequence of latent vectors, $\mathbf{Z}_c = \{\mathbf{z}_c^1, \mathbf{z}_c^2, \ldots, \mathbf{z}_c^T\}$, where each $\mathbf{z}_c^t \in \mathbb{R}^G$ represents a compressed encoding of the behaviour of the system at time t during case c.

Dropout introduces Subsection 3.1.5 is applied between the LSTM layers of the encoder and decoder to reduce overfitting and improve generalisation. The dropout rate is set to 0.4, slightly below the generic default of 0.5 recommended by Srivastava et al. [85], because it yields good results with minimal signs of overfitting.

Decoder

The decoder takes the latent sequence $\mathbf{Z}_c = \{\mathbf{z}_c^1, \mathbf{z}_c^2, \ldots, \mathbf{z}_c^T\}$ and attempts to reconstruct the original input sequence, denoted as $\widehat{\mathbf{X}}_c = \{\widehat{\mathbf{x}}_c^1, \widehat{\mathbf{x}}_c^2, \ldots, \widehat{\mathbf{x}}_c^T\}$, where each reconstructed vector $\widehat{\mathbf{x}}_c^t \in \mathbb{R}^F$. The decoder mirrors the encoder in structure, maintaining a symmetric AE design including dropout. Figure 4.3 gives an schematic view of the full auto-encoder: the input and reconstructed sequences are shown in grey, the OC-informed LSTM is shown in red (encoder) and blue (decoder), the bottleneck is shown in green.

Loss function

The model is trained using the MSE loss function between the original input $\mathbf{x}$ and the reconstructed sequence $\hat{\mathbf{x}}$:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{T \cdot F} \sum_{t=1}^{T} \sum_{f=1}^{F} \left(x_{i,t,f} - \hat{x}_{i,t,f} \right)^2 \tag{4.7}$$

where $x_{i,t,f}$ and $\hat{x}_{i,t,f}$ represent the true and reconstructed value of feature f at time step t in the i-th input sequence, respectively.

Early stopping as introduced (Subsection 3.1.5) is employed for the training. The training optimisation terminates if the validation loss has not improved from the MSE error of the test data for five consecutive epochs (with an overall limit of 100 epochs) and the weights are rolled back to the best-performing epoch.

4.1.2. Health indicator construction

The model is trained exclusively on healthy data to learn normal behaviour. It does not learn to represent degraded behaviour and is less able to predict the sensor values accurately. When the system degrades, the reconstruction error between the predicted and actual values increases for specific sensors. Therefore, the reconstruction error of these sensors can be used to construct a reconstruction-error HI. In Subsection 4.1.3 these reconstruction-error HI will be normalised.

This study aggregates the reconstruction error on a *daily* basis. Sensor selection is guided by domain knowledge, retaining only those variables whose readings are expected to rise as the component degrades.

Let $\mathcal{L}_d^t$ denote the mean reconstruction error at time t on day d, calculated as:

$$\mathcal{L}_d^{c,s} = \frac{1}{n_d - 1} \sum_{c,s=1}^{C,S} \left\| \hat{\mathbf{x}}_{c,s}^t - \mathbf{x}_{c,s}^t \right\| \tag{4.8}$$

where $\mathbf{x}_{c,s}^t$ is the true sensor vector and $\hat{\mathbf{x}}_{c,s}^t$ is the reconstructed vector at time step t, and n_d is the number of time steps recorded on day d for case c and sensor s.

Let λ_d^c be the HI for case c on day d can be calculated as the average reconstruction error across the selected rotor bearing sensors:

$$\lambda_d^c = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \mathcal{L}_d^{c,s} \tag{4.9}$$

where $\mathcal{S}$ is the set of selected sensors. This results in a time series $\lambda^c = \{\lambda_1^c, \lambda_2^c, \ldots, \lambda_d^c\}$, the sequence is the reconstruction-error HI sequence for that case c.

4.1.3. Z-score normalisation applied to the health indicator

The HI should be normalised to improve interpretability and transferability, because the reconstruction-error HI (λ^c) lacks an intuitive scale. In the literature, HIs usually range between 0 and 1, where 0 denotes a healthy state and 1 marks the failure point [27], [36], [41]. Milani et al. [27] achieve this by deriving the HI from the latent space, bounding it with a sigmoid activation, and constraining it to start at 0 and end at 1. De Pater et al. [28] obtain a similar effect using min–max normalisation. Encalada-Davila et al. [41] increment a weekly counter whenever a threshold is crossed, yet this still fails to identify a clear failure point. These existing approaches are not feasible here, as the HI is not drawn from the latent space and has only limited references with varying maxima.

In this thesis, z-score normalisation is aplied. This normalisation enables meaningful comparison of HI values across turbines by referencing each turbine's own statistics. Z-score normalisation centres the axis at the mean (set to 0) and scales it so that one unit corresponds to one standard deviation, expressed mathematically as:

$$z_i^c = \frac{\lambda_d^c - \mu_H^c}{\sigma_H^c} \tag{4.10}$$

where λ_d^c is the reconstruction error of case c on day d,, and μ_H^c, σ_H^c are the corresponding mean and standard deviation of the reconstruction error calculated during the healthy period.

This normalisation ensures that similar degradation trends produce comparable values, regardless of the baseline of each HI. In threshold-based maintenance, it offers a consistent and interpretable basis for decision-making. Especially since the change of an HI exceeding a certain value is the same arccos turbines, which will be discussed in the next section (Subsection 4.1.4)

4.1.4. Anomaly detection using Chebyshev's inequality

In condition monitoring, Chebyshev's inequality based thresholds are commonly used to detect anomalies in the literature [28], [90], [91]. Chebyshev's inequality provides an upper bound on the probability that a random variable deviates more than k standard deviations from its mean. The Chebyshev's inequality [90] states:

$$\Pr\left(|\lambda_d^c - \mu_H^c| \geq k\sigma_H^c\right) \leq \frac{1}{k^2}. \tag{4.11}$$

Here, λ_d^c is the reconstruction error of case c on day d, μ_H^c and σ_H^c are respectively the mean and standard deviation of that error during the healthy training period, and k is a chosen number of standard deviations that defines the alarm threshold.

A turbine is diagnosed as unhealthy if the HI crosses this threshold n times in succession [91]. Assuming the successive observations are independent, the probability of witnessing n consecutive crossings is bounded by $(1/k^2)^n$ [36]. Exceeding this limit signals an anomaly, which means that the model continues to run, but the turbine is no longer considered healthy. The threshold is turbine-specific to account for unit-to-unit behavioural differences.

Because the z-score-normalised HI is expressed in units of standard deviation, centred around the mean, any threshold on the HI directly defines the corresponding k in Chebyshev's inequality. For example, if the normalised HI exceeds 5 ones, it represents a deviation of $k = 5$ standard deviations from the mean. From Chebyshev's inequality (see Equation 4.11), the probability of observing such a deviation is bounded above by $1/25 = 4\%$.

Health indicator for the healthy cases

To assess the ability of the model to distinguish between healthy and anomalous behaviour, the AE is also applied to data from healthy turbines. A rotating approach was used: for each healthy case, the model was trained on the remaining five healthy datasets and then evaluated on the excluded case. This ensures that the test data is not seen during training, maintaining the integrity of the evaluation.

4.1.5. Evaluation matrix for health indicator

Bearing degradation is known to be a highly monotonic process, driven by irreversible and accumulating damage mechanisms that originate at the microscale and propagate to failure [27]. Therefore, a good HI is expected to show a clear upward trend over time, referred to as *trendability* ($\mathcal{T}$). A good HI is expected and to follow this trend consistently, captured by the *monotonicity* ($\mathcal{M}$) metric. Both are bounded between 0 and 1, with values closer to 1 indicating stronger and more reliable trends. Monotonicity ($\mathcal{M}$) and trendability ($\mathcal{T}$) are commonly used to address the quality of a HI [27], and are computed as follows.

Monotonicity ($\mathcal{M}$)
Monotonicity measures how consistently the HI increases or decreases over time. For a case c with a sequence of HI values λ^c, the monotonicity $\mathcal{M}$ is calculated as follows [28]:

$$\mathcal{M} = \frac{1}{F_c - 1} \left| \sum_{d=1}^{F_c - 1} I(\lambda_{d+1}^c - \lambda_d^c) - I(\lambda_d^c - \lambda_{d+1}^c) \right| \tag{4.12}$$

where d is the day, F_c the number of datapoints, and $I(x)$ is an indicator function defined as:

$$I(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

Trendability ($\mathcal{T}$)
Trendability assesses whether the HI follows a clear upward or downward trend across the full time series. It is calculated using the Spearman correlation coefficient *Corr* between the ranked HI values λ^e and the corresponding time vector T^c for an case c [27]:

$$\mathcal{T} = Corr(\lambda^c, T^c) \tag{4.13}$$

The HI is assessed only once the anomaly threshold has been exceeded. Before degradation begins, the HI should remain relatively stable. Interest in trendability and monotonicity therefore centres on ensuring a clear and consistent trajectory after anomaly detection, so that the HI can be trusted for threshold setting without subsequently falling below the threshold again.

P-F interval
In addition, the P-F interval is used to evaluate performance. The P-F interval is defined as the time between the first detection of abnormal behaviour (P) and the failure event (F) [31]. The earlier a potential failure is detected by the monitoring system, the smaller the damage suffered [31], therefore, it is important for the HI to have a large P-F interval. In this case, P is detected by the Chebyshev's inequality equation presented in Subsection 4.1.4.

4.2. Maintenance

This section links the HI to offshore maintenance. The HI trajectory is divided into four maintenance zones (A-D) which are degradation states in which certain maintenance activity is linked. $TH_{\text{minor}}, TH_{\text{major}}, TH_{\text{repl}}$ are the three fixed thresholds that separate these zones and signal when the HI crosses from one level of degradation to the next. TH_{minor} coincides with the earliest detection point P, whereas functional failure is marked by point F.

Four maintenance strategies build on this framework. *Corrective replacement* acts only after failure, *preventive replacement* installs a new component at the beginning of severe degradation, *CBM* follows the thresholds in real time and applies the activity that matches the zone reached, and *perfect CBM* mirrors the CBM but defers each task to the last feasible weather window.

The remainder of this section defines the zones (Subsection 4.2.1), expands on the strategies (Subsection 4.2.3), sets the thresholds and life-extension factors (Subsection 4.2.2), outlines the operational constraints (Subsection 4.2.4), and develops the cost model (Subsection 4.2.5).

4.2.1. Maintenance zones

In this theses, maintenance activities are structured according to the degradation state observed in the HI. The zones are based on Horenbeek et al. [31], who segment the deterioration process into four distinct zones. The HI is ment to reprecent the degradation process, which give the zones as illustrated in Figure 4.4. Each of these zones corresponds to a specific maintenance activity. The separate zones are divided into zones A to D, bounded by three HI thresholds, which mean:

- **Zone A**: Represents the early stage of degradation, where minor corrective actions can still extend the component's lifetime. This zone is entered when the HI exceeds the threshold TH_{minor}, which the same as the earliest point of detection P [31].
- **Zone B**: Indicates significant degradation requiring more substantial maintenance actions, called major repairs. This zone begins once the HI crosses the threshold TH_{major}.
- **Zone C**: This zone corresponded to severe degradation. Consequential damage, damage to other components, is possible. Full preventive replacement is necessary. This zone is triggered once the HI exceeds the threshold TH_{repl}.
- **Zone D**: This zone is defined after the failure point, the point F. Here, the component is no longer functional and consequential damage might have also accured. This zone corresponds to a corrective maintenance action.

4.2.2. Determining the thresholds

The maintenance zones described in Subsection 4.2.1 are entered after crossing the static thresholds $TH_{\text{minor}}, TH_{\text{major}}, TH_{\text{repl}}$, as depicted in Figure 4.4. Here, we discuss the setting of these thresholds.

In setting TH_{minor}, it is assumed that the anomaly detection point P occurs within Zone A, which means $TH_{\text{minor}} = P$. The anomaly detection point is based on Chebyshev's inequality (Subsection 4.1.4), such that $P = k$, with k being the threshold derived from the inequality. This k corresponds directly to the z-score value k .

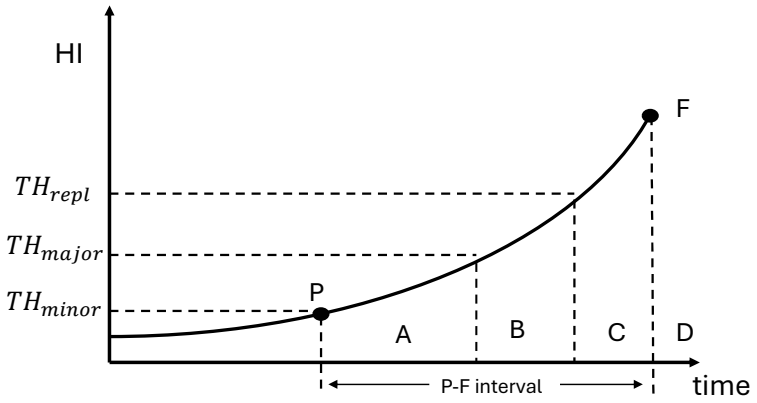


Figure 4.4: Maintenance zones based on HI thresholds

For this thesis, TH_{major} and TH_{repl}, are defined based on a reference turbine. The maximum HI value observed in this reference turbine is divided into three equal intervals to establish the main and replacement thresholds, under the assumption of the same progression of degradation across these phases. The division starts from zero, as the degradation trend begins here, even though the degradation is only detected later at point P. These thresholds are visualised in Figure 4.5.

Life extension factors quantify the relative increase in component lifespan achieved through different maintenance activities. These values are adopted from [16], with factors of 0.3 for minor repairs, 0.5 for major repairs, and 1.0 for full replacements, the latter reflecting the installation of a new component. An overview of these values is provided in Table 4.1.

These life extensions cannot extend the lifetime indefinitely. According to Deao et al. [17], the life of a roller bearing can become approximately 200% the original lifetime by maintaining a high lubrication cleanliness. For this thesis, it is assumed that such an extension can be achieved through well-timed maintenance activities. Therefore, the maximum possible lifetime extension is set to 1 (Table 4.1), which would double the lifetime.

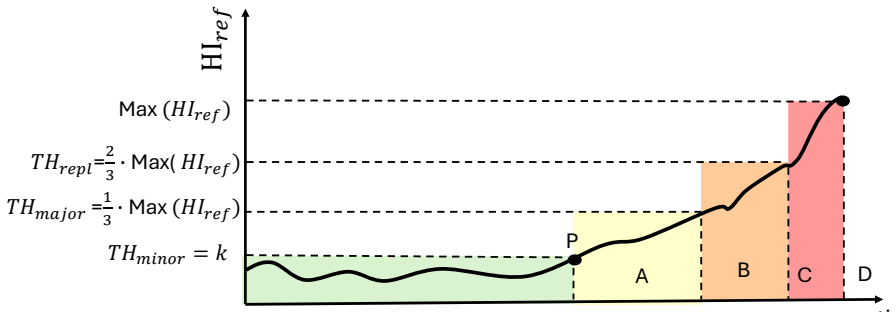


Figure 4.5: Thresholds based on the theoretical reference turbine

Table 4.1: Maintenance thresholds and life extension factors [16], [17]

Maintenance type	Relative threshold	Life extension factor	Maximum extension
Minor repair	$3 + 0 \cdot \max(\mathrm{HI}_{ref})$	0.3	1
Major repair	$\frac{1}{3}\ \max(\mathrm{HI}_{ref})$	0.5	1
Corrective Replacement	$\frac{2}{3}\ \max(\mathrm{HI}_{ref})$	1	–
Preventive Replacement	–	1	–

4.2.3. Maintenance strategies

The main CBM strategy is evaluated alongside the corrective, preventive, and perfect CBM strategies to assess its performance. Maintenance zones, as defined in Subsection 4.2.1, are entered after crossing the corresponding static thresholds TH_{minor}, TH_{major}, and TH_{repl}, as described in Subsection 4.2.2. Each strategy is triggered by a different threshold and is described below on the basis of its corresponding degradation zone and response logic.

For all strategies, once a maintenance action is triggered, a lead time L is introduced. The lead time is defined as the period between the maintenance decision and the scheduled execution, including the time required to mobilise the maintenance team, purchase spare parts and prepare for repairs [18]. After this lead time, maintenance is executed during the first suitable weather window. The specific activity carried out depends on the degradation zone in which the turbine resides on the execution day.

Corrective replacement

In this strategy, corrective replacement is performed after the component has failed and the HI reaches the failure point F. As a result, maintenance takes place in Zone D. The turbine is non-operational from failure until the replacement is completed, leading to high downtime and potential secondary damage. It serves as a baseline for comparing other strategies.

Preventive replacement

This strategy involves only replacement actions, bypassing Zones A and B. Maintenance is activated when the HI exceeds TH_{repl}, placing the component in Zone C, or in Zone D if a failure occurs before the activity can be executed. If maintenance is performed before failure, replacement is preventive. Otherwise, it defaults to a corrective intervention.

Condition-based maintenance

This is the primary strategy proposed in this thesis. Maintenance is initiated as soon as the HI exceeds the first threshold, TH_{minor}. The type of maintenance performed corresponds to the degradation zone in which the turbine is located on the day of execution, which is the day with the first suitable weather window after the lead time.

For minor or major maintenance, it is assumed that each maintenance action restores the component to a condition equivalent to its state at before the time of intervention. As a result, the same fractional lifetime extension can be applied again, as the HI will follow the same tradectery. This enabling multiple consecutive maintenance actions until the maximum permitted extension is reached.

Let $\mathcal{L}_0$ denote the lifetime of the component at the moment of the first action, life_ext(a) the fractional extension achieved per maintenance and $\text{life\_ext}_{\text{max}}$ be the maximum lifetime extension. The maximum number of repeated actions is calculated as:

$$N_{\text{repairs}} = \left\lfloor \frac{\text{life\_ext}_{\text{max}}}{\text{life\_ext}(a)} \right\rfloor$$

The total lifetime is then given by:

$$\mathcal{L} = \mathcal{L}_0 \cdot \left(1 + N_{\text{repairs}} \cdot \text{life\_ext}(a)\right)$$

Once the maximum extension has been reached, a full replacement is performed. This ensures a fair comparison of annual life cycle costs across all strategies, with preventive replacement performed when further lifetime extension is no longer feasible.

Perfect condition-based maintenance

This strategy mirrors CBM as it preforms maintenance in the same zones, but optimises the timing of the intervention. Rather than initiating maintenance immediately upon threshold crossing, the action is scheduled during the last feasible weather window within the respective zone. This approach represents the theoretical optimum for CBM, with the objective of maximising component utilisation while minimising lifecycle costs.

4.2.4. Operational constraints

Offshore maintenance operations are strongly influenced by environmental conditions, particularly wave height and wind speed. In this study, SCADA historical SCADA wind data recorded at each turbine site is used to simulate operational feasibility, under the assumption of perfect forecasts. Unfortunately, wave height data are not available in the dataset. However, wave and wind conditions are known to be correlated, as seen in Gintautas et al. [32]. Consequently, this study assumes that periods of high wind coincide with high-wave conditions. This assumption likely leads to an overestimation of maintenance accessibility, and in real-world conditions the wave forecast should be included in future research.

Furthermore, wind data is only available up to the observed failure point. Beyond this point, wind conditions are synthetically modelled by repeating the preceding wind profile. Specifically, this means that after failure, the wind time series is reset to its initial value and loops from the beginning.

Wind data were taken with a resolution of one hour, consistent with the approach of Hu et al.[92]. Since the wind speed is measured with the hub height and the constraints are with a height of 10 m, the wind speed is adjusted to the height of 10 m using the logarithmic wind profile, as detailed in the Appendix A.1. The hub height in these calculations is assumed to be 80 meters.

Accessing window

This thesis adopts the first version model of Hu et al. [92], as illustrated in Figure 4.6. For this model, the jack-up time is included, but this only applies when a jack-up vessel is used. The technician transfer phases are integrated into the turbine maintenance period, which means that transfer is assumed to occur during the maintenance window rather than separately.

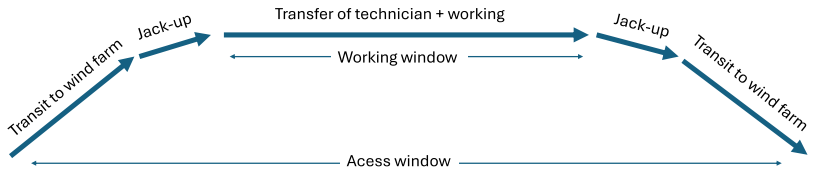


Figure 4.6: Accessing window, adapted from: [92]

Vessels

The vessels used in this thesis are selected based on the required maintenance intensity and capabilities, as summarised in Table 4.2. Crew Transfer Vessels (CTVs) are the lowest ranked vessels, suitable for minor maintenance activities. For more complex and resource-intensive tasks, Service Operation Vessels (SOVs) are used for major maintenance. Jack-up vessels are required for preventive and corrective replacements because they can lift up the blades and hub, needed for bearing replacements. Heavy lifting vessels are capable of performing the most extensive offshore operations. Although they are technically suitable for replacement of main bearings, they are considered excessive and are not economically justified for this purpose. Therefore, they are not used for main bearing repairs in this study. How these vessels are modeled is explained bellow.

Crew Transfer Vessel and Service operating vessel

CTVs and SOVs operate in 12-hour shifts, from 07:00 to 19:00 [92]. At the end of each shift, the vessel returns to the port. The effective working time at the turbine is reduced by the transit duration, and if maintenance is not completed within one shift, it resumes on the next available weather window. Weather suitability is required during the departure and return periods. Specifically, wind speeds must remain below the operational threshold from 07:00 until the vessel leaves the harbour, and again from the time it leaves the turbine until 19:00. Partial deployment, such as delayed departure or early retrieval due to weather, is not considered in this study.

Jack-up Vessel

In contrast to CTVs and SOVs, jack-up vessels operate continuously in 24-hour shifts, with technicians working in three rotations to enable uninterrupted repair activities [93]. A valid weather window is defined as a continuous period during which all operational phases can be completed sequentially without interruption. This differs from CTVs and SOVs, where weather constraints apply primarily during transit and technician transfer.

Vessel Type	Rank	Maintenance type
CTV (Crew Transfer Vessel)	I	Minor maintenance
SOV (Service Operation Vessel)	II	Major maintenance
Jack-up Vessel	III	Preventive and corrective replacements
Heavy Lift Vessel	IV	Used for the highest capacity maintenance tasks, not aplicible for rotor bearings

Table 4.2: Vessel hierarchy and associated maintenance tasks for main bearings [16]

4.2.5. Cost breakdown

Total costs

The formulation of the total maintenance cost is adapted from Borsotti et al. [16], where the total cost C_{tot} associated with a maintenance action a is given by:

$$C_{\text{tot}}(a, T) = C_{\text{mat}}(a) + C_{\text{tech}}(a) + C_{\text{dt}}(T) + C_{\text{trans}}(a) \tag{4.14}$$

This equation defines the total maintenance cost $C_{\text{tot}}(a, T)$ for a maintenance activity a at time T. It includes four components: material cost $C_{\text{mat}}(a)$, technician cost $C_{\text{tech}}(a)$, downtime cost $C_{\text{dt}}(T)$, and vessel transfer cost $C_{\text{trans}}(a)$. Each is detailed in the following subsections.

Material cost

It is assumed that material costs depend only on the type of maintenance performed. Smaller maintenance activities require less materials, and failure leads to consequential damage, which results in a higher cost for corrective maintenance compared to preventive maintenance [31]. For any maintenance action a, the material cost is defined as:

$$C_{\text{mat}}(a) = \begin{cases} C_{\text{minor}}, & \text{if } a = \text{minor} \\ C_{\text{major}}, & \text{if } a = \text{major} \\ C_{\text{prev}}, & \text{if } a = \text{preventive} \\ C_{\text{corr}}, & \text{if } a = \text{corrective} \end{cases} \tag{4.15}$$

Here, $C_{\text{minor}}, C_{\text{major}}, C_{\text{prev}}$, and C_{corr} represent the fixed material costs associated with minor repair, major repair, preventive replacement, and corrective replacement, respectively.

Technician cost

Technician costs are proportional to the number of personnel and time required to complete the maintenance. The total technician cost is computed as:

$$C_{\text{tech}}(a) = N_{\text{tech}}(a) \cdot R_{\text{hour}} \cdot T_{\text{repair}}(a) \tag{4.16}$$

where $N_{\text{tech}}(a)$ is the number of technicians required for the maintenance type a, R_{hour} is the hourly wage per technician and $T_{\text{repair}}(a)$ is the duration of repair in hours for the task.

Downtime cost

Downtime costs result from lost electricity production during maintenance. In this thesis, no flexible production are taken into account for this. Therefore, the downtime cost is calculated based on the average power output of the dataset and the electricity market price:

$$C_{\text{dt}}(T) = P_{\text{avg}} \cdot C_{\text{el}} \cdot T_{\text{dt}} \tag{4.17}$$

where P_{avg} is the average power of the turbine, C_{el} is the cost of electricity per MWh, and T_{dt} is the total downtime in hours.

Vessel transfer cost

The cost of transfer of the vessel includes mobilisation, fuel for travel, and daily operating costs. The cost structure depends on the type of maintenance activity and the required vessel for this activity.

For replacement activities (both preventive and corrective), which require a jack-up vessel to remain on-site throughout the repair, the transfer cost is given by:

$$C_{\text{trans}}(a) = C_{\text{mob}}(a) + 2 \cdot D_{\text{travel}} \cdot F(a) + \frac{T_{\text{repair}}(a) + 2 \cdot T_{\text{travel}}(a) + 2 \cdot T_{\text{jack-up}}}{24} \cdot R_{\text{day}}(a) \tag{4.18}$$

where $C_{\text{mob}}(a)$ is the mobilisation cost, D_{travel} is the one-way travel distance, $F(a)$ is the fuel consumption per kilometre, $R_{\text{day}}(a)$ is the daily vessel rate, $T_{\text{jack-up}}$ is the time required for jacking up and down, and $T_{\text{repair}}(a)$ is the time required to complete maintenance task a. The travel time $T_{\text{travel}}(a)$ is calculated as:

$$T_{\text{travel}}(a) = \frac{D_{\text{travel}}}{v_{\text{vessel}}(a)} \tag{4.19}$$

where $v_{\text{vessel}}(a)$ is the average speed of the vessel used for task a.

For minor and major maintenance, which are executed using CTVs or SOVs that return to port daily, the vessel cost accounts for multiple operational days and is calculated as:

$$C_{\text{trans}}(a) = C_{\text{mob}}(a) + 2 \cdot D_{\text{travel}} \cdot F(a) \cdot N_{\text{days}}(a) + N_{\text{days}}(a) \cdot R_{\text{day}}(a) \tag{4.20}$$

Here, $N_{\text{days}}(a)$ is the number of days required to complete the repair, estimated from available shift time and access duration as:

$$N_{\text{days}}(a) = \frac{T_{\text{repair}}(a) - 2 \cdot T_{\text{travel}}}{12} \tag{4.21}$$

where T_{travel} is the one-way travel time in hours and $T_{\text{repair}}(a)$ is the time required to complete maintenance task a.

Annualized cost evaluation

To assess the cost-effectiveness of each maintenance strategy, total costs are converted into annualised costs. The annualized maintenance cost is calculated as follows:

$$C_{\text{annual}} = \sum_{n=1}^{N_{rapair}+1} \frac{C_{\text{tot}}(a)}{\mathcal{L}_0 \cdot \text{life\_ext}(a)} \cdot 365 \tag{4.22}$$

where:

- C_{annual} is the annual maintenance cost in euros.
- $C_{\text{tot}}(a)$ is the total cost of a single maintenance action a, including material, labour, downtime, and transport costs.
- $\mathcal{L}_0$ is the lifetime at the moment of the first maintenance action.
- life_ext(a) is the fractional life extension achieved by maintenance action a.
- N_{repairs} is the total number of identical maintenance actions applied before reaching the maximum allowed lifetime extension. One preventive replacement activity follows after these repeated actions.

Summary

This chapter introduces a two–part framework that links data-driven condition monitoring to maintenance economics: (i) the construction of a HI and (ii) its use in different offshore maintenance strategies.

(i) Health indicator: An LSTM AE is used to create the HI, trained on healthy data to learn normal turbine behaviour. Each sensor window passes through the encoder, is compressed into a latent space, and is then reconstructed by the decoder. The LSTM layer captures slow degradation trends while taking the operating conditions as extra input, so that these operating conditions are used informatively, but not reconstruction. The weighted reconstruction error of selected sensors, chosen on the basis of domain knowledge, forms the reconstruction-error HI. Because each turbine has unique behaviour, the reconstruction-error HI is z-score normalised with the mean and standard deviation of the healthy period, supporting consistent threshold selection across turbines. An anomaly is flagged when the normalised HI exceeds k standard deviations, in accordance with Chebyshev's inequality. After detection, the HI is assessed with three metrics: monotonicity, trendability, and the length of the P–F interval.

(ii) Maintenance: These HI provide valuable information for maintenance decisions. In order to quantify the effect of this information on increasing lifetime and reducing cost, the HIs will be integrated in maintenance planning. One HI will be taken as a reference to define thresholds that the degradation curve into four zones (A–D). These thresholds are defined by dividing the maximum HI of the reference turbine into equal thirds, with the lowest threshold matching the anomaly point P. Each zone triggers an escalating intervention: minor repair (A), major repair (B), preventive replacement (C) or corrective replacement (D). Minor or major repairs reset the HI and can therefore be applied

repeatedly until the maximum lifetime extension is reached. After this a full replacement is required. The framework supports four strategies: *corrective* performs maintenance after failure in Zone D, *preventive* replaces the bearing on entry to Zone C, *CBM* raises a work order in TH_{minor} and performs the appropriate action for the zone reached on the service date, and *perfect CBM* follows the same zone logic but defers action to the last feasible weather window. A life-cycle cost model sums material, labour, downtime, vessel cost, and provides schedules dates.

The next chapter introduces the case study used to train and evaluate the LSTM-AE model. The full methodology is then applied to this case study, with the corresponding results presented in Chapter 6.

5

Case Study Dataset

High-quality open-source dataset including failure logs are scarce in the wind industry, the recently published CARE to Compare dataset [44] is a notable attempt to close this gap. This dataset is a comprehensive and anonymised collection of SCADA time series and failure logs from three offshore wind farms. Wind farm B is selected from the three wind farms introduced, as it includes three documented rotor bearing failures across nine turbines over one year. Three failures may seem modest, yet a trio of comparable, well-documented rotor-bearing events offers a richer test bed than most other open-source datasets. These cases have not yet been broadly analysed in the literature. The chapter further introduces the dataset in Section 5.1 and documents the preprocessing steps taken for the data from wind farm B in Section 5.2.

5.1. Dataset description

CARE to Compare is a benchmark dataset of real-world SCADA time series and failure logs from three wind farms (wind farm A, B, C), recently published by Gück et al. in November 2024 [44].Wind farm A, also known as the EDP dataset, is a high-quality offshore dataset located in Portugal. EDP has been extensively used in previous research, such as by Jankauskas et al. [24] and Udo et al. [26]. In contrast, Wind Farms B and C, situated offshore in Germany, are newly introduced and remain largely unexplored. Table 5.1 provides an overview of the three wind farms and end their key characteristics such as number of turbines, anomalies, and sensors.

Data have been anonymised by removing location details, turbine specifications, manufacturer information, and other identifying factors. For example, the rated power is unknown as it is normalised, and wind turbines are given a generic ID such as 1. Additionally, the timestamps have been shifted so that all datasets start within the year 2022. Seasonal consistency has been ensured.

The wind farms are made up of cases, each containing approximately one year of SCADA data recorded at 10-minute intervals. At the end of each case there is an event window which is labelled as either anomalous or normal, depending on whether a failure occurs. For failure events, the end time corresponds to the point of failure, while the start time reflects the best estimate by Gück et al. of the earliest possible point of detection. These timestamps were determined using a combination of operator feedback, service report documentation, and model-based analysis [44]. Furthermore, it is possible for cases with the same turbine to overlap, which requires additional care during evaluation.

Table 5.1: Summary of the three Wind Farms in CARE to Compare [44]

	Wind Farm A	**Wind Farm B**	Wind Farm C
Turbines	5	**9**	22
Datasets	22	**15**	58
Anomaly events	11	**6**	27
Normal behaviour	11	**9**	31
Features	86	**257**	957
Sensors	54	**63**	238

The selection of the wind farm for this thesis focused on the newly introduced datasets. After evaluating wind farms B and C, wind farm B is chosen due to the occurrence of documented failures. Although limited in number, the wind farm includes three failures related to the rotor bearings: two failures on rotor bearing 2 and one on rotor bearing 1. As confirmed by the authors of the CARE dataset, all three bearing failures were critical bearing failures. In addition to bearing failures, there were three reports of high temperatures in the transformer cell. These are not explored in this thesis.

Wind Farm B consists of nine turbines and fifteen cases, as summarised in Table 5.2. For this wind farm, the anomalous and normal cases from the same turbine show significant overlap. The anomalous case spans a longer period, including several additional months of data, while the normal case sometimes contains earlier data. To ensure consistency during model training, these cases are combined, where applicable. The cases are referred to by the larger dataset to avoid confusion, i.e. case 7, 34, 19, 27, 53, 77. Case 27, case 53, and case 77 have a rotor bearing failure and are considered faulty. The remaining turbines are considered healthy and used for training, including those with high transformer cell temperatures, as these are not expected to influence rotor bearing performance.

The rotor bearing temperatures for the three failure cases are shown in Figure 5.1. The failure of rotor bearing 2 from case 77 is the most clearly visible, showing a distinct rise in temperature. Case 53, also involving the rotor bearing 2, also exhibits a noticeable temperature increase. In case 27, which concerns rotor bearing 1, the temperature anomaly is less pronounced. However, when comparing the temperatures of both rotor bearings, a deviation becomes apparent. The event start and end times shown in Figure 5.1 refer to the CARE authors' best guess of early detection. For case 27, 53, 77 these lengths are 41, 61 and 60 respectively. Notably, a temperature increase is observable even before the indicated start time.

The features of the Wind Farm B dataset are derived from 63 sensors, each capturing four statistical metrics: mean, minimum, maximum, and standard deviation. These sensors can be categorised into four primary groups according to their function [94]:

1. **Condition parameters:** These include wind speed (measured by multiple anemometers), wind direction, and ambient temperature and are useful to assess external conditions influencing turbine performance and power generation.
2. **Health Parameters:** These include measurements such as bearing temperatures, gearbox temperatures, generator temperatures, and vibration levels. Monitoring these parameters helps assess the physical integrity of the turbine and detect early signs of mechanical wear or failure.

Table 5.2: Overview of Wind Farm B, its wind turbine assets and their associated cases

Turbine ID	Cases	Failure
0	21	No
2	13	No
5	82	No
13	7 (+ 2)	High transformer cell temperature
14	34 (+ 52)	High transformer cell temperature
11	19 (+ 74)	High transformer cell temperature
7	27 (+87)	Rotor bearing 1
6	53 (+23)	Rotor bearing 2
12	77 (+86)	Rotor bearing 2

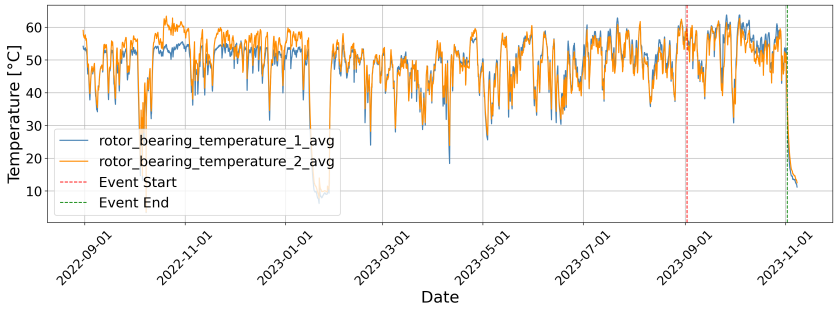
3. **Performance parameters:** This group includes parameters that measure operational performance such as rotor speed, generator speed, grid voltage and frequency, active and reactive power, and electrical currents.
4. **Controlling parameters:** This category includes measurements related to automated control systems of the turbine, including pitch angle, nacelle direction, and inverter torque.

The power curve for the raw data from Wind Farm B, specifically for turbine 13 (case 7), over the course of one year, is shown in Figure 5.2. As observed, a significant portion of the turbine's operation appears to be curtailed. This is evident from deviations below the expected power curve, where the turbine generates less power than theoretically possible for the given wind speed. This pattern of curtailed operation is also observed in other turbines, with a similar portion of data affected, though not necessarily during the same periods. Additionally, it becomes apparent from the image that some active power values exceed 1 or fall below 0, even though the power has been normalised by the rated power to preserve anonymity. This phenomenon is discussed in more detail in Subsection 5.2.5.

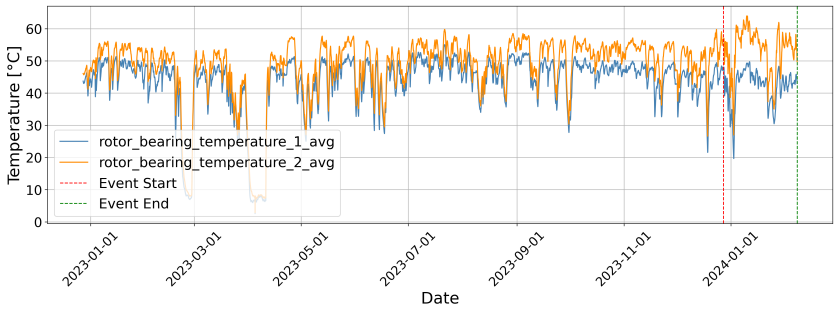
5.2. Data preprocessing

Effective data preprocessing is essential to build a model that accurately represents healthy turbine behaviour. This section outlines the key steps taken to prepare the SCADA dataset for analysis and modelling, specifically the steps listed below:

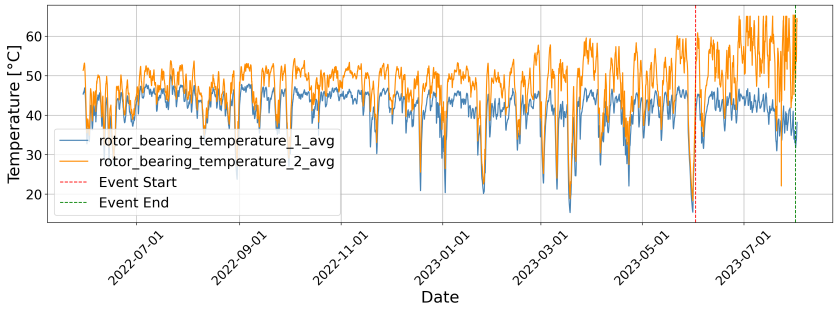
1. Feature selection
2. Data cleaning
3. Resampling
4. Feature reduction
5. Normalise data
6. Divide data into train and test sets



(a) Case 27, Asset 7, rotor bearing 1 failure



(b) Case 53, Asset 6, rotor bearing 2 failure



(c) Case 77, Asset 12, rotor bearing 2 failure

Figure 5.1: Rotor bearing temperature 1 and 2 plotted for the three failure cases

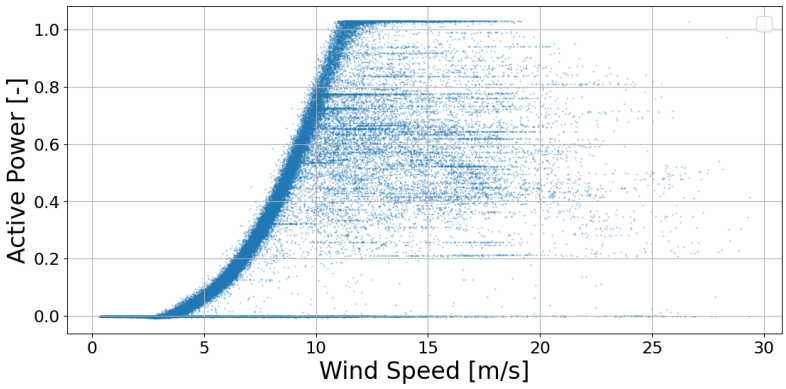


Figure 5.2: Raw data power curve for Case 7, Asset 13

Table 5.3: Selected SCADA variables and corresponding ranges

Sensor Name	Description	Range	Units
sensor_8	Outside temperature	[-5, 40]	°C
sensor_22	Hub temperature	[0, 50]	°C
sensor_25	Rotor speed	>0	rpm
sensor_51	Rotor bearing temperature 1	[33.4, 120]	°C
sensor_52	Rotor bearing temperature 2	[38.1, 120]	°C
wind_speed_61	Wind speed	>0	m/s
power_62	Active power	>0	-

5.2.1. Feature selection

The original SCADA dataset comprises 257 features collected from 63 sensors. This needs to be reduced to reduce dimensionality and improve model performance. As a first step, only 10-minute average values were retained, while maximum, minimum, and standard deviation values were excluded.

The initial narrowing of features is guided by expert knowledge and previous studies, following Encalada-Dávila et al. [41] and Yang et al. [62], who identified key indicators for rotor bearing faults. For example, Yang et al. [62] emphasised the importance of the miscorrelation between the main bearing temperature and the rotor speed or the generated power as a potential indicator of bearing failures. Certain useful signals, such as the low speed shaft temperature, grease level, and external bearing temperatures, were not available in the SCADA dataset. The hub temperature is included as an alternative to the external bearing temperatures, as it is closest. Wind speed and outside temperature are included as condition parameters.

The dataset also included the vibrational signals drive train vibration axis Z, tower vibration axis X, tower vibration axis Y, with z being the vertical direction. All three signals were plotted in the time leading up to the failure, and no significant improvement of the peak vibrations were visible. Therefore, these features were excluded from the model. A full list of the selected features, along with their descriptions, typical ranges, and units, is provided in Table 5.3.

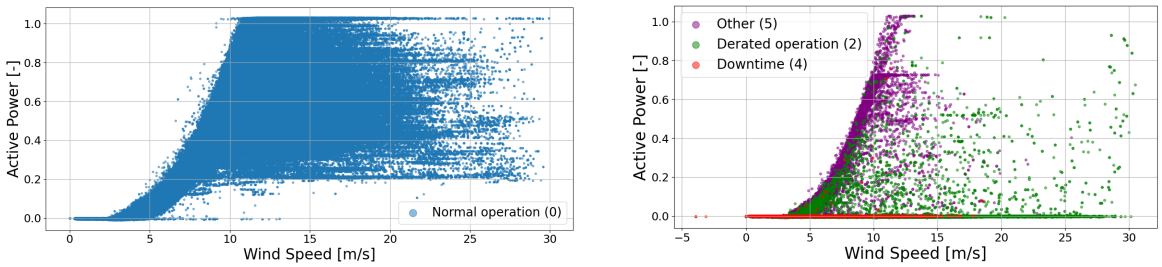
5.2.2. Data cleaning

Data cleaning is performed to remove non-physical values, outliers, and retain only representative operating conditions. Anomalies, data points that deviate significantly from typical patterns, can distort relationships within the data set and introduce bias into predictive models [95]. Such anomalies may arise from entirely different mechanisms from those that govern normal behaviour, which makes it essential to recognise and exclude them [95]. However, this process must be carried out with care to avoid removing significant variations. The next paragraphs describe the data cleaning steps taken in this thesis.

1: Normal Operating Data
Normal operating data are selected on the basis of predefined labels in the CARE dataset. Only data recorded under normal operating conditions are retained, specifically status codes 0. Data associated with status codes 2 (derated operation), 3 (Service), 4 (downtime) and 5 (other) are excluded. In the

dataset operating mode 1 (idling) and 3 (service) do not have any data points. The power curves for normal and abnormal operations are plotted Figure 5.3, only the data points from Figure 5.3a are retained and all points in Figure 5.3b are excluded.

Some non-representative values still remain in the dataset. Specifically, standstill data is removed because it causes a drop in bearing temperatures, which is difficult for the model to predict accurately. In contrast, data corresponding to curtailed operation is retained, as curtailment occurs frequently in the dataset. Excluding it would result in the loss of a substantial portion of data and disrupt the time sequences that are essential for training the LSTM model.



(a) Normal operating modes

(b) Abnormal operating modes

Figure 5.3: Power curves separated by operational mode, for the raw data

2: Missing Values
In the CARE dataset, missing values are recorded as zeros [44], which requires careful handling. It is observed that in cases where certain values, such as rotor bearing temperature, were zero when they physically cannot be, other sensor values were also recorded as zero. These values were excluded from the dataset. Fortunately, these instances are limited to only ten.

3: Physical Inconsistencies and Outliers
Martí-Puig et al. [96] investigated the effects of outlier removal in wind turbine fault diagnostics by comparing the performance of Extreme Studentised Deviate, quantile-based, and Hampel filters applied to SCADA data. Their study demonstrates that systematically removing statistical outliers can negatively impact model generalisation, as failure events often appear as statistical anomalies. They conclude that a more effective approach is to use manually defined static or dynamic value ranges for filtering rather than relying solely on automated statistical methods. Additionally, if required, the Extreme Studentised Deviate filter is found to offer the best results.

For this reason, outliers are identified based on predefined operational ranges. The outside temperature range is set from -5°C to 40°C, based on Encalada-Dávila et al. [41]. They also definned an upper limit for rotor bearing temperatures of 120°C. Due to the lack of precise operational specifications, no upper limits were applied to rotor speed or wind speed. Active power is excluded from this filtering step, as it will be addressed later through renormalisation, as discussed in Subsection 5.2.3. The hub temperature range is conservatively set between 0°C and 50°C, based on inspection of the dataset.

Finally, rotor bearing temperatures were filtered more strictly since the model has difficulty accurately predicting low temperature values that result from, for example, start-up after periods of standstill. Since the reconstruction of these predictions will form the basis of the HI, this inaccuracy introduced unwanted noise. Therefore, they were bounded using the Extreme Studentised Deviation method following the recommendation from Mart-Puig et al. [96]. Specifically, values below the mean minus

two standard deviations were removed. This corresponds to lower bounds of 33.4°C and 38.1°C for rotor bearings 1 and 2, respectively. All selected operational ranges are summarised in Table 5.3.

Outliers are flagged as NaN and data columns recorded under abnormal operating conditions are excluded, as they do not reflect typical turbine behaviour. After filtering for normal operating conditions, removing missing values and interpolating outliers and physically inconsistent entries, the data set is reduced from 859,065 to 660,922 data points. This represents a reduction of approximately 23.1 % from the original dataset, although this figure is further reduced after resampling as is discussed in Subsection 5.2.3.

5.2.3. Resampling

Rotor bearing temperature is the main health parameter used in this research, which is expected to gradually increase over time leading to failure. Therefore, preprocessed signals are resampled at six-hour intervals, as used by Milani et al. [50]. This resampling allowes for significantly faster model training as a result of the reduced number of data points. In addition, it helpes decrease the ratio of missing data points, which is important for the temporal aspect of the model. Different resampling rates are tried, as high sampling rates can negatively impact model output quality [50]. The 6 hours resampling window resulted in 20,653 data points, corresponding to a data loss of 13. 5 %, compared to 23. 1 % before resampling.

5.2.4. Feature reduction

The selected features from Subsection 5.2.1 are tested for correlation, as the model can have difficulty with highly correlated inputs. A correlation matrix is computed for the selected features using the Pearson correlation coefficient, as shown in Figure 5.4. Features with a correlation coefficient greater than 0.9 are considered too strongly correlated to be included together, as this could reduce the model's ability to generalise. Between the outside temperature and the hub temperature, only the outside temperature is kept.

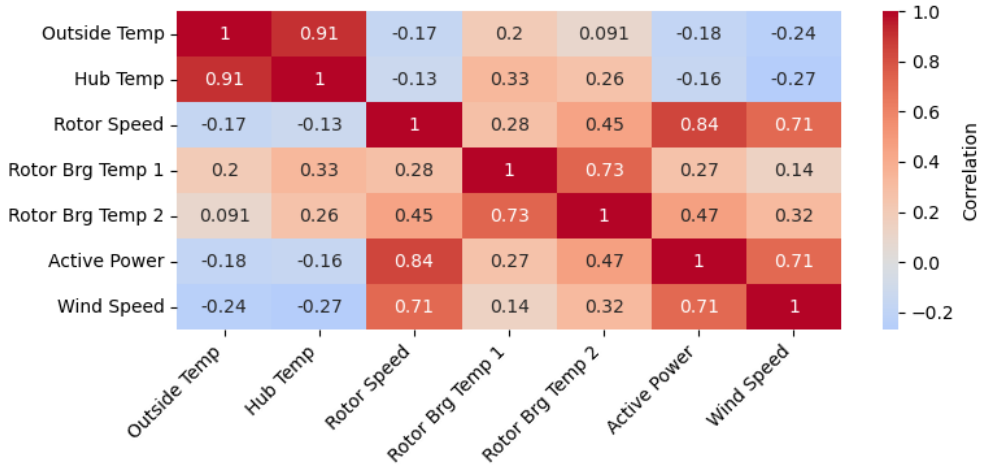


Figure 5.4: Correlation matrix after cleaning for all selected features

5.2.5. Normalise data

Normalisation techniques are applied to scale the data within a predefined range, typically [0,1]. This transformation ensures that all features contribute proportionally to the model, preventing variables with larger numerical ranges from dominating those with smaller ranges. In this research a Min-Max normalisation is performed using the following formula [97]:

$$X' = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{5.1}$$

where:

- X is the original feature value,
- X_{min} and X_{max} are the minimum and maximum values of the feature, respectively,
- X' is the normalized value.

The maximum values used for normalisation are taken from the entire dataset, not per turbine. This ensures that the same absolute values are interpreted consistently across different turbines, both before and after normalisation. It assumes that all turbines operate under similar conditions, which means that a specific value represents the same level of performance or stress for each turbine.

Power normalising

Although the dataset is already normalised with respect to the rated power, 24% of the data contained values outside the expected range of 0 to 1 as can be seen in Figure 5.2. According to the authors of CARE, the values above one are instances where the power is greater than the rated power, and the instances below zero are due to standby consumption. Values below 0 are already deleted, as standstill data is deleted in Subsection 5.2.2. The power values are again normalised to have the maximum value of 1.

5.2.6. Division into train and test sets

The dataset used in this study consists of six cases of healthy wind turbine operation, each representing roughly one year of data. This represents a relatively limited amount of data. To maximise the amount of usable data for model testing, healthy segments of faulty turbine data are also included in the test set.

For training, the healthy part of the three faulty cases is used. These cases cannot be used for training, as they are needed to perform z-score normalisation on the corresponding turbine data. Indications of main bearing faults are consistently found to be faulty 5 to 6 months prior to failure [49]. For that reason, all data up to six months before failure in the faulty turbines is selected for testing.

This data split allows for an extended test set, which improves model performance. Although the train set may not be fully representable compared to random splits, it is acceptable for the purpose of overfitting detection. The test set is mainly used for monitoring generalisation during training, with early stopping applied to prevent overfitting. A growing performance gap between training and test data is interpreted as a sign of overfitting. Model performance is evaluated on the basis of the behaviour of the HIs, and not with the train set.

Summary

This chapter documents the preparation of the CARE to Compare Wind Farm B dataset for rotor-bearing fault analysis. CARE to Compare is a benchmark of real-world SCADA time series and failure logs from three anonymised offshore wind farms. From this benchmark, Wind Farm B which is one of the two newly released German farms is selected because it contains three confirmed critical rotor-bearing failures, which remain largely unexplored in the literature. The wind farm consists of nine turbines. In this wind farm, there are 15 cases, each case supplies about one year of ten-minute SCADA measurements followed by a labelled event window. The overlab within the same turbines is merged with a result of nine cases. The cases with a rotor bearing failure and the end are deemed as faulthy, and the other six are healthy and used for model training.

Of the 257 original SCADA variables (63 sensors each providing four statistics), only the 10-minute means are retained. Expert judgement then reduced the feature set to seven key signals: ambient and hub temperatures, rotor speed, wind speed, active power, and the two rotor bearing temperatures. Data cleaning involved three steps: (i) removal of abnormal operating modes, (ii) deletion of missing values (encoded as zeros) and physically impossible readings, and (iii) exclusion of outliers by applying static operational ranges together with the Extreme Studentised Deviate test for low bearing temperatures. These operations reduced the data set from 859 065 to 660 922 rows. To improve data coverage and reduce the computational load, the cleaned series are resampled at six-hour intervals, limiting overall data loss to 13.5 % compared with the 23.1 % discarded during the initial cleaning. The coarser resolution still preserves the gradual temperature rise that is expected with a rotor bearing failure.

To prepare the data for modelling the highly correlated features are removed, keeping outside temperature instead of hub temperature. The outside temperature and wind speed now serve as the operating conditions variables fed to the LSTM cells informed of the operating conditions introduced in Subsection 4.1.1. All variables are then min–max normalised with global minima and maxima, and active power was normalised again as significant values are above 1. The training set contains every healthy segment from the six fault-free turbines, while the test set combines their remaining data with the healthy parts of the three fault cases, each truncated six months before the failure point. This split maximises the training data, while the test data are still valuable in detecting overfitting of the model. In the next chapter, this case is integrated into the method presented in Chapter 4 and the corresponding results are shown and discussed.

6
Results and Discussion

This chapter reports the results of the methodology introduced in Chapter 4 when applied to the case study described in Chapter 5. First, Subsection 6.1.1 evaluates the HI obtained from the LSTM AE. Next, Section 6.2 integrates HI into a multi-threshold CBM strategy and compares it against other strategies. The chapter closes with a discussion of performance, implications and limitations in Section 6.3, with particular emphasis on Section 6.3.

6.1. Health indicator

This section details how the HI is generated and evaluated. Subsection 6.1.1 begins by revisiting the case-specific model design. After which the subsection quantifies the AE's reconstruction performance, which forms the basis for computing the reconstruction-error HI also discussed in the same subsection. Finally, Subsection 6.1.2 presents the z-score–normalised HIs for both faulty and healthy turbines and assesses their performance and suitability for early warning diagnostics and maintenance planning.

6.1.1. Autoencoder

Case-specific model configuration

The LSTM–AE is tailored to the present case study through the parameters listed in Table 6.1. As detailed in Chapter 5, four SCADA sensors are selected (*rotor-bearing temperature 1*, *rotor-bearing temperature 2*, *active power*, and *rotational speed*), and two operating condition variables (*ambient temperature* and *wind speed*). For the AE, these can be represented as:

$$\mathbf{x}_c^t = \begin{bmatrix} T_{\text{rotor1}} \\ T_{\text{rotor2}} \\ P_{\text{active}} \\ \omega \end{bmatrix} \in \mathbb{R}^{F=4}, \qquad \mathbf{o}_c^t = \begin{bmatrix} T_{\text{amb}} \\ v_{\text{wind}} \end{bmatrix} \in \mathbb{R}^{O=2}.$$

The AE employs hidden LSTM layers with an 128 hidden dimension, to process the four input features ($F = 4$). Their output is projected onto a latent space of dimension $G = 2$. $G = 2$ was chosen because $G \ll F$ [27] and because a single latent neuron could not reproduce the full operating behaviour. A dropout rate of 0.4 between the LSTM layers. The Adam optimiser minimises the MSE loss.

Table 6.1: Model hyperparameters and training configuration, after being aplied to the Case study

Parameter	Value / Description
Input features (F)	4
Operating condition features (O)	2
LSTM sequence length (T)	250
Sliding window size (n_{sw})	250
Stride (s)	1
Hidden dimension (H)	128
Latent space dimension (G)	2
Dropout rate	0.4
Batch size	24
Max number of epochs	100
Early stopping patience	5
Loss function	MSE
Optimizer	Adam
Activation functions	Sigmoid, Tanh (used within LSTM)

Each training sequence spans $T = 250$ time steps, which is about two months of data. This length enables the LSTM to learn slow degradation trends. To achieve this a sliding window of the same length ($n_{\text{sw}} = 250$) and a stride of $s = 1$ is generated.

Model training

With the hyperparameters fixed, the model is trained to minimised the MSE between each input window x_c^t and its reconstruction $\hat{x}_c^t$. Training automatically stopped after 19 epochs due to early stopping, after 5 epochs of improvement in the validation loss. The progression of MSE loss (Equation 4.7) throughout the training process is illustrated in Figure 6.1, and the final loss values at the best performing epoch are summarised in Table 6.2.

At epoch 14, the model achieved a training MSE of 0.00225 and a validation MSE of 0.00229. These low values indicate effective learning of healthy data patterns, while the consistent, small difference suggests good generalisation. It is important to note that the validation set contains data from different turbines, which vary in predictability.

Reconstruction temperatures

The AE model can now predict an reconstructed signal $\hat{\mathbf{x}}_c^t$ from the input signal $\mathbf{x}_c^t$. The underlying assumption of the model is that as the system begins to degrade, the model's ability to accurately reconstruct the input diminishes, resulting in increased reconstruction error. Specificity, rotor bearings temperature 1 and 2 expected to increase. Figure 6.2 illustrates this principle through reconstructed temperature signals for two rotor bearing temperature sensors across the three different failure cases. In each subplot, the actual temperature signal is shown in blue (temperature 1) or grey (temperature 2), while the predicted (reconstructed) signal is shown in red.

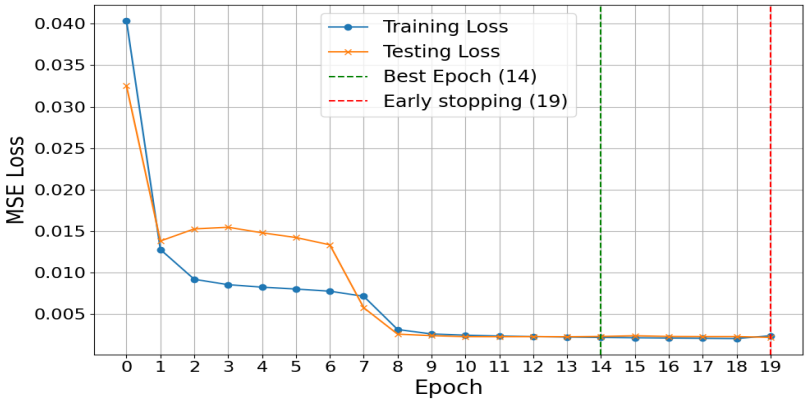


Figure 6.1: MSE loss vs epochs

Metric	Epoch	Training Set	Validation Set
MSE Loss	14	0.00225	0.00229

Table 6.2: Final mean squared error (MSE) loss values at epoch 14.

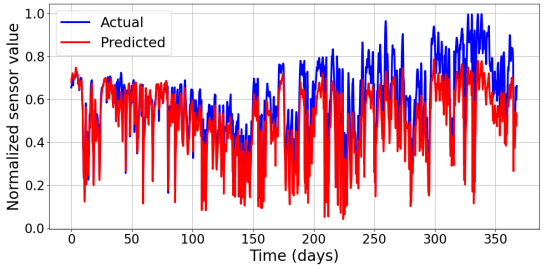
Case 53 and case 77 both experienced failures in rotor bearing 2, with the resulting increase in the actual temperature 2 values towards the end of the time series, as can be seen in Figure 6.2d and Figure 6.2f. The AE performs well during the earlier, healthy phase of operation, accurately reconstructing both temperature signals. However, as degradation begins, the model struggles to follow the upward trend in temperature 2. Interestingly, the model significantly overestimates temperature 1 in these instances, as shown in Figure 6.2c and 6.2e. Additionally, the actual temperature 1 signal also shows a general dip toward the end.

Case 27 involves a failure in rotor bearing 1. As shown in Figure 6.2a, the degradation is more pronounced in Temperature 1, and again the model fails to fully capture the increasing trend. In contrast, the model continues to predict Temperature 2 accurately, as depicted in Figure 6.2b.

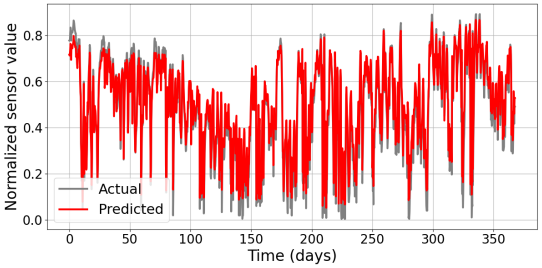
Reconstruction-error health indicator

The reconstruction-error HIs are computed using Equation 4.9 as the daily average of the reconstruction errors across the selected temperature sensors 1 and 2 (Figure 6.2). The sensors are combined to prevent bias towards the failing side, based on the assumption that operators can identify the degrading bearing using temperature measurements. The reconstruction-error HIs resulting are shown in Figure 6.3. In all three cases, the HIs start at low values during healthy operation and gradually increase as the degradation progresses. This upward trend is smooth due to the long LSTM sequence length. As expected, the timing and shape of the increase vary between cases, reflecting differences in the failure modes and the affected components.

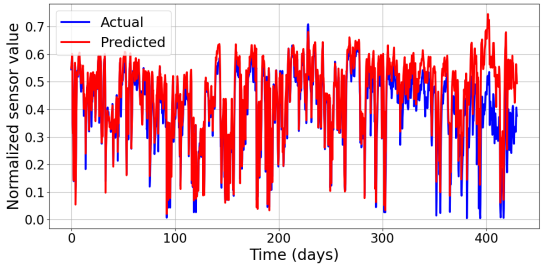
As discussed earlier in this section, temperature 1 of Cases 53 and 77 decreases towards the end of the series, while the model continues to predict higher values. Since the HI is based on the absolute reconstruction error of both temperature sensors, this divergence amplifies the value of the HIs. Case 77 displays the highest absolute HI, due to a larger increase in temperature 2.



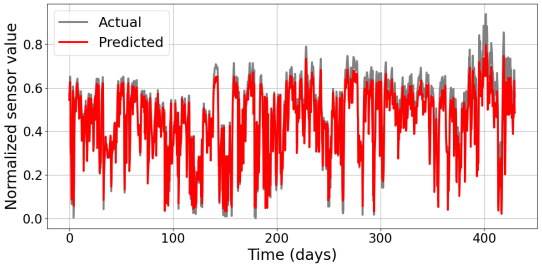
(a) Temp 1 - Asset 7, Case 27



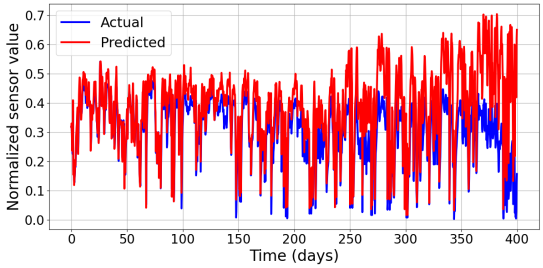
(b) Temp 2 - Asset 7, Case 27



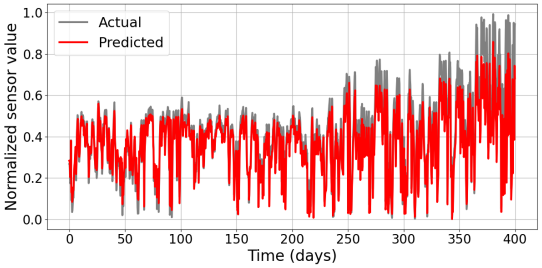
(c) Temp 1 - Asset 6, Case 53



(d) Temp 2 - Asset 6, Case 53

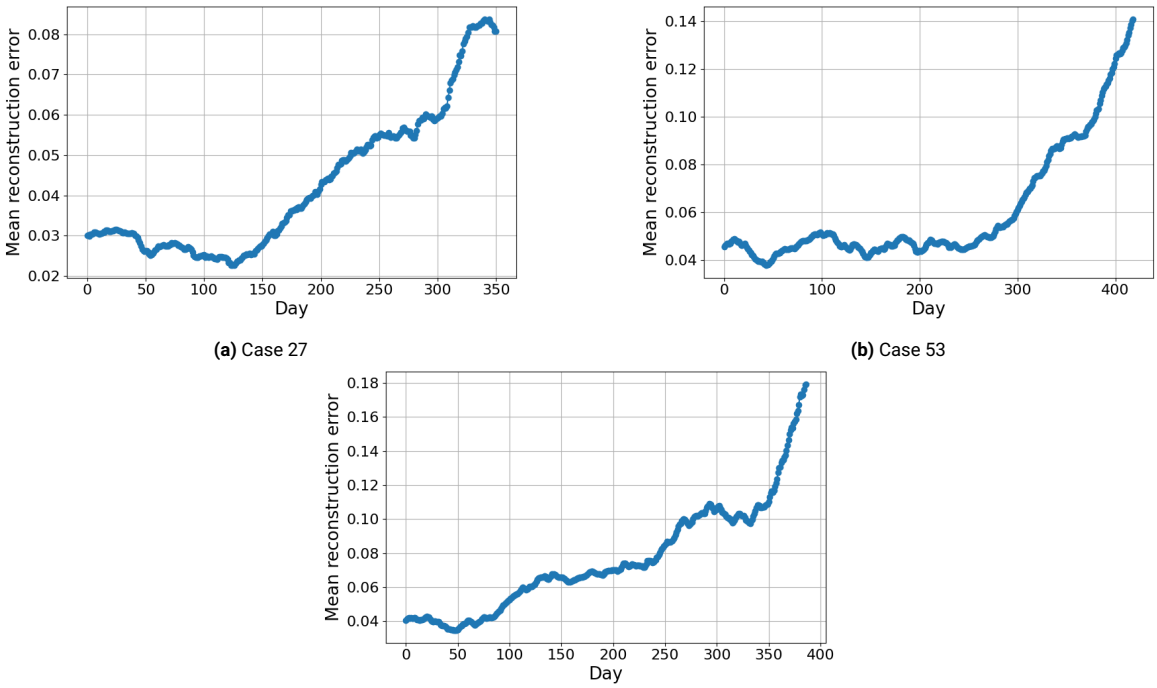


(e) Temp 1 - Asset 12, Case 77



(f) Temp 2 - Asset 12, Case 77

Figure 6.2: Predicted and actual normalized rotor bearing temperatures for all three cases



(a) Case 27

(b) Case 53

(c) Case 77

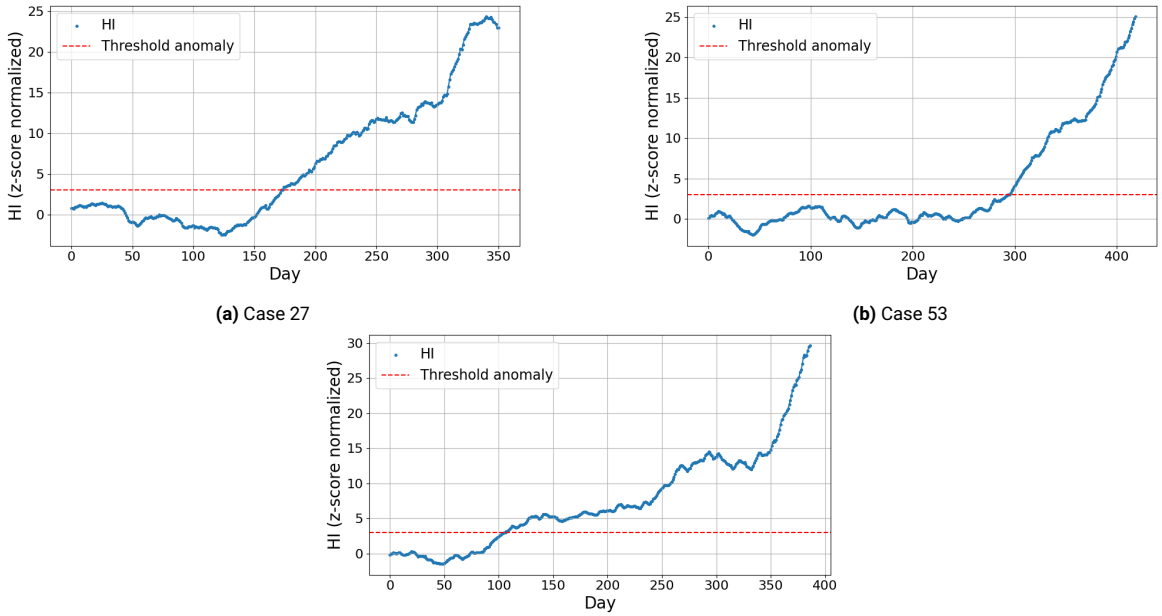
Figure 6.3: Mean reconstruction error of rotor bearing temperature 1 and 2

With the rotor bearing 1 failure, Case 27, the model had no same difficulty in predicting the other rotor bearing temperature. The upward trend in the reconstruction-error HI is mainly due to the increase in the error of the rotor bearing temperature 1, resulting in a lower value of the mean reconstruction error. Despite differences in absolute levels, all three cases exhibit a similar relative increase over time, which supports the application of z-score normalisation for consistent degradation detection.

6.1.2. Health indicator normalised

The first 100 days of turbine operation are assumed to represent the healthy baseline. After applying Z-score normalisation, as described in Subsection 4.1.3, the resulting HIs are shown in Figure 6.4. In these plots, the vertical axis represents the Z-score: a value of 0 corresponds to the mean of the first 100 days, and each unit indicates one standard deviation from this mean.

Anomalies are defined based on Chebyshev's inequality (Subsection 4.1.4), with parameters $k = 3$ and $n = 1$. Setting $k = 3$ corresponds to detecting anomalies at a threshold of three standard deviations above the mean, which is common in literature [90], [91]. In all three failure cases, this threshold is exceeded and not re-entered. For that reason, the value $n = 1$ is chosen. It is also notable that the final HI magnitudes are comparable between cases, with maximum Z scores of 24.3, 25.0, and 29.6 for cases 27, 53, and 77, respectively.



(a) Case 27

(b) Case 53

(c) Case 77

Figure 6.4: Health Indicator Z-score normalised based on the first 100 days

Evaluation health indicator

As discussed in Subsection 4.1.5, the HI is evaluated using the length of the P-F interval, as well as the *monotonicity* ($\mathcal{M}$) and *trendability* ($\mathcal{T}$) metrics after the anomaly threshold is crossed. These results are summarised in Table 6.3 for all three failure cases, along with their averages.

It is crucial for the model to provide early warnings, as this enables wind farm operators to act before anomalies worsen and become more costly to repair. The P–F intervals, which represent the time between the first detection and the actual failure, were 179, 125, and 281 days for Case 27, Case 53, and Case 77, respectively. This results in an average early warning time of 195 days (6.5 months). These results are high compared to CARE's event windows, which is their indication for early detection. These windows were 61, 41, 60 days for Case 27, Case 53, and Case 77, respectively. Notably, Case 53 was the latest detected in both instances. The P-F intervals do align with findings in the literature using vibration and temperature data, where major bearing failures are often detected 5–6 months in advance [49]. Encalada-Dávila et al. [41] also investigated (a single) main bearing failure using only healthy SCADA data for training and were able to detect the fault 106 days in advance. This shows the effectiveness of the model in providing early warnings, even compared to the results reported in the literature.

Furthermore, HI in this study demonstrates a strong and consistent increase trend over time, as reflected in a high average trendability score of 0.99. This value remains consistently high across all three failure cases, showing that the HI reliably increases over time once degradation begins. The average monotonicity score is 0.49, which is realistic, as monotonicity is known to be more sensitive to short-term fluctuations and noise in the signal [27].

Table 6.3: Monotonicity and trendability of HIs after Z-score threshold crossing

Case	Threshold day	P-F interval	Monotonicity	Trendability
27	172	179	0.36	0.99
53	294	125	0.79	1
77	106	281	0.31	0.97
Mean	–	195	0.49	0.99
Mean [28]	–	–	0.38	0.94

After the anamaly threshold is croshed, Case 53 exhibits a very monotonic progression with a score of 0.79 and perfect trendability of 1. Case 77 shows slightly lower values, with a monotonicity of 0.31 and trendability of 0.97, as a result of visible fluctuations in the middle of the HI signal. Case 27 presents a near-perfect trendability at 0.99 and a monotonicity of 0.36. The monotonicity could potentially be improved through post-processing or by evaluating the HI at a lower frequency. For example, weekly rather than daily would increase the average monotonicity to 0.62 by reducing the impact of short-term noise. However, the HI is retained at daily resolution to support maintenance planning.

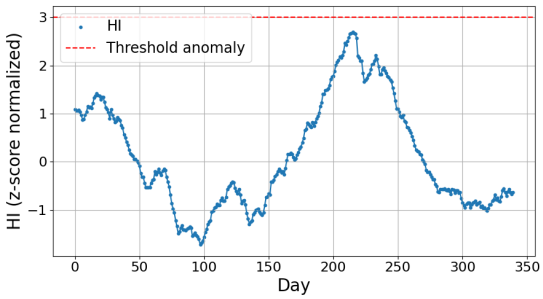
De Pater and Mitici [28] reported a mean trendability of 0.94 and monotonicity of 0.38, confirming that the results of this thesis are good. It is important to note that they investigate different datasets, which can have a great effect on the results. Therefore, a direct comparison is difficult.

Healthy cases

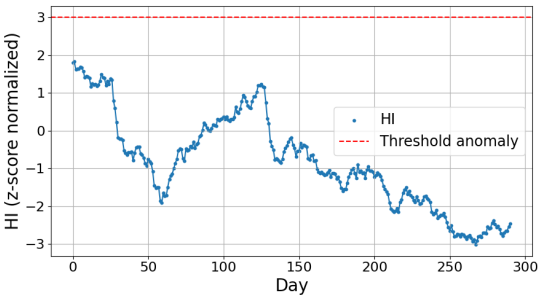
To assess the ability of the model to distinguish between healthy and degrading behaviour, it was also evaluated on data from turbines with no known rotor bearing faults. For each healthy case, the model was trained on the remaining five healthy datasets and then tested on the excluded case to ensure that the test data were not seen during training. The resulting HIs are presented in Figure 6.5. In this, In particular, none of the healthy cases exceed the anomaly threshold, indicating that the model does not raise false alarms under normal operations. In addition, these cases show fluctuating behaviour, which is expected in the absence of progressive degradation.

As in the faulty cases, Z-score normalisation is applied using statistics from the first 100 days, which are assumed to represent healthy behaviour. As a result, the threshold can not be crossed in these early days and can not be detected. In Case 34 (Figure 6.5c), Case 82 (Figure 6.5e), and Case 83 (Figure 6.5f), a gradual upward trend is observed within this initial period. If this upward trend had occurred beyond the 100-day normalisation window,it could have triggered a false alarm.

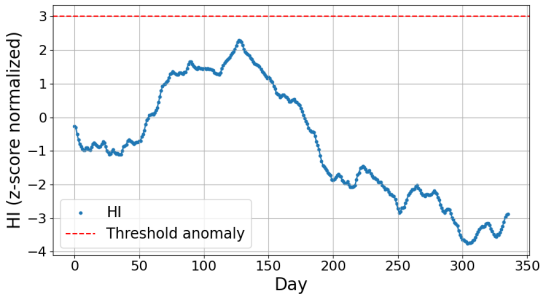
Case 7 (Figure 6.5a), the HI approaches the threshold but remains below it. In general, the fact that all HIs remain below the anomaly threshold confirms both the appropriateness of the chosen threshold and the performance of the model. Moreover, none of the HIs display a clear upward trend over time, which is expected for healthy turbines and further supports the model's reliability in distinguishing between normal and deteriorating conditions.



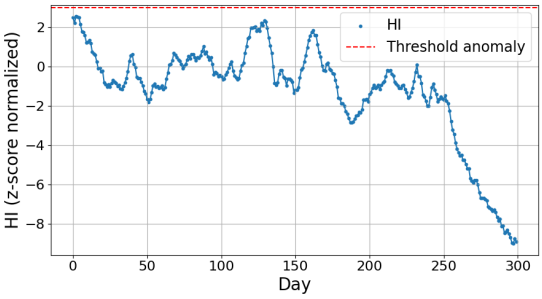
(a) Asset 13 – Case 7



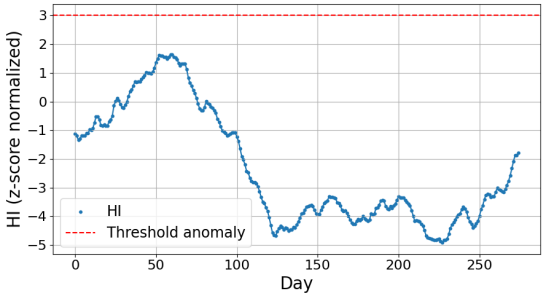
(b) Asset 0 – Case 21



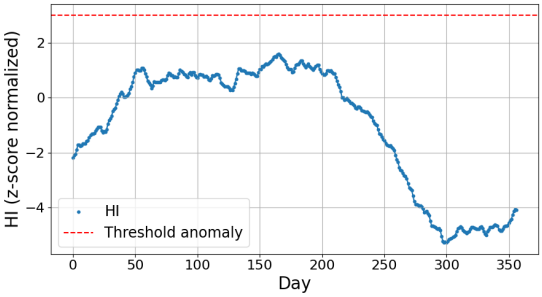
(c) Asset 14 – Case 34



(d) Asset 11 – Case 19



(e) Asset 5 – Case 82



(f) Asset 2 – Case 83

Figure 6.5: HIs for the six healthy turbine, z score normalised

6.2. Maintenance

Given the promising HI results, this section demonstrates their potential in a maintenance context. It first outlines the parameters and assumptions used in the maintenance method in Section 4.2. Next, it presents the CBM planning results in Subsection 6.2.2, followed by a comparison with alternative maintenance strategies in Subsection 6.2.3.

6.2.1. Maintenance parameters

This subsection defines the maintenance parameters and assumptions used in the analysis. These are grouped into five categories: thresholds, wind farm parameters, lifetime assumptions, maintenance task parameters, and vessel-specific parameters.

Thresholds

In Subsection 4.2.2, the thresholds were initially defined based on a single reference turbine. With the inclusion of Case 53 in Subsection 6.1.2, this case was selected as the real reference. It represents an average scenario in terms of maximum HI values across the three failure cases and was therefore chosen to set the threshold values. As illustrated in Figure 6.6, the resulting thresholds are: $TH_{\text{minor}} = 3$, $TH_{\text{major}} = 8.3$, and $TH_{\text{repl}} = 16.7$. The value for TH_{minor} is based on Chebyshev's inequality, using the common choice of $k = 3$, as discussed in Subsection 6.1.2.

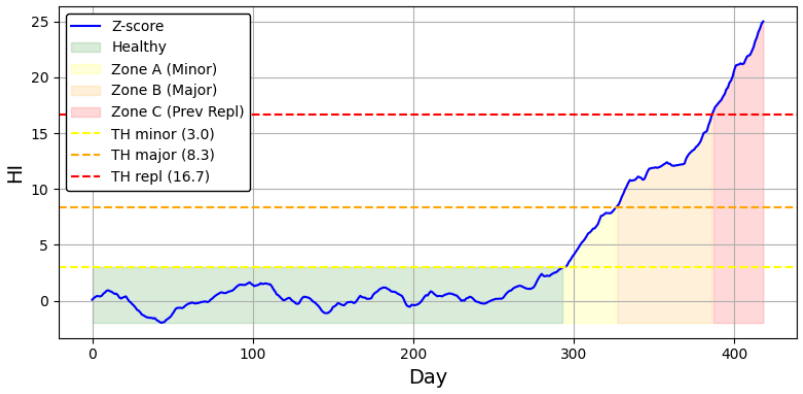


Figure 6.6: Thresholds based on the reference case 53

Wind farm parameters

Since the exact location of the wind farm is unknown, the values are based on the values of Borsotti et al. [16]. They took a generic offshore Wind Farm in the North Sea, 25km from the onshore base, with 3MW wind turbines. From the SCADA data, the capacity factor is taken to calculate the average power of the data. This capacity factor is calculated by the avarage power divided by the rated power, which is 0.346 for the CARE B wind farm. The wind farm parameters are shown in Table 6.4.

Table 6.4: Wind Farm Parameters [16]

Description [Unit]	Value
Rated power per turbine [MW]	3.00
Average capacity factor [-]:	0.346
Average power output per turbine [MW]	1.04
Hub height [m]	80
Distance from shore [km]	25
Cost of electricity [€/MWh]	150

lifetime assumptions

In the CARE case study, the exact lifetime of the main bearing is unknown, as the data only captures the final degradation phase. To estimate the full lifetime required for annualised cost evaluation, a typical bearing lifetime from literature is used as a baseline. According to Tian et al. [18], bearing failures can be modelled using a Weibull distribution with a shape parameter of 2 and a scale parameter of 3750 days. This corresponds to an average lifetime of approximately 3323 days.

To align this with the observed data, the average length of the HIs in the case study (386 days) is subtracted from the assumed full lifetime. This results in an initial degraded lifetime of roughly 2937 days before the data begins. This adjustment ensures that, on average, the bearings in the case study are assumed to fail at the same point as bearings in the reference distribution, allowing for consistent and realistic annualised cost comparisons across maintenance strategies.

Maintenance task parameters

The costs per maintenance activity, along with the corresponding repair times and technician requirements, are summarised in Table 6.5. These values are specific to the maintenance of the rotor bearing and are based on data provided by Borsotti et al. [16].

Table 6.5: Cost components and repair requirements for different rotor-bearing maintenance actions [16]

Maintenance type	C_{mat} [€]	C_{fix} [€]	T_{repair} [h]	N_{tech}
Minor repair	1 350	3 982.5	6	2
Major repair	3 750	10 455	12	4
Preventive replacement	15 000	43 755	36	8
Corrective replacement	60 000	43 755	36	8

Vessel-specific parameters: All vessel–related limits and costs used in the planning model are taken directly from the literature and are listed in Table 6.6.

For SOVs and CTVs, the wind speed limits are taken from Hu et al. [92], with thresholds of 17m/s for CTVs and 12m/s for SOVs at 10 metres height. In addition, no lead time is needed for CTV and SOV vessels [67].

For Jack-up vessels the operational wind speed limits for jack-up vessels vary depending on the vessel type and the specific activity, ranging from 10 to 16 m/s [98]. For this study, a conservative wind speed threshold of 10 m/s was adopted and applied uniformly throughout all phases of the operation: transfer, jacking-up, repair and jacking-down. Each jacking-up or jacking-down operation is assumed to require 3 hours [93]. As in Tian et al. [18], a lead time of 30 days is assumed for both the vessel and bearing replacement components.

Table 6.6: Operational parameters for maintenance vessels [16], [18], [67], [92], [93], [98]

Vessel Type	Day Rate [€]	Mobilization [€]	Fuel [L/h]	Speed [km/h]	Max Wind [m/s]	Jack-up Time [h]	Lead time [d]
CTV	8,000	0	5	25	12	–	0
SOV	18,000	0	20	17.3	17	–	0
Jack-up Vessel	50,000	57,000	100	11.9	10	3	30

6.2.2. Condition-based maintenance planning

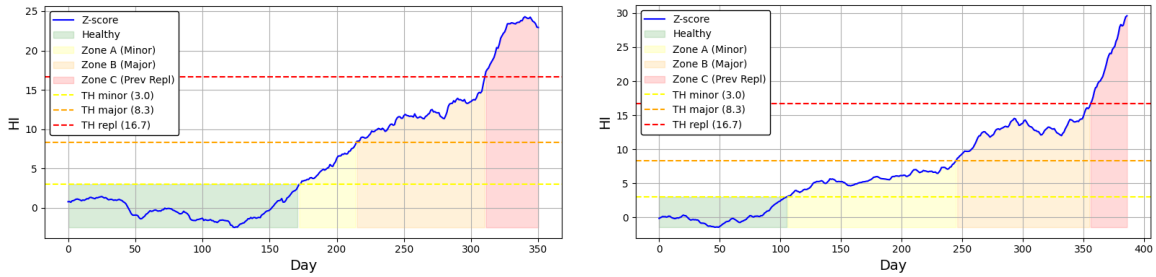
Cases 27 and 77 are used for scheduling and Case 53 as a reference. Figure 6.7 illustrates how the defined thresholds correspond to each of the scheduling HIs and shows the progression of the maintenance zones. Notably, once any threshold is exceeded, the HI does not revert below it. This characteristic is important for planning maintenance without introducing doubt ad a later stage about whether an intervention is still necessary. Table 6.7 summarises the precise calendar days on which each threshold is first crossed, and the failure days.

Table 6.7 also displays important information for the CBM strategy introduced in Subsection 4.2.3. Recall that CBM schedules maintenance after the HI crosses TH_{minor}. Since there is no lead time for minor maintenance, a weather window is searched immediately. The search for a weather window is shown in Figure 6.8, where favourable wind speeds are indicated in light green and unfavourable ones in light red (according to the wind speeds thresholds documented in Table 6.6). This image starts on the TH_{minor} day and ends on the failure day day. Upon TH_{major} and TH_{repl} the zones change as indicated by orange and red. The access day is defined as the first day of the maintenance activity. For Case 77 the acces day is directly after crossing the threshold, were for Case 27, the first days the wind speed is above the threshold wind speed, as marked by the red. Consequently, the accessing day is delayed by 8 days. This results in acess days for Case 27 and 77 occur on day 172 and 106, respectively.

Both are still well below TH_{major}, as becomes clear in Figure 6.8 where TH_{major} starts the orange major repair zone. For this reason, the maintenace activy is minor maintenance. Once the minor intervention is complete, the remaining life is extended by the life extension factor. This is repeated until the life is extended to a maximum of 200 %, after which preventive replacement is performed (as discussed in Subsection 4.2.3).

6.2.3. Strategy comparison

The CBM strategy is compared to corrective replacements, preventive replacements, and perfectly timed CBM maintenance, as described in Section 4.2. The resulting performance metrics are shown in Figure 6.9a, which includes three subplots. Subsection 4.2.3 shows the average lifespan of the two turbines in days, where corrective replacement and preventive replacement are coloured orange and blue, and the CBM and the perfect CBM are shown in green and purple, respectively. Figure 6.9b



(a) Case 27

(b) Case 77

Figure 6.7: Maintenance thresholds and zones indicated in the HIs

Table 6.7: Overview of important days per case

Case	TH_{minor}	TH_{major}	TH_{repl}	Failure day	Access day	Maintenance type
27	172	215	311	351	180	minor
77	106	246	356	387	106	minor

presents the total downtime per turbine in hours, using the same colour scheme. Figure 6.9c displays the number and types of interventions per turbine, with colours indicating the type of intervention: yellow for minor repairs, orange for major repairs, red for preventive replacements and dark red for corrective replacements. Figure 6.10 shows the annual maintenance cost per turbine for each strategy. Each bar is divided into cost components: material (red), transfer (purple), technician (blue), and downtime (yellow). The total cost is indicated at the top of each bar.

All strategies, except corrective replacement, successfully schedule maintenance before failure, as can be seen in Figure 6.9c since no corrective replacements are counted. In corrective and preventive replacement, the rotor bearing is replaced directly, so only one maintenance intervention occurs. In contrast, both CBM stratergies implements multiple minor interventions. Specifically, three minor maintenance activities occur before the component reaches its maximum lifetime, at which point a preventive replacement takes place. None of the four strategies includes major maintenance activities. These proactive interventions extend component life substantially (Figure 6.9a): average turbine lifetimes increase from 3338 days under corrective replacement or 3270 days under preventive

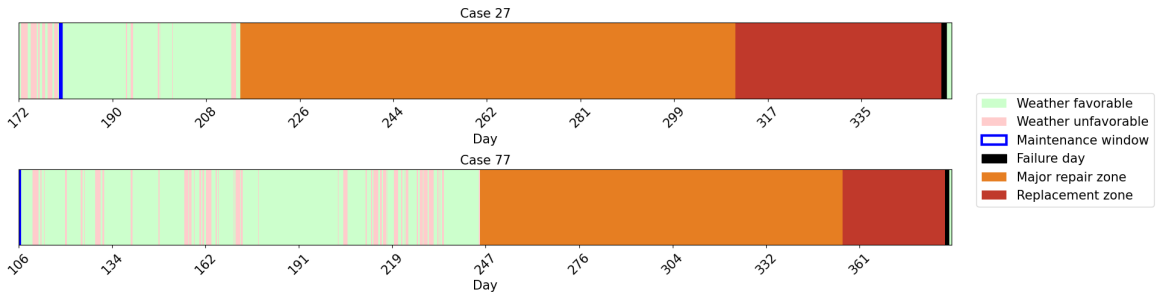
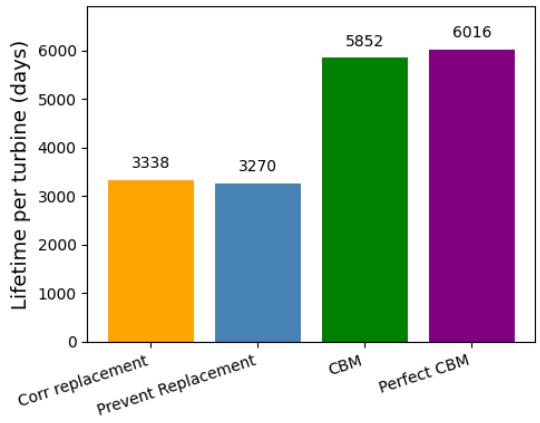


Figure 6.8: CBM planning for minor maintenance for Case 27 and Case 77

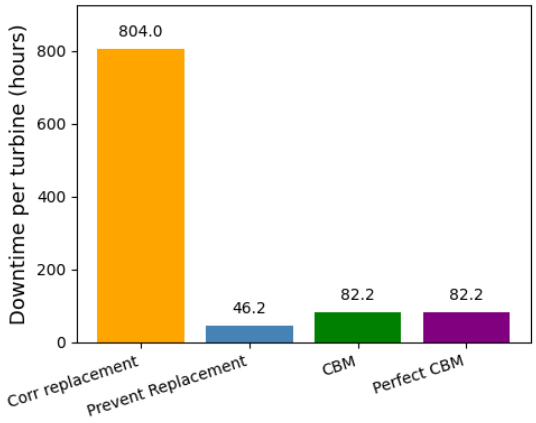
replacement to 5852 days under standard CBM (+ 75%), and 6016 days under perfect CBM (+80%). Optimally timing maintenance yields an additional 2.8% improvement compared to standard CBM.

In corrective replacement, the turbine remains offline until it fails, resulting in 804 hours (33.5 days) of unplanned downtime over its useful life (Figure 6.9b). In contrast, preventive replacement limits downtime to just 46.2, and both CBM strategies incur 82.2 hours of total downtime (Figure 6.9b).

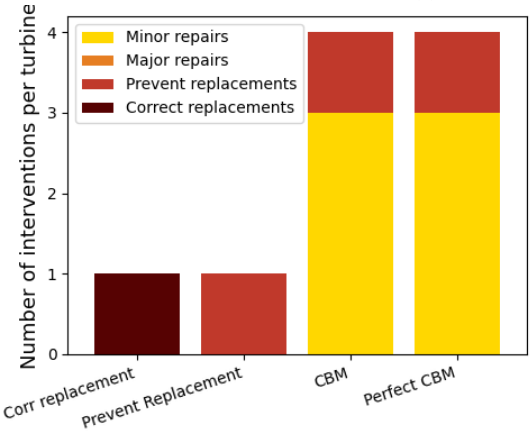
Quantitatively, the annual maintenance cost per turbine (Figure 6.10) falls from €40 073 under corrective replacement to €22 710 with preventive replacement (−43.3%), highlighting the benefit of scheduling replacements to avoid additional material and downtime costs. Extending the turbine life by small increments reduces the cost further to €15 017 with CBM (−62.5% versus corrective; −33.8% versus preventive). Perfectly timed CBM yields a modest additional savings, reducing the cost to €14 607 (−2.7% versus CBM), consistent with the extended lifetime discussed earlier.



(a) Average lifetime per turbine (days)



(b) Total downtime per turbine (hours)



(c) Number and types of intervention per turbine

Figure 6.9: Average turbine comparison for maintenance strategies regarding: (a) wasted lifetime, (b) downtime, and (c) number of and types interventions

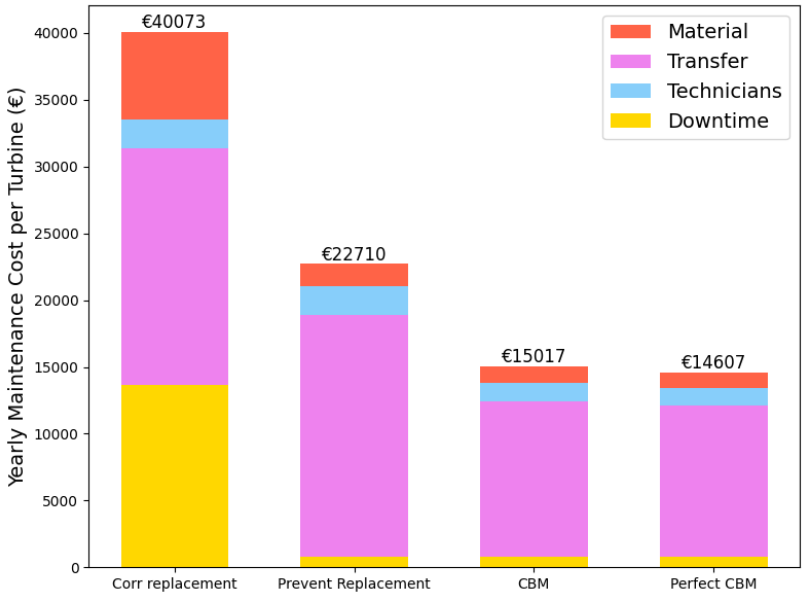


Figure 6.10: Cost comparison between corrective replacements, preventive maintenance, CBM and perfectly timed CBM

6.3. Discussion

This thesis aims to provide an end-to-end framework for CBM by using an unsupervised LSTM AE trained on SCADA data from the new CARE to compare dataset [44] to develop HIs. HIs are developed and evaluated for their performance, and these HIs are integrated into a multi-threshold maintenance model to demonstrate the potential of such a HI in extending component life and reducing downtime and costs.

Health Indicator generation and performance

This study develops an unsupervised HI using an LSTM AE trained exclusively on healthy SCADA data. For the three documented rotor bearing failures, the model raised an anomaly approximately 195 days in advance on average, while no false alarms were triggered in six healthy turbines. This level of performance is at the upper end of what has been reported in the literature using temperature and vibration data, where main bearing faults are typically detected 5–6 months prior to failure [49]. Based on only healthy SCADA data, as in this study, Encalada-Dávila et al. [41] were able to detect a single main bearing fault 106 days before failure. Moreover, CARE's event window, defined as the most likely early detection region, was on average 54 days, although temperature increases were observed earlier. These comparisons highlight the effectiveness of the unsupervised AE model presented in this study in delivering early fault warnings using only SCADA data. In addition, after crossing the early detection thresholds, the HIs have shown a high average trendability of 0.99 and monoticity of 0.49. This result indicates a clear upward trend after the anomaly is detected, with limited variation from the trend, and therefore supports the reliability of the HI.

A key strength of the unsupervised approach proposed in this work is its generalisability across different failure modes. Different failure modes are expected to show a number of possible degradation patterns [41]. The three failure cases in this study each exhibited distinct temporal profiles, which shows that the degradation pattern does not have to follow a uniform trend across the turbines. Supervised HI models often rely on a predefined degradation shape, such as a linear decline [51] or a quadratic curve [52], which would not have captured the progression observed here. In contrast, the HI based on the reconstruction error can adapt to the underlying failure trajectory without requiring any prior assumptions about its form. This allows for a more realistic and generalisable representation of varying degradation behaviours across assets.

This study used the newly released CARE to Compare dataset [44], which consists of real-world SCADA data. The data set includes labelled cases that distinguish between healthy and faulty cases, allowing the model to be trained exclusively on data assumed to reflect healthy operation. However, the absence of information on operational lifetimes makes it difficult to verify the true condition of the training data. Temperature levels varied considerably between turbines, with some showing high values even during presumed healthy periods. This variability is reflected in the loss, which differed between turbines. Notably, test loss was similar to training loss, despite the expectation that the model would perform better on familiar data. These inconsistencies are likely due to environmental differences, particularly location-specific wake effects caused by prevailing wind directions, which are common in this wind farm and can have a significant influence on average power output [99] and therefore the bearing temperature. If degradation had been present in the training data, a clear, monotonic increase in HI is expected, which is not observed in the healthy cases (Figure 6.5).

Due to varying baseline temperature levels across turbines, reconstruction errors also differed because higher temperatures tended to produce higher residuals. This makes the trend of reconstruction errors more important. To ensure that the thresholds had a consistent interpretation, HI was z score normalised based on the first 100 healthy days of each turbine. This approach enabled the setting of thresholds with the same statistical meaning to Chebyshev's inequality (described in Subsection 4.1.4), providing a statistical meaning to the thresholds in maintenance planning. Although HI values bounded between 0 and 1 are more common in the literature for advantages in interpretability and RUL prediction, such normalisation was unsuitable here due to the small number of failures and inconsistent end-of-life values.

Maintenance implications

The long P-F interval, combined with high trendability and monotonicity, allowed the use of a multithreshold maintenance strategy. This strategy categorises the severity of the degradation into zones for minor repair, major repair, and complete replacement. In all tested cases, maintenance was successfully scheduled within the zone corresponding to the set thresholds, demonstrating that early warnings provide sufficient time. Additionally, once any HI crosses a threshold, it does not fall below it again, due to the high trendability and monotonicity.

This end-to-end framework demonstrates the practical development and deployment of an unsupervised HI, showing that its integration can reduce downtime, reduce costs, and trigger earlier interventions than conventional approaches. Those earlier interventions allow an extension of the lifetime of the components by relatively inexpensive actions such as maintaining high lubrication cleanliness, which can extend the life of a roller bearing by 200% [17]. Such an extension of life contributes to more responsible use of material and directly reduces the LCOE.

From an operator's perspective, the approach requires only low-frequency SCADA temperature data and can be integrated into existing maintenance workflows as a supplementary indicator. No additional cost for CMS is needed as most large-scale turbines already provide SCADA data [40], making the approach attractive to any operator. Although not all types of failure types will manifest as clearly in the temperature signals (for example plastic deformation or corrosion) and thus may not be detected by the model, the model can effectively identify those that do, offering substantial potential for reducing unplanned downtime and associated costs through early detection.

Limitations

Despite promising results, this thesis is subject to several limitations that should be considered when interpreting the findings. These are discussed in two parts: the first focuses on limitations related to the data and the construction of the HI, and the second addresses assumptions made in the maintenance planning framework.

Health Indicator Generation

Z-score normalisation proved useful in this thesis, but it is sensitive to noise, model parameters, and the choice of the reference window. Changes in training configuration can lead to more fluctuations in the normalised HI compared to the reconstruction error. In principle, Min-Max Normalisation would offer a more robust in this sense.

Furthermore, Z-score normalisation is unable to pick up trends in the first part of the data, as the first 100 days are used for the normalisation. In healthy Case 52, Case 82, Case 83 (Figures 6.5c, 6.5e, 6.5f), the HI showed a rising trend during the initial reference period. If these same patterns had occurred later in the time series, they may have triggered false alarms.

The selection of both the reconstruction error of rotor bearings 1 and 2 combined is not ideal. In de Pater and Mitici [28], sensors were selected based on their Spearman correlation during healthy operation. However, in the year of available data, the sensors did not show a clear increase over time. Consequently, the selection of the rotor bearing temperatures in this study is based on domain knowledge. Both sensors were combined to avoid bias towards impending failure, with the underlying idea that operators can identify the failing bearing based on temperature values.

However, this choice inflated the reconstruction-error HI for Cases 53 and 77. The predicted values of temperature 1 increased even though the fault occurred in bearing 2, which raised the reconstruction-error HI (see Subsection 6.1.1). By contrast, this effect did not arise in Case 27, where the failure was in bearing 1. This behaviour is difficult to explain; one plausible reason is that temperature 2 influences the prediction of temperature 1 more strongly than the reverse. A more robust approach would be to first classify the faulty bearing and then base the HI solely on the reconstruction error of the corresponding sensor.

Finally, the model used a relatively long LSTM sequence length of approximately two months. Although this helps filter out short-term noise and improves trendability, it also reduces sensitivity to sudden changes. Therefore, the current approach is best suited for slowly developing failure modes and will be less suitable for rapidly developing fault modes.

Maintenance

The maintenance model was designed primarily to explore the potential benefits of integrating a HI. As such, it remained relatively simple and included several simplifications, such as regarding assumptions about constant lead times and lifetime extensions and maximum lifetimes. Static thresholds are used instead of optimised ones. Furthermore, the weather forecast is assumed to be perfectly known and is based on the SCADA data from the turbine. However, no wave information was included. This can overestimate weather windows and needs to be included for further research.

The maintenance thresholds are inherently arbitrary, as their values are based on equal divisions of the HI from a reference turbine without confirmation of the underlying physical degradation state. In practice, wind turbine operators would need to perform inspections to calibrate these thresholds with more accuracy.

The CBM model exhibits unrealistic behaviour by prescribing minor repairs in sequence until the system reaches its maximum life extension. This stems from the model's assumption that each intervention resets the HI to its previous trajectory. In addition, the model assumes that a preventive replacement can always be scheduled once every life extension is complete, before any renewed risk of failure. This assumption relies on the ability to perform preventive maintenance in time based on HI, as became apparent from the preventive replacement strategy. However, there is no guarantee that the HIs will perform similarly after multiple maintenance activities have already been carried out.

Additionally, the model assumes that performing minor and major maintenance immediately before failure can extend the useful life of the components to 200 %. This assumption relies on maintaining the cleanliness of the lubricant at 200 % of the normal level [17]. Some failure modes are directly related to the lubricant, such as surface-initiated fatigue or wear. However, with failure modes such as plastic deformation, electrical erosion, fractures, or subsurface fatigue, the effect of lubricant will not be so significant. In addition, when degradation becomes detectable, any wear or damage may already be irreversible. Consequently, even for lubricant-related failure modes, a 200 % life extension may be overly optimistic. Upon inspection, it is important for the techniques to determine the failure mode and the suited repair.

Finally, the current model does not include time-based inspections. In practice, operators inspect offshore wind turbines about every six months [100]. Such scheduled visits enable maintenance actions similar to those in the CBM strategy. Inspections can also detect failures at an early stage, although their accuracy is imperfect and their timing must be appropriate [100]. The three critical failures observed in the dataset show that inspections do not always achieve this goal. Nevertheless, adding time-based inspections would provide a more realistic baseline than a purely corrective strategy, so future work should include them to allow a fair comparison. In that comparison, CBM could reduce the number of routine visits while still lowering the failure rate [101].

7

Conclusion and Recommendations

Smart, sensor-based maintenance solutions are essential to reduce the LCOE of offshore wind and ensure clean, reliable energy. This thesis presents an end-to-end CBM framework to model rotor bearing degradation via an SCADA-based HI that is integrated into a multi-threshold CBM framework. HIs are developed using an operating condition-informed LSTM AE that learns normal behaviour by training on healthy data. The results show that even with limited run-to-failure data, it is feasible to develop a high-quality HI from SCADA data that detects failures months in advance and use this information in a practical maintenance strategy. As the method relies solely on SCADA data, it offers a non-intrusive and cost-efficient alternative to CMS.

This chapter provides an answer to the main research question:

How can rotor bearing health indicators for wind turbines be developed from SCADA data and integrated into a condition-based maintenance strategy?

This question is answered by answering the following four sub-questions:

1. *Which method can effectively derive rotor bearing health indicators and anomalies from SCADA data when run-to-failure cases are scarce?*

   SCADA systems provide abundant operational data but typically contain few labelled run-to-failure records. Data-driven approaches, particularly deep learning methods, excel at extracting patterns from such large datasets with complex non-linear relationships. Unsupervised deep learning techniques, such as AEs, suit situations with limited run-to-failure data because they can train solely on healthy data to model normal behaviour. An AE encodes input signals into a compact latent space that preserves essential features and then reconstructs the original signal. As degradation sets in, the model's predictions for rotor-bearing temperature diverge from the actual values, and the reconstruction error increases. This error serves as a proxy for component health. However temperature signals, and thus the reconstruction errors, are noisy. Employing a LSTM architecture that embeds operating conditions within its cell state captures long-term trends while filtering short-term variability. Finally, z-score normalisation, based on the mean and standard deviation from the first 100 healthy days of each turbine. This yields a standardised HI that supports threshold setting for CBM, as z-score values align with probability bounds derived from Chebyshev's inequality.

2. *What evaluation metrics are appropriate for the health indicators, and how well do the derived health indicators perform on the CARE dataset?*

   To assess the preformance of the derived HIs, three metrics are used: trendability, monotonicity, and the duration of the P-F interval. Trendability measures how consistently the HI increases over time and is quantified using the Spearman rank correlation. Monotonicity reflects the proportion of strictly increasing steps, indicating the smoothness of the signal. The P-F interval captures how early the HI raises an alarm before failure, determining the available reaction time for maintenance planning. All three criteria are commonly used in the literature. Applied to three bearing failure cases, the HI preformed well with an average trendability of 0.99 and a monotonicity of 0.49 after the anomaly threshold was crossed (on a scale from 0 to 1). High trendability and monotonicity are essential to ensure that the HI remains above the alarm threshold once degradation is detected, which is the case for all three HIs. Alarms were raised 179, 125, and 281 days before failure, giving a mean P-F interval of 195 days. Importantly, no false alarms occurred across six healthy turbines. These results lie at the upper end of values reported in the literature and demonstrate that the temperature-based HI performs well on the failures from CARE and provide actionable early warnings.

3. *How can a practical condition-based maintenance framework for offshore wind turbines be developed based on the health indicators?*

   Because the HI offers a long P-F interval, the CBM strategy can distinguish multiple degradation zones within it. The proposed framework defines three degradation zones: minor repair, major repair, and full replacement. Each zone is bounded by a static HI threshold estimated from one turbine taken as reference turbine. Once the first threshold (the anomaly threshold) is exceeded, a maintenance task is scheduled in the next suitable weather window, based on the recorded wind data from the SCADA system. For replacements an extra lead time of 30 day is assumed before planning the intervention. The zone in which the task occurs determines the type of action together with the associated cost and life extension. Minor and major zones allow multiple consecutive repairs. After each repair the component is assumed to return to its original HI trajectory, whereby it follow the same trajectory. The component lifetime can be extended by a maximum of 200 %, after which preventive replacement is required. This structure favours early low-cost interventions over reactive replacements while considering offshore logistical constraints such as vessel availability and weather downtime.

4. *How does the proposed framework compare to corrective replacement, preventive replacement and perfectly timed maintenance strategies?*

   Three benchmark strategies are considered. In *corrective replacement*, maintenance is only performed after bearing failure. In *preventive replacement*, the bearing is planned as the replacement threshold is met. The *perfectly timed CBM* strategy mirrors CBM but schedules the intervention at the latest suitable weather window while the system is still in the same degradation zone. In all cases (except corrective), the long P-F interval allows timely maintenance planning in the same maintenance zone, without the HI dropping below or rising above the associated thresholds. Embedding the HI in a multi-threshold CBM stratergy the annual maintenance cost per turbine to €25.1 thousand, which is 62.5% lower than corrective replacement and 33.9% below preventive replacement. Downtime over the turbine lifetime decreased by 722 hours compared to corrective replacement. The bearing's operational life is extended by 75%

compared to corrective replacement. The hypothetical perfectly timed CBM scenario yields a further 2.2% yearly cost saving due the 2.2% life extension. This demonstrates the theoretical benefit of implementing HIs for rotor bearings as relatively inexpensive interventions can be used to significantly extend component lifetime.

Recommendations for Further Research

To further realise the potential of the proposed framework and address the limitations of this study, future work is recommended to pursue the following directions:

1. **Expand the AE framework to include other failure data within the CARE dataset.** Wind farms A and C in the CARE dataset include additional drivetrain faults. Incorporating these fault types into the AE framework could allow a broader assessment of its applicability and facilitate evaluation of its transferability across different components.
2. **Develop RUL estimators.** The HI now lacks information on a clear failure point, a robust RUL estimation could provide the wind turbine operator with additional information. In this study, robust RUL estimation was not feasible due to the limited number and inconsistent progression of degradation of failure cases, as well as the lack of clearly defined failure points. These are challenges that limit the applicability of regression-based techniques. With access to larger datasets, NN or similarity-based methods could work to predict time-to-failure.
3. **Incorporate time-based inspections.** Time-based inspections are standard practice in the offshore wind industry but are not yet included in the CBM comparison. Adding them would offer a more realistic baseline than a purely corrective approach. The resulting comparison could show how CBM reduces routine visits and potentially lowers failure rates.
4. **Enhance the maintenance framework.** The model could be improved by implementing dynamic, rather than static, thresholds. In addition, incorporating wave height forecasts alongside wind forecasts could improve the accuracy of access modelling. The inclusion of historical power production from SCADA data and dynamic electricity prices could allow for more precise estimation of downtime costs and interesting maintenance opportunities.

A
Appendix

A.1. log Law

The mean wind speed at a given height z can be estimated nu using the logarithmic wind profile with assumed neuteral atmospheric boundary layer stability correction, given by [102]:

$$U(z) = \frac{u_*}{k}\left[\ln\left(\frac{z}{z_0}\right)\right]$$

where u_* is the friction velocity, $k \approx 0.4$ is the von Kármán constant, z_0 is the surface roughness length, and c is a stability-dependent correction term (positive for unstable and negative for stable atmospheric conditions). This formulation accounts for vertical mixing effects and is more physically grounded than the simpler power law approach, which assumes a fixed exponent and is less accurate under varying stability conditions.

Zo for open sea is a typlical value 0.0002m

To estimate wind speed at 10 meters from a known value at 80 meters under neutral conditions, we compute:

$$\frac{U(10)}{U(80)} = \frac{\ln(10/0.0002)}{\ln(80/0.0002)} = \frac{\ln(50000)}{\ln(400000)} \approx \frac{10.82}{12.90} \approx 0.839.$$

Therefore, the wind speed at 10 meters can be approximated by multiplying the 80-meter wind speed by a factor of 0.839.

The mean wind speed at a given height z can be estimated nu using the logarithmic wind profile. This logatritmic wind profile with assumed neuteral atmospheric boundary layer stability correction, given by [102]:

$$U(z) = \frac{u_*}{k}\ln\left(\frac{z}{z_0}\right),$$

where $U(z)$ is the mean wind speed at height z, u_* is the friction velocity, $k \approx 0.4$ is the von Kármán constant, and z_0 is the surface roughness length. For offshore environments such as the North Sea, a typical value of z_0 is 0.0002 m [103].

To convert wind speeds from 80 meters (assumed hub height) to 10 meters under neutral conditions, the following ratio is used:

$$\frac{U(10)}{U(80)} = \frac{\ln(10/0.0002)}{\ln(80/0.0002)} \approx \frac{10.82}{12.90} \approx 0.839.$$

This implies that the wind speed at 10 meters can be estimated by multiplying the 80-meter wind speed by approximately 0.839.

Bibliography

[1] United Nations Framework Convention on Climate Change, *Paris agreement*, Report of the Conference of the Parties, 21st Session, Paris, 2015. [Online]. Available: https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement.

[2] European Commission, "Repowereu plan," European Commission, Tech. Rep., 2018. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52022DC0230.

[3] HM Government, "Net zero strategy: Build back greener," UK Government, Tech. Rep., 2021. [Online]. Available: https://www.gov.uk/government/publications/net-zero-strategy.

[4] A. V. Litră, E. Nichifor, I. B. Chiţu, A. Zamfirache, and G. Brătucu, "The Dilemma of the European Integration Principle—Ensuring Energy Independence of the European Union," *Sustainability*, vol. 15, no. 21, p. 15 560, Jan. 2023, ISSN: 2071-1050. DOI: 10.3390/su152115560.

[5] Z. Ren, A. S. Verma, Y. Li, J. J. E. Teuwen, and Z. Jiang, "Offshore wind turbine operations and maintenance: A state-of-the-art review," *Renewable and Sustainable Energy Reviews*, vol. 144, p. 110 886, Jul. 2021. DOI: 10.1016/j.rser.2021.110886.

[6] Global Wind Energy Council, "Global wind report 2023," GWEC, Tech. Rep., 2023. [Online]. Available: https://gwec.net/global-wind-report-2023.

[7] K. Tartt, A. M. Kazemi-Amiri, A. R. Nejad, J. Carroll, and A. McDonald, "Life extension of wind turbine drivetrains by means of SCADA data: Case study of generator bearings in an onshore wind farm," *Results in Engineering*, vol. 24, p. 102 921, Dec. 2024, ISSN: 2590-1230. DOI: 10.1016/j.rineng.2024.102921.

[8] A. Memija. "Dutch gov't shelves two offshore wind tenders, plans single site auction." Accessed: 2025-07-30, Offshore Wind. (May 19, 2025), [Online]. Available: https://www.offshorewind.biz/2025/05/19/dutch-govt-shelves-two-offshore-wind-tenders-plans-single-site-auction/.

[9] A. Memija. "Denmark cancels 3 gw offshore wind tender, gov't eyes auction with state subsidies." Accessed: 2025-07-30, Offshore Wind. (Jan. 31, 2025), [Online]. Available: https://www.offshorewind.biz/2025/01/31/denmark-cancels-3-gw-offshore-wind-tender-govt-eyes-auction-with-state-subsidies/.

[10] T. Stehly, P. Beiter, and P. Duffy, "2019 Cost of Wind Energy Review," *Renewable Energy*, 2020.

[11] H. Fox, A. C. Pillai, D. Friedrich, M. Collu, T. Dawood, and L. Johanning, "A Review of Predictive and Prescriptive Offshore Wind Farm Operation and Maintenance," *Energies*, vol. 15, no. 2, p. 504, Jan. 2022. DOI: 10.3390/en15020504.

[12] M. Shafiee and J. D. Sørensen, "Maintenance optimization and inspection planning of wind energy assets: Models, methods and strategies," *Reliability Engineering & System Safety*, vol. 192, p. 105 993, Dec. 2019. DOI: 10.1016/j.ress.2017.10.025.

[13] C. A. Irawan, D. Ouelhadj, D. Jones, M. Stålhane, and I. B. Sperstad, "Optimisation of maintenance routing and scheduling for offshore wind farms," *European Journal of Operational Research*, vol. 256, no. 1, pp. 76–89, Jan. 2017. DOI: 10.1016/j.ejor.2016.05.059.

[14] C. J. Crabtree, D. Zappalá, and S. I. Hogg, "Wind energy: UK experiences and offshore operational challenges," *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, vol. 229, no. 7, pp. 727–746, Nov. 1, 2015, ISSN: 0957-6509. DOI: 10.1177/0957650915597560.

[15] M. Borsotti, R. Negenborn, and X. Jiang, "Model predictive control framework for optimizing offshore wind O&M," in *Advances in Maritime Technology and Engineering*, 1st ed. London: CRC Press, May 8, 2024, pp. 533–546, ISBN: 978-1-003-50876-2. DOI: 10.1201/9781003508762-65.

[16] M. Borsotti, R. R. Negenborn, and X. Jiang, "Stochastic optimization of predictive maintenance scheduling for offshore wind farms," SSRN Electronic Journal, Tech. Rep., 2024, Preprint, available at https://ssrn.com/abstract=5189581. DOI: 10.2139/ssrn.5189581.

[17] C. D. Dao, B. Kazemtabrizi, C. J. Crabtree, and P. J. Tavner, "Integrated condition-based maintenance modelling and optimisation for offshore wind turbines," *Wind Energy*, vol. 24, no. 11, pp. 1180–1198, Nov. 2021. DOI: 10.1002/we.2625.

[18] Z. Tian, T. Jin, B. Wu, and F. Ding, "Condition based maintenance optimization for wind power generation systems under continuous monitoring," *Renewable Energy*, vol. 36, no. 5, pp. 1502–1509, May 2011. DOI: 10.1016/j.renene.2010.10.028.

[19] A. Lee. "Offshore wind deaths and injuries grow amid 'new and evolving risks'," Recharge. (Jun. 12, 2025), [Online]. Available: https://www.rechargenews.com/wind/offshore-wind-deaths-and-injuries-grow-amid-new-and-evolving-risks/2-1-1831894.

[20] M. J. Kabir, A. M. T. Oo, and M. Rabbani, "A brief review on offshore wind turbine fault detection and recent development in condition monitoring based maintenance system," in *2015 Australasian Universities Power Engineering Conference (AUPEC)*, Sep. 2015, pp. 1–7. DOI: 10.1109/AUPEC.2015.7324871. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7324871 (visited on 07/29/2025).

[21] L. Yang, Y. Chen, and X. Ma, "A state-age-dependent opportunistic intelligent maintenance framework for wind turbines under dynamic wind conditions," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 10, pp. 10 434–10 443, Oct. 2023. DOI: 10.1109/TII.2023.3240727.

[22] M. Florian and J. D. Sørensen, "Wind Turbine Blade Life-Time Assessment Model for Preventive Planning of Operation and Maintenance," *Journal of Marine Science and Engineering*, vol. 3, no. 3, pp. 1027–1040, 3 Sep. 2015. DOI: 10.3390/jmse3031027.

[23] M. Arias Chao, C. Kulkarni, K. Goebel, and O. Fink, "Fusing physics-based and deep learning models for prognostics," *Reliability Engineering & System Safety*, vol. 217, p. 107 961, Jan. 2022. DOI: 10.1016/j.ress.2021.107961.

[24] M. Jankauskas, A. Serackis, M. Šapurov, *et al.*, "Exploring the Limits of Early Predictive Maintenance in Wind Turbines Applying an Anomaly Detection Technique," *Sensors*, vol. 23, no. 12, p. 5695, 12 Jan. 2023. DOI: 10.3390/s23125695.

[25] Q. Z. Jianing Man Zijun Zhang, "Data-driven predictive analytics of unexpected wind turbine shut-downs," *IET Renewable Power Generation*, vol. 15, pp. 1833–1842, Apr. 2019. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-rpg.2018.5520.

[26] W. Udo and Y. Muhammad, "Data-Driven Predictive Maintenance of Wind Turbine Based on SCADA Data," *IEEE Access*, vol. 9, pp. 162 370–162 388, 2021. DOI: 10.1109/ACCESS.2021.3132684.

[27] A. E. Milani, D. Zappalá, and S. J. Watson, "A hybrid Convolutional Autoencoder training algorithm for unsupervised bearing health indicator construction," *Engineering Applications of Artificial Intelligence*, vol. 139, p. 109 477, Jan. 2025, ISSN: 0952-1976. DOI: 10.1016/j.engappai.2024.109477.

[28] I. de Pater and M. Mitici, "Developing health indicators and RUL prognostics for systems with few failure instances and varying operating conditions using a LSTM autoencoder," *Engineering Applications of Artificial Intelligence*, vol. 117, p. 105 582, Jan. 2023, ISSN: 0952-1976. DOI: 10.1016/j.engappai.2022.105582.

[29] W. Yu, I. Y. Kim, and C. Mechefske, "An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme," *Reliability Engineering & System Safety*, vol. 199, p. 106 926, Jul. 2020, ISSN: 0951-8320. DOI: 10.1016/j.ress.2020.106926.

[30] T. Han, J. Pang, and A. C. C. Tan, "Remaining useful life prediction of bearing based on stacked autoencoder and recurrent neural network," *Journal of Manufacturing Systems*, vol. 61, pp. 576–591, Oct. 2021, ISSN: 0278-6125. DOI: 10.1016/j.jmsy.2021.10.011.

[31] A. V. Horenbeek, J. V. Ostaeyen, J. R. Duflou, and L. Pintelon, "Quantifying the added value of an imperfectly performing condition monitoring system—Application to a wind turbine gearbox," vol. 111, ISSN: 0951-8320. DOI: 10.1016/j.ress.2012.10.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832012002104.

[32] T. Gintautas and J. D. Sørensen, "Improved Methodology of Weather Window Prediction for Offshore Operations Based on Probabilities of Operation Failure," *Journal of Marine Science and Engineering*, vol. 5, no. 2, p. 20, Jun. 2017, ISSN: 2077-1312. DOI: 10.3390/jmse5020020.

[33] M. Hofmann and I. B. Sperstad, "NOWIcob – A Tool for Reducing the Maintenance Costs of Offshore Wind Farms," *Energy Procedia*, vol. 35, pp. 177–186, 2013. DOI: 10.1016/j.egypro.2013.07.171.

[34] M. Mitici, I. De Pater, A. Barros, and Z. Zeng, "Dynamic predictive maintenance for multiple components using data-driven probabilistic RUL prognostics: The case of turbofan engines," *Reliability Engineering & System Safety*, vol. 234, p. 109 199, Jun. 2023. DOI: 10.1016/j.ress.2023.109199.

[35] J. Lee and M. Mitici, "Deep reinforcement learning for predictive aircraft maintenance using probabilistic Remaining-Useful-Life prognostics," *Reliability Engineering & System Safety*, vol. 230, p. 108 908, Feb. 2023. DOI: 10.1016/j.ress.2022.108908.

[36] I. De Pater, A. Reijns, and M. Mitici, "Alarm-based predictive maintenance scheduling for aircraft engines with imperfect Remaining Useful Life prognostics," *Reliability Engineering & System Safety*, vol. 221, p. 108 341, May 2022. DOI: 10.1016/j.ress.2022.108341.

[37] K. T. P. Nguyen and K. Medjaher, "A new dynamic predictive maintenance framework using deep learning for failure prognostics," *Reliability Engineering & System Safety*, vol. 188, pp. 251–262, Aug. 2019. DOI: 10.1016/j.ress.2019.03.018.

[38] J. Chatterjee and N. Dethlefs, "Scientometric review of artificial intelligence for operations & maintenance of wind turbines: The past, present and future," *Renewable and Sustainable Energy Reviews*, vol. 144, p. 111 051, Jul. 2021. DOI: 10.1016/j.rser.2021.111051.

[39] R. Pandit, D. Astolfi, J. Hong, D. Infield, and M. Santos, "Scada data for wind turbine data-driven condition/performance monitoring: A review on state-of-art, challenges and future trends," *Wind Engineering*, vol. 47, no. 2, pp. 422–441, Apr. 30, 2023. DOI: 10.1177/0309524X221124031. [Online]. Available: https://doi.org/10.1177/0309524X221124031.

[40] J. Tautz-Weinert and S. J. Watson, "Using SCADA data for wind turbine condition monitoring - A review," *IET Renewable Power Generation*, vol. 11, no. 4, pp. 382–394, Mar. 15, 2017. DOI: 10.1049/iet-rpg.2016.0248.

[41] Á. Encalada-Dávila, B. Puruncajas, C. Tutivén, and Y. Vidal, "Wind Turbine Main Bearing Fault Prognosis Based Solely on SCADA Data," *Sensors*, vol. 21, no. 6, p. 2228, Jan. 2021, ISSN: 1424-8220. DOI: 10.3390/s21062228.

[42] W. Zhang, J. Vatn, and A. Rasheed, "A review of failure prognostics for predictive maintenance of offshore wind turbines," in *Journal of Physics: Conference Series*, vol. 2362, Institute of Physics, 2022. DOI: 10.1088/1742-6596/2362/1/012043.

[43] I. Vasilev, *Advanced Deep Learning with Python: Design and implement advanced next-generation AI solutions using TensorFlow and PyTorch*. Birmingham, UK: Packt Publishing Ltd, 2019, p. 468, ISBN: 1789952719.

[44] C. Gück, C. M. A. Roelofs, and S. Faulstich, "CARE to Compare: A Real-World Benchmark Dataset for Early Fault Detection in Wind Turbine Data," *Data*, vol. 9, no. 12, p. 138, Nov. 23, 2024, ISSN: 2306-5729. DOI: 10.3390/data9120138.

[45] C. Dao, B. Kazemtabrizi, and C. Crabtree, "Wind turbine reliability data review and impacts on levelised cost of energy," *Wind Energy*, vol. 22, no. 12, pp. 1848–1871, 2019. DOI: 10.1002/we.2404.

[46] D. Zappalá and P. J. Tavner, "Wind turbine reliability - maintenance strategies," *Comprehensive Renewable Energy (Second Edition)*, pp. 353–370, 2021. DOI: 10.1016/B978-0-12-819727-1.00154-0.

[47] E. Artigao, S. Martín-Martínez, A. Honrubia-Escribano, and E. Gómez-Lázaro, "Wind turbine reliability: A comprehensive review. towards effective condition monitoring development," *Applied Energy*, vol. 228, pp. 1569–1583, Oct. 2018. DOI: 10.1016/j.apenergy.2018.07.037.

[48] M. Rezamand, M. Kordestani, R. Carriveau, D. S.-K. Ting, M. E. Orchard, and M. Saif, "Critical Wind Turbine Components Prognostics: A Comprehensive Review," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, pp. 9306–9328, Dec. 2020. DOI: 10.1109/TIM.2020.3030165.

[49] E. Hart, B. Clarke, G. Nicholas, *et al.*, "A review of wind turbine main bearings: Design, operation, modelling, damage mechanisms and fault detection," *Wind Energy Science*, vol. 5, no. 1, pp. 105–124, Jan. 15, 2020, ISSN: 2366-7443. DOI: 10.5194/wes-5-105-2020.

[50] A. E. Milani, D. Zappalá, F. Castellani, and S. Watson, "Boosting field data using synthetic SCADA datasets for wind turbine condition monitoring," *Journal of Physics: Conference Series*, vol. 2767, no. 3, p. 032 033, Jun. 2024, ISSN: 1742-6596. DOI: 10.1088/1742-6596/2767/3/032033.

[51] H. Wang, X. Zhang, X. Guo, T. Lin, and L. Song, "Remaining useful life prediction of bearings based on multiple-feature fusion health indicator and weighted temporal convolution network," *Measurement Science and Technology*, vol. 33, no. 10, p. 104 003, Jul. 2022, ISSN: 0957-0233. DOI: 10.1088/1361-6501/ac77d9.

[52] G. Jiang, W. Li, W. Fan, Q. He, and P. Xie, "TempGNN: A Temperature-Based Graph Neural Network Model for System-Level Monitoring of Wind Turbines With SCADA Data," *IEEE Sensors Journal*, vol. 22, no. 23, pp. 22 894–22 907, Dec. 2022, ISSN: 1558-1748. DOI: 10.1109/JSEN.2022.3213551.

[53] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li, "Predictive maintenance in the Industry 4.0: A systematic literature review," *Computers & Industrial Engineering*, vol. 150, p. 106 889, Dec. 2020. DOI: 10.1016/j.cie.2020.106889.

[54] M. H. Do and D. Söffker, "State-of-the-art in integrated prognostics and health management control for utility-scale wind turbines," *Renewable and Sustainable Energy Reviews*, vol. 145, p. 111 102, Jul. 2021. DOI: 10.1016/j.rser.2021.111102.

[55] A. Stetco, F. Dinmohammadi, X. Zhao, *et al.*, "Machine learning methods for wind turbine condition monitoring: A review," *Renewable Energy*, vol. 133, pp. 620–635, Apr. 2019. DOI: 10.1016/j.renene.2018.10.047.

[56] L. Zhuang, A. Xu, and X.-L. Wang, "A prognostic driven predictive maintenance framework based on Bayesian deep learning," *Reliability Engineering & System Safety*, vol. 234, p. 109 181, Jun. 2023. DOI: 10.1016/j.ress.2023.109181.

[57] G. d. N. P. Leite, A. M. Araújo, and P. A. C. Rosas, "Prognostic techniques applied to maintenance of wind turbines: A concise and specific review," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 1917–1925, Jan. 2018. DOI: 10.1016/j.rser.2017.06.002.

[58] K. Abid, M. S. Mouchaweh, and L. Cornez, "Fault prognostics for the predictive maintenance of wind turbines: State of the art," in *Communications in Computer and Information Science*, vol. 967, Springer Verlag, 2019, pp. 113–125, ISBN: 978-3-030-14879-9. DOI: 10.1007/978-3-030-14880-5_10.

[59] J. Kang, J. Sobral, and C. G. Soares, "Review of condition-based maintenance strategies for offshore wind energy," *Journal of Marine Science and Application*, vol. 18, no. 1, pp. 1–16, Mar. 2019. DOI: 10.1007/s11804-019-00080-y.

[60] J. A. Andrawus, J. Watson, and M. Kishk, "Wind Turbine Maintenance Optimisation: Principles of Quantitative Maintenance Optimisation," *Wind Engineering*, vol. 31, no. 2, pp. 101–110, Mar. 2007. DOI: 10.1260/030952407781494467.

[61] M. I. H. Tusar and B. R. Sarker, "Maintenance cost minimization models for offshore wind farms: A systematic and critical review," *International Journal of Energy Research*, vol. 46, no. 4, pp. 3739–3765, Jul. 2022. DOI: 10.1002/er.7425.

[62] W. Yang, R. Court, and J. Jiang, "Wind turbine condition monitoring by the approach of SCADA data analysis," *Renewable Energy*, vol. 53, pp. 365–376, May 2013. DOI: 10.1016/j.renene.2012.11.030.

[63] SKF Group, *Bearing damage and failure analysis*, https://www.skf.com/binaries/pub12/Images/0901d1968064c148-Bearing-failures—14219$_2$ − $EN_tcm_1$2 − 297619.pdf, SKF, Gothenburg, Sweden, 2017.

[64] *En 13306:2017 maintenance terminology*, https://www.ds.dk, 2017.

[65] M. Ben-Daya, U. Kumar, and D. Murthy, "A review of maintenance optimization models for multi-unit systems," *Computers & Industrial Engineering*, vol. 135, pp. 1125–1134, 2019. DOI: 10.1016/j.cie.2018.10.025.

[66] M. Shafiee, "Maintenance logistics organization for offshore wind energy: Current progress and future perspectives," *Renewable Energy*, vol. 77, pp. 182–193, May 2015. DOI: 10.1016/j.renene.2014.11.045.

[67] "The silver lining: Utilising curtailment in offshore wind as an opportunity for operation and maintenance," vol. 330, ISSN: 0029-8018. DOI: 10.1016/j.oceaneng.2025.121190. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801825009035.

[68] E. Pérez, L. Ntaimo, and Y. Ding, "Multi-component wind turbine modeling and simulation for wind farm operations and maintenance," *SIMULATION*, vol. 91, no. 4, pp. 360–382, Apr. 2015. DOI: 10.1177/0037549715572490.

[69] I. Bakir, M. Yildirim, and E. Ursavas, "An integrated optimization framework for multi-component predictive analytics in wind farm operations & maintenance," *Renewable and Sustainable Energy Reviews*, vol. 138, p. 110 639, Mar. 2021. DOI: 10.1016/j.rser.2020.110639.

[70] P. Y. P. Zhou, "An opportunistic condition-based maintenance strategy for offshore wind farm based on predictive analytics," *Renewable and Sustainable Energy Reviews*, vol. 109, pp. 1–9, Jul. 2019. DOI: 10.1016/j.rser.2019.03.049.

[71] M. Yildirim, N. Z. Gebraeel, and X. A. Sun, "Integrated Predictive Analytics and Optimization for Opportunistic Maintenance and Operations in Wind Farms," *IEEE Transactions on Power Systems*, vol. 32, no. 6, pp. 4319–4328, Nov. 2017. DOI: 10.1109/TPWRS.2017.2666722.

[72] J. Wang, Y. Liang, Y. Zheng, R. X. Gao, and F. Zhang, "An integrated fault diagnosis and prognosis approach for predictive maintenance of wind turbine bearing with limited samples," *Renewable Energy*, vol. 145, pp. 642–650, Jan. 2020. DOI: 10.1016/j.renene.2019.06.103.

[73] Y. Wang, Y. Zhao, and S. Addepalli, "Remaining Useful Life Prediction using Deep Learning Approaches: A Review," *Procedia Manufacturing*, vol. 49, pp. 81–88, 2020. DOI: 10.1016/j.promfg.2020.06.015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2351978920316528.

[74] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. DOI: 10.1038/nature14539.

[75] J. Zhou, J. Yang, S. Xiang, and Y. Qin, "Remaining useful life prediction methodologies with health indicator dependence for rotating machinery: A comprehensive review," *IEEE Transactions on Instrumentation and Measurement*, pp. 1–1, 2025, ISSN: 1557-9662. DOI: 10.1109/TIM.2025.3556919.

[76] P. Bangalore and M. Patriksson, "Analysis of SCADA data for early fault detection, with application to the maintenance management of wind turbines," *Renewable Energy*, vol. 115, pp. 521–532, Jan. 2018. DOI: 10.1016/j.renene.2017.08.073.

[77] K. Suzuki, Ed., *Artificial Neural Networks - Methodological Advances and Biomedical Applications*. InTech, Apr. 11, 2011. DOI: 10.5772/644. [Online]. Available: http://www.intechopen.com/books/artificial-neural-networks-methodological-advances-and-biomedical-applications.

[78] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org.

[79] H. Kaur, S. Ul Ahsaan, B. Alankar, and V. Chang, "A proposed sentiment analysis deep learning algorithm using gated recurrent unit and lstm,"

[80] D. Hughes and N. Correll, "Distributed machine learning in materials that couple sensing, actuation, computation and communication," *arXiv*, 2016. DOI: 10.48550/arXiv.1606.03508. arXiv: 1606.03508 [cs.LG]. [Online]. Available: https://doi.org/10.48550/arXiv.1606.03508.

[81] J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C.-H. Lee, "On Mean Absolute Error for Deep Neural Network Based Vector-to-Vector Regression," *IEEE Signal Processing Letters*, vol. 27, pp. 1485–1489, 2020, ISSN: 1558-2361. DOI: 10.1109/LSP.2020.3016837.

[82] F. Qiao, Y. Ma, L. Ma, S. Chen, H. Yang, and P. Ma, "Research on SCADA data preprocessing method of Wind Turbine," pp. 1606–1610, Apr. 2021. DOI: 10.1109/ACPEE51499.2021.9437009.

[83] R. Abdulkadirov, P. Lyakhov, and N. Nagornov, "Survey of optimization algorithms in modern neural networks," *Mathematics*, vol. 11, no. 11, p. 2466, 2023. DOI: 10.3390/math11112466.

[84] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," *arXiv preprint arXiv:1512.05287*, 2016.

[85] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[86] H. Chen, H. Liu, X. Chu, Q. Liu, and D. Xue, "Anomaly detection and critical SCADA parameters identification for wind turbines based on LSTM-AE neural network," *Renewable Energy*, vol. 172, pp. 829–840, Jul. 2021, ISSN: 0960-1481. DOI: 10.1016/j.renene.2021.03.078.

[87] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

[88] Z. Ye and J. Yu, "Health condition monitoring of machines based on long short-term memory convolutional autoencoder," *Applied Soft Computing*, vol. 107, p. 107 379, Aug. 2021, ISSN: 1568-4946. DOI: 10.1016/j.asoc.2021.107379.

[89] Y. Liu, Z. Liu, H. Zuo, H. Jiang, P. Li, and X. Li, "A DLSTM-Network-Based Approach for Mechanical Remaining Useful Life Prediction," *Sensors*, vol. 22, no. 15, p. 5680, Jan. 2022, ISSN: 1424-8220. DOI: 10.3390/s22155680.

[90] P. Shakya, M. S. Kulkarni, and A. K. Darpe, "A novel methodology for online detection of bearing health status for naturally progressing defect," *Journal of Sound and Vibration*, vol. 333, no. 21, pp. 5614–5629, Oct. 13, 2014, ISSN: 0022-460X. DOI: 10.1016/j.jsv.2014.04.058.

[91] X. Kong and J. Yang, "Remaining Useful Life Prediction of Rolling Bearings Based on RMS-MAVE and Dynamic Exponential Regression Model," *IEEE Access*, vol. 7, pp. 169 705–169 714, 2019, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2954915.

[92] B. Hu, S. Swamy, and K. Hermans, "Including realistic vessel and human behaviour in simulations of offshore wind farm operations," TNO, Tech. Rep. TNO 2018 R10937, Sep. 3, 2018.

[93] Y. Dalgic, I. Lazakis, O. Turan, and S. Judah, "Investigation of optimum jack-up vessel chartering strategy for offshore wind farm O&M activities," *Ocean Engineering*, vol. 95, pp. 106–115, Feb. 1, 2015, ISSN: 0029-8018. DOI: 10.1016/j.oceaneng.2014.12.011.

[94] Y. Zhao, D. Li, A. Dong, D. Kang, Q. Lv, and L. Shang, "Fault Prediction and Diagnosis of Wind Turbine Generators Using SCADA Data," *Energies*, vol. 10, Aug. 2017, ISSN: 1996-1073. DOI: 10.3390/en10081210.

[95] R. Morrison, X. Liu, and Z. Lin, "Anomaly detection in wind turbine SCADA data for power curve cleaning," *Renewable Energy*, vol. 184, pp. 473–486, Jan. 2022, ISSN: 0960-1481. DOI: 10.1016/j.renene.2021.11.118. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148121017134.

[96] P. Marti-Puig, A. Blanco-M, J. J. Cárdenas, J. Cusidó, and J. Solé-Casals, "Effects of the pre-processing algorithms in fault diagnosis of wind turbines," *Environmental Modelling & Software*, Special Issue on Environmental Data Science and Decision Support: Applications in Climate Change and the Ecological Footprint, vol. 110, pp. 119–128, Dec. 2018, ISSN: 1364-8152. DOI: 10.1016/j.envsoft.2018.05.002.

[97] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data Preprocessing for Supervised Leaning," vol. 1, no. 1, 2006.

[98] D. Ahn, S.-c. Shin, S.-y. Kim, H. Kharoufi, and H.-c. Kim, "Comparative evaluation of different offshore wind turbine installation vessels for Korean west–south wind farm," *International Journal of Naval Architecture and Ocean Engineering*, vol. 9, no. 1, pp. 45–54, Jan. 1, 2017, ISSN: 2092-6782. DOI: 10.1016/j.ijnaoe.2016.07.004.

[99] F. Porté-Agel, Y.-T. Wu, and C.-H. Chen, "A Numerical Study of the Effects of Wind Direction on Turbine Wakes and Power Losses in a Large Wind Farm," *Energies*, vol. 6, no. 10, pp. 5297–5313, Oct. 2013, ISSN: 1996-1073. DOI: 10.3390/en6105297. [Online]. Available: https://www.mdpi.com/1996-1073/6/10/5297.

[100] O. T. Bindingsbø, M. Singh, K. Øvsthus, and A. Keprate, "Fault detection of a wind turbine generator bearing using interpretable machine learning," *Frontiers in Energy Research*, vol. 11, Dec. 13, 2023, ISSN: 2296-598X. DOI: 10.3389/fenrg.2023.1284676.

[101] G. Rinaldi, P. R. Thies, and L. Johanning, "Current Status and Future Trends in the Operation and Maintenance of Offshore Wind Turbines: A Review," *Energies*, vol. 14, no. 9, p. 2484, Jan. 2021, ISSN: 1996-1073. DOI: 10.3390/en14092484.

[102] E. L. Petersen, N. G. Mortensen, L. Landberg, J. Højstrup, and H. P. Frank, "Wind power meteorology. part i: Climate and turbulence," *Wind Energy*, vol. 1, no. 1, pp. 25–46, 1998.

[103] G. van Bussel and W. Bierbooms, *Course offshore wind farm design oe 5662*, Lecture material, 2004. [Online]. Available: https://ocw.tudelft.nl.