HAZEM ATEF MOHAMED ALY

INFLUENCE OF EXTREME WEATHER CONDITIONS ON WIND TURBINE FAILURES FOR ENHANCED ENERGY PRODUCTION

UNIVERSIDADE DO ALGARVE
Instituto Superior de Engenharia
September 2025

HAZEM ATEF MOHAMED ALY

INFLUÊNCIA DAS CONDIÇÕES METEOROLÓGICAS EXTREMAS NAS FALHAS DE TURBINAS EÓLICAS PARA UMA PRODUÇÃO DE ENERGIA MELHORADA

Mestrado em Engenharia Mecânica

(Especialidade em Energia, Climatização e Refrigeração)

Trabalho realizado sob a orientação de:

Profª. Cláudia Dias Sequeira

UNIVERSIDADE DO ALGARVE
Instituto Superior de Engenharia
September 2025

INFLUÊNCIA DAS CONDIÇÕES METEOROLÓGICAS EXTREMAS NAS FALHAS DE TURBINAS EÓLICAS PARA UMA PRODUÇÃO DE ENERGIA MELHORADA

Declaration of authorship of the work

I hereby declare to be the author of this work, which is original and unpublished. Authors and works consulted are properly cited in the text and included in the reference list.

HAZEM ATEF MOHAMED ALY

Acknowledgements

Words fall short of expressing my gratitude to those who believed in me and gave me their time, effort, and support.

Starting with professor Cláudia Sequeira. This work would have never been finished without your guidance, and patience above all.

To all my professors at the university of Algarve. Thank you all! Not only for your academic teachings but also for being humble, patient and always supportive.

To my colleagues, you made my experience much easier with your welcoming attitude and the willingness to help one another.

To Gloria, thanks for believing in me even in the hardest times and for teaching me that kindness pays off sooner or later.

Finally, for my parents, thanks for the unconditional love. You are major partners in this achievement and every other one.

RESUMO

Esta dissertação analisa a relação entre as condições meteorológicas e as falhas em turbinas eólicas, com foco nas substituições de multiplicadores, através do alinhamento de registos de manutenção com dados meteorológicos horários do parque eólico de Lousã II (Portugal) entre 2019 e 2024. O conjunto de dados integrou ordens de trabalho, indicadores mensais de desempenho obtidos a partir de SCADA (velocidade do vento e potência) e variáveis do Meteostat (temperatura, humidade relativa, velocidade do vento e rajadas). Foram identificados oito eventos de substituição de multiplicadores, comparados com períodos de controlo sazonais equivalentes.

A metodologia incluiu a rotulagem supervisionada de variáveis meteorológicas em classes interpretáveis, a aplicação do algoritmo k-means para identificar regimes de funcionamento, e duas abordagens de aprendizagem automática: uma Árvore de Decisão, que gera regras condicionais do tipo se–então, e a Mineração de Regras de Associação (Apriori), que deteta combinações frequentes de condições meteorológicas antes das falhas. Em complemento, foi calculado um indicador simples de perda de energia, comparando a produção real com a teórica, e criado um OilScore para sintetizar resultados laboratoriais sobre a condição dos lubrificantes.

Os resultados mostraram que a maioria das falhas ocorreu em condições meteorológicas comuns, sobretudo em ventos calmos a moderados (velocidade < 15 m/s) e rajadas moderadas (≈15–25 m/s). Rajadas fortes ou extremas (> 35 m/s) estiveram associadas a poucas falhas, em linha com paragens automáticas de segurança. Humidade relativa elevada (≥80%) surgiu como contexto recorrente, em especial nos sistemas de passo e guinada. No paruqe éolico, verificou-se uma associação positiva entre perdas mensais de energia e temperatura, sendo que os meses mais quentes apresentaram maiores desvios face à curva de referência. Nas turbinas com substituições de multiplicadores, picos de perda de energia precederam várias intervenções, sugerindo utilidade para aviso prévio.

A hipótese de que as falhas aumentam em condições extremas não foi confirmada. Os riscos estão antes ligados à exposição prolongada a regimes húmidos e ventos moderados, o que reforça a importância de integrar métricas de perda de energia, dados meteorológicos e condição do óleo em estratégias de manutenção preditiva.

Palavras Chave: turbinas eólicas; caixa multiplicadora; falhas; manutenção

ABSTRACT

This dissertation investigates the relationship between weather and wind turbine failures, with emphasis on gearbox replacements, using maintenance records aligned with hourly meteorological data from the Lousã II wind farm (Portugal) for 2019–2024. The dataset integrates event-based work orders, monthly performance indicators derived from SCADA wind speed and power, and Meteostat weather variables (temperature, relative humidity, wind speed, and gusts). Eight gearbox replacement events were analysed alongside season-matched control periods. Methods included supervised labelling of weather into bins, k-means clustering to reveal operating regimes, and two machine-learning approaches: a Decision Tree to derive if–then rules, and Association Rule Mining (Apriori) to detect frequent co-occurring weather conditions preceding failures. In parallel, a simple energy-loss metric compared actual to theoretical production, and an OilScore summarised lubricant health.

Failures across components occurred mainly under ordinary weather conditions rather than extremes. Counts were highest in calm-to-low wind bins and moderate gusts (≈15–25 m/s), while very few events occurred during strong or extreme gusts—consistent with protective shutdowns. Higher relative humidity (≥80%) was a recurring backdrop, particularly for pitch and yaw systems. Fleet-level analysis revealed a positive link between monthly energy loss and temperature, with warmer months showing greater deviations from the reference curve. For turbines with gearbox replacements, pronounced energy loss spikes often preceded interventions, suggesting operational value for early-warning. In the gearbox-focused analyses, neither the Decision Tree nor the association rules revealed a single dominant trigger. Patterns involving low gusts and high humidity appeared before some events but should be regarded as exploratory due to limited data.

The evidence does not support the hypothesis that failures increase during extreme weather. Instead, risks align with prolonged exposure to humid, moderate-wind regimes. The study highlights opportunities for predictive maintenance by combining labelled weather, energy loss metrics, and oil condition data, and points to future work using exposure-based risk models, high-frequency SCADA features, and cross-site validation.

Keywords: wind turbines; gearbox failures; weather conditions, maintenance.

TABLE OF CONTENTS

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

Acronym	Definition
AOA	Angle Of Attack
AEP	Annual Energy Production
ARM	Association Rule Mining
CSV	Comma-Separated Values (data file format)
DT	Decision Tree
ISO VG	International Standards Organization Viscosity Grade (lubricant classification)
k-means	k-means clustering algorithm (unsupervised learning method)
ML	Machine Learning
MLxtend	Machine Learning Extensions (Python library for data mining and rule extraction)
NumPy	Numerical Python (library for scientific computing)
OT	Work Order (*Ordem de Trabalho*)
Pandas	Python Data Analysis Library
PQ index	Particle Quantifier Index (oil debris indicator in lubricants)
Python	Programming language used for analysis
SCADA	Supervisory Control and Data Acquisition
WT	Wind Turbine
wspd_mean	Mean wind speed (m/s)
wspd_max	Maximum wind speed (m/s)
wpgt_max	Maximum peak gust (m/s)
gust_factor	Ratio of maximum gust to mean wind speed
temp_mean	Mean air temperature (°C)
temp_min	Minimum air temperature (°C)
temp_max	Maximum air temperature (°C)
rhum_mean	Mean relative humidity (%)

rhum_max	Maximum relative humidity (%)
prcp_sum	Total precipitation (mm)
wdir_circ_std	Circular standard deviation of wind direction (°)

1

INTRODUCTION

Wind energy is now a major part of the power mix. New projects are larger, and the fleet already in the ground must run reliably to keep costs down. Failures still happen, and they are expensive. Weather is often blamed, but the exact link between day-to-day weather and specific failures is hard to determine and quantify. This thesis looks at that link using real maintenance records from a single wind farm and matching weather data. The objective is to describe what actually happens in the field where the exterior environment usually has a major role in the power production process. The number of variables is very large and the combinations among them are infinite.

When it comes to setting a definition for “Extreme weather condition”, two major points of view can be considered.

Firstly, a basic approach that is generally direct and straightforward such as wind speed higher than a certain value or temperature lower than a certain value. It looks at weather parameters from an individual one-dimensional perspective.

Secondly, a more realistic approach that considers the complexity of the real-world circumstances. This is a scenario where natural unpredictable combinations of wind speed, wind peak gusts, temperature, relative humidity, and many other variables interact with one another. In this view, what is “extreme” is the combination of specific patterns of weather conditions that might not cause any harm when occurring separately, however when occurring simultaneously they form an “extreme condition” relative to the components of the wind turbine. To make meaningful analysis of the available data, both approaches must be considered and combined where necessary as this takes the analysis one step closer to identify any recurring patterns, if they exist.

This study is interdisciplinary to an extent and requires some knowledge of modern data analysis methods and tools to experiment with and test different hypotheses. The use of python codes provided a flexible framework that allowed for an iterative process of searching for patterns applying different data analysis techniques and deciding which is useful and which is not relevant for this case.

1.1 OBJECTIVES AND CHARACTERISTICS OF THE DISSERTATION

This dissertation has three major objectives. The first is to build a clean and aligned dataset that matches component failures with the corresponding recent weather states. The second is to describe the data using simple labels and small K-clusters to identify common operating regimes. The third objective is to analyze if there are clear and repeatable weather patterns that occur in the periods prior to the replacement events of major components.

The focus is to discover how much these patterns might affect the maintenance and consequently, the reliability and power production of the wind turbine. The study is conducted from the point of view of mechanical and energy engineering utilizing the tools of data analysis to provide a clear reading of the evidence, based on the site's maintenance logs and weather variables.

1.2 CONTEXT OF THE WORK

To guide the analysis along the way to reaching the above-mentioned objectives, practical assumptions and hypotheses were set as starting point. One hypothesis is that failure counts increase during periods of strong wind speeds and in the event of unexpected wind gusts. Another hypothesis is that a big increase in one weather parameter (humidity for example) tends to occur more frequently around failures in specific subsystems. Finally, it is possible to identify a clear and repeatable pre-failure weather signature in the case of gearbox replacements.

This dissertation attempts to find whether these hypotheses can be proven based on evidence from the data analysis and their interpretation.

1.3 ORGANIZATION OF THE REPORT

The Introduction establishes the context of the study, outlines the objectives, and presents the central research question. It also discusses the motivation for investigating the influence of weather on wind turbine reliability and explains why gearbox failures are given particular attention.

Following the introduction, the Literature Review examines previous research on wind turbine failures, reliability challenges, maintenance strategies, and the role of environmental factors. It positions the present study within the broader academic and industrial context, with special emphasis on gearboxes and on the application of statistical and machine-learning methods for predictive maintenance.

The Methodology chapter then describes the datasets and the analytical framework adopted in the dissertation. It covers the cleaning and alignment of maintenance records, the extraction of weather data from Meteostat, the computation of SCADA-based energy loss indicators, and the development of an OilScore to summarize lubricant condition. It also presents the analytical methods applied, including supervised labelling, k-means clustering, Decision Tree analysis, and Association Rule Mining.

The Results and Discussion chapter presents the findings of these methods, first showing descriptive distributions of failures across labelled weather categories, then highlighting clusters of operating regimes linked to failures. It then focuses on the gearbox, presenting the rules generated by Decision Tree analysis and the frequent patterns identified by Association Rule Mining. Results on SCADA-based energy loss and OilScore are also integrated, providing a broader picture of the operational and environmental context of failures. The discussion interprets these results in light of the central hypothesis, evaluates the reliability of the observed patterns, and highlights both the strengths and limitations of the analysis.

The report concludes with the Conclusions chapter, which summarizes the main contributions of the study, revisits the central hypothesis, and reflects on the broader significance of the results. It also sets out directions for Future Work, outlining how the research can be extended through advanced modelling, integration of high-frequency SCADA and weather data, and cross-site validation, with the goal of strengthening predictive maintenance strategies in wind energy.

2

LITERATURE REVIEW

The production of wind energy has been expanding rapidly in response to climate change all over the world. It is estimated that the global wind power production has almost doubled from 2000 to 2017 as major economies such as USA, China and Germany, among others, expanded their wind energy capacity (Arshad & O'Kelly, 2019). Due to the growing risks of global warming, depletion of fossil fuel reserves, and the enforcement of more stringent environmental policies in the global energy market and society, the use of renewable energy sources has increased dramatically over the last ten years. With their rapid global growth, solar and wind energy have emerged as the most economically viable renewable energy sources of all those now accessible (Edenhofer et al., 2012).

There are two primary types of wind turbines: vertical axis wind turbines (VAWTs) and horizontal axis wind turbines (HAWTs), which are distinguished by their rotor structure and position in the airflow. Because of their increased efficiency, HAWTs have become quite popular in the commercial energy production sector. This is because a lot of time and effort has been put into their research and development. This study is entirely focused on the HAWTs, which are widely used in wind farms and have a wider acceptance rate. Numerous factors affect a HAWT's efficiency, such as the wind speed while it is operating, the blades' length, the tower's height, the casing's design, and the surrounding environment, which includes the weather and potential collisions with insects and birds. Therefore, the weather conditions have a significant impact on the efficiency of the wind turbine (Papież et al., 2019).

Weather conditions are a determining factor in the efficiency of the wind turbine. Wind turbines are prone to some productivity problems when operating harsh weather conditions which lead to damage or deformation in aerodynamic quality of the components. Usually this

occurs when the wind turbines are exposed to extreme weather and airborne particles. The overall performance of the wind turbine can, therefore, be affected as the blade surface aerodynamic quality degrades and deviates from the original design over time (Kelly et al., 2022). As a result, the annual energy production (AEP) of the turbines drops below the designed power.

Understanding how harsh climatic conditions affect airfoil performance is essential due to the aerodynamic properties of wind turbine blades and airfoils. Extreme conditions like hailstorms, rain, and icing, among others, pose unique challenges that significantly affect the basic aerodynamics of wind turbine blades and airfoils.

The challenges that face the operation of wind turbines due to weather conditions differ greatly depending on the environment and in this literature review the common problems are discussed.

Sand and dust are common airborne particles in areas with warm, dry climates, and they may give rise to LE erosion issues. However, this issue might not be present in more humid and greener settings. Additionally, sand erosion may pose serious risks to coastal regions. Generally speaking, dust buildup on the blade surface can decrease lift force and increase drag force, which lowers power production.

The study of (Khalfallah & Koliub, 2007) was the first to examine in detail the effects of dust accumulation-induced blade surface roughness on HAWT performance in the sandy dry region of Hurghada, Egypt close to the coast of the Red Sea. Their experimental study examined the mechanism of dust formation and building on the blade surface over the turbine's operating time, concentrating on a stall-regulated, 300 kW HAWT. Additionally, the study assessed how dust affected a pitch-regulated 100 kW HAWT's performance and contrasted the results with those of a 100 kW stall-regulated HAWT. They came to the conclusion that a number of variables, such as rotor speed and specifications, nacelle height above the ground, power-regulation type, and wind farm site characteristics, determine how dust affects HAWTs. Also, the study estimated the mean power loss of a wind turbine due to accumulated dust over different periods of time as shown in figure (1) below.

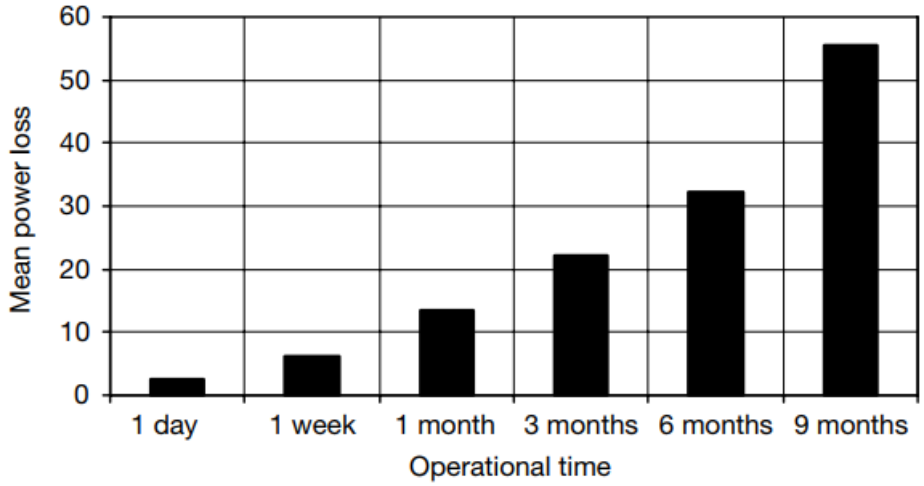


Figure 1. Percentage of Mean Power Loss due to dust effect (Khalfallah & Koliub, 2007)

The results of the study showed that the mean power loss was approximately 57% over a 9-month period.

The effect of sand particles on the primary airfoil's performance in a HAWT was also investigated by (Khakpour et al., 2007). The study considered major parameters such as the Angle Of Attack "AOA" which is defined as the angle between the chord line of the wind turbine blade and the relative wind direction. Examining various AOAs, they made a comparison between the flow pattern and pressure distribution under dusty and uniform flow circumstances. One of the their conclsuions was that coarse particles produce the most erosion at zero AOA, whereas small particles produce the highest erosion rate at high AOAs.

Rainfall is one of the many climatic factors that might affect HAWT performance during operation; it reduces aerodynamic lift and increases drag. Additionally, there is a lot of water vapor condensation in the low-pressure area above the airfoil, which releases latent heat from water droplets. On the airfoil surface, the residual droplets create a thin layer of water that is then impacted by other rains. This contact effectively roughens the airfoil surface and increases drag by forming craters and an uneven coating (Cao et al., 2014).

Despite rain being a common occurrence in some areas, little research has been done on how it affects HAWT operational effectiveness. Although a lot of research has been done on how rain affects aviation applications, there aren't many studies that particularly address the performance of wind turbines in such conditions.

(Corrigan & DeMiglio, 1985) carried out the first comprehensive investigation of how precipitation affects HAWT power output. They investigated the two-bladed Mod-0 HAWT experimentally using three distinct rotor configurations and demonstrated how rain affected

its functionality. According to the data, rain can cause performance to deteriorate by up to 20% in light rainfall and up to 30% in heavy rainfall. Additionally, at low wind speeds, the performance deterioration increased to as much as 36% when snow and drizzle were combined. Furthermore, an analysis was conducted to predict how rain might affect HAWT performance. They used a BEM code for their analysis, adding modified airfoil properties to take the rain effect into consideration. In high winds and moderate rainfall rates, the analysis estimated a 31% performance loss; these predicted outcomes were in good accordance with the experimental data.

Regarding the location in question in this dissertation, it is important to note that the location of Lousã II receives a moderate and predictable amount of precipitation almost every year which rarely reach any extreme level. Therefore, precipitation data was later excluded from the analysis as their effect was not considered of great impact for the purpose of this study.

Developments in CFD over the last few decades have greatly improved our comprehension of two-phase flow dynamics. Since rainfall is a two-phase flow, additional transport equations for the second phase must be addressed using CFD codes. Additionally, these codes make it easier to handle interaction terms that include mass, momentum, and energy transfers between phases. (Valentine & Decker, 1995) and (Durst et al., 1984) have researched and developed two popular methods for modeling fluid-particle dynamics in detail.

Following these, (Luers, 1985) analyzed theoretically, the aerodynamic performance of the same HAWT, and the computational results aligned well with the experimental data from (Corrigan and Demiglio, 1985). According to (Luers, 1985') findings, there is a significant 25% power loss at a wind speed of 10 m/s and a rain rate of 50 mm/h. Additionally, notable performance penalties are observed even under less intense rain conditions and lower wind speeds. It has been determined that the deterioration in performance is attributed to the aerodynamic roughness caused by the impact of raindrops and the waviness of the water film on the HAWT's blades.

(LUERS, 1985) found that at a wind speed of 10 m/s with a rain rate of 50 mm/h, there is a notable 25% loss in power. Furthermore, even with less severe rain and lower wind speeds, significant performance losses are seen. The aerodynamic roughness brought on by raindrop impact and the waviness of the water film on the HAWT's blades has been identified as the reason of the performance decline.

(Anh & Duc, 2019) developed a model and analytical technique that replicated the physical processes of raindrop formation on HAWT blades. Their model calculated the ideal wetness,

predicted the impact of precipitation, and then assessed power and performance in a range of rainy weather scenarios while taking into account the varied geometrical shapes of the turbine blades. They discovered that the impact force of rain was correlated with the wetness on HAWT blades, and that the ideal wind speed minimized both the impact force and power loss. Furthermore, a bigger raindrop size considerably decreased power output, which had an impact on the HAWT blades' rotation speed.

In a subsequent study, (Anh et al., 2022) suggested an analytical approach to examine a HAWT's power output and the pitch control system's performance in different wet scenarios. By simulating the physics-based process of raindrops landing on the blade surfaces, their model made it possible to calculate the ideal wetness levels based on the geometry of the swept region. Pitch angles should be reduced during rainy conditions when the wind speed surpasses the predetermined threshold value, according to their analysis of the HAWT's power generation and performance in wet situations.

Icing happens when supercooled water droplets harden on the blade's surface, increasing load and causing structural alterations that can seriously reduce the airfoil's aerodynamic properties. When there is ice on the surface, the wind becomes more turbulent, which increases drag. Consequently, the blade's capacity to generate power is reduced as the lift it produces diminishes.

Because of their advantageous, plentiful wind resources, many cold-climate nations currently use wind energy and are planning to further expand their wind energy capacity. These areas provide a naturally favorable setting for the use of wind energy. The difficult conditions brought on by atmospheric icing occurrences must be taken into consideration by HAWT manufacturers, especially airfoil designers for turbine blades, in order to guarantee optimal power production. Wind farms at higher elevations in colder climates show the most promise because wind speeds typically rise by 0.1 $m \cdot s^{-1}$ for every 100 m of altitude above the initial 1000 m (Parent & Ilinca, 2011). Additionally, because of their higher air density, colder climates have a wind power potential that is around 10% higher than that of other regions. The wind's kinetic energy is increased by the denser, colder air, which increases HAWT power generation (Hochart et al., 2008).

Icing is a physical phenomenon that has a major effect on HAWT performance and is a considerable challenge in cold climatic locations. Glaze ice and rime ice are the two main forms of ice creation. Glaze ice usually forms at 0°C when rain freezes on icy surfaces. It spreads over wide distances as ice sheets and has a transparent appearance. However, when very cold atmospheric moisture droplets come into touch with a cold surface, rime ice forms.

At temperatures lower than 0°C, rime ice starts building up. Ice on aerodynamic surfaces impairs HAWT performance and may cause sensor malfunctions or erroneous anemometer and wind vane readings. Additionally, ice accumulation increases the potential of falling ice hazards to personnel and can cause power line disruptions, rotor imbalances, and weakened aerodynamic braking (Manwell et al., 2009).

(Bose, 1992) investigated the natural icing events of a 1.05 m diameter HAWT intended for battery charging during the winter of 1990–1991. Glaze ice accumulation, which was caused by freezing rain and drizzle, was one of the most severe icing episodes that was recorded. At the blade section's leading edge (LE), the icing process showed a qualitatively similar pattern to the formation seen on fixed airfoil sections in icing wind tunnels under similar circumstances. More ice accumulated at the blade tips than at the blade roots as a result of the blades' circular action. A sizable area of the suction side of the sections was ice-free, but the pressure side was covered in ice.

A number of parameters, including temperature, wind velocity, mean volume diameter (MVD), and liquid water content (LWC), as well as geometrical features like size and shape, influence the buildup of ice on HAWT blades. Ice with varying quantities and forms is formed as a result of different temperatures and heat distribution throughout the HAWT blades.
(Seifert & Richert, 1997) carried out an experimental study on the NACA 4415 airfoil, a common airfoil found in HAWT blades. The study looked at the airfoil's aerodynamic coefficients both with and without different kinds of ice forms at the LE. The more ice that formed on the LE, the worse the aerodynamic deterioration. After three months of operation, the energy loss was estimated for a typical 300 kW turbine using aerodynamic data from the iced airfoil as input for load and power calculation algorithms. Depending on the ice development, the AEP decreased by 6% to 19%.

(Zhao et al., 2009) used both theoretical and experimental research to look into the fault mechanism of HAWT blade icing. First, icing and normal conditions were used to mimic the S830 airfoil's aerodynamic properties. According to the simulation results, the airfoil's aerodynamics are significantly impacted by the uneven ice covering, which lowers the lift coefficient and raises the drag coefficient, which directly lowers power output. Moreover, the HAWT experienced power frequency vibrations due to an exciting force produced by icing-induced mass unbalance. They came to the conclusion that there are two ways to detect icing: directly and indirectly. They found that power frequency vibrations and drops in power output are indicators of icing.

An important aspect in the formation of icing is the temperature and the droplet size. The effects of variations in atmospheric temperature and droplet size on the rate and shape of ice accretion and the resulting flow field properties were examined numerically by (Homola et al., 2010) for a 5 MW pitch controlled HAWT with blades made by the NACA 64-618 airfoil. The results showed that whereas ice lowers lift coefficients in all situations, the degree of this decrease differed. In particular, the cases that produced streamlined ice shapes that resembled rime ice showed only slight lift changes, while the cases that produced horn-shaped glazed ice showed more noticeable lift decreases.

Similar to the precipitation case, no analysis was made within the study of this thesis regarding the phenomenon of icing as it was not possible to identify the occurrence of icing in the location of Lousã II especially given its moderate climate and mild winters.

3

METHODOLOGY

The study investigates how weather conditions relate to wind-turbine failures by aligning maintenance events with meteorological states and then analyzing those events with a combination of data analysis methods of supervised labelling, unsupervised clustering, and targeted component studies.

The analysis in this work is based on a large set of raw data collected from the Lousa II wind farm between 2019 and 2024. The dataset combines continuous SCADA measurements with event-based maintenance and laboratory reports. The SCADA files are organized by parameter (e.g., bearing temperature, oil temperature, rotor speed, wind speed, etc.) and are provided in yearly Excel files with hourly resolution, covering all 20 turbines at the site. These records capture the operational and environmental conditions experienced by each machine over the six-year period. In addition, the dataset includes laboratory analyses of oil and grease samples, performed at irregular intervals, as well as a log of maintenance work orders that document failures, repairs, and component replacements. A site layout plan in PDF format provides contextual information on the physical arrangement of the turbines. Table 1 below summarizes the main sources of raw data, their format, and resolution.

Table 1 Summary of the available raw data of the wind farm

Data Source	Format	Variables (per turbine)	Resolution	Notes
Bearing Temp Gearbox	Excel	Gearbox bearing temperature [°C]	Hourly	Continuous SCADA data
OIL TEMP	Excel	Gearbox oil temperature [°C]	Hourly	Continuous SCADA data

Power	Excel	Active electrical power [kW]	Hourly	Includes curtailments and downtime
Rotor Speed	Excel	Rotor rotational speed [RPM]	Hourly	Continuous SCADA data
Temp Ambiente	Excel	Ambient air temperature [°C]	Hourly	Site-level, recorded per turbine position
Wind Speed	Excel	Wind speed [m/s]	Hourly	Continuous SCADA data
Wind Direction	Excel	Wind direction [°]	Hourly	Continuous SCADA data
Oil/Grease Analysis	Excel	Viscosity, water content, wear metals	Event-based	~292 lab reports, irregular intervals
Work Orders (failures)	CSV	Date, turbine ID, component, intervention	Event-based	Includes gearbox replacements, repairs, inspections
Site Layout	PDF	Turbine locations and access roads	–	Reference document, not time-series

In addition to SCADA data, a maintenance log in the form of a CSV file was provided, covering the same period from 2019 to 2024. This file contains detailed work order records for each intervention carried out at the Lousa II wind farm. Each entry corresponds to a single maintenance action and includes information such as the date and time, the turbine affected, the component involved, and the type of intervention (e.g., preventive inspection, corrective repair, or replacement). Descriptive notes and status fields indicate the reason for the intervention and its outcome, while time-related fields capture the duration of the maintenance activity.

Table 2 Example of the .CSV file containing the failure records

#PT_06.LDO_1WN=G010	2022-02-22-16.03.00.000000	2022-02-28-16.03.00.000000	Manutenção Anual - Tipo 2	OTHER
#PT_06.LDO_1WN=G005 CKA10	2022-02-22-17.01.00.000000	2022-02-22-17.01.00.000000	Substituir Sensor de incêndio	REPAIR
#PT_06.LDO_1WN	2022-02-24-17.56.00.000000	2022-02-24-17.56.00.000000	Auditoria APA	HSE
#PT_06.LDO_1WN=W300	2022-02-24-18.00.00.000000	2022-05-12-21.00.00.000000	BFD's danificados	HSE
#PT_06.LDO_1WN=W300	2022-03-01-12.08.00.000000	2022-06-23-12.08.00.000000	Limpeza de FGC sob linha MT/AT	OTHER
#PT_06.LDO_1WN=G016 MDK20 TL001	2022-03-02-09.33.00.000000	2022-03-02-17.44.00.000000	Inspeção à gearbox	OTHER
#PT_06.LDO_1WN=G015	2022-03-02-18.16.00.000000	2022-03-07-18.16.00.000000	Manutenção Anual - Tipo 2	OTHER
#PT_06.LDO_1WN=G010 MDL10	2022-03-03-18.32.00.000000	2022-03-16-18.02.00.000000	Substituir Yaw gears	REPAIR
#PT_06.LDO_1WN=G003 MDA20	2022-03-04-18.33.00.000000	2022-03-05-18.33.00.000000	Substituir pitch inverter	REPAIR
#PT_06.LDO_1WN=G015 MDK20 TL001	2022-03-07-18.13.00.000000	2022-12-05-19.27.00.000000	Recolha de amostra de óleo da Gearbox- 2ª fase	OIL
#PT_06.LDO_1WN=G015 MDK20 TL001	2022-03-07-18.42.00.000000	2022-12-05-19.05.00.000000	Recolha de amostra de óleo da Gearbox- 1ª fase	OIL
#PT_06.LDO_1WN=G014 MDL20	2022-03-08-19.04.00.000000	2023-03-15-18.33.00.000000	Ruído excessivo ao fazer yaw	REPAIR
#PT_06.LDO_1WN	2022-03-09-17.26.05.078000	2022-03-31-11.11.00.000000	PE Lousã II - Inspeção mensal às Turbinas Eólicas	
#PT_06.LDO_1WN=G001 MDA10	2022-03-09-17.26.05.265000	2022-04-05-11.38.58.860000	Inspeção ao sistema de pás mensal	
#PT_06.LDO_1WN=G001 MDA10	2022-03-09-17.26.05.375000	2022-04-05-11.38.59.001000	TE exterior - Verificar o estado das pás	
#PT_06.LDO_1WN=G001 MDA20	2022-03-09-17.26.05.453000	2022-04-05-11.38.59.064000	Inspeção ao hub mensal	

That work orders dataset was the cornerstone of this research and formed the basis for all subsequent analysis. While the SCADA measurements offered a detailed picture of turbine operation and environmental conditions, they remained abstract without a clear link to actual failures or interventions. The maintenance log provided this link by documenting the dates, components, and types of actions performed on each turbine. This made it possible to identify failure events, define the corresponding pre-failure operating windows, and distinguish corrective actions from routine inspections. In practice, the work orders file functioned as the reference dataset to which all other information was aligned.

- ACTSTART: Start time of maintenance or failure event.
- ACTFINISH: End time of the event.
- DESCRIPTION: Free-text description of the maintenance or issue.
- ZSOURCETYPE: Type of source triggering the task (e.g., REPAIR).
- LOCATION: Identification of the turbine involved.

3.1 DATA PREPROCESSING

In any data-driven research, preprocessing is a critical step that lays the foundation for reliable analysis and valid results. Given the complexity of the datasets involved in this study—comprising meteorological data and wind turbine maintenance logs, elaborate preprocessing steps were required to ensure the datasets were accurate, clean, and compatible for correlation analysis.

In this step the historical data was cleaned and organized to ensure that the failures considered were actual failures affecting the components of the WT. The main purpose of this step was to exclude the irrelevant data such as regular periodic maintenance actions, periodic inspections, specimen collections etc.

The preprocessing started with setting some sort of criteria for what would be considered a failure. Routine inspections, planned maintenance, sampling/specimen tasks, and administrative entries were excluded. Remaining records were assigned to components based on the log description. Thereafter, the historical data was systematically divided into three main categories:

Actual Failure Data: These are logs of incidents that necessitated maintenance time and cost, encompassing all part replacements and repairs. This data is further separated by the affected component. The main components analyzed include Gearbox, Generator, Yaw

system, Pitch system, and Blades, in addition to an "Other" category for unspecified components or remaining actions.

Oil Replacements and Hydraulic System Data: This category contains entries related to oil replacement.

Irrelevant Data: Any data not relevant to the study's scope was excluded.

Following the filtering process, out of the 67639 raw entries, a total of 869 maintenance actions were identified as relevant for this study. The distribution of these actions across components is detailed as follows in table 3 below:

Table 3 Count of relevant failures per component

Component	Number of filtered failures
Blades	146
Yaw system	57
Pitch system	52
Generator	25
Gearbox	23 (Including 8 replacement events of 8 different gearboxes)
Other	566

3.2 TOOLS FOR THE ANALYSIS

To conduct the data analysis, a set of specialized computational tools was employed, with a focus on open-source libraries within the Python ecosystem. The choice of tools was guided by their robustness, flexibility, and capacity to handle large-scale, time-stamped datasets efficiently.

Python was the primary programming language used in this analysis. Its widespread adoption in scientific computing, combined with a vast ecosystem of libraries tailored for data processing and machine learning, makes it particularly suitable for tasks involving structured data, temporal alignment, and statistical modelling. Python’s syntax is both expressive and concise, facilitating reproducibility and collaboration in research settings. Furthermore, its integration capabilities allow seamless interaction with external data sources, such as weather APIs and spreadsheet formats, which was essential for this project.

To retrieve historical weather data corresponding to the recorded failure events, the Meteostat library was used in this study. It provides open-access, high-resolution meteorological records from global weather stations and reanalysis models. In this work it was employed to obtain hourly data on temperature (*temp*, °C), relative humidity (*rhum*, %) , wind speed (*wspd*, m/s) , and peak gusts (*wpgt*, m/s) at the exact geographic coordinates of the wind farm. The weather data was then aligned temporally with the maintenance records by querying conditions ten minutes prior to each failure, ensuring consistency between operational and environmental datasets.

Meteostat's Python API facilitated seamless integration into the data-processing workflow, while its standardized formats and built-in quality checks reduced preprocessing needs and improved the reliability and reproducibility of the analysis.

It is important to mention that by default, Meteostat provides wind speed and gust values in kilometers per hour (km/h). However, for consistency with engineering standards and wind turbine design specifications, these were converted to meters per second (m/s).

Other Python libraries were employed to manage, visualize, and analyze the data. Pandas served as the core tool for data manipulation, enabling the loading and cleaning of Excel files, standardization of date formats, merging of weather and maintenance datasets, and creation of derived variables. Its ability to handle missing values and align heterogeneous data by timestamps was particularly useful for synchronizing records. For visualization, Matplotlib and Seaborn were used to generate bar charts, scatter plots, and pair plots that illustrated both the distribution of failures across weather categories and the outcomes of clustering analysis. Finally, Scikit-learn provided the framework for applying machine learning techniques, most notably k-means clustering, along with preprocessing steps such as feature scaling and evaluation metrics, allowing the identification and interpretation of hidden patterns in the weather–failure relationships.

3.3 DATA ANALYSIS

The analysis began with a combined use of supervised labeling and unsupervised clustering to study the operating conditions of the wind farm. Supervised labeling was applied to categorize weather variables into intuitive classes, such as cold, mild, or hot temperature ranges, and dry, moist, or corrosive humidity levels. These categories were then used to align with maintenance data for different turbine components, including gearboxes, blades, yaw systems, generators and other key subsystems. In parallel, unsupervised clustering techniques, particularly k-means, were employed to group failure cases and operational states without predefined labels,

helping to reveal hidden patterns in how turbines respond to environmental stresses. This dual approach provided both structure, through labeled categories that could be directly interpreted, and discovery, through clustering that exposed less obvious correlations across the dataset. Figure 2 below demonstrates the framework used for data processing.

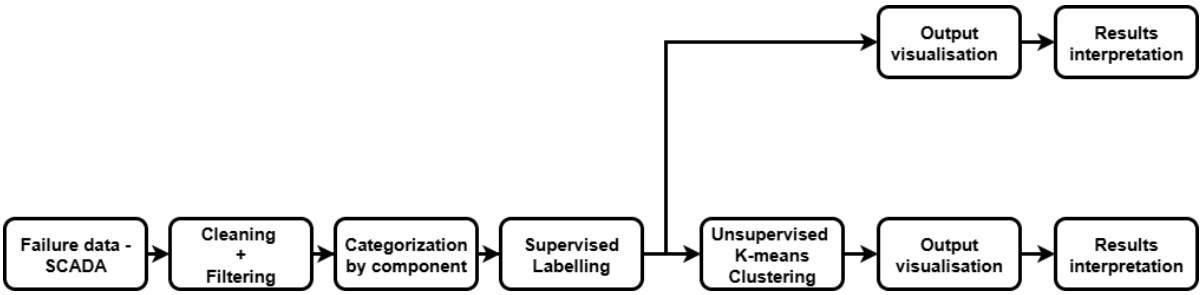


Figure 2. Diagram of the framework used for processing the data for all components

Given their high cost and operational significance, gearboxes received a more detailed analysis. For the eight documented replacement events, additional investigations were carried out to better understand pre-failure conditions. In these cases, decision tree models were used to identify which weather and operational variables contributed most to the occurrence of failures, providing interpretable rules that highlight the relative importance of different stressors. In addition, association rule mining was applied to uncover frequent co-occurrences between specific weather categories and gearbox replacements, revealing patterns that may not emerge through classical statistical methods. This focused analysis offered deeper insight into the unique mechanisms behind gearbox failures and complemented the broader patterns observed across the turbine fleet. Figure 3 below shows the additional analysis techniques used specifically for the gearboxes.

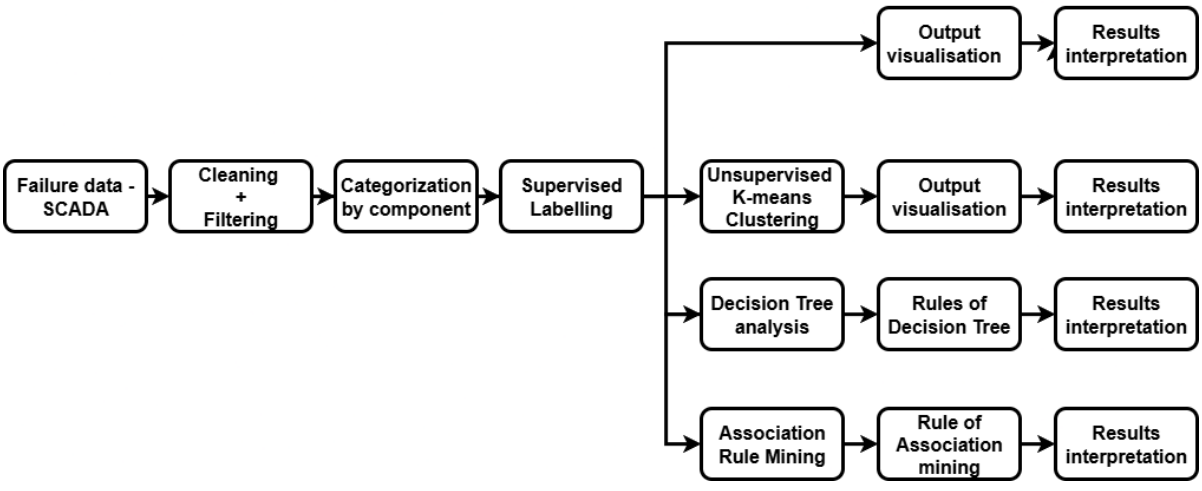


Figure 3. Diagram of the framework used to further investigation of the gearbox replacement events

However, it is important to note that while the methods were designed to be clear and reproducible, there are some limits that affect how the results should be read:

- The study assumes that a recorded maintenance action reflects the timing and nature of the actual failure. In reality, the event date may refer to when the repair was scheduled or logged, not when the fault occurred.
- Failure counts are reported directly, without normalizing by how long the turbines operated under each weather condition. Higher counts in "normal" weather may simply reflect longer exposure times.
- Gearbox and generator events are few, which limits the strength of any patterns found with these sub-systems.
- Using small values for K in the K-Means analysis helps keep the results readable but it may not capture all possible regimes in the data.
- The supervised labels are based on engineering judgement and convenience. Different threshold values might produce slightly different groupings.

These limitations do not prevent the analysis from meeting its main aim — to describe the observed patterns — but they should be kept in mind when interpreting the findings or applying them to other sites

3.3.1 SUPERVISED LABELLING ANALYSIS

Supervised labeling is a data analysis technique in which raw numerical values are assigned to predefined categories, making complex datasets easier to interpret and analyze. Instead of working with continuous weather variables such as temperature, humidity, or wind speed—which can take on thousands of different values—supervised labeling groups these measurements into meaningful classes. This process transforms data from being purely quantitative to also being qualitative, so that patterns can be more easily recognized and compared. For example, instead of looking at exact temperatures like 12.3 °C or 28.7 °C, the data can be grouped into categories such as "cold," "mild," or "hot." The benefit of this approach is that it provides a structured framework where relationships between weather conditions and turbine performance can be directly tested, and it reduces the noise that comes with granular data.

In this thesis, supervised labeling was used to classify weather conditions that are known to influence wind turbine reliability, focusing on temperature, relative humidity, and wind speed. The labeling framework was primarily based on the study of Reder et al. (2018), which defined scientifically grounded thresholds for environmental conditions associated with turbine failures. For example, their scheme included relative humidity labels such as dry air (20–40%), moist air (40–60%), corrosive (60–80%), highly corrosive (80–98%), and precipitation (100%); temperature ranges from freezing (−10–0°C) to very hot (35–40°C); and wind speed bands aligned with turbine operating regimes: calm (<3 m/s), low (3–10 m/s), high (10–26 m/s), and storm (>26 m/s). As they note, "Supervised labelling is used to define thresholds and assign labels to the input parameters based on expert judgements and findings from literature… wind speed labelling was carried out using labels according to common literature findings and the typical cut-in and cut-out wind speeds of WTs… relative humidity can be labelled in terms of its resulting corrosiveness".

While this thesis adopted Reder et al.'s categorization as the scientific baseline, slight adjustments were made to better reflect the conditions of the Lousã II wind farm and the characteristics of the dataset used here. These adjustments were necessary to account for local climatic ranges and to ensure that the categories were balanced in terms of data representation. In this way, the labeling scheme remained both scientifically justified and directly relevant to the data under study, providing a robust foundation for later analyses such as clustering, decision trees, and association rule mining. Table 4 below explains how the labels were adapted to this study.

Table 4 Adapted weather thresholds compared to Reder et al. (2018), adjusted for Portuguese climate and Nordex N90/2500 operation.

Parameter	**Reder et al. (2018) thresholds**	**Adapted thresholds (this thesis)**	**Notes**
Relative Humidity (%)	Dry air: 20–40% Moist air: 40–60% Corrosive: 60–80% Highly corrosive: 80–98% Precipitation: 100%	Dry air: <40% Moist air: <60% Corrosive: <80% Highly corrosive: ≥80%	Simplified to four categories; thresholds shifted slightly lower to reflect higher average humidity in Portugal.
Temperature (°C)	Freezing: −10–0 Very cold: 0–5 Cold: 5–10 Cool: 10–15 Mild: 15–20 Room temp.: 20–25 Warm: 25–30 Hot: 30–35 Very hot: 35–40	Cold: ≤10 Cool: ≤15 Mild: ≤20 Room temp.: ≤25 Warm: ≤30 Hot: >30	Reduced from 9 to 6 categories; adjusted to reflect Portuguese climate (rare subzero temps, more emphasis on mild to warm ranges).
Wind Speed (m/s)	Calm: <3 Low: 3–10 High: 10–26 Storm: >26	Calm: <3 Low: 3–10 High: 10–26 Storm: >26	The same labels are used.
Wind Gust (m/s)	*Not explicitly included*	Light: <15 Moderate: <25 Strong: <35 Extreme: ≥35	New variable added in this thesis to capture short-term gust effects, considered important for failure analysis.

Finally, to visually explore the relationship between environmental stress and component failures, the script generated a set of grouped bar charts. These charts display the frequency of failures under each labelled weather condition, broken down by turbine component. The entire supervised labelling process was done by a python code provided in Appendix A.

3.3.2 UNSUPERVISED CLUSTERING USING K-MEANS

Unsupervised clustering is a family of methods used to discover natural groupings in data without relying on predefined labels. Among these methods, k-means clustering is one of the most widely applied because of its simplicity and effectiveness in handling large datasets. The basic idea of k-means is to partition a dataset into k groups, or "clusters," in such a way that points within the same cluster are more similar to each other than they are to points in other clusters. Similarity is measured using distance, most often the Euclidean distance, which

makes the method intuitive: each data point is assigned to the cluster whose center (called a "centroid") it lies closest to, and the centroids are recalculated until the clusters stabilize. In practice, this means that the algorithm is able to take a large amount of complex, continuous data and structure it into a limited number of categories that reflect patterns inherent in the data itself, rather than ones imposed by the analyst.

In this thesis, k-means clustering was applied to the weather and maintenance data in order to identify hidden patterns that could not be revealed through supervised labeling alone. While labeling provided clear, interpretable categories based on thresholds (such as "cold" or "corrosive"), it still relied on human judgment and assumptions about what boundaries were important. Clustering, in contrast, let the data itself determine how conditions should be grouped. This was particularly valuable for variables like wind speed, temperature, and humidity, where conditions do not always fall neatly into fixed categories and where failures may be triggered by combinations of factors rather than single thresholds. By running the k-means algorithm, environmental variables were grouped into clusters that captured their typical ranges and co-occurrences. These clusters then served as the basis for further analysis, allowing us to ask whether certain types of environmental patterns, emerging directly from the data, were linked to higher frequencies of component failures.

The clustering was carried out by feeding the cleaned and time-aligned dataset into the algorithm, specifying the number of clusters k using established criteria such as the elbow method and silhouette scores. In practice, this ensured that the chosen k provided a balance between over-simplifying the data (too few clusters) and over-fragmenting it (too many clusters). For example, the weather parameters leading up to each failure event were grouped into three or four clusters, depending on the variable, which represented typical operational "modes" of the wind farm. Each failure record could then be associated with a particular cluster, making it possible to investigate whether gearboxes, blades, or yaw systems tended to fail under specific environmental clusters more than others. This process offered a more data-driven view of operating conditions, complementing the more interpretable but less flexible supervised labeling. Figures 4 and 5 below demonstrate the process of creating the clusters.

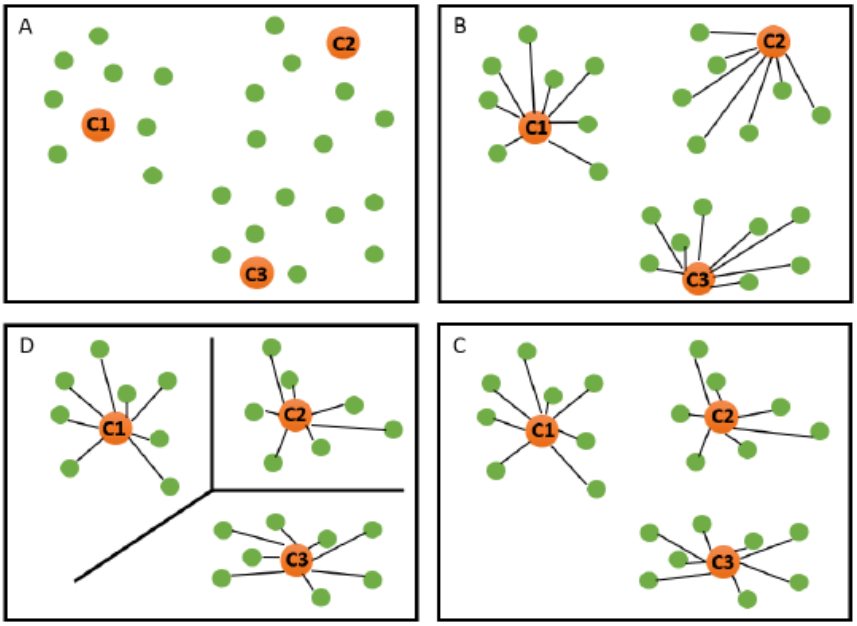


Figure 4. The K-Means clustering process: Three centroids are randomly chosen, showing the initialization of centroids, assignment of points, updating of centroids, and convergence into stable clusters. Adapted from (Kandali et al., 2021)

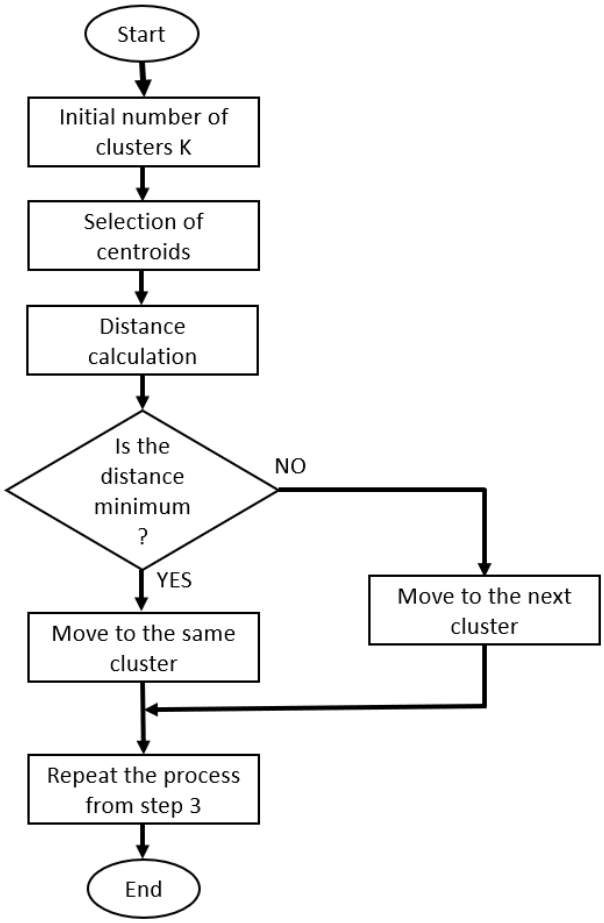


Figure 5. Flowchart of the k-means clustering process. Adapted from Kandali et al. (2021).

The value of k-means clustering in this study lies in its ability to reduce complexity while remaining objective. Rather than relying on manually set thresholds, the algorithm allowed the structure of the dataset itself to guide the categorization of conditions. This was particularly important in the context of wind turbine reliability, where failures are often influenced by subtle and cumulative interactions between multiple environmental variables. By applying clustering separately for each component—such as blades, gearboxes, and pitch systems—it became possible to see whether particular weather profiles were more commonly associated with certain types of failures. This approach provided insights that could not be obtained through labeling alone. The strength of this method is also reflected in the wider literature. Clare et al. (2024) highlights the power of k-means for discovering weather regimes relevant to turbine performance, while Clifton and Lundquist (2012) show how clustering environmental variables can reveal meaningful patterns in wind behavior. More generally, the effectiveness and best practices of clustering techniques are well documented, as reviewed by Jain (2010). By combining supervised labeling and unsupervised clustering, the analysis in this thesis gained the dual benefit of interpretability and discovery—labels made results easy to communicate, while clusters revealed hidden patterns that might otherwise have remained unnoticed. The K-means clustering process was performed by a python code provided in Appendix B.

3.3.3 DECISION TREE ANALYSIS FOR GEARBOX REPLACEMENT EVENTS

Decision Tree analysis is a supervised learning method that organizes data into a series of branching *if–then* rules. The concept is straightforward but powerful: the algorithm scans through available variables—such as wind speed, gust intensity, temperature, and humidity—and searches for thresholds that best split the data into groups with different outcomes, in this case gearbox failure versus non-failure. At each step, the algorithm selects the variable and threshold that maximise the separation between the two groups, gradually building a branching structure that resembles a tree. The end points of the branches, or *leaves*, represent conditions that are strongly associated with one outcome or the other. What makes decision trees particularly appealing is their interpretability: they ultimately produce explicit rules that can be expressed in plain language.

In this study, the Decision Tree method was applied to investigate whether specific thresholds or combinations of weather parameters could distinguish periods preceding gearbox failures from periods without failures. Gearboxes are critical components of wind turbines,

and their replacements represent significant operational and financial costs, making the identification of the environmental conditions leading up to these events an important task. While unsupervised clustering and supervised labelling provided a general picture of environmental influences, they did not reveal the exact combinations of variables most closely associated with failures. The Decision Tree approach helped to fill this gap by identifying the decision boundaries where risk appeared to shift.

The analysis considered eight recorded gearbox replacement events in the wind farm:

- G008 – 2021-04-30
- G012 – 2022-05-23
- G016 – 2022-06-20
- G006 – 2023-03-17
- G011 – 2023-04-10
- G004 – 2023-04-20
- G015 – 2024-06-14
- G003 – 2024-11-04

For each event, two observation windows were defined:

- Short-term window: the 7 days preceding the failure.
- Medium-term window: the 30 days preceding the failure.

To provide a comparative baseline, an equal number of control periods were selected from times when no gearbox replacements occurred. These control periods were matched to similar seasonal conditions in order to reduce the influence of seasonal bias. Each observation period—whether a pre-failure window or a control window—was then assigned a binary label:

- Failure = 1 for pre-failure windows.
- Failure = 0 for control windows.

From the Meteostat hourly dataset, each observation window was summarised using aggregated weather variables considered most relevant to mechanical loading and environmental stress on wind turbines. The selected parameters were:

- wspd_mean – Mean wind speed (m/s)
- wspd_max – Maximum wind speed (m/s)
- wpgt_max – Maximum peak gust (m/s)
- temp_mean, temp_min, temp_max – Mean, minimum, and maximum temperature (°C)
- rhum_mean, rhum_max – Mean and maximum relative humidity (%)

Each metric was calculated separately for both short-term and medium-term windows, producing a dataset of pre-failure and control conditions. The algorithm then split the dataset

into subgroups by identifying thresholds in these variables that maximised the distinction between failure and non-failure periods. Each split took the form:

- *If variable ≤ threshold → follow left branch; else → follow right branch.*

This recursive process continued until one of three stopping conditions was met:

1. A branch contained only one class (pure node),
2. No further statistically meaningful splits were possible, or
3. The maximum allowed depth of the tree was reached.

This approach can be valuable for two main reasons. First, it demonstrates that weather parameters do not act in isolation but may combine in subtle ways to increase stress on turbine components. For instance, a single episode of strong gusts might not be sufficient to cause damage, but when combined with sustained high humidity the likelihood of mechanical stress could be higher. Second, the method produces explicit rules that are both interpretable and communicable, making them useful for academic analysis as well as for practical applications such as predictive maintenance.

The full Python code used to perform the Decision Tree analysis is provided in Appendix C.

3.3.4 ASSOCIATION RULE MINING FOR GEARBOX REPLACEMENT EVENTS

The Association Rule Mining (ARM) analysis was applied to the same dataset described in the Decision Tree methodology. This included the eight gearbox replacement events and their corresponding control periods, with each observation window summarized using Meteostat data. The same 7-day short-term pre-failure windows were used, ensuring direct comparability between methods.

While the Decision Tree aims to create a hierarchical model that predicts outcomes, ARM instead focuses on uncovering frequent co-occurrences of conditions that appear disproportionately in pre-failure periods. In this case, the analysis was limited to the same weather variables previously highlighted as most relevant: wind speed, wind gusts, temperature, and relative humidity.

To make the continuous data suitable for ARM, each variable was discretised into Low, Medium, and High categories. Thresholds for these categories were determined directly from the data distribution by dividing the observed values into three equal-frequency intervals. For example, in this dataset:

- wpgt_max=Low corresponded to maximum gusts of ≤ 11.30 m/s,
- wspd_mean=Low referred to mean wind speeds of ≤ 2.50 m/s,

- rhum_max=High referred to maximum humidity values above 97.67%.

The data were then transformed into a "transaction" format, with each window represented as a set of conditions (e.g., *wspd_mean=Low, rhum_max=High*) along with the outcome label (*Failure=Yes* or *Failure=No*).

To analyse these transactions, the Apriori algorithm was applied. This algorithm systematically searches through the dataset to identify combinations of conditions, or *itemsets*, that occur more frequently than would be expected by chance. It works in a stepwise manner: first detecting frequent single conditions, then combining them into larger sets only if the smaller ones are also frequent. This property—known as the *Apriori principle*—ensures that the search remains computationally efficient. From these frequent itemsets, association rules were generated with "Failure=Yes" as the consequent. Each rule was then evaluated using standard ARM metrics:

- Support – how often the condition occurs in all periods,
- Confidence – the probability of failure when the condition is present,
- Lift – how much more likely a failure is under the condition compared to chance.

The Python implementation of this procedure is provided in Appendix D.

3.3.5 ENERGY LOSS AGAINST WEATHER CONDITIONS

This section quantifies how weather relates to monthly energy losses for and checks whether turbines with gearbox replacements show distinctive patterns. The approach is intentionally simple, reproducible, and based on SCADA records aligned with public weather data. All outputs were designed for clarity, with common axes across figures and explicit sample counts. The data used in the analysis is summarized in table 5 below. Also, key assumptions and considerations are summarized in table 6.

Table 5 Overview of data sources, variables, resolution, and study periods used in the analysis.

Source	Variables	Resolution	Period	Notes
SCADA (farm)	Active Power (kW), Nacelle Wind Speed (m/s) per WT	Hourly	2019–2024	Single sheet with WTxx_Power(kW) and WTxx_Wind(m/s) columns plus DateTime.
Public weather (Meteostat)	Temperature (°C), Relative Humidity (%), mean wind (m/s), gust (m/s), precipitation (mm)	Hourly	Matched to SCADA window	Data converted to Europe/Lisbon local time.
Reference curve	Nordex N90/2500 theoretical power curve	—	—	Used as production reference. A standalone figure of the curve is included and cited.
Maintenance log	Gearbox replacements	Event	2021–2024	Events marked for WT08, WT12, WT16, WT06, WT11, WT04, WT15, WT03.

Table 6 Key assumptions and filtering rules applied to performance and loss calculations.

Item	Value / Rule	Rationale
Rated power	2,500 kW	From manufacturer's brochure
Cut-in speed	4 m/s	From manufacturer's curve
Downtime / curtailment filter	Exclude rows where Wind ≥ 3 m/s and Power < 10 kW	Avoid counting stops/curtailment as "weather loss"
Time basis	Monthly aggregation of hourly records	Smooths noise and aligns with operational practice
Common y-axis for losses	0–40%	Ensures comparability across turbines

For each hourly nacelle wind speed , theoretical power was interpolated on the Nordex N90/2500 curve:

$$P_{\text{theo}}(v) = \begin{cases} 0 & v < 3\,\text{m/s} \\ \text{linear interpolation on curve} & 3 \le v \le 25\,\text{m/s} \\ 2500 & v > 25\,\text{m/s} \end{cases}$$

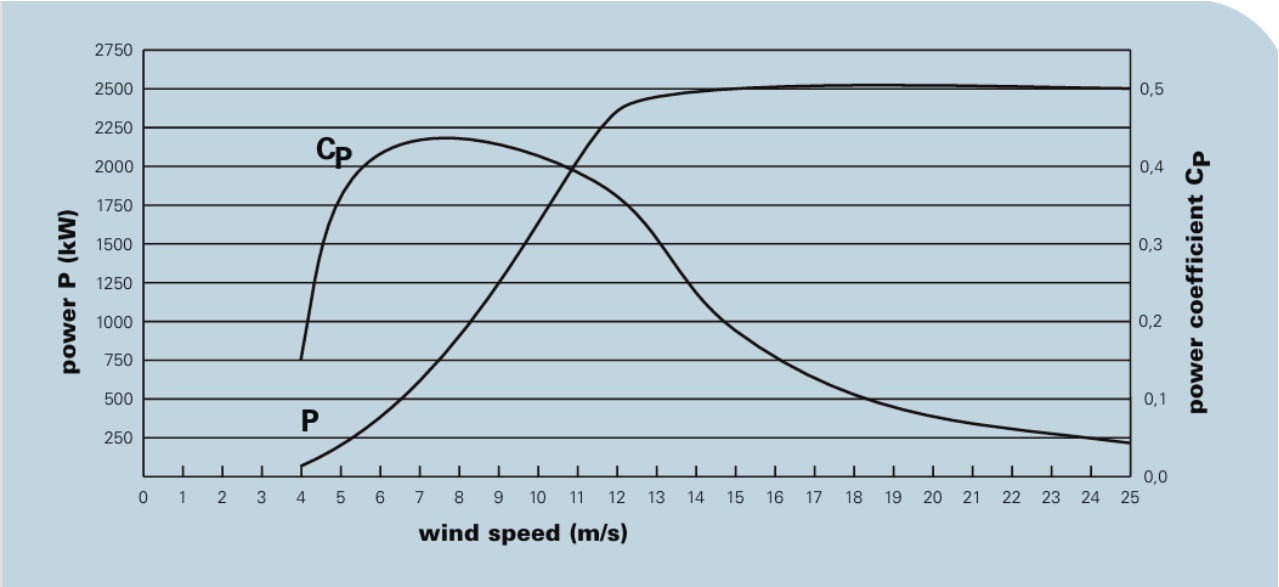


Figure 6. Nordex N90/2500 theoretical power curve – source: manufacturer brochure

Hourly rows with Wind ≥ 3 m/s and Power < 10 kW were dropped to avoid misclassifying downtime as weather-related loss.

For each turbine t and month m the actual and theoretical energies were accumulated:

$$E_{\text{act}}(t,m) = \sum \text{Power}_{\text{act}}\,[kW] \;\div 1000 \;\rightarrow \text{MWh}$$

$$E_{\text{theo}}(t,m) = \sum P_{\text{theo}}(v)\,[kW] \;\div 1000 \;\rightarrow \text{MWh}$$

Monthly energy loss was then defined as:

$$\text{Loss}(t,m) = \begin{cases} \dfrac{E_{\text{theo}}(t,m) - E_{\text{act}}(t,m)}{E_{\text{theo}}(t,m)} \times 100, & E_{\text{theo}}(t,m) > 0 \\ \text{NaN}, & \text{otherwise} \end{cases}$$

The number of valid hourly records N(t, m) was preserved for each month. This was later used in plots (marker size) and in the summary table (median monthly N, abbreviated medN).

Hourly Meteostat observations were aligned with SCADA timestamps and aggregated monthly. The following indicators were computed:

- T_{mean} (°C)
- RH_{mean} (%)
- $GustRatio_{\text{mean}} = \text{gust}/\text{mean wind}$ (–)
- $Rain_{\text{total}}$ (mm/month)

These were merged with turbine-month losses to yield a single joined dataset containing: Turbine, Year, Month, E_{act}, E_{theo} , Loss%, Hours, and weather metrics.

Gearbox replacements were annotated on time-series plots as vertical dashed lines. In scatter plots of Loss vs Temperature, points were color-coded as "pre" or "post" intervention to make potential performance shifts visible.

The methodology produces four complementary outputs:

- Fleet time-panel: a 4×5 grid of monthly energy losses, with medN annotated and gearbox events indicated.
- Comparative fleet panels: loss vs temperature, humidity, gust ratio, and rainfall. Marker size is proportional to sample count; turbines with interventions show pre/post coloring.
- Per-turbine figures: individual monthly timelines with gearbox markers, scatter plots for each weather driver, and loss vs temperature with a 95% confidence band.
- Summary table: per turbine, total filtered hours, median monthly samples, number of months, and mean/min/max losses, with gearbox presence indicated.

The calculation of losses was based directly on the manufacturer's reference curve without correction for site-specific air density. This choice was intentional. Rather than attempting to predict absolute energy production, the objective was to quantify deviations from a fixed benchmark. By keeping the curve unadjusted, environmental effects such as warm-season density reductions or temperature-driven derating appear naturally as positive loss percentages. Later in the thesis, a robustness check with a density-corrected curve is presented, to illustrate how much of the observed temperature signal can be attributed to aerodynamic conditions versus operational or mechanical factors.

All figures clearly state the time basis ("monthly averages, downtime-filtered") and the analysis window. Loss axes are fixed to 0–40% to ensure comparability. Legends appear only when needed (e.g. pre/post intervention). Captions explain what is shown, when, and why it matters. The workflow is fully scripted in Python, ensuring reproducibility (Appendix E).

Regarding limitations, nacelle wind speeds differ from free-stream wind and Meteostat data represents area-level rather than on-tower conditions. Monthly aggregation, however, reduces

these mismatches and provides stable signals. The downtime filter is intentionally simple; a more detailed flagging system (curtailments, fault codes) could refine exclusions. Trendlines in scatter plots are exploratory and are presented with raw data and sample counts to avoid overstating precision.

3.3.6 OIL ANALYSIS AND ENERGY LOSS

The methodological approach adopted in this study began with the collection of laboratory oil and grease analysis reports, which were originally provided in the form of nearly three hundred PDF documents covering the six-year operational period of the wind farm. Each report contained measurements of chemical and physical properties of the gearbox lubricants, as well as trace metals, contamination levels, and laboratory engineer comments. In their raw form, these documents were not suitable for systematic analysis, so the first task consisted of designing a workflow to automatically parse and standardize them into a structured dataset. To achieve this, a Python script was developed that read each PDF file, extracted the relevant tables, and identified key parameters such as iron (Fe), copper (Cu), the Particle Quantifier index (PQ), water content in parts per million, viscosity at 40 °C, oxidation level, and ISO cleanliness class. An additional step was the extraction of metadata, particularly the turbine identifier, which was encoded in the file name (e.g., “wtg01” indicating turbine 01). This allowed each row of the resulting dataset to be tied unambiguously to a specific machine and sampling date.

Once the parsing was complete, the dataset contained several hundred records with consistent column names and turbine identifiers. The reports themselves followed a common template and included a wide range of variables, which can be grouped into several categories. Table 7 summarizes the main types of information recorded by the laboratory. This structure reflects the richness of the raw reports, even if only a subset of the variables were ultimately employed in the OilScore calculation. Also an example of the lab reports is provided in the Appendices F and G.

Table 7 Oil analysis variables reported across categories of wear, contamination, condition, and maintenance context.

Category	Variables reported
Sample and machine info	Turbine identifier, gearbox manufacturer, serial number, sampling date, test date
Wear metals	Iron (Fe), Copper (Cu), Lead (Pb), Chromium (Cr), Nickel (Ni), Tin (Sn), Molybdenum (Mo), Aluminium (Al), etc.
Contamination	Water (ppm), Silicon (Si), Sodium (Na), Potassium (K), ISO 4406 cleanliness class
Oil condition	Viscosity at 40 °C and 100 °C, oxidation index, nitration index, infrared index
Additive package	Zinc (Zn), Phosphorus (P), Calcium (Ca), Magnesium (Mg), Barium (Ba), Boron (B)
Diagnostic markers	Particle Quantifier (PQ index), Total Base Number (TBN, when available)
Maintenance context	Operating hours, production since last oil change, date of last oil change
Laboratory comments	Narrative assessment of oil state and recommendations

Given the diversity of reported metrics and their different scales, it was necessary to develop a synthetic index that could capture the overall state of the oil in a way that would allow comparison across turbines and over time. For this purpose, an "OilScore" was defined. The OilScore was designed to summarize five key "stress" components derived from the lab reports.

These were:

- **Wear:** Fe concentration (mg/kg) and PQ index
- **Contamination:** Water content (ppm)
- **Ageing:** Oxidation index
- **Condition:** percentage deviation of viscosity at 40 °C from the nominal ISO VG 320 grade, calculated as |Viscosity_40C – 320| / 320 × 100

Each of these components was standardized using a z-score:

$$z_X = \frac{X - \mu_X}{\sigma_X}$$

where μx and σx are the mean and standard deviation of that component across the dataset. This transformation placed all metrics on a common, dimensionless scale. A positive z-score

indicates that the measurement was above the fleet average (and therefore worse, given that higher wear, contamination, or oxidation values are undesirable), whereas a negative z-score indicates a condition better than the average. The OilScore for each sample was then calculated as the mean of the available z-scores, ignoring missing values:

$$\text{OilScore} = \text{mean}\,(z_{Fe}, z_{PQ}, z_{Water}, z_{Oxid}, z_{ViscDev})$$

Higher values correspond to worse oil condition (more wear, more contamination, higher oxidation, or larger viscosity deviation), while negative values indicate samples that were healthier than the fleet average. Around zero is considered "typical."

A few clarifications help interpret the score. First, viscosity was not used in its raw form; instead, the deviation from the nominal 320 cSt grade was taken, so being close to 320 counts as healthy. Second, when a component was missing from a report, the OilScore was calculated as the mean of the remaining z-scores, which ensured that all samples could still be included. Third, because sampling frequency was not uniform, with sometimes multiple samples per month for a turbine, the monthly OilScore was defined conservatively as the maximum of that month's values. At the farm level, the monthly OilScore was taken as the median across all turbines, which balances outliers while still reflecting the central tendency of the fleet.

Because samples were not taken continuously but sporadically, usually a few times per year per turbine, an aggregation step was required to align them with monthly performance data. In parallel to the oil dataset, a second dataset had been prepared containing monthly records of actual versus theoretical energy production together with average meteorological conditions, specifically temperature and relative humidity. From this, a monthly loss percentage was derived by comparing observed and expected energy production, and each month was further annotated according to whether it fell into a "hot," "humid," "hot+humid," or "normal" category. These categories were defined by taking the upper quartile thresholds of the six-year temperature and humidity distributions, ensuring that only the most extreme 25% of values were classified as hot or humid months.

The two data sources were then merged by month and turbine, yielding a combined view that links internal oil condition, external weather stressors, and energy performance. In addition, known gearbox replacement events were encoded directly into the workflow so that their timing could be highlighted in subsequent visualizations. This complete dataset made it possible to investigate whether degraded lubricant condition acted as a mediator between extreme weather and production losses.

To make this workflow clearer, the process can be summarized as follows:

1. Raw PDF oil/grease analysis reports (≈290 documents, 2019–2024)

2. Automated parsing with Python script (extraction of metals, contamination, condition metrics, turbine ID)
3. Consolidated structured dataset (samples × variables)
4. Computation of OilScore (z-score normalization of Fe, PQ, Water, Oxidation, Viscosity deviation; averaged into single index)
5. Monthly aggregation by turbine and fleet
6. Merge with production losses and weather dataset (monthly energy loss %, average temperature and humidity; hot/humid bins)
7. Integration of gearbox replacement events (hard-coded dates)
8. Graphical analysis (scatterplots of Loss vs OilScore, time-series with replacement markers, fleet view small multiples).

The pipeline begins with the raw PDF oil and grease analysis reports and proceeds through the parsing step, the construction of a structured dataset, the calculation of OilScore, and its monthly aggregation by turbine and by farm. This is then merged with the weather and energy loss dataset, enriched with gearbox replacement markers, and ultimately transformed into graphical outputs. The graphical analysis took two complementary forms: time-series plots of OilScore, in which red dashed lines indicated the timing of gearbox replacements, and scatterplots of monthly energy loss against OilScore, where points were colored according to the weather bin of the corresponding month. These graphs do not always reveal a neat linear relationship, which is expected given the complexity of turbine operation and the limited sampling frequency, but they provide a qualitative and intuitive means of exploring whether higher OilScores, particularly during hot or humid months, coincide with elevated losses or precede major failures.

4

RESULTS AND DISCUSSION

This chapter presents the results of the analytical methods applied to investigate the relationship between weather conditions and gearbox failures in the wind farm dataset. Four complementary approaches were employed, each addressing the problem from a different perspective. First, supervised labelling was used to categorize weather variables into interpretable groups, providing a structured basis for subsequent analyses. Second, k-means clustering was applied as an unsupervised method to detect natural groupings in the data without relying on predefined labels, offering insights into recurring weather patterns. Building on these foundations, two machine learning techniques were used to explore the conditions specifically preceding gearbox replacements. The Decision Tree analysis sought to identify thresholds and combinations of variables that best separated failure and non-failure periods, producing interpretable rules in plain language. Finally, Association Rule Mining was employed to uncover frequent co-occurrences of weather conditions disproportionately present before failures. Taken together, these methods provide both descriptive and exploratory perspectives, allowing the discussion to address not only the patterns directly linked to gearbox replacements but also the broader environmental contexts in which they occur.

4.1 SUPERVISED LABELLING

An analysis of failure frequencies across relative humidity categories reveals distinct vulnerabilities among turbine components (see Figure 7) Blade failures are notably concentrated under *corrosive* (60–80% RH) and *high corrosive* (>80% RH) conditions, peaking in the former category.

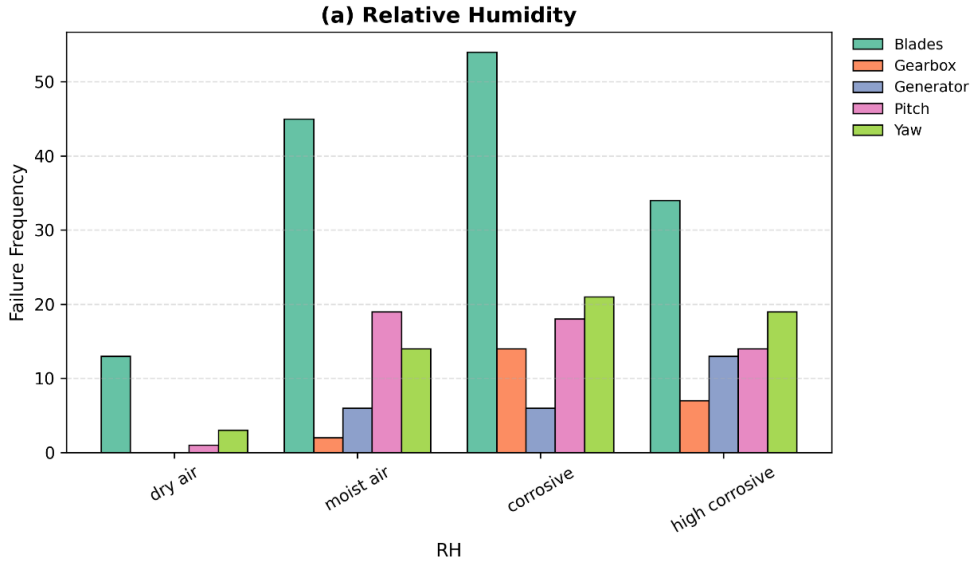


Figure 7. Visual representation of supervised labelling analysis. Number of failures in different conditions of Relative Humidity

This pattern suggests moisture-driven degradation, such as surface erosion or internal condensation. Similarly, failures of pitch and yaw systems increase markedly with elevated humidity, likely reflecting corrosion of electromechanical parts like bearings or sensor housings. In contrast, gearbox failures show a more even distribution across humidity levels, with only a moderate rise under harsher humidity conditions—potentially indicating secondary effects such as lubricant breakdown rather than direct exposure. All components display minimal failure rates in *dry air*, reinforcing the premise that humid environments exacerbate failure risk. These findings are consistent with (Pelka & Fischer, 2023) , who, in a statistically representative field-data based study, identified humidity as a significant predictor of failure in turbine power converters, often surpassing wind speed in explanatory power. Furthermore, (Fischer et al., 2021) demonstrated that ambient and internal converter humidity exhibit strong seasonal patterns closely aligned with elevated failure incidents across diverse turbine fleets. Collectively, these results validate the integration of relative humidity as a key variable in weather-driven failure models and maintenance forecasting.

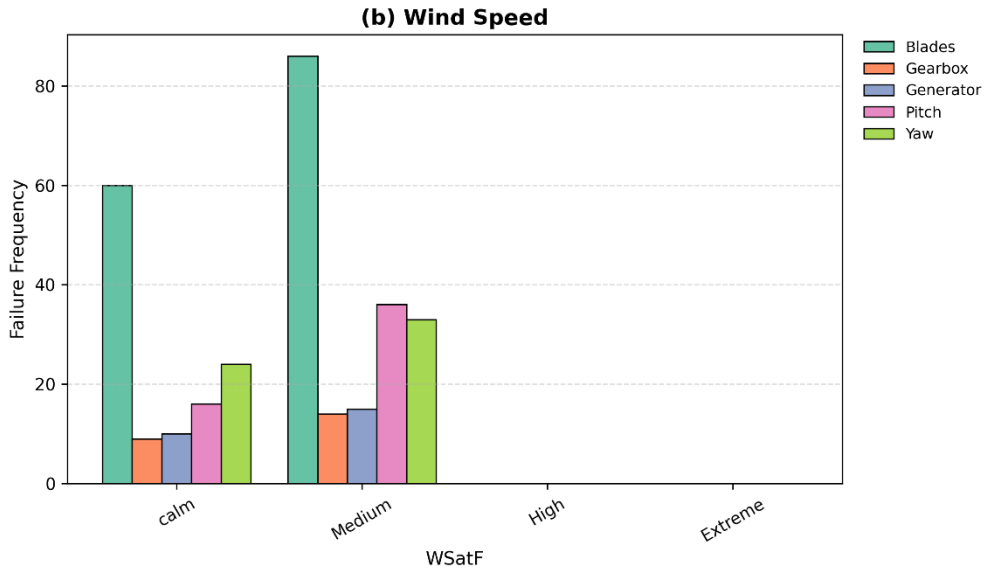


Figure 8. Visual representation of supervised labelling analysis. Number of failures in different conditions of Wind speed

The results show that the majority of failures occurred during calm and medium wind conditions, while no failures were recorded in the high or extreme categories. Among components, blade failures were by far the most frequent, peaking in the medium wind category with more than 80 recorded events, and also showing substantial numbers under calm winds. Other components such as gearboxes, generators, pitch, and yaw systems exhibited lower overall failure frequencies, though they followed a similar trend of being more common under calm to medium wind conditions. Notably, pitch and yaw failures increased in the medium category compared to calm conditions, suggesting that these subsystems may be particularly sensitive to moderate operational loading. The absence of failures during extreme winds indicates either that turbines were shut down before reaching such conditions or that extreme events were too infrequent within the dataset to leave a measurable impact. Overall, the figure highlights that failures are not confined to periods of severe wind loading but can occur predominantly under normal operating conditions.

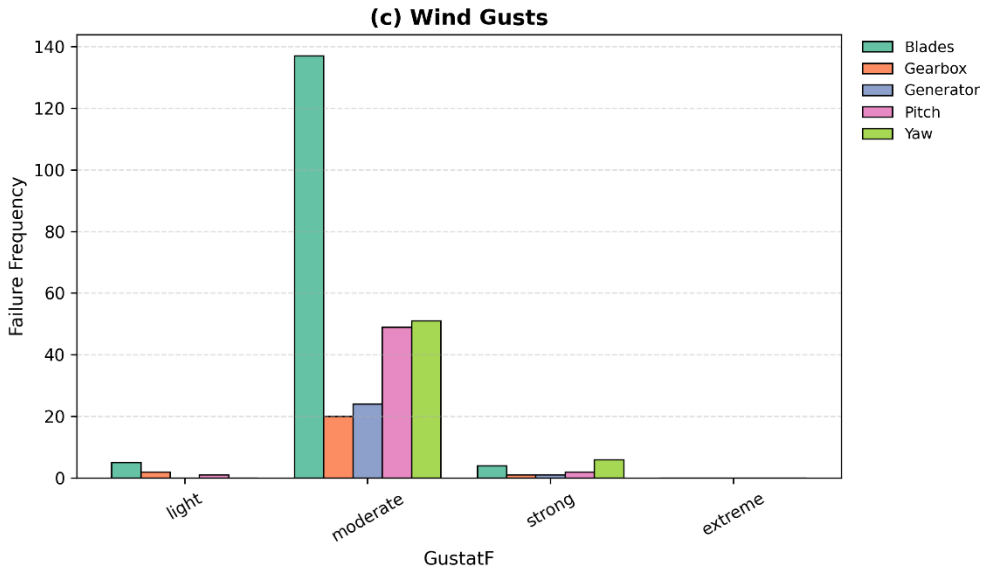


Figure 9. Visual representation of supervised labelling analysis. Number of failures with different ranges of wind peak gusts

The wind gust chart (Figure 9) shows that the majority of failures across all components occurred under *moderate* gust conditions (15–25 m/s), with the most pronounced peak observed in blade failures. This suggests that repeated exposure to moderate gusts may contribute more to long-term fatigue than short-duration extreme gusts. Pitch and yaw systems also show significantly elevated failure counts under moderate gusts, which may be due to the frequent movement and adjustments the turbine makes when wind directions and speeds change rapidly. These patterns highlight the role of operational turbulence and cycling rather than peak gust intensity as a driver of wear and failure.

Notably, *strong* (25–35 m/s) and *extreme* (>35 m/s) gusts are associated with very few failures across all components. This can likely be explained by automatic shutdown procedures that deactivate turbine operation once gusts exceed safety thresholds, thereby protecting the system from damage. This result aligns with observations by (Hahn et al., 2007) , who showed that moderate but frequent gusts result in more cumulative damage than rare extremes, particularly in blades and pitch systems.

The data suggests that moderate gust conditions—common during storm build-up or shifting fronts—may be more critical for reliability planning and maintenance forecasting than previously assumed. These insights support the inclusion of gust intensity and variability as key indicators in weather-related failure risk models.

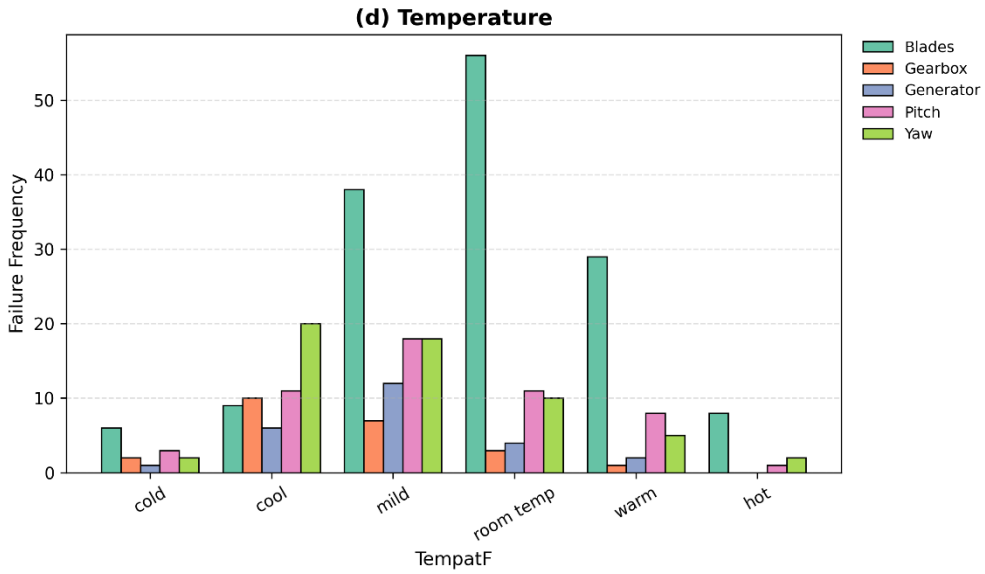


Figure 10: Visual representation of supervised labelling analysis. Number of failures in different Temperatures.

The temperature chart (Figure 10) shows blade failures most frequently occurred under *room temperature* (20–25 °C) conditions, with notable counts also at *mild* (15–20 °C) and *warm* (25–30 °C) categories. Unexpectedly, failure occurrence does not increase in *cold* (<10 °C) or *hot* (>30 °C) ranges where thermal stress is typically presumed to be more critical. A plausible explanation is that room temperatures often coincide with spring and autumn, seasons characterized by higher wind turbulence and production activity—leading to greater cyclical stress on blades even under moderate temperatures. This matches observations in (Antoniou et al., 2020) , who demonstrated that thermal residual stresses and operational loading during nominal temperature ranges can substantially accelerate fatigue damage in blade trailing.

For other components—gearbox, generator, pitch, and yaw—the distribution of failures across temperature classes is comparatively uniform. Few failures occur at *cold* or *hot* extremes, possibly due to operational curtailment during extreme temperatures or the limited frequency of such conditions in Lousã, Portugal. These results suggest that ambient temperature alone is not a primary failure driver, but it may influence failure risk when combined with other stressors, such as wind loading and humidity.

4.2 UNSUPERVISED K-MEANS CLUSTERING

The clustering output for the Gearbox component (Figure 11) shows how failure-related weather conditions are grouped into two distinct clusters using K-Means.

Figure 11. Output visualization of the k-means clustering analysis for Gearbox failures

Temperature (temp): Cluster 0 (green) is concentrated around moderate temperatures (approximately 10–22 °C), while Cluster 1 (orange) includes a slightly broader range, with some events occurring at lower or higher temperatures. Both clusters show overlap around 15–20 °C.

Relative Humidity (rhum): Cluster 0 shows a clear tendency toward higher humidity levels (75–90%), while Cluster 1 includes events with lower humidity, some below 60%. This suggests that gearbox failures may be more frequent in humid conditions, which is consistent with known issues of corrosion and lubricant degradation in wind turbine gearboxes, especially under moist environmental exposure (Tavner, 2012).

Wind Speed (wspd) and Gust (wpgt): Cluster 0 is associated with moderate wind speeds (3–6 m/s) and gusts (5–12 m/s), whereas Cluster 1 occurs in calmer conditions, with wind speeds below 3 m/s and gusts below 7 m/s. This difference may reflect fatigue-related stress on gearboxes, as repeated loading in moderate wind conditions has been shown to accelerate wear and reduce gear life over time (Qiu et al., 2017).

These two clusters indicate distinct weather-related stress profiles associated with gearbox failures:

- Cluster 0 likely reflects failure events influenced by high humidity and moderate wind, pointing to gradual degradation such as seal wear, moisture ingress, and lubricant breakdown.
- Cluster 1 likely includes failures less affected by weather, potentially due to internal mechanical faults or age-related issues.

Together, these observations support the broader understanding that gearbox reliability is influenced not only by mechanical design but also by environmental factors, particularly humidity and sustained operational loading (Tavner, 2012; Qiu et al., 2017).

The generator clustering output (Figure 12) shows two distinct weather-related clusters among the failure events.

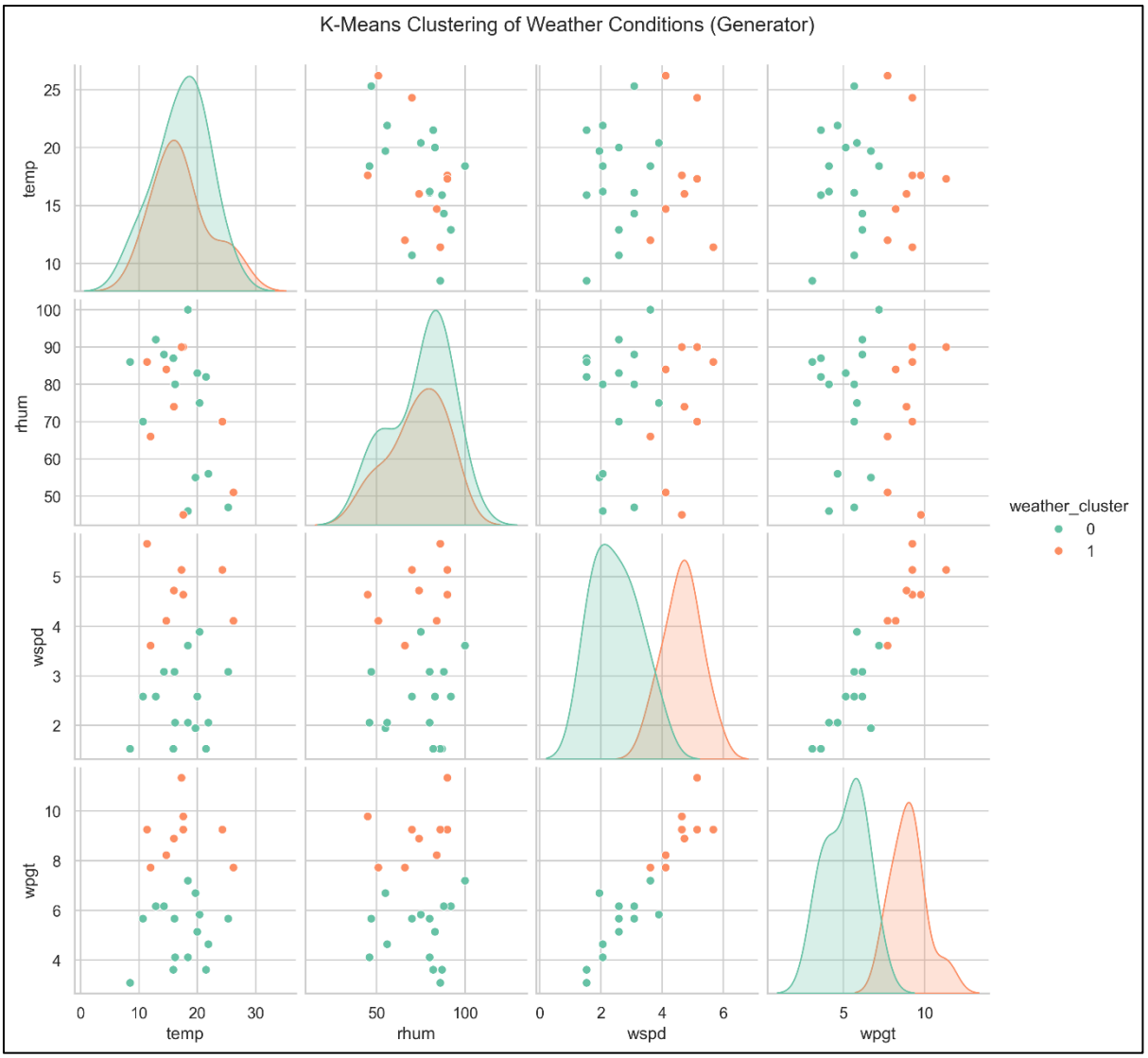


Figure 12. Output visualization of the k-means clustering analysis for Generator failures

Temperature (temp): Both clusters occupy a similar moderate range (~15–22 °C) with substantial overlap. This suggests that ambient temperature alone is not a significant differentiator for generator failures.

Relative Humidity (rhum): There is some separation—Cluster 0 tends towards higher humidity levels (>75%), whereas Cluster 1 spreads across lower values too (50–70 %). Elevated humidity may contribute to early electrical component degradation, such as insulation breakdown or corrosion in generators (Tavner, 2012).

Wind Speed (wspd) and Gust (wpgt): Notably, Cluster 0 corresponds to lower wind speeds and gusts (wspd: ~2–4 m/s; wpgt: ~4–7 m/s), while Cluster 1 trends toward higher loading conditions (wspd up to 6 m/s; gpgt up to 10 m/s). These differences suggest that higher wind-related stress might play a role in generator failure risk, particularly through mechanical and thermal loading in persistent operational states. This aligns with evidence that generator component fatigue correlates with extended periods of high loading derived from SCADA data (Qiu et al., 2017). While the clustering shows modest separation, the patterns point toward two probable failure regimes for generators:

- Cluster 0: Higher humidity and lower wind/gust exposure—failure might be driven by environmental moisture stress, affecting electrical insulation or bearings.
- Cluster 1: Lower humidity but higher wind-related loading—failure may be driven by thermal and fatigue stress from sustained power generation.

These insights support the broader thesis that generator reliability is influenced by both humidity and loading conditions.

The clustering output for the Blades component (Figure 13) shows three distinct clusters of weather conditions, each representing a unique regime observed around blade failure events.

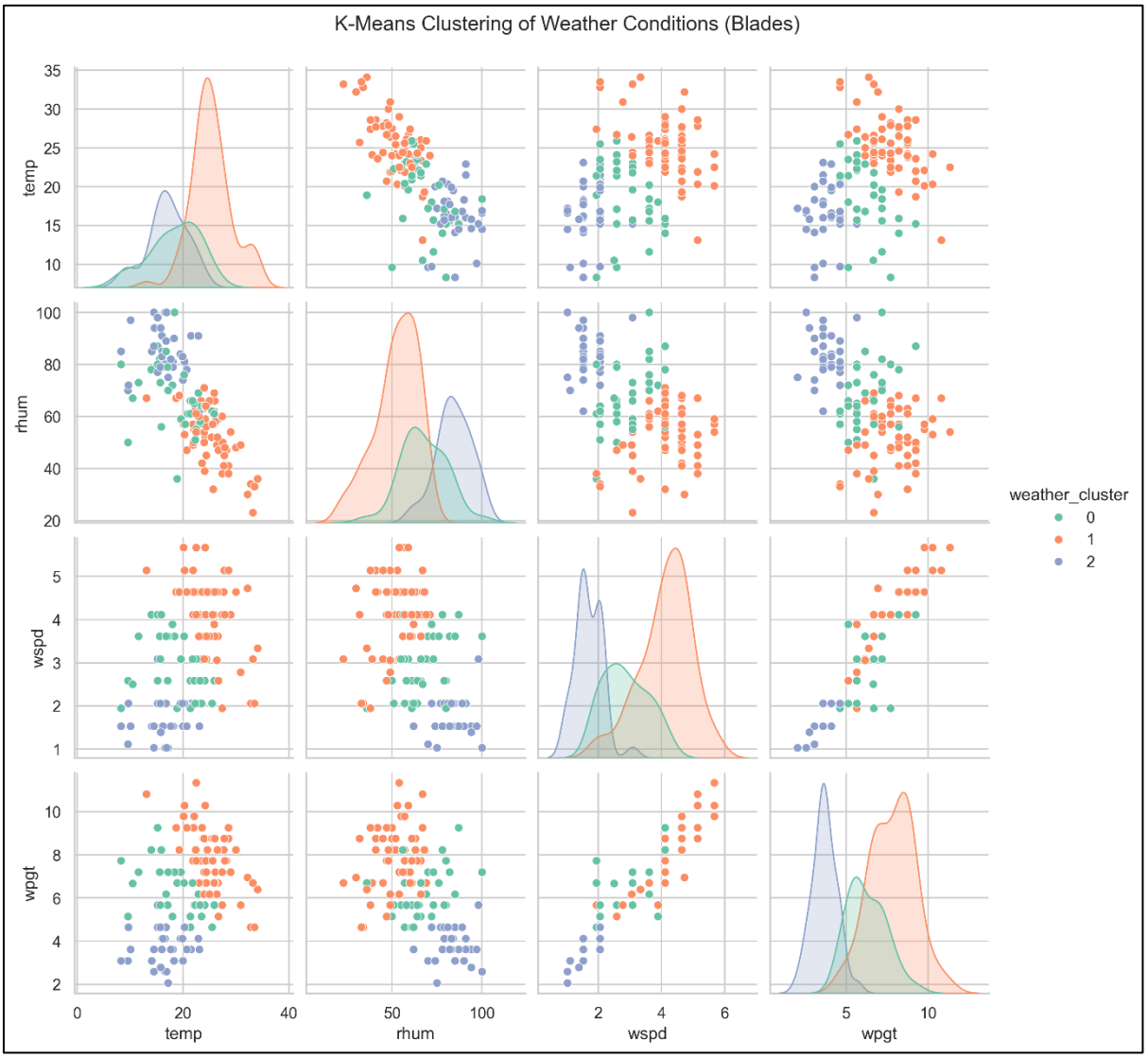


Figure 13. Output visualization of the k-means clustering analysis for Blades failures

Temperature (temp): Cluster 1 (orange) dominates the higher temperature range, centering around 20–30 °C, while Cluster 2 (blue) is associated with cooler conditions, mostly below 20 °C. Cluster 0 (green) spans both mid and low temperatures, suggesting a transitional weather profile.

Relative Humidity (rhum): Cluster 2 (blue) is clearly associated with higher humidity values (>80%), while Cluster 1 (orange) has a much wider distribution, spanning 40–85%. This separation suggests that humidity is a meaningful factor in blade failure environments. High humidity can promote moisture ingress, leading to internal delamination or bonding

issues in composite materials—a known failure mechanism in wind turbine blades (Brøndsted et al., 2005).

Wind Speed (wspd) and Gust (wpgt): Cluster 1 shows the highest wind speeds and gusts, typically between 4–6 m/s and 7–10 m/s respectively. Cluster 2 again dominates the lower end of both variables, while Cluster 0 occupies an intermediate zone. The separation of Cluster 1 highlights that blade failures often occur during periods of moderate to high aerodynamic loading, consistent with fatigue stress mechanisms (Ciang et al., 2008).

These clusters represent three major regimes of weather exposure associated with blade failures:

- Cluster 1: *Warm, turbulent, and moderately humid conditions* – likely related to fatigue-driven failure due to repetitive stress from wind loading.
- Cluster 2: *Cool and very humid conditions* – may correspond to moisture-related degradation, such as internal delamination or erosion from condensation and icing cycles.
- Cluster 0: *Transitional or mixed conditions*, with no dominant extreme, possibly representing generalized wear or degradation from standard operation.

These patterns support the understanding that blade failures are influenced by both mechanical stress (wind loading) and environmental stress (moisture, humidity), reinforcing findings from previous research that blade reliability is highly sensitive to weather conditions, especially for offshore and high-humidity inland environments.

The clustering results for the Yaw system (Figure 14) reveal three distinct weather condition groups at the time of yaw-related failure events. The yaw system plays a critical role in wind turbine orientation and is subject to both mechanical wear and weather-induced operational stress.

Figure 14. Output visualization of the k-means clustering analysis for Yaw failures

Temperature (temp): Cluster 2 (blue) contains the majority of failure events, centered tightly around moderate temperatures (~15–25 °C). Cluster 0 (green) includes slightly lower temperature events, while Cluster 1 (orange) spans a broader but smaller set of entries. This suggests that temperature alone is not the strongest discriminating variable across failures.

Relative Humidity (rhum): Humidity ranges are relatively broad across all clusters, but Cluster 2 is slightly skewed toward higher humidity (>70%), while Cluster 1 includes events

occurring in drier conditions (~40–60%). This may indicate some sensitivity of the yaw mechanism to corrosive environments, particularly in systems with insufficient sealing or lubrication.

Wind Speed (wspd) and Gust (wpgt): A strong separation is observed here. Cluster 2 consistently exhibits higher wind speeds and gusts (wspd: 3–6 m/s; wpgt: 7–12 m/s), whereas Clusters 0 and 1 correspond to lower aerodynamic loading conditions. Since the yaw system is responsible for adjusting the nacelle orientation during wind shifts, this pattern supports the idea that yaw-related failures are more common during dynamic wind conditions, requiring frequent actuation and motor-driven adjustments. High yaw activity has been associated with increased mechanical wear, motor overheating, and brake system fatigue (Wilkinson, M. et al., 2007)

The clusters can be interpreted as follows:

- Cluster 2 likely reflects failures under operational stress, where the yaw system is adjusting frequently due to moderate–high wind and gust conditions.
- Clusters 0 and 1 appear more related to environmental degradation, especially in lower wind and high-humidity scenarios that can accelerate bearing wear or corrosion.

This supports existing research suggesting that yaw system reliability is sensitive to both cumulative operational use and environmental exposure, especially in turbulent inland locations (Tavner, 2012).

The clustering output for the Pitch system (Figure 15) shows three distinct clusters of weather conditions associated with pitch-related failure events. The pitch system is responsible for adjusting the angle of the blades to regulate aerodynamic loads, optimize energy capture, and protect the turbine during high wind events. As a result, it operates under both environmental exposure and mechanical stress

Figure 15. Output visualization of the k-means clustering analysis for Pitch failures

Temperature (temp): Cluster 0 (green) contains the majority of failure events at higher ambient temperatures (20–30 °C), while Clusters 1 (orange) and 2 (blue) span cooler conditions. Though all three clusters overlap in the 15–25 °C range, the concentration of failures in warm and humid conditions may indicate elevated thermal or electrical stress on pitch control electronics and actuators.

Relative Humidity (rhum): Cluster 0 also aligns with higher humidity values, often above 75%. This suggests that moisture ingress or condensation could be contributing to control or actuator failures—an effect that has been linked to degradation in sealing systems and controller enclosures (Reder & Melero, 2017)

Wind Speed (wspd) and Gust (wpgt): Cluster 2 shows higher wind speeds and gust levels, while Cluster 1 occurs during quieter wind periods. These patterns likely correspond to different stress modes: failures under high gusts are often due to fatigue from rapid and repeated pitch adjustments, while failures under calm conditions may stem from control lock-up or mechanical wear during standby periods (González-González et al., 2018) .

The three clusters suggest multiple pathways for pitch system failure:

- Cluster 0: High temperature and humidity – may lead to thermal degradation or electronic control issues.
- Cluster 1: Calm and dry – may reflect stagnant operating modes that cause internal wear or sensor drift.
- Cluster 2: Windy and gusty – associated with active aerodynamic regulation, possibly resulting in actuator fatigue or sensor errors from rapid pitch cycles.

These observations are consistent with previous findings that weather conditions such as low temperatures, strong wind, humidity, and turbulence contribute to pitch system faults (Reder & Melero, 2022), and that low wind availability can induce mechanical risks in pitch controllers (González-González et al., 2018).

4.3 DECISION TREE ANALYSIS

The Decision Tree model was trained on the labelled dataset of pre-failure and control windows. Its performance on the test set achieved an overall accuracy of 61.5%, which, while modest, provides insight into the patterns of separation achieved. The model demonstrated very high sensitivity to failure cases. All failure windows were correctly identified (recall = 1.00), meaning the tree did not miss any gearbox replacements. However, this came at the expense of specificity, as many control windows were misclassified as failures, leading to a lower recall for normal operation (0.545). This imbalance reflects a common trade-off in small, imbalanced datasets, where the model tends to prioritise the identification of rare events (failures) at the cost of introducing false positives.

The tree constructed a set of *if–then* rules that describe the most relevant decision boundaries. The maximum peak gust over the medium-term (30-day) window

(med_wpgt_max) was identified as the most influential variable, with subsequent splits refined by maximum temperature and mean relative humidity. The principal rules were as follows:

1. Low gusts + moderate temperature
    - *If med_wpgt_max ≤ 13.36 m/s and med_temp_max ≤ 31.05 °C → one failure case, no controls.*
    - This rule coincided with one pre-failure window, representing a potentially relevant condition.
2. Low gusts + high temperature
    - *If med_wpgt_max ≤ 13.36 m/s and med_temp_max > 31.05 °C → no cases present.*
    - This was a theoretical branch with no matching observations.
3. Higher gusts + lower humidity
    - *If med_wpgt_max > 13.36 m/s and med_rhum_mean ≤ 82.35% → almost exclusively controls.*
    - This describes operating conditions typical of normal, non-failure periods.
4. Higher gusts + higher humidity
    - *If med_wpgt_max > 13.36 m/s and med_rhum_mean > 82.35% → no cases present.*

Only the first rule aligned with a gearbox replacement, while the other branches primarily represented conditions under which turbines continued to operate without incident.

The threshold values identified by the tree, such as 13.36 m/s for medium-term gusts, should not be interpreted as physical or engineering limits. These values are statistical cut-offs, chosen automatically by the algorithm as the points that best separated failures from controls in this dataset.

The algorithm tests all possible split points within the observed data and selects the one that maximises classification accuracy at that step. The specific value is therefore data-dependent and could shift if additional observations were included. For example, a new control period with gusts around 13.5 m/s might cause the threshold to move closer to that value. The correct interpretation is therefore relative: gearbox failures in this dataset were more likely to occur during periods of lower medium-term gusts compared to non-failure periods, rather than at exactly 13.36 m/s.

The analysis suggests that gearbox failures were not consistently preceded by extreme weather events. Instead, one observed pattern indicated that failures sometimes occurred under relatively calm wind conditions combined with moderate temperatures. By contrast, periods

characterised by higher gusts and moderate humidity were generally associated with continued, non-failure operation. These results are informative but should be treated with caution. The limited sample size, combined with the multifactorial nature of gearbox failures (which depend on wear, lubrication, design, and maintenance in addition to weather), means that the identified rules are exploratory. They provide potential insights into environmental contexts that may influence failures, but they do not establish causal relationships.

The Decision Tree approach did not reveal a single dominant combination of weather parameters that preceded all failures. One observed pattern involved low medium-term peak gusts (≤ 13.36 m/s) combined with moderate maximum temperatures (≤ 31 °C), which coincided with one gearbox replacement but not with any control periods. Given the limited sample size, these patterns are hypothesis-generating rather than definitive.

4.4 ASSOCIATION RULE MINING

The ARM analysis identified recurring combinations of weather conditions that were disproportionately associated with gearbox failures. Unlike the Decision Tree, which produced a single branching structure, ARM generated a list of rules ranked by support, confidence, and lift. The most notable rules included:

1. Low peak gusts (wpgt_max=Low) were linked to failures, with a lift of 1.43. This indicates that gearbox replacements were more likely to occur when short-term maximum gusts were unusually low (≤ 11.30 m/s).
2. High humidity combined with low gusts (rhum_max=High AND wpgt_max=Low) was also associated with failures, again with a lift of 1.43. This suggests that calm and humid conditions sometimes co-occurred before replacement events.
3. Low mean wind speeds (wspd_mean=Low) appeared in several rules, pointing to a broader tendency for failures to coincide with relatively calm operating conditions rather than high wind loading.

Across multiple rules, the recurring theme was that failures were not consistently preceded by extreme weather events. Instead, they tended to appear in periods characterized by low wind speeds, weak gusts, and high humidity. These patterns are not predictive formulas but statistical associations that highlight possible environmental contexts worth investigating further. Given the limited sample size, they should be considered exploratory findings that generate hypotheses for future studies.

4.5 ENERGY LOSS AGAINST WEATHER CONDITIONS

This analysis produced a large number of graphical outputs, including fleet-level panels, comparative scatter plots, and individual turbine figures. While each plot contributes to a detailed picture of turbine performance and weather influences, it would not be practical to present them all in this section. For the sake of clarity, only the most noticeable and relevant results are discussed here, with a focus on the patterns that are most consistent across the fleet or that stand out in connection with specific turbines and interventions. These selected graphs highlight the key findings of the study, while the full set of outputs remains available in the appendix and supplementary material.

Figure 16 below presents the relationship between monthly energy loss and mean temperature for all twenty turbines in the farm. Each subplot corresponds to one turbine, covering the period 2019–2024, with marker size proportional to the number of valid hourly records contributing to each monthly point. Turbines that experienced gearbox replacements have their data color-coded into pre- and post-intervention months. A simple linear trend line is fitted to each turbine for clarity.

Across the fleet, a clear and consistent pattern emerges: energy losses tend to increase with higher mean monthly temperatures. The slope of the regression lines is positive for nearly all turbines, indicating that warmer months are systematically associated with greater deviations from the theoretical power curve. This confirms the expectation that environmental factors linked to temperature, such as lower air density or thermal derating of components, play a significant role in shaping performance. The effect is not subtle: in many turbines the difference between cooler months (10–12 °C) and warmer months (20–22 °C) corresponds to a change of 5–10 percentage points in loss, which is operationally meaningful at the scale of monthly production.

While the overall fleet shows a coherent response to temperature, the magnitude and clarity of the pattern vary by turbine. Machines such as WT01, WT02, WT05, and WT17 exhibit very regular scatter clouds with a strong upward slope, suggesting that their performance is particularly sensitive to seasonal heating. In contrast, some turbines such as WT09, WT14, and WT20 show more dispersed points, with the upward trend still visible but less sharply defined. This variation may reflect differences in turbulence exposure, yaw alignment, or other localized factors that amplify or obscure the role of temperature.

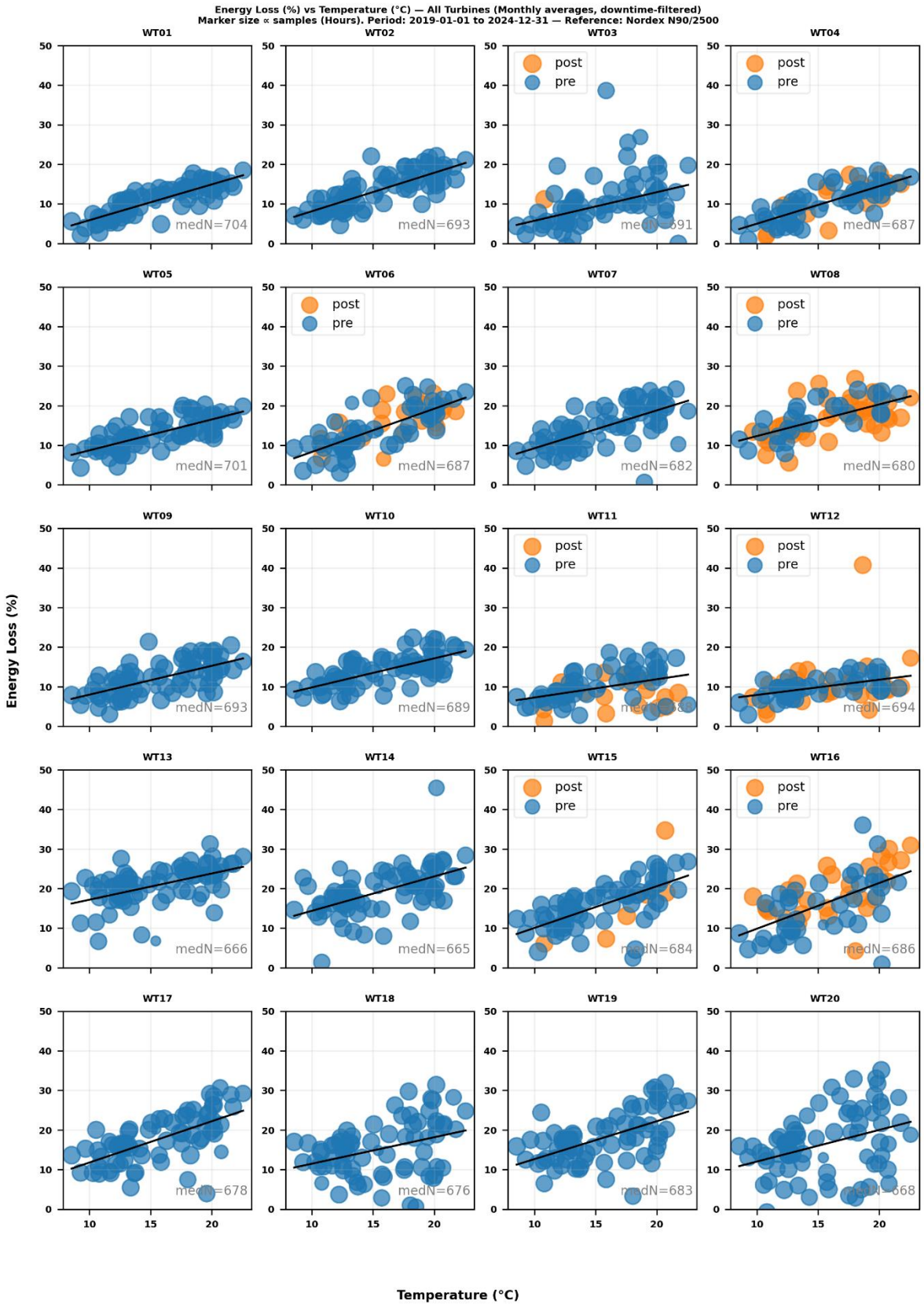


Figure 16. Relationship between temperature and energy losses (%) across all turbines (2019–2024), with monthly averages and downtime filtering applied.

Regarding the gearbox replacements, the turbines that underwent gearbox replacements (WT03, WT04, WT06, WT08, WT11, WT12, WT15, and WT16) do not show a simple pre/post split in their temperature–loss behavior. The pre- and post-intervention points (distinguished in orange and blue) often overlap, and the fitted trend remains positive on both sides of the intervention. This suggests that while gearbox replacements are essential for restoring mechanical reliability, they do not fundamentally alter the underlying temperature sensitivity of the turbine. In other words, the thermal and aerodynamic effects driving seasonal losses are broader phenomena that affect all machines, regardless of intervention history.

Another point worth noting is the spread of points around the regression lines. Even though temperature is a dominant driver, it is not the only one. For example, at a given mean temperature of 15 °C, loss values may range from 5% to 20% depending on the turbine and month. This indicates that additional weather variables, such as humidity or turbulence intensity, and operational factors also modulate the outcome. Nevertheless, the strength and consistency of the temperature effect across the fleet makes it the single most noticeable signal in the dataset.

Taken together, this figure demonstrates two key findings. First, the farm as a whole experiences greater energy losses in warmer months, with a broadly consistent relationship across turbines. Second, while interventions such as gearbox replacements are important from a maintenance perspective, they do not erase the seasonal imprint of temperature on performance. This reinforces the idea that temperature-driven energy losses should be considered a systematic effect, rather than an anomaly of specific turbines.

Figures 17 and 18 below show the monthly evolution of energy loss across the fleet from 2019 to 2024. Each subplot corresponds to one turbine, with vertical dashed red lines marking the timing of gearbox replacements. The y-axis is fixed at 0–40% to enable direct comparison across machines, while the median number of valid hourly samples per month (medN) is annotated for transparency.

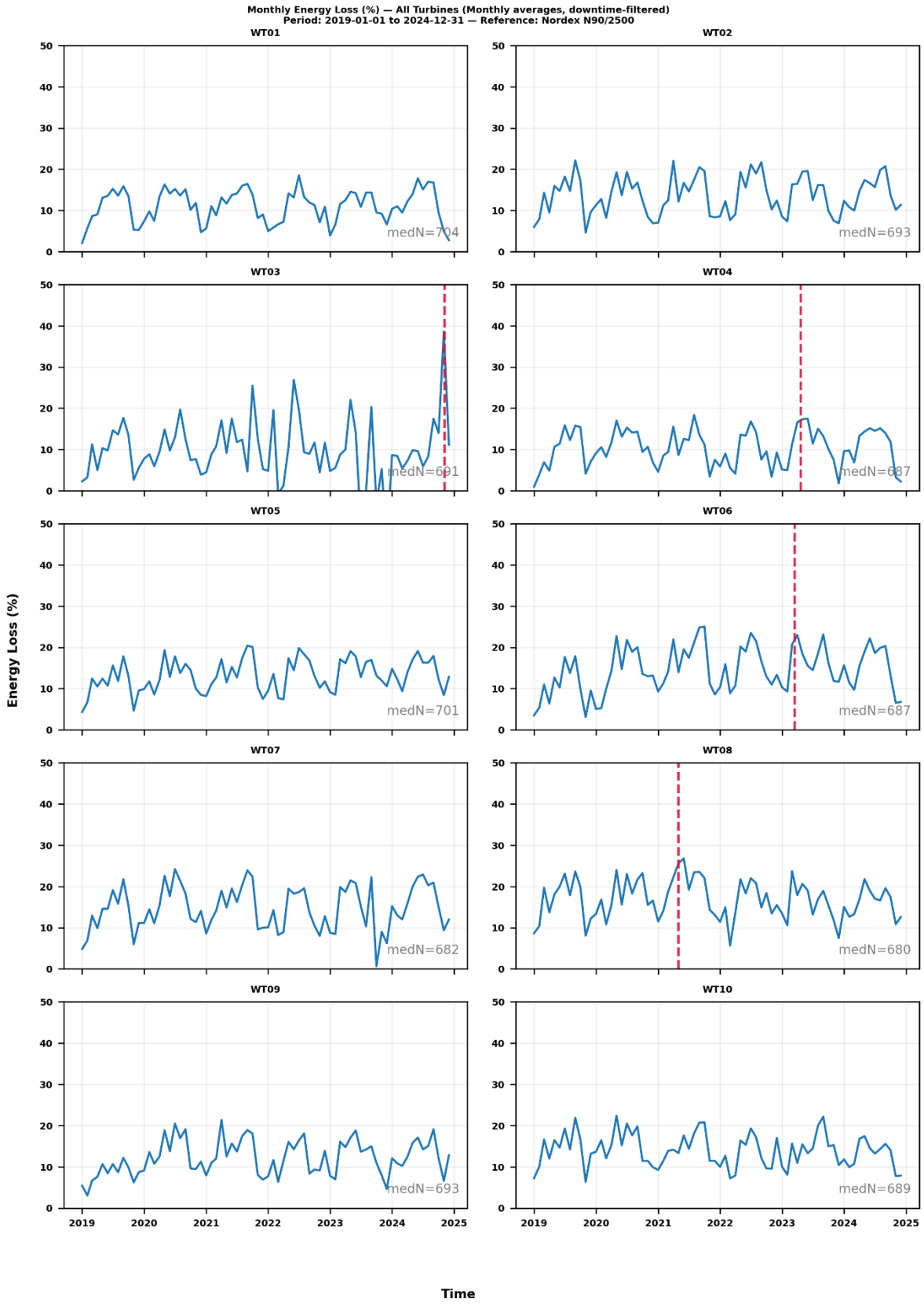


Figure 17. Monthly energy loss (%) for WT 01 – WT 10 (2019–2024), with downtime filtering applied. Red dashed lines indicate gearbox replacement events.

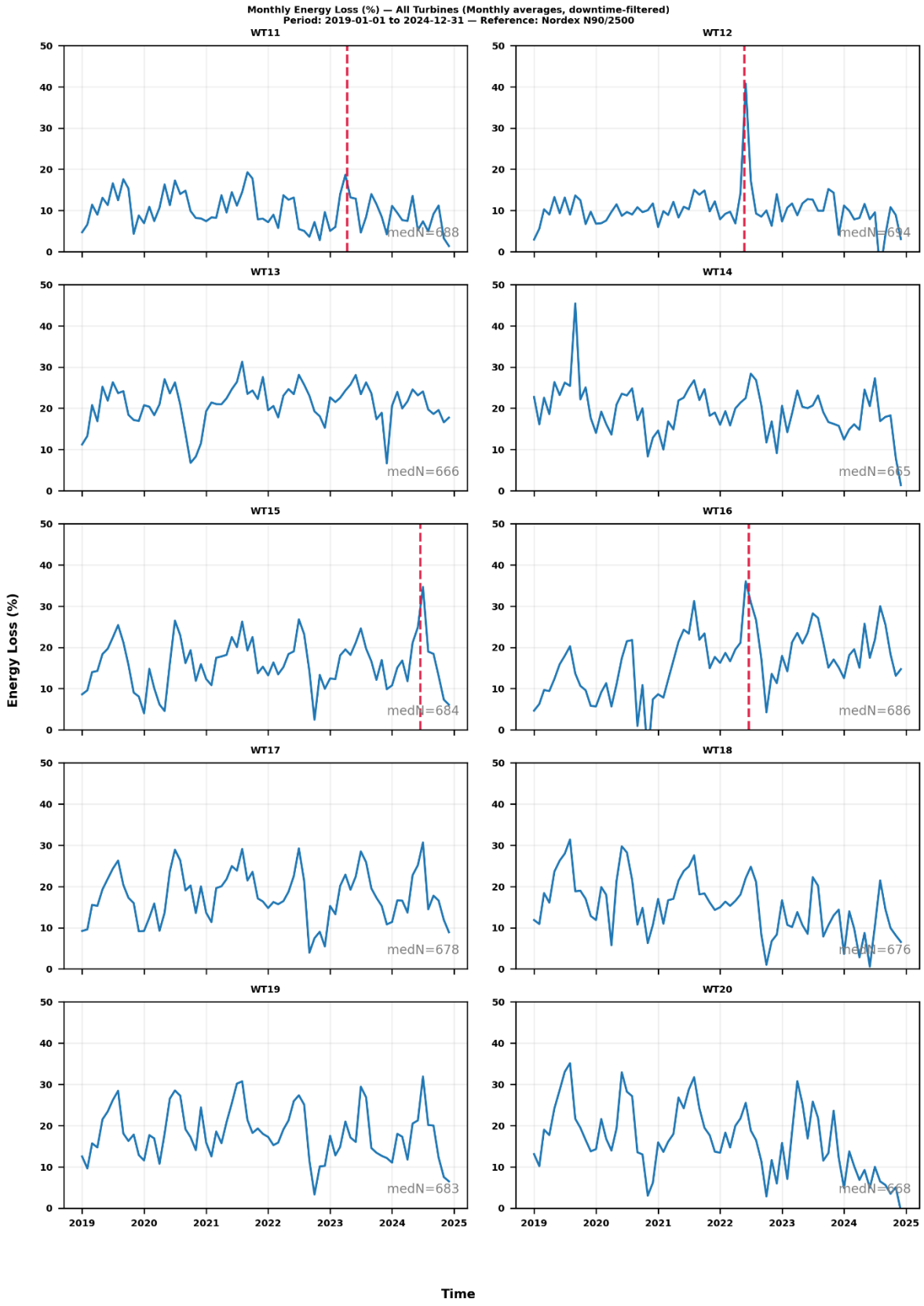


Figure 18. Monthly energy loss (%) for WT 11 – WT 20 (2019–2024), with downtime filtering applied. Red dashed lines indicate gearbox replacement events.

A strong seasonal rhythm is evident across almost all turbines: losses rise in the summer months and decline during the cooler winter months, reflecting the temperature effect already highlighted in the scatter plots. What makes this figure particularly valuable, however, is the way it reveals the interaction between these seasonal swings and the timing of major mechanical interventions.

In several turbines (notably WT03, WT08, WT12, WT16, and WT15), gearbox replacements were carried out immediately after or during a period of unusually high losses. For instance, WT16 exhibits a pronounced spike in early 2022, with monthly losses briefly exceeding 30%, followed by a gearbox replacement. A similar pattern is observed in WT12 in mid-2022, where a sudden jump in losses precedes the intervention. WT03 and WT15 also display extreme peaks shortly before their respective replacements, standing well above the background seasonal variability. These coincidences are unlikely to be accidental: they suggest that rising energy losses served as an operational signal of severe mechanical degradation, ultimately triggering the decision to replace the gearbox.

From an engineering perspective, this makes sense. Gearbox wear leads to reduced mechanical efficiency, increased vibration, and thermal stress, all of which can lower the effective energy conversion from rotor to generator. As the degradation worsens, the turbine struggles to maintain its theoretical output, leading to sustained or spiking loss values. Once these losses cross a threshold that is both technically concerning and financially significant, operators are more likely to schedule major corrective maintenance. The observed alignment between loss peaks and gearbox replacements therefore validates the energy loss metric as not just a weather-related indicator, but also as a diagnostic proxy for mechanical health.

It is also instructive to note that the replacements generally occurred after peaks rather than before them. In other words, operators appear to react to the problem once it is already visible in performance metrics. This reactive approach highlights a limitation of current maintenance practices: rather than anticipating failures, interventions are triggered once the energy loss is high enough to be noticed in production reports or alarms. In the context of predictive maintenance, the patterns visible here suggest that monitoring loss trajectories could provide earlier warning, allowing gearbox interventions to be scheduled before performance deteriorates to the point of extreme monthly losses.

On the other hand, not all turbines that underwent gearbox replacements show a clean, isolated peak before the intervention. For example, WT04 and WT11 display more moderate increases leading up to their replacements, embedded within the usual seasonal variability. In

these cases, the decision to intervene may have been informed by additional signals such as vibration monitoring, oil analysis, or condition-based alerts, which complement the production-based view. This underlines the importance of integrating SCADA-derived loss indicators with other condition monitoring tools.

Looking across the entire fleet, an important deduction is that while seasonal cycles affect all turbines, extreme peaks are not uniformly distributed. They cluster in the machines that required major gearbox work, reinforcing the link between mechanical stress and elevated losses. This strengthens the argument that deviations from the theoretical curve, particularly when they manifest as sustained or exceptional monthly losses, can serve as a practical proxy for underlying mechanical health.

Finally, the fleet perspective makes clear that gearbox replacements, once carried out, do not eliminate the broader environmental influence on performance. After the interventions, the turbines continue to follow the same seasonal loss rhythm as their peers. This indicates that gearbox health determines the severity of losses but does not erase the underlying weather-driven pattern. In other words, mechanical condition and environmental stressors act together: degradation amplifies losses, while interventions restore the baseline, but the seasonal effect remains.

To illustrate how weather conditions, energy losses, and gearbox replacements intersect, two turbines are presented as case studies: WT12 and WT16. Both machines experienced gearbox replacements during the study period, and both show distinctive patterns in their loss profiles that highlight the interaction between environmental stress and mechanical degradation. WT12 provides one of the clearest examples of how energy losses can anticipate mechanical interventions. In the monthly loss time series (Figure 19) the turbine exhibits relatively stable losses between 5–15% for most of the period, following the same seasonal rhythm observed across the fleet. However, in early 2022 there is a dramatic and isolated spike where monthly losses exceed 35%. This extreme deviation is immediately followed by the gearbox replacement, as indicated by the vertical dashed line.

This sequence strongly suggests that the intervention was triggered reactively, in response to a sudden collapse in performance. The fact that losses returned to normal levels afterwards reinforces the interpretation that the gearbox was the critical failure point. From an operational perspective, this validates the use of energy loss as a performance-based diagnostic indicator.

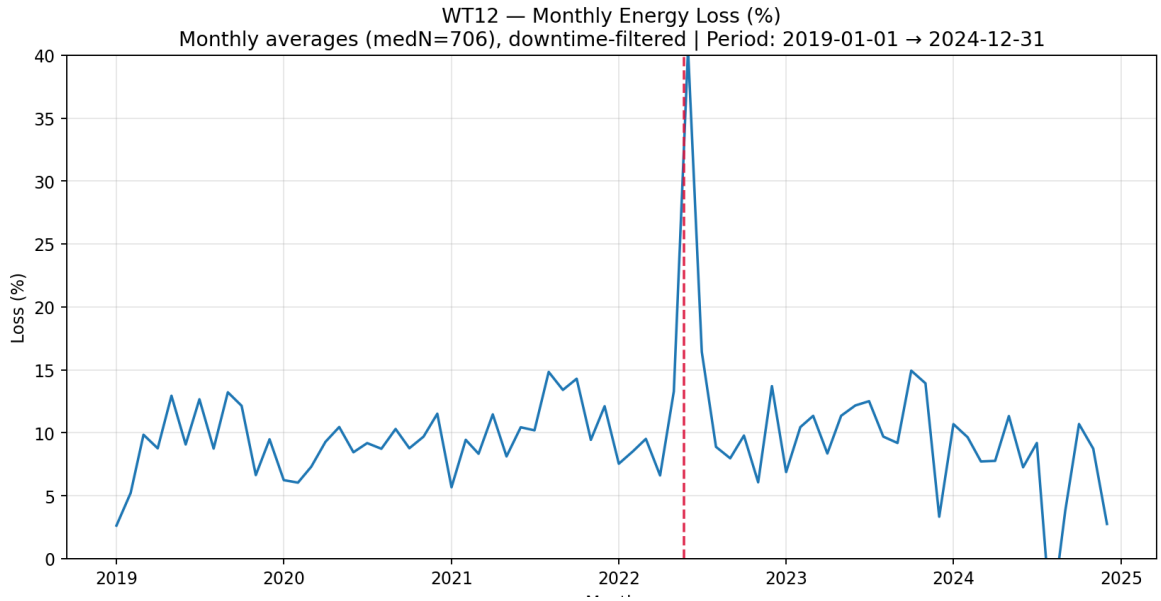


Figure 19. Monthly energy loss (%) for WT12 (2019–2024), showing a sharp peak before the gearbox replacement (red dashed line.

When examining WT12 against temperature (Figure 20) the data show the expected positive association: warmer months tend to correspond with higher losses, although the trend is less steep than for some other turbines. What stands out, however, is that even within this broad relationship, the extraordinary spike seen in the time series cannot be explained by temperature alone. This underlines an important point: weather sets the background conditions for performance, but mechanical failures can generate acute deviations that go far beyond environmental effects.

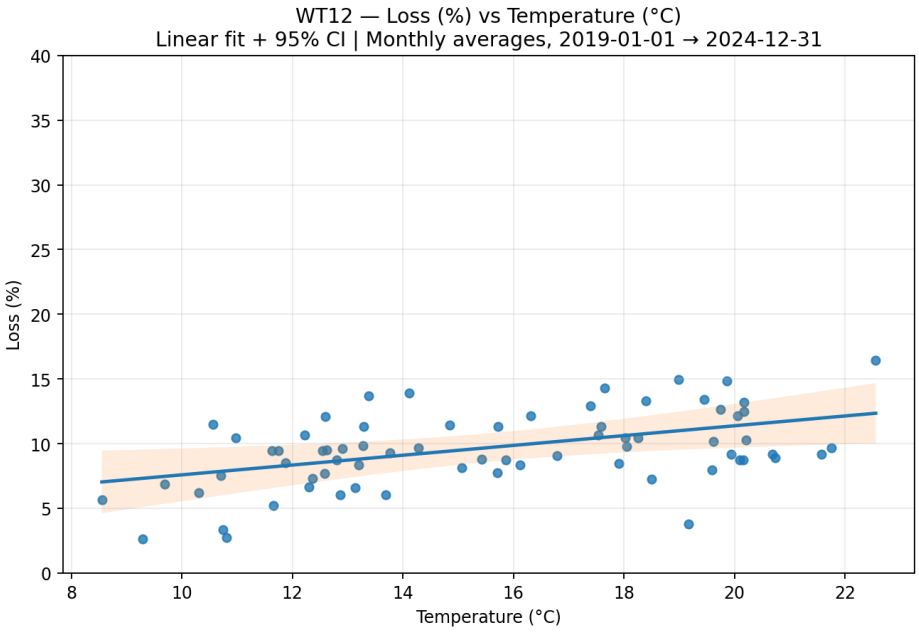


Figure 20. Relationship between temperature and energy loss (%) for WT12 (2019–2024), with monthly averages and linear regression fit (95% confidence interval).

WT12 demonstrates the dual nature of energy losses: a systematic seasonal sensitivity to temperature and an exceptional peak linked directly to gearbox failure.

WT16 presents a complementary case. Unlike WT12, which showed an abrupt spike, WT16's losses rose more gradually before the gearbox replacement in mid-2022. As seen in the time series (Figure 21), monthly losses climbed steadily, reaching values above 30% just before the intervention. After the replacement, losses dropped but continued to exhibit seasonal variation.

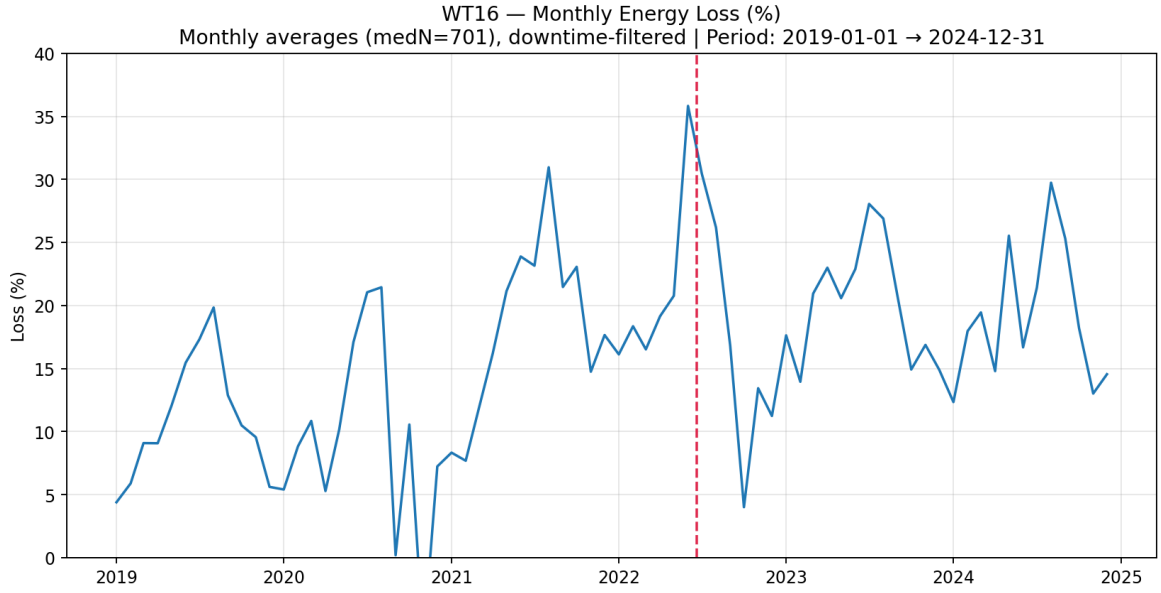


Figure 21. Monthly energy loss (%) for WT16 (2019–2024), showing a marked peak before the gearbox replacement (red dashed line).

The scatter plot of WT16 losses versus temperature (Figure 22) reinforces this interpretation. Compared to WT12, the points are more dispersed, but the positive slope with temperature is clearer and stronger. Losses above 20% occur almost exclusively during months with mean temperatures above 17–18 °C. This suggests that the turbine was particularly vulnerable to environmental stress, with mechanical degradation amplifying the seasonal effect of warm months.

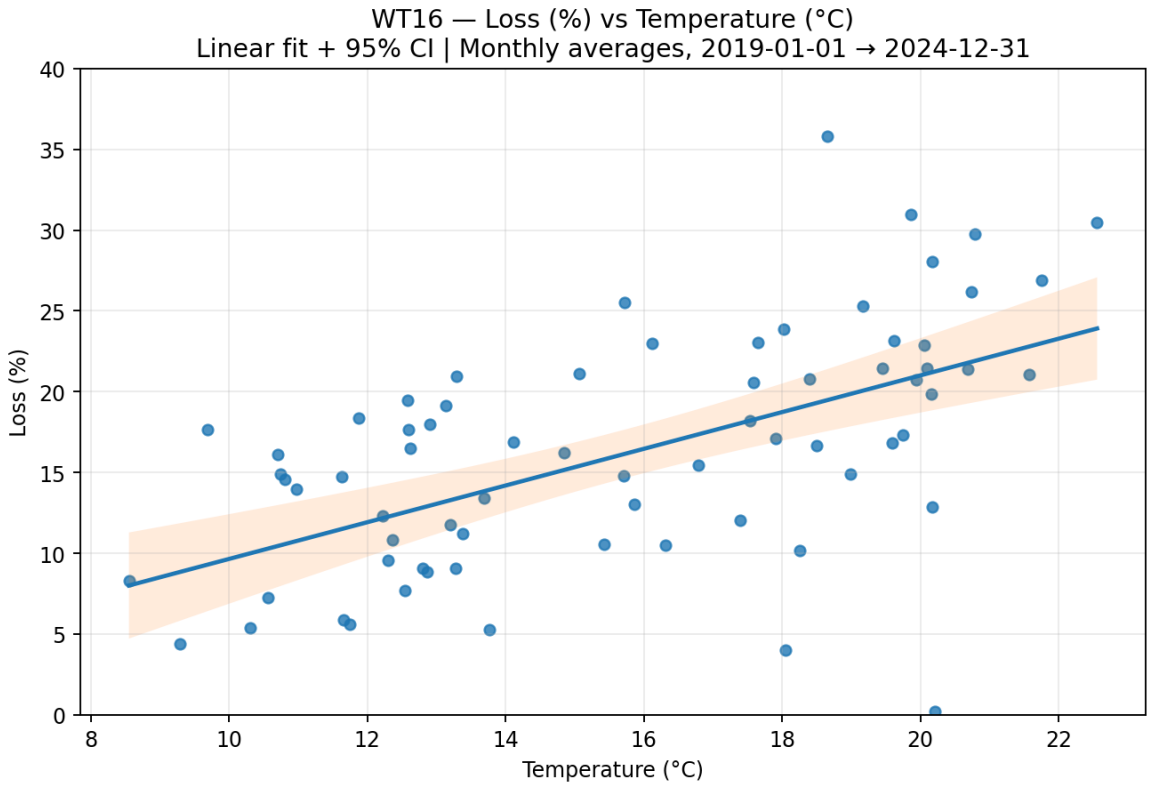


Figure 22. Relationship between temperature and energy loss (%) for WT16 (2019–2024), with monthly averages and linear regression fit (95% confidence interval).

WT16 therefore illustrates a different but equally important pathway: instead of a single dramatic failure point, gradual degradation magnified the underlying weather-driven losses, ultimately necessitating a gearbox replacement.

Taken together, WT12 and WT16 highlight two modes by which energy losses, weather, and mechanical health interact. In WT12, the gearbox failure was sudden and obvious, producing a spike well above the background seasonal variation. In WT16, degradation was more progressive, with losses steadily rising until intervention became unavoidable. Both cases confirm that while temperature shapes the baseline loss profile, mechanical failures can either exacerbate or dominate this signal. From a practical standpoint, monitoring monthly losses against the theoretical power curve provides operators with a clear, quantitative tool to distinguish normal seasonal variability from signs of emerging mechanical failure.

4.6 OIL ANALYSIS AND ENERGY LOSS

The results of the oil and grease analysis are presented in the form of an aggregated OilScore for each turbine, tracked over time and aligned with the dates of gearbox replacements. This approach provides a fleet-wide overview of lubricant condition and allows us to explore whether deterioration in oil health precedes major component interventions. The use of z-score normalization makes it possible to compare across turbines and across different laboratory metrics, with positive values indicating oil conditions worse than the fleet average and negative values reflecting healthier-than-average samples. By examining the temporal evolution of OilScore, we aim to identify whether common deterioration patterns can be observed across the fleet and to assess whether gearbox replacements coincide with episodes of elevated oil stress.

The fleet view graph (Figure 23) displays twenty small multiples, each corresponding to one turbine. The x-axis represents time, covering the six-year study period, while the y-axis shows the OilScore. Vertical dashed red lines mark the timing of gearbox replacement events for the relevant turbines. Several patterns can be deduced from this visualization. First, it is evident that OilScore is not constant but fluctuates considerably over time, reflecting both natural variation in operating conditions and the discrete timing of sampling. Many turbines, such as WT01, WT05, and WT13, show a gradual upward trend in OilScore, with intermittent peaks that suggest worsening lubricant condition followed by either oil changes or natural recovery. Other turbines, such as WT17 and WT18, show predominantly negative or near-zero OilScores across the period, indicating that their oil samples were consistently healthier than the fleet average and did not exhibit extreme stress.

Most striking are the turbines where gearbox replacements occurred. For WT04, WT06, WT11, WT12, WT15, and WT16, the dashed red lines marking replacements coincide closely with local peaks in OilScore. In WT04, a sharp rise in OilScore precedes the replacement, peaking around three standard deviations above the mean, which strongly suggests that oil condition was registering severe stress prior to the intervention. WT06 shows a similar pattern, with a pronounced spike in OilScore followed immediately by the replacement event. WT11 and WT12 both display rising OilScores leading up to their replacements, although the absolute values are more moderate, suggesting that the oil was registering deterioration but not necessarily extreme outliers. WT15 and WT16, on the other hand, show a shift from negative to positive OilScores prior to the replacements, which may indicate that these turbines transitioned from below-average stress to above-average stress as they approached failure.

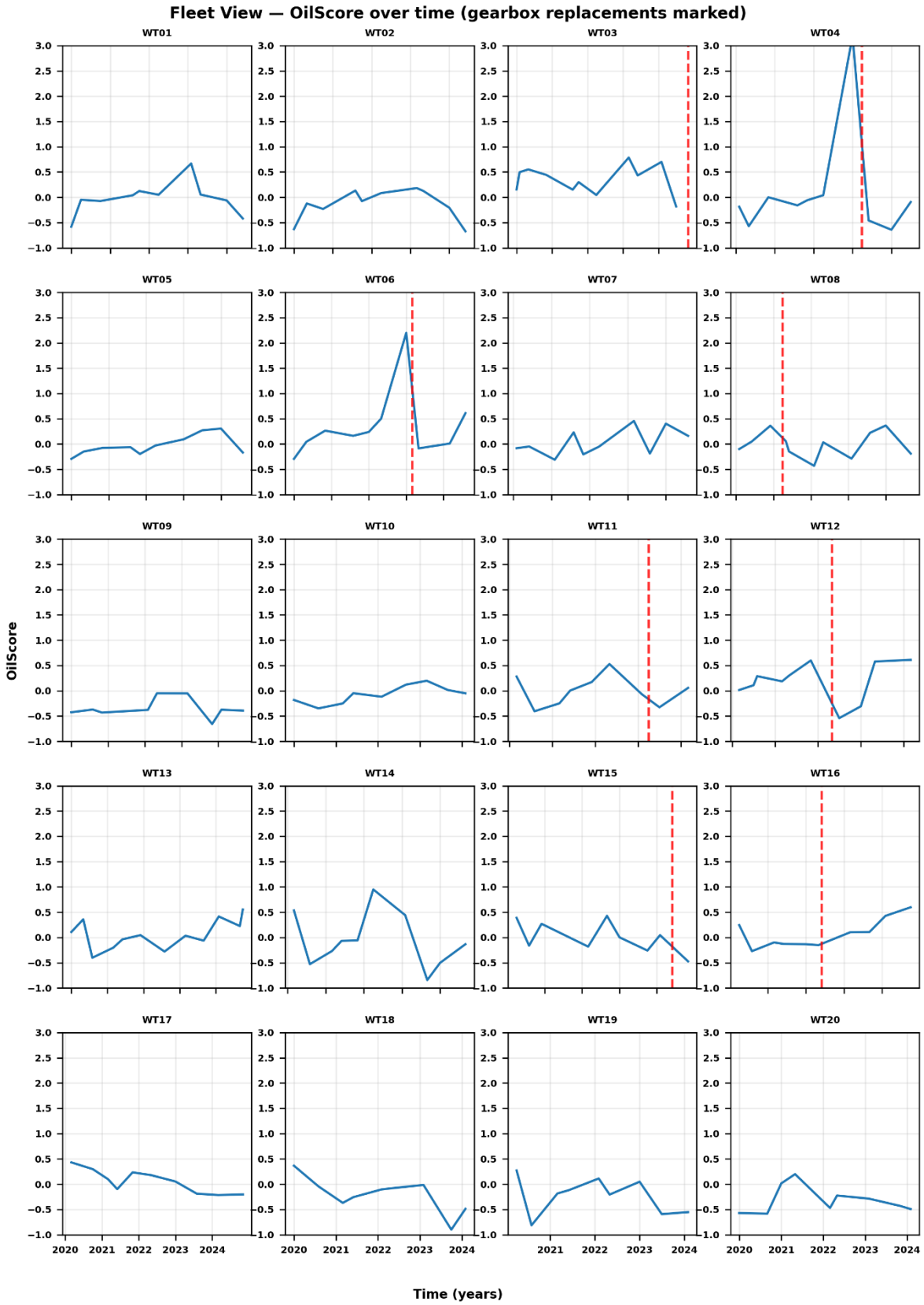


Figure 23. Fleet view - OilScore over time with gearbox replacements marked in red dashed line

Another interesting observation is the diversity of OilScore trajectories across turbines that did not experience replacements. For example, WT02 shows fluctuating scores around zero, with no clear sustained rise, while WT17 exhibits a gradual decline in OilScore, suggesting improvement or consistently benign conditions. This diversity indicates that not all turbines experience lubricant stress in the same way, underscoring the role of site-specific factors, operating histories, and potentially weather exposure in shaping oil condition.

Overall, the fleet view suggests that while OilScore does not always increase monotonically before a gearbox replacement, there is a recurring pattern of elevated or spiking values in the months leading up to major interventions. This finding supports the use of lubricant analysis as an early warning tool, particularly when combined with weather annotations and loss data, as it demonstrates that oil condition captures meaningful signs of gearbox stress at the turbine level.

5

CONCLUSIONS AND FUTURE WORK

This dissertation set out to explore whether weather conditions could be linked to failures in wind turbines, with a specific focus on gearbox replacements. The analysis was based on maintenance records aligned with hourly weather data from Meteostat for the Lousã II wind farm, covering the period from 2019 to 2024. The objective was to investigate whether failures occurred more often under extreme conditions, or whether they coincided with more common, everyday operating environments.

5.1 ANALYSIS OF THE WORK DONE

After cleaning the original maintenance export, a total of 869 actions were retained as relevant failures across components. These included 146 blade events, 57 yaw events, 52 pitch events, 25 generator events, 23 gearbox events, and 566 classified as "other." Within these, eight gearbox replacements were confirmed during the study window and became the focal point of the advanced weather-failure analysis.

Across all components, the evidence showed that most failures occurred under ordinary weather conditions rather than extremes. Wind-related counts were concentrated in the calm to low wind bins (wspd < 15 m/s) and in the moderate gust range (15–25 m/s). By contrast, very few failures were recorded during extreme gusts (> 35 m/s), a result consistent with turbine protective systems that shut down operations once safety thresholds are exceeded.

Humidity appeared as a recurring context across components. Several charts showed that relative humidity bins above 80% coincided with elevated counts of failures. While this does not demonstrate causation, it was one of the more consistent patterns to emerge across the descriptive analyses. Temperature, in contrast, showed more component-specific patterns.

Blade events were more frequent under mild to warm conditions, while other subsystems, such as the gearbox and generator, showed flatter distributions without a single dominant temperature range.

The clustering analysis confirmed that there was no single “risky-weather” state that explained most failures. Instead, the k-means clusters suggested multiple operating regimes. For example, failures often separated into humid–moderate wind clusters versus drier–calmer clusters. For pitch and yaw, clusters with higher humidity were more populated, while for gearbox and generator failures the separation was visible but less clear, due to the smaller number of samples.

The gearbox-focused analyses—Decision Tree and Association Rule Mining—highlighted the challenge of working with a small number of failure events. The Decision Tree analysis achieved perfect recall of failure periods but produced rules that were not stable across all events. One observed split suggested that lower medium-term gusts (≤ 13.36 m/s) combined with moderate maximum temperatures (≤ 31 °C) coincided with one gearbox replacement but not with controls. Other branches described conditions more typical of normal operation. These results should be read as statistical thresholds chosen by the algorithm rather than physical limits for gearbox damage.

The Association Rule Mining analysis, which searched for frequent co-occurrences of conditions, produced rules consistent with this picture. Failures were disproportionately linked with low gusts and high humidity, for example: wpgt_max = Low or rhum_max = High AND wpgt_max = Low. These associations had modest support and confidence values, but the lift values above 1 indicated that they occurred more often in failure periods than would be expected by chance. Once again, the evidence pointed toward ordinary, humid, low-gust operating environments rather than extreme weather.

In addition to the weather-based analyses, two complementary perspectives were explored. First, a simple energy loss metric was derived by comparing actual production with the theoretical Nordex N90/2500 power curve. At the fleet level, a clear positive correlation was observed between monthly energy loss and temperature, with warmer months systematically showing higher deviations. For turbines that underwent gearbox replacements, pronounced energy loss spikes or sustained elevations were often visible in the months preceding the intervention, suggesting that this indicator could provide an operationally useful early-warning signal. Second, laboratory analyses of lubricants were summarised in an OilScore, which aggregated variables such as wear metals, water content, oxidation, and viscosity deviations.

This index allowed a compact assessment of oil health, highlighting degradation processes that may reflect environmental stresses as well as internal mechanical wear.

Regarding the central hypothesis—whether failures increase in extreme weather conditions—the results across all methods clearly indicate that this was not the case. There was no proven increase in failures during extreme wind or gust categories. In fact, most failures were concentrated in moderate ranges, likely reflecting both exposure time (turbines operate more of the time in these conditions) and the role of automatic shutdown procedures during the most severe storms. Relative humidity emerged as a recurrent backdrop for failures, particularly for pitch and yaw systems, although the relationship remained associative rather than causal.

For gearbox replacements, the limited number of events meant that no specific weather–failure rule could be confirmed with confidence. Patterns involving low gusts and humid conditions did emerge, but these should be treated as exploratory insights rather than predictive indicators.

Therefore, the hypothesis that failures increase under extreme weather was not confirmed. Within the evidence available, the study concluded that most failures occurred under ordinary operating weather conditions. Humidity, while not a direct cause, stood out as a repeated contextual factor in several subsystems.

Taken together, the findings suggest that the main risks to turbine reliability do not stem from rare extremes but from prolonged operation in common weather regimes, particularly those that combine moderate winds, low gusts, and high humidity. These are the conditions in which turbines spend most of their operating time. The cumulative effects of cyclic mechanical loading, moisture-assisted degradation, lubrication challenges, and frequent control-system adjustments under these states may drive wear and stress on components more than rare storm events.

The conclusion is therefore twofold:

(i) Extreme weather is not the main driver of observed failures in this dataset.

(ii) Everyday conditions, especially those involving high humidity and moderate winds, deserve closer attention in reliability models and maintenance planning.

This perspective does not agree with the assumption that only high-impact storms pose risks to turbine components. Instead, it suggests that condition monitoring and predictive maintenance should account for cumulative exposure to humid and moderate-wind regimes, using complementary indicators such as energy loss and oil condition to anticipate failures and reduce life-cycle costs.

5.2 FUTURE WORK

The findings of this study highlight several promising directions for future research and development. Rather than focusing only on the limitations of the dataset, future work should aim to expand the scope of analysis, introduce new methodologies, and bring together perspectives from both data science and wind turbine engineering. This will allow the exploratory results presented here to be transformed into practical strategies for predictive maintenance and improved turbine reliability.

One clear direction is the development of more advanced modelling approaches. This study relied on interpretable methods such as labelling, clustering, Decision Trees, and Association Rule Mining. While valuable, these approaches are not sufficient to capture all the complex interactions between weather and turbine operation. Future studies could test ensemble models such as Random Forests or Gradient Boosting, as well as survival analysis methods that estimate the time-to-failure under different conditions. Hybrid physical–statistical models that combine weather variables with fatigue load calculations could also bridge the gap between engineering understanding and statistical prediction.

Another important step is the integration of higher-frequency SCADA data with weather features. Turbines generate continuous signals for torque, vibration, bearing temperatures, oil condition, and power output. Combining these operational measurements with weather data would make it possible to separate external stresses from internal wear processes. Techniques such as sensor fusion and time-series feature engineering could be applied to capture how environmental and operational factors interact in shaping failure risk.

The additional analyses carried out in this work on energy loss and oil condition also open new possibilities. The energy loss metric, which compared actual output with the Nordex N90/2500 reference curve, showed that several gearbox replacements were preceded by unusual loss spikes. Formalising this metric and linking it to cumulative weather exposure and SCADA anomalies could turn it into an operationally useful early-warning tool. Similarly, the OilScore demonstrated how multiple laboratory indicators can be combined into a compact measure of lubricant health. Expanding this into dynamic oil health models, where oil degradation is linked to weather exposure and turbine operation, would strengthen predictive maintenance by integrating lubrication into the same framework as mechanical and environmental factors.

Future work should also shift from simple counts of failures to exposure-adjusted metrics. Failures should be evaluated in relation to the cumulative time turbines spend in specific

regimes, such as humid and low-gust conditions. Survival models like Cox proportional hazards or accelerated failure time models could provide more precise estimates of how prolonged exposure accelerates failure. In parallel, more attention should be given to variability and transitions. Rapid humidity changes, gust fluctuations, or daily cycles may play an important role in stressing components, even if average values remain within "normal" ranges. Methods from time-series and change-point analysis could be used to study these dynamics.

Validation across multiple wind farms is another priority. The present study focused on a single site, which limits the generalisability of the results. Applying the same methodology across different wind farms and turbine models would help distinguish site-specific patterns from general industry-wide ones. Larger and more diverse datasets would also make it possible to train more powerful models while maintaining interpretability.

The ultimate goal of this line of research is to improve predictive maintenance systems. The interpretable rules and associations identified here, such as the recurrence of humid and low-gust regimes before failures, could be built into monitoring dashboards. These dashboards could combine weather exposure indicators with SCADA measurements, energy loss signals, and OilScore trends to provide operators with real-time "watch-list" alerts. Coupled with digital twin simulations, such systems could also forecast how different weather trajectories affect component lifetimes, allowing operators to optimise inspection schedules and spare-parts planning.

Finally, future studies should broaden the environmental dimensions considered. Beyond wind, gusts, temperature, and humidity, other stressors such as air density, icing, salt exposure in coastal sites, and pollutant levels may also influence reliability. Linking these external factors with economic and operational impacts—such as downtime costs, repair expenses, and insurance claims—would ensure that predictive indicators are tied not only to technical outcomes but also to business decisions.

In summary, future work should move beyond filling data gaps and focus on building richer, more practical predictive frameworks. By integrating weather, SCADA, energy loss, and oil condition data, and validating across multiple sites, research can transform exploratory insights into actionable tools. This will support earlier failure detection, more efficient maintenance, and ultimately a reduction in the cost of energy from wind power.

REFERENCES

Anh, N. T., & Duc, N. H. (2019). A STUDY ON POWER OUTPUT OF HORIZONTAL AXIS WIND TURBINES UNDER RAIN. Vietnam Journal of Science and Technology, 57(3), 356. https://doi.org/10.15625/2525-2518/56/3/12721

Anh, N. T., Duc, N. H., & Gmsarn, /. (2022). Effect Analysis of Performance and Pitch Controller Operation for Wind Turbine under Rain. In International Journal (Vol. 16).

Antoniou, A., Rosemeier, M., Tazefidan, K., Krimmer, A., & Wolken-Möhlmann, G. (2020). Impact of Site-Specific Thermal Residual Stress on the Fatigue of Wind-Turbine Blades. AIAA Journal, 58(11), 4781–4793. https://doi.org/10.2514/1.J059388

Arshad, M., & O'Kelly, B. (2019). Global status of wind power generation: theory, practice, and challenges. International Journal of Green Energy, 16(14), 1073–1090. https://doi.org/10.1080/15435075.2019.1597369

Bose, N. (1992). Icing on a small horizontal-axis wind turbine — Part 1: Glaze ice profiles. Journal of Engineering and Industrial Aerodynamics, 45(1), 75–85. https://doi.org/10.1016/0167-6105(92)90006-V

Brøndsted, P., Lilholt, H., & Lystrup, A. (2005). COMPOSITE MATERIALS FOR WIND POWER TURBINE BLADES. Annual Review of Materials Research, 35(1), 505–538. https://doi.org/10.1146/annurev.matsci.35.100303.110641

Burton, T. (Ed.). (2001). Wind energy: Handbook. J. Wiley.

Cao, Y., Wu, Z., & Xu, Z. (2014). Effects of rainfall on aircraft aerodynamics. Progress in Aerospace Sciences, 71, 85–127. https://doi.org/10.1016/j.paerosci.2014.07.003

Ciang, C. C., Lee, J.-R., & Bang, H.-J. (2008). Structural health monitoring for a wind turbine system: A review of damage detection methods. Measurement Science and Technology, 19(12), 122001. https://doi.org/10.1088/0957-0233/19/12/122001

Clare, M. C. A., Warder, S. C., Neal, R., Bhaskaran, B., & Piggott, M. D. (2024). An Unsupervised Learning Approach for Predicting Wind Farm Power and Downstream Wakes Using Weather Patterns. Journal of Advances in Modeling Earth Systems, 16(2), e2023MS003947. https://doi.org/10.1029/2023MS003947

Clifton, A., & Lundquist, J. K. (2012). Data Clustering Reveals Climate Impacts on Local Wind Phenomena. Journal of Applied Meteorology and Climatology, 51(8), 1547–1557. https://doi.org/10.1175/JAMC-D-11-0227.1

Corrigan, R., & DeMiglio, R. (1985). Effect of precipitation on wind turbine performance. (Final report). https://doi.org/10.2172/5801463

Diaconita, A. I., Andrei, G., & Rusu, E. (2022). Estimation of the Tower Shape Effect on the Stress–Strain Behavior of Wind Turbines Operating under Offshore Boundary Conditions. Inventions, 7(1), 11. https://doi.org/10.3390/inventions7010011

Douvi, D., Douvi, E., & Margaris, D. P. (2021). The Operation of a Three-Bladed Horizontal Axis Wind Turbine under Hailstorm Conditions—A Computational Study Focused on Aerodynamic Performance. Inventions, 7(1), 2. https://doi.org/10.3390/inventions7010002

Durst, F., Miloievic, D., & Schönung, B. (1984). Eulerian and Lagrangian predictions of particulate two-phase flows: a numerical study. Applied Mathematical Modelling, 8(2), 101–115. https://doi.org/10.1016/0307-904X(84)90062-3

Edenhofer, Ottmar., Pichs Madruga, R., & Sokona, Y. . (2012). Renewable energy sources and climate change mitigation : special report of the Intergovernmental Panel on Climate Change. Cambridge University Press.

Fiore, G., Camarinha Fujiwara, G. E. C., & Selig, M. S. (2015, January 5). A Damage Assessment for Wind Turbine Blades from Heavy Atmospheric Particles. 53rd AIAA Aerospace Sciences Meeting. https://doi.org/10.2514/6.2015-1495

Fischer, K., Steffes, M., Pelka, K., Tegtmeier, B., & Dörenkämper, M. (2021). Humidity in Power Converters of Wind Turbines—Field Conditions and Their Relation with Failures. Energies, 14(7), 1919. https://doi.org/10.3390/en14071919

G. Wilson, & D. McMilan. (2014). Safety, reliability and risk analysis: Beyond the horizon: proceedings of the European Safety and Reliability Conference, Esrel 2013, Amsterdam, The

Netherlands, 29 September - 2 October 2013, Modeling the relationship between wind turbine failure modes and the environment. CRC Press.

González-González, A., Jimenez Cortadi, A., Galar, D., & Ciani, L. (2018). Condition monitoring of wind turbine pitch controller: A maintenance approach. Measurement, 123, 80–93. https://doi.org/10.1016/j.measurement.2018.01.047

Hahn, B., Durstewitz, M., & Rohrig, K. (2007). Reliability of Wind Turbines. In J. Peinke, P. Schaumann, & S. Barth (Eds), Wind Energy (pp. 329–332). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-33866-6_62

Hau, E. (2013). Wind turbines: Fundamentals, technologies, application, economics (Third, translated edition). Springer.

Hochart, C., Fortin, G., Perron, J., & Ilinca, A. (2008). Wind turbine performance under icing conditions. Wind Energy, 11(4), 319–333. https://doi.org/10.1002/we.258

Homola, M. C., Virk, M. S., Wallenius, T., Nicklasson, P. J., & Sundsbø, P. A. (2010). Effect of atmospheric temperature and droplet size variation on ice accretion of wind turbine blades. Journal of Wind Engineering and Industrial Aerodynamics, 98(12), 724–729. https://doi.org/10.1016/j.jweia.2010.06.007

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8), 651–666. https://doi.org/10.1016/j.patrec.2009.09.011

Kandali, K., Bennis, L., & Bennis, H. (2021). "A New Hybrid Routing Protocol Using a Modified K-Means Clustering Algorithm and Continuous Hopfield Network for VANET." IEEE Access, 9, 47169–47183. https://doi.org/10.1109/ACCESS.2021.3068074

Katsaprakakis, D. Al., Papadakis, N., & Ntintakis, I. (2021). A Comprehensive Analysis of Wind Turbine Blade Damage. Energies, 14(18), 5974. https://doi.org/10.3390/en14185974

Kelly, J., Vogel, C., & Willden, R. (2022a). Impact and mitigation of blade surface roughness effects on wind turbine performance. Wind Energy, 25(4), 660–677. https://doi.org/10.1002/we.2691

Khakpour, Y., Bardakji, S., & Nair, S. (2007). Aerodynamic Performance of Wind Turbine Blades in Dusty Environments. Volume 8: Heat Transfer, Fluid Flows, and Thermal Systems, Parts A and B, 483–491. https://doi.org/10.1115/IMECE2007-43291

Khalfallah, M. G., & Koliub, A. M. (2007). Effect of dust on the performance of wind turbines. Desalination, 209(1–3), 209–220. https://doi.org/10.1016/j.desal.2007.04.030

Letson, F., Barthelmie, R. J., & Pryor, S. C. (2020). Radar-derived precipitation climatology for wind turbine blade leading edge erosion. Wind Energy Science, 5(1), 331–347. https://doi.org/10.5194/wes-5-331-2020

LUERS, J. K. (1985, January 14). RAIN INFLUENCES ON A WIND TURBINE THEORETICAL DEVELOPMENT AND APPLICATIONS. 23rd Aerospace Sciences Meeting. https://doi.org/10.2514/6.1985-256

Manwell, J. F., McGowan, J. G., & Rogers, A. L. (2011). Wind energy explained: Theory, design and application (2. ed., repr. with cor). Wiley.

Manwell, J. F., McGowan, J. G., & Rogers, A. L. (2009). Wind Energy Explained. Wiley. https://doi.org/10.1002/9781119994367

Mishnaevsky, L. (2022). Root Causes and Mechanisms of Failure of Wind Turbine Blades: Overview. Materials (Basel, Switzerland), 15(9), 2959. https://doi.org/10.3390/ma15092959

Papież, M., Śmiech, S., & Frodyma, K. (2019). Factors affecting the efficiency of wind power in the European Union countries. Energy Policy, 132, 965–977. https://doi.org/10.1016/j.enpol.2019.06.036

Parent, O., & Ilinca, A. (2011). Anti-icing and de-icing techniques for wind turbines: Critical review. Cold Regions Science and Technology, 65(1), 88–96. https://doi.org/10.1016/j.coldregions.2010.01.005

Pelka, K., & Fischer, K. (2023). Field-data-based reliability analysis of power converters in wind turbines: Assessing the effect of explanatory variables. Wind Energy, 26(3), 310–324. https://doi.org/10.1002/we.2800

Punge, H. J., & Kunz, M. (2016). Hail observations and hailstorm characteristics in Europe: A review. Atmospheric Research, 176–177, 159–184. https://doi.org/10.1016/j.atmosres.2016.02.012

Qiu, Y., Chen, L., Feng, Y., & Xu, Y. (2017). An Approach of Quantifying Gear Fatigue Life for Wind Turbine Gearboxes Using Supervisory Control and Data Acquisition Data. Energies, 10(8), 1084. https://doi.org/10.3390/en10081084

Reder, M., & Melero, J. J. (2017). Modelling Wind Turbine Failures based on Weather Conditions. Journal of Physics: Conference Series, 926, 012012. https://doi.org/10.1088/1742-6596/926/1/012012

Seifert, H., & Richert, F. (1997). Aerodynamics of iced airfoils and their influence on loads and power production. https://www.researchgate.net/publication/242175233

Tavner, P. J. (2012). Offshore wind turbines: Reliability, availability and maintenance (Online-Ausg). Institution of Engineering and Technology.

Valentine, J. R., & Decker, R. A. (1995). A Lagrangian-Eulerian scheme for flow around an airfoil in Multiphase Flow, 21(4), 639–648. https://doi.org/10.1016/0301-9322(95)00007-K

Wilkinson, M., Harman, K., & Spinato, F. (2007). Condition monitoring of wind turbine drive trains. European Wind Energy Conference (EWEC).

Zhao, M., Jiang, D., & Li, S. (2009). Research on fault mechanism of icing of wind turbine blades. 2009 World Non-Grid-Connected Wind Power and Energy Conference, 1–4. https://doi.org/10.1109/WNWEC.2009.5335772

APPENDICES

- Appendix A – Supervised labelling python code
- Appendix B – Unsupervised K-means clustering python code
- Appendix C – Decision tree python code
- Appendix D – Association Rule Mining python code
- Appendix E – Energy Loss Against Weather Conditions Python Code
- Appendix G – Example Of Lab Report For Oil Analysis

A SUPERVISED LABELLING PYTHON CODE

```
import pandas as pd
from datetime import timedelta
from meteostat import Hourly, Point
import matplotlib.pyplot as plt
import numpy as np
import os

# --------------------------
# LABEL DEFINITION FUNCTIONS
# --------------------------
def label_temp(t):
    if pd.isna(t): return 'unknown'
    elif t <= 10: return 'cold'
    elif t <= 15: return 'cool'
    elif t <= 20: return 'mild'
    elif t <= 25: return 'room temp'
    elif t <= 30: return 'warm'
    else: return 'hot'

def label_rhum(h):
    if pd.isna(h): return 'unknown'
    elif h < 40: return 'dry air'
    elif h < 60: return 'moist air'
    elif h < 80: return 'corrosive'
    else: return 'high corrosive'

def label_wspd(w):
    if pd.isna(w): return 'unknown'
    elif w < 3: return 'calm'
    elif 3 < w < 10: return 'Medium'
    elif 10 < w < 25: return 'High'
    else: return 'Extreme'

def label_gust(g):
    if pd.isna(g): return 'unknown'
```

```
    elif g < 3: return 'light'
    elif 3 < g < 10: return 'moderate'
    elif 10 < g < 25 : return 'strong'
    else: return 'extreme'

# -------------------------
# GROUPED BAR PLOT FUNCTION
# -------------------------
def plot_grouped_bar(data, xlabel, ylabel, title, filename):
    import matplotlib.pyplot as plt
    import numpy as np
    import os

    fig, ax = plt.subplots(figsize=(10, 5))

    categories = data.columns.tolist()
    components = data.index.tolist()
    n_categories = len(categories)
    n_components = len(components)
    bar_width = 0.8 / n_components
    x = np.arange(n_categories)

    # Professional color palette
    colors = ['#66c2a5', '#fc8d62', '#8da0cb', '#e78ac3', '#a6d854',
'#ffd92f']

    for i, component in enumerate(components):
        offset = (i - n_components / 2) * bar_width + bar_width / 2
        ax.bar(
            x + offset,
            data.loc[component],
            bar_width,
            label=component,
            color=colors[i % len(colors)],
            edgecolor='black',
            linewidth=0.8
        )

    ax.set_xticks(x)
    ax.set_xticklabels(categories, rotation=30, fontsize=10)
    ax.set_xlabel(xlabel, fontsize=11)
    ax.set_ylabel(ylabel, fontsize=11)
    ax.set_title(title, fontsize=13, weight='bold')
    ax.grid(axis='y', linestyle='--', alpha=0.4)

    ax.legend(loc='upper left', bbox_to_anchor=(1.02, 1), borderaxespad=0,
fontsize=9, frameon=False)

    plt.tight_layout(rect=[0, 0, 0.85, 1])
```

```
    os.makedirs("final_formatted_charts_V4", exist_ok=True)
    plt.savefig(f"final_formatted_charts_V4/{filename}", dpi=300)
    plt.close()

# -------------------------
# MAIN PROCESSING FUNCTION
# -------------------------
def analyze_multiple_failures(input_files, output_excel_combined):
    all_data = []

    for file in input_files:
        component_name = os.path.splitext(os.path.basename(file))[0]
        df = pd.read_excel(file)
        df['ACTSTART'] = pd.to_datetime(df['ACTSTART'], format='%Y-%m-%d-
%H.%M.%S.%f', errors='coerce')
        df = df.dropna(subset=['ACTSTART'])

        df['COMPONENT'] = component_name
        df['weather_time'] = df['ACTSTART'] - timedelta(minutes=10)
        df['weather_time_rounded'] = df['weather_time'].dt.floor('h')

        all_data.append(df)

    df_all = pd.concat(all_data, ignore_index=True)

    location = Point(40.1167, -8.2492)
    start = pd.to_datetime(df_all['weather_time'].min()).floor('h')
    end = pd.to_datetime(df_all['weather_time'].max()).ceil('h')

    print("Fetching weather data...")
    weather = Hourly(location, start, end, model=True).fetch()

    df_merged = pd.merge(df_all, weather, left_on='weather_time_rounded',
right_index=True, how='left')

    # ⚠️ Meteostat returns wind speed and gust in km/h — convert to m/s
    df_merged['wspd'] = df_merged['wspd'] / 3.6
    df_merged['wpgt'] = df_merged['wpgt'] / 3.6

    # Apply labels
    df_merged['temp_label'] = df_merged['temp'].apply(label_temp)
    df_merged['rhum_label'] = df_merged['rhum'].apply(label_rhum)
    df_merged['wspd_label'] = df_merged['wspd'].apply(label_wspd)
    df_merged['gust_label'] = df_merged['wpgt'].apply(label_gust)

    # Save combined dataset
    df_merged.to_excel(output_excel_combined, index=False)
    print(f"\n✅ Combined labelled dataset saved to:
{output_excel_combined}")
```

```
    # Prepare grouped data for bar charts
    def prepare_chart_data(label_col, sorted_labels):
        grouped = df_merged.groupby(['COMPONENT',
label_col]).size().unstack().fillna(0)
        grouped = grouped.reindex(columns=sorted_labels, fill_value=0)
        return grouped

    wind_data = prepare_chart_data('wspd_label', ['calm', 'Medium', 'High',
'Extreme'])
    rhum_data = prepare_chart_data('rhum_label', ['dry air', 'moist air',
'corrosive', 'high corrosive'])
    temp_data = prepare_chart_data('temp_label', ['cold', 'cool', 'mild',
'room temp', 'warm', 'hot'])
    gust_data = prepare_chart_data('gust_label', ['light', 'moderate',
'strong', 'extreme'])

    # Plot and save final charts
    plot_grouped_bar(rhum_data, 'RH', 'Failure Frequency', '(a) Relative
Humidity', 'humidity_final_V4.png')
    plot_grouped_bar(wind_data, 'WSatF', 'Failure Frequency', '(b) Wind
Speed', 'wind_speed_final_V4.png')
    plot_grouped_bar(temp_data, 'TempatF', 'Failure Frequency', '(d)
Temperature', 'temperature_final_V4.png')
    plot_grouped_bar(gust_data, 'GustatF', 'Failure Frequency', '(c) Wind
Gusts', 'gusts_final_V4.png')

# --------------------------
# RUN SCRIPT
# --------------------------
if __name__ == "__main__":
    input_files = [
        "Yaw.xlsx",
        "Gearbox.xlsx",
        "Generator.xlsx",
        "Blades.xlsx",
        "Pitch.xlsx"
    ]
    output_excel_combined = "All_Labelled_Failures_V4.xlsx"
    analyze_multiple_failures(input_files, output_excel_combined)
```

B UNSUPERVISED K-MEANS CLUSTERING PYTHON CODE

```
import os
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import timedelta
from meteostat import Point, Hourly
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from kneed import KneeLocator

# === 1. Load and standardize time format ===
input_file = "Yaw.xlsx"  # <- replace with your filename
df = pd.read_excel(input_file)

df['ACTSTART'] = pd.to_datetime(df['ACTSTART'], format='%Y-%m-%d-
%H.%M.%S.%f', errors='coerce')
df = df.dropna(subset=['ACTSTART'])

# Create weather alignment time
df['weather_time'] = df['ACTSTART'] - timedelta(minutes=10)
df['weather_time_rounded'] = df['weather_time'].dt.floor('h')

# === 2. Fetch Meteostat weather data ===
location = Point(40.1167, -8.2492)  # Lousã coordinates
start = df['weather_time'].min().floor('h')
end = df['weather_time'].max().ceil('h')

print("Fetching weather data...")
weather = Hourly(location, start, end, model=True).fetch()
weather = weather[['temp', 'rhum', 'wspd', 'wpgt']]  # Include wind gusts

# === 3. Merge failure + weather data ===
df_merged = pd.merge(df, weather, left_on='weather_time_rounded',
right_index=True, how='left')
df_valid = df_merged.dropna(subset=['temp', 'rhum', 'wspd', 'wpgt'])

# === 4. Standardize variables ===
features = ['temp', 'rhum', 'wspd', 'wpgt']
```

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df_valid[features])

# === 5. Elbow method to choose k ===
inertia = []
K_range = range(1, 11)
for k in K_range:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled)
    inertia.append(kmeans.inertia_)

knee = KneeLocator(K_range, inertia, curve='convex', direction='decreasing')
optimal_k = knee.elbow or 3
print(f"Optimal number of clusters: {optimal_k}")

# === 6. Apply K-means ===
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
df_valid['weather_cluster'] = kmeans.fit_predict(X_scaled)

# === 7. Merge cluster back into full dataset ===
df_merged['weather_cluster'] = -1
df_merged.loc[df_valid.index, 'weather_cluster'] =
df_valid['weather_cluster']

# === 8. Export clustered dataset ===
output_file = "Clustered_Yaw_with_Gusts.xlsx"
df_merged.to_excel(output_file, index=False)
print(f"Saved: {output_file}")

# === 9. Pairplot visualization ===
sns.set(style="whitegrid")
plot = sns.pairplot(df_valid, vars=features, hue='weather_cluster',
palette='Set2')
plot.fig.suptitle("K-Means Clustering of Weather Conditions (Yaw)", y=1.03)
plot.savefig("Yaw_weather_clusters_pairplot.png")
plt.show()

print("Visualization saved as: Yaw_weather_clusters_pairplot.png")
```

C DECISION TREE PYTHON CODE

```
import os
from datetime import datetime, timedelta
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from meteostat import Point, Hourly
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.utils import check_random_state

# ----------------------------
# 0) Site & events
# ----------------------------
LAT = 40.1167
LON = -8.249
ELEV = 150
LOCATION = Point(LAT, LON, ELEV)

EVENTS = [
    ("G008", "2021-04-30"),
    ("G012", "2022-05-23"),
    ("G016", "2022-06-20"),
    ("G006", "2023-03-17"),
    ("G011", "2023-04-10"),
    ("G004", "2023-04-20"),
    ("G015", "2024-06-14"),
    ("G003", "2024-11-04"),
]

SHORT_DAYS = 7
MEDIUM_DAYS = 30
CONTROLS_PER_EVENT = 5
CONTROL_GAP_DAYS = 45
RANDOM_SEED = 42
OUT_DIR = "gearbox_patterns_out"
os.makedirs(OUT_DIR, exist_ok=True)
```

```
# ----------------------------
# Helpers
# ----------------------------
def fetch_hourly(start_dt, end_dt, location=LOCATION):
    start_dt = pd.Timestamp(start_dt).tz_localize(None)
    end_dt   = pd.Timestamp(end_dt).tz_localize(None)
    df = Hourly(location, start_dt, end_dt).fetch()

    for col in ["wspd", "wpgt", "temp", "rhum"]:
        if col not in df.columns:
            df[col] = np.nan

    df["wspd"] = df["wspd"] / 3.6
    df["wpgt"] = df["wpgt"] / 3.6
    return df

def window_stats(df):
    if df is None or len(df) == 0:
        return {k: np.nan for k in [
            "wspd_mean","wspd_max","wpgt_max",
            "temp_mean","temp_min","temp_max",
            "rhum_mean","rhum_max"
        ]}
    return {
        "wspd_mean": df["wspd"].mean(),
        "wspd_max": df["wspd"].max(),
        "wpgt_max": df["wpgt"].max(),
        "temp_mean": df["temp"].mean(),
        "temp_min": df["temp"].min(),
        "temp_max": df["temp"].max(),
        "rhum_mean": df["rhum"].mean(),
        "rhum_max": df["rhum"].max()
    }

def build_window(end_date_str, days):
    e = datetime.strptime(end_date_str, "%Y-%m-%d")
    return e - timedelta(days=days), e - timedelta(seconds=1)

def features_for_event(event_date):
    s_short, e_short = build_window(event_date, SHORT_DAYS)
    s_med, e_med = build_window(event_date, MEDIUM_DAYS)

    df_short = fetch_hourly(s_short, e_short)
    df_med = fetch_hourly(s_med, e_med)

    f_short = {f"short_{k}": v for k, v in window_stats(df_short).items()}
    f_med   = {f"med_{k}": v for k, v in window_stats(df_med).items()}
    return {**f_short, **f_med}
```

```
def sample_control_dates(events, controls_per_event, gap_days, rng):
    exclude = []
    years = set()
    for _, d in events:
        e = datetime.strptime(d, "%Y-%m-%d")
        years.add(e.year)
        exclude.append((e - timedelta(days=gap_days), e +
timedelta(days=gap_days)))

    min_year, max_year = min(years), max(years)
    all_days = pd.date_range(f"{min_year}-01-01", f"{max_year}-12-31",
freq="D")

    def allowed(dt):
        for a, b in exclude:
            if a <= dt <= b:
                return False
        return True

    controls = []
    for _, d in events:
        e = datetime.strptime(d, "%Y-%m-%d")
        month_days = [pd.Timestamp(x).to_pydatetime().date() for x in
all_days if (x.month == e.month)]
        rng.shuffle(month_days)
        picked = 0
        for cand in month_days:
            cand_dt = datetime(cand.year, cand.month, cand.day)
            if allowed(cand_dt):
                controls.append(cand_dt.strftime("%Y-%m-%d"))
                picked += 1
            if picked >= controls_per_event:
                break
    return controls

# ----------------------------
# 1) Build dataset
# ----------------------------
rng = check_random_state(RANDOM_SEED)
control_dates = sample_control_dates(EVENTS, CONTROLS_PER_EVENT,
CONTROL_GAP_DAYS, rng)

rows = []
for tid, d in EVENTS:
    feats = features_for_event(d)
    rows.append({"turbine": tid, "end_date": d, "y": 1, **feats})
for d in control_dates:
    feats = features_for_event(d)
    rows.append({"turbine": "CTRL", "end_date": d, "y": 0, **feats})
```

```
df = pd.DataFrame(rows).dropna(axis=1, how="all")
df.to_csv(os.path.join(OUT_DIR, "features_gearbox_windows.csv"), index=False)

# -----------------------------
# 2) Train Decision Tree
# -----------------------------
feature_cols = [c for c in df.columns if c not in ["turbine","end_date","y"]]
X = df[feature_cols].values
y = df["y"].values

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=RANDOM_SEED, stratify=y
)

clf = DecisionTreeClassifier(max_depth=3, min_samples_leaf=2,
                             class_weight="balanced",
random_state=RANDOM_SEED)
clf.fit(X_train, y_train)

print(classification_report(y_test, clf.predict(X_test), digits=3))

# -----------------------------
# 3) Export rules
# -----------------------------
def extract_rules_from_tree(tree, feature_names, node_index=0,
rule_prefix=None):
    if rule_prefix is None:
        rule_prefix = []
    left = tree.children_left[node_index]
    right = tree.children_right[node_index]
    thr = tree.threshold[node_index]
    feat = tree.feature[node_index]
    if left == right:
        value = tree.value[node_index][0]
        samples = int(tree.n_node_samples[node_index])
        neg, pos = int(value[0]), int(value[1])
        prob_pos = pos / max(1, pos+neg)
        return [{
            "rule": " AND ".join(rule_prefix) if rule_prefix else "(always)",
            "samples": samples, "pos": pos, "neg": neg, "pos_rate": prob_pos
        }]
    fname = feature_names[feat]
    rules = []
    rules += extract_rules_from_tree(tree, feature_names,
left,  rule_prefix+[f"{fname} <= {thr:.3f}"])
    rules += extract_rules_from_tree(tree, feature_names, right,
rule_prefix+[f"{fname} > {thr:.3f}"])
    return rules
```

```
rules = extract_rules_from_tree(clf.tree_, feature_cols)
for r in rules:
    print(r)
```

D ASSOCIATION RULE MINING PYTHON CODE

```
# Association Rule Mining for Gearbox Failure Patterns (wind, gusts, temp,
humidity only)

import os
from datetime import datetime, timedelta
import numpy as np
import pandas as pd
from meteostat import Point, Hourly
from sklearn.utils import check_random_state
from mlxtend.frequent_patterns import apriori, association_rules

LAT = 40.1167
LON = -8.249
ELEV = 150
LOCATION = Point(LAT, LON, ELEV)

EVENTS = [
    ("G008", "2021-04-30"),
    ("G012", "2022-05-23"),
    ("G016", "2022-06-20"),
    ("G006", "2023-03-17"),
    ("G011", "2023-04-10"),
    ("G004", "2023-04-20"),
    ("G015", "2024-06-14"),
    ("G003", "2024-11-04"),
]

SHORT_DAYS = 7
CONTROLS_PER_EVENT = 5
CONTROL_GAP_DAYS = 45
RANDOM_SEED = 42
OUT_DIR = "gearbox_assoc_rules"
os.makedirs(OUT_DIR, exist_ok=True)

def fetch_hourly(start_dt, end_dt, location=LOCATION):
    start_dt = pd.Timestamp(start_dt).tz_localize(None)
    end_dt   = pd.Timestamp(end_dt).tz_localize(None)
    df = Hourly(location, start_dt, end_dt).fetch()
    for col in ["wspd", "wpgt", "temp", "rhum"]:
```

```
        if col not in df.columns:
            df[col] = np.nan
    df["wspd"] = df["wspd"] / 3.6
    df["wpgt"] = df["wpgt"] / 3.6
    return df

def window_stats(df):
    return {
        "wspd_mean": df["wspd"].mean(),
        "wspd_max": df["wspd"].max(),
        "wpgt_max": df["wpgt"].max(),
        "temp_mean": df["temp"].mean(),
        "temp_min": df["temp"].min(),
        "temp_max": df["temp"].max(),
        "rhum_mean": df["rhum"].mean(),
        "rhum_max": df["rhum"].max()
    }

def build_window(end_date_str, days):
    e = datetime.strptime(end_date_str, "%Y-%m-%d")
    return e - timedelta(days=days), e - timedelta(seconds=1)

def features_for_date(date_str):
    s_short, e_short = build_window(date_str, SHORT_DAYS)
    df_short = fetch_hourly(s_short, e_short)
    return window_stats(df_short)

def sample_control_dates(events, controls_per_event, gap_days, rng):
    exclude = []
    years = set()
    for _, d in events:
        e = datetime.strptime(d, "%Y-%m-%d")
        years.add(e.year)
        exclude.append((e - timedelta(days=gap_days), e +
timedelta(days=gap_days)))
    min_year, max_year = min(years), max(years)
    all_days = pd.date_range(f"{min_year}-01-01", f"{max_year}-12-31",
freq="D")
    def allowed(dt):
        for a, b in exclude:
            if a <= dt <= b: return False
        return True
    controls = []
    for _, d in events:
        e = datetime.strptime(d, "%Y-%m-%d")
        month_days = [pd.Timestamp(x).to_pydatetime().date() for x in
all_days if x.month == e.month]
        rng.shuffle(month_days)
        picked = 0
```

```
        for cand in month_days:
            cand_dt = datetime(cand.year, cand.month, cand.day)
            if allowed(cand_dt):
                controls.append(cand_dt.strftime("%Y-%m-%d"))
                picked += 1
            if picked >= controls_per_event: break
    return controls

rng = check_random_state(RANDOM_SEED)
control_dates = sample_control_dates(EVENTS, CONTROLS_PER_EVENT,
CONTROL_GAP_DAYS, rng)

rows = []
for tid, d in EVENTS:
    feats = features_for_date(d)
    rows.append({"turbine": tid, "date": d, "y": 1, **feats})
for d in control_dates:
    feats = features_for_date(d)
    rows.append({"turbine": "CTRL", "date": d, "y": 0, **feats})

df = pd.DataFrame(rows).dropna(axis=1, how="all")
df.to_csv(os.path.join(OUT_DIR, "assoc_features.csv"), index=False)

# Bin continuous vars
thresholds = []
binned = pd.DataFrame()
for col in df.columns:
    if col in ["turbine","date","y"]:
        binned[col] = df[col]
    else:
        cats, edges = pd.cut(df[col], bins=3, retbins=True, labels=False,
include_lowest=True)
        thresholds.append({"Variable": col, "Low_max": edges[1],
"Medium_max": edges[2], "High_min": edges[2]})
        binned[col] = cats
pd.DataFrame(thresholds).to_csv(os.path.join(OUT_DIR,
"label_thresholds.csv"), index=False)

# Transactions
transactions = []
for _, row in binned.iterrows():
    items = []
    for col in df.columns:
        if col in ["turbine","date","y"]: continue
        if pd.isna(row[col]): continue
        cat = ["Low","Medium","High"][int(row[col])]
        items.append(f"{col}={cat}")
    items.append("Failure=Yes" if row["y"]==1 else "Failure=No")
    transactions.append(items)
```

```
# One-hot encode
all_items = sorted({item for trans in transactions for item in trans})
onehot = pd.DataFrame(0, index=range(len(transactions)), columns=all_items)
for i, trans in enumerate(transactions):
    onehot.loc[i, trans] = 1

# ARM
freq_items = apriori(onehot, min_support=0.1, use_colnames=True)
rules = association_rules(freq_items, metric="lift", min_threshold=1.0)
rules = rules[rules['consequents'].apply(lambda x: 'Failure=Yes' in x)]
rules.sort_values(by=["lift","confidence"], ascending=False).to_csv(
    os.path.join(OUT_DIR, "assoc_rules.csv"), index=False
)
```

E ENERGY LOSS AGAINST WEATHER CONDITIONS PYTHON CODE

```
from pathlib import Path
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from meteostat import Hourly, Point
from datetime import datetime

# ---------------------- Config ----------------------
SCADA_FILENAME = "WindFarm_Tidy_2019_2024_AllYears.xlsx"
OUTPUT_XLSX    = "monthly_eloss_weather_V3.xlsx"
PLOTS_DIR      = "plots_V3"

LAT, LON = 40.1167, -8.2492
TIMEZONE = "Europe/Lisbon"

YMAX_LOSS = 40  # fixed y-axis for comparability
RATED_POWER_KW = 2500.0
CUT_IN_MS = 4.0

# Save options to prevent text clipping
SAVE_KW = dict(bbox_inches="tight", pad_inches=0.2, dpi=170)

# Nordex N90/2500 curve (as used throughout)
PC_WS = np.array([3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
19, 20, 21, 22, 23, 24, 25], dtype=float)
PC_P  = np.array([0,
70,180,400,700,1000,1350,1750,2100,2350,2450,2500,2500,2500,2500,2500,2500,25
00,2500,2500,2500,2500,2500], dtype=float)

# Gearbox replacements (WT IDs → timestamps)
def _parse_gbx(dtstr: str) -> pd.Timestamp:
```

```
    return pd.to_datetime(datetime.strptime(dtstr, "%Y-%m-%d-%H.%M.%S.%f"))

GBX = {
    "WT08": _parse_gbx("2021-04-30-10.00.00.000000"),
    "WT12": _parse_gbx("2022-05-23-10.00.00.000000"),
    "WT16": _parse_gbx("2022-06-20-12.56.00.000000"),
    "WT06": _parse_gbx("2023-03-17-18.13.00.000000"),
    "WT11": _parse_gbx("2023-04-10-18.48.00.000000"),
    "WT04": _parse_gbx("2023-04-20-19.05.00.000000"),
    "WT15": _parse_gbx("2024-06-14-16.07.00.000000"),
    "WT03": _parse_gbx("2024-11-04-15.59.00.000000"),
}

# ---------------------- Helpers ----------------------
def interp_power(ws: np.ndarray) -> np.ndarray:
    ws = np.asarray(ws, dtype=float)
    p = np.interp(ws, PC_WS, PC_P, left=0.0, right=RATED_POWER_KW)
    p[ws < CUT_IN_MS] = 0.0
    return p

def fetch_weather(lat: float, lon: float, tzname: str, start, end) ->
pd.DataFrame:
    loc = Point(lat, lon)
    dfw = Hourly(loc, start, end).fetch()
    if dfw.empty:
        raise RuntimeError("Meteostat returned no hourly weather. Check
internet/date range.")
    dfw = dfw[['temp','rhum','wpgt','wspd','prcp']].copy()
    dfw = dfw.rename(columns={'temp':'T_C', 'rhum':'RH_pct',
'wpgt':'gust_ms', 'wspd':'wind_ms', 'prcp':'rain_mm'})
    dfw.index = dfw.index.tz_localize('UTC').tz_convert(tzname)
    dfw['DateTime'] = dfw.index.tz_localize(None)
    with np.errstate(divide='ignore', invalid='ignore'):
        dfw['gust_ratio'] = dfw['gust_ms'] / dfw['wind_ms']
    return
dfw.reset_index(drop=True)[['DateTime','T_C','RH_pct','gust_ms','wind_ms','ra
in_mm','gust_ratio']]

def melt_scada(scada: pd.DataFrame) -> pd.DataFrame:
    wind_cols = [c for c in scada.columns if c.endswith('_Wind(m/s)')]
    turbines = sorted({c.split('_')[0] for c in wind_cols})
    frames = []
    for tid in turbines:
        wcol = f"{tid}_Wind(m/s)"; pcol = f"{tid}_Power(kW)"
        sub = scada[['DateTime', wcol, pcol]].copy()
        sub = sub.rename(columns={wcol:'Wind_ms', pcol:'Power_kW'})
        sub['Turbine'] = tid
        frames.append(sub)
    return pd.concat(frames, ignore_index=True)
```

```
def monthly_eloss(scada_long: pd.DataFrame) -> pd.DataFrame:
    df = scada_long.copy()
    ok = ~((df['Wind_ms'] >= CUT_IN_MS) & (df['Power_kW'] < 10.0))
    df = df.loc[ok].copy()
    df['P_theo'] = interp_power(df['Wind_ms'])
    dt = pd.to_datetime(df['DateTime'])
    df['Year'] = dt.dt.year; df['Month'] = dt.dt.month
    grp = df.groupby(['Turbine','Year','Month'], as_index=False).agg(
        E_act_kWh=('Power_kW','sum'),
        E_theo_kWh=('P_theo','sum'),
        Hours=('Power_kW','size')
    )
    grp['E_act_MWh'] = grp['E_act_kWh'] / 1000.0
    grp['E_theo_MWh'] = grp['E_theo_kWh'] / 1000.0
    grp['E_loss_%'] = np.where(grp['E_theo_kWh'] > 0,
                               (grp['E_theo_kWh'] -
grp['E_act_kWh'])/grp['E_theo_kWh']*100.0,
                               np.nan)
    return
grp[['Turbine','Year','Month','Hours','E_act_MWh','E_theo_MWh','E_loss_%']]

def monthly_weather(dfw: pd.DataFrame) -> pd.DataFrame:
    dt = pd.to_datetime(dfw['DateTime'])
    dfw['Year'] = dt.dt.year; dfw['Month'] = dt.dt.month
    return dfw.groupby(['Year','Month'], as_index=False).agg(
        Temp_mean=('T_C','mean'),
        RH_mean=('RH_pct','mean'),
        GustRatio_mean=('gust_ratio','mean'),
        Rain_total=('rain_mm','sum'),
    )

# ---------------------- Plot utilities ----------------------
def save_power_curve_figure(out_dir: Path):
    out_dir.mkdir(parents=True, exist_ok=True)
    fig, ax = plt.subplots(figsize=(6.6, 4.6), constrained_layout=True)
    ax.plot(PC_WS, PC_P, linewidth=2)
    ax.set_title("Nordex N90/2500 Theoretical Power Curve\nReference used for
Loss (%) calculations")
    ax.set_xlabel("Wind speed (m/s)"); ax.set_ylabel("Power (kW)")
    ax.grid(alpha=0.3)
    fig.savefig(out_dir / "power_curve_N90_2500.png", **SAVE_KW);
plt.close(fig)

def _period_col(df):
    return pd.to_datetime(df['Year'].astype(str) + '-' +
df['Month'].astype(str) + '-01')

def _draw_gbx_vline(ax, tid, start, end):
```

```
    if tid in GBX:
        ts = GBX[tid]
        if start <= ts <= end:
            ax.axvline(ts, color='crimson', linestyle='--', linewidth=1.5,
alpha=0.9)

def _legend_if_any(ax, **kwargs):
    h, l = ax.get_legend_handles_labels()
    if l:
        ax.legend(**kwargs)

def _safe_linfit_and_plot(ax, x, y):
    x = np.asarray(x, dtype=float); y = np.asarray(y, dtype=float)
    mask = np.isfinite(x) & np.isfinite(y)
    x, y = x[mask], y[mask]
    if len(x) < 3 or np.nanstd(x) == 0.0:
        return
    try:
        A = np.vstack([x, np.ones_like(x)]).T
        a, b = np.linalg.lstsq(A, y, rcond=None)[0]
        xx = np.linspace(x.min(), x.max(), 100)
        yy = a*xx + b
        ax.plot(xx, yy, color='black', linewidth=1.2)
    except Exception:
        return

# ---------------------- Figures: fleet ----------------------
def make_fleet_panel_time(joined: pd.DataFrame, out_dir: Path, start, end):
    j = joined.copy(); j['Period'] = _period_col(j)
    turbines = sorted(j['Turbine'].unique())
    rows, cols = 4, 5
    fig, axes = plt.subplots(rows, cols, figsize=(17, 10.5), sharex=True,
sharey=True, constrained_layout=True)
    axes = axes.ravel()
    for i, tid in enumerate(turbines):
        ax = axes[i]; sub = j[j['Turbine']==tid].sort_values('Period')
        ax.plot(sub['Period'], sub['E_loss_%'], linewidth=1.3)
        _draw_gbx_vline(ax, tid, start, end)
        medN = int(np.nanmedian(sub['Hours'])) if len(sub) else 0
        ax.text(0.98, 0.06, f"medN={medN}", transform=ax.transAxes,
ha='right', va='bottom', fontsize=7, color='gray')
        ax.set_title(tid, fontsize=9); ax.set_ylim(0, YMAX_LOSS);
ax.grid(alpha=0.2)
    for k in range(i+1, rows*cols): axes[k].axis('off')
    fig.suptitle(f"Monthly Energy Loss (%) — All Turbines (Monthly averages,
downtime-filtered)\n"
                 f"Period: {start.date()} to {end.date()} — Reference: Nordex
N90/2500",
                 fontsize=12)
```

```
    for ax in axes[(rows-1)*cols:]: ax.set_xlabel("Month")
    fig.text(0.04, 0.5, "Loss (%)", va='center', rotation='vertical')
    fig.savefig(out_dir / "fleet_panel_eloss.png", **SAVE_KW); plt.close(fig)

def _fleet_scatter(joined: pd.DataFrame, out_dir: Path, xcol: str, xlabel:
str, fname: str, start, end):
    j = joined.copy(); j['Period'] = _period_col(j)
    turbines = sorted(j['Turbine'].unique())
    rows, cols = 4, 5
    fig, axes = plt.subplots(rows, cols, figsize=(17, 10.5), sharex=False,
sharey=True, constrained_layout=True)
    axes = axes.ravel()

    for i, tid in enumerate(turbines):
        ax = axes[i]
        sub = j[j['Turbine']==tid].dropna(subset=[xcol, 'E_loss_%']).copy()
        if tid in GBX:
            gbx_t = GBX[tid]
            sub['phase'] = np.where(sub['Period'] < gbx_t, 'pre', 'post')
            colors = {'pre':'tab:blue', 'post':'tab:orange'}
            for ph, g in sub.groupby('phase'):
                if len(g) == 0: continue
                denom = float(g['Hours'].max()) if g['Hours'].max() and
np.isfinite(g['Hours'].max()) else 1.0
                sizes = np.clip((g['Hours'] / denom)*120, 20, 120)
                ax.scatter(g[xcol], g['E_loss_%'], s=sizes, alpha=0.7,
label=ph, color=colors[ph])
            _safe_linfit_and_plot(ax, sub[xcol], sub['E_loss_%'])
            _legend_if_any(ax, fontsize=7, loc='upper left', framealpha=0.3)
        else:
            if len(sub):
                denom = float(sub['Hours'].max()) if sub['Hours'].max() and
np.isfinite(sub['Hours'].max()) else 1.0
                sizes = np.clip((sub['Hours'] / denom)*120, 20, 120)
                ax.scatter(sub[xcol], sub['E_loss_%'], s=sizes, alpha=0.7)
                _safe_linfit_and_plot(ax, sub[xcol], sub['E_loss_%'])

        medN = int(np.nanmedian(sub['Hours'])) if len(sub) else 0
        ax.text(0.98, 0.06, f"medN={medN}", transform=ax.transAxes,
ha='right', va='bottom', fontsize=7, color='gray')
        ax.set_title(tid, fontsize=9); ax.set_ylim(0, YMAX_LOSS);
ax.grid(alpha=0.2)

    for k in range(i+1, rows*cols): axes[k].axis('off')
    fig.suptitle(f"Loss (%) vs {xlabel} — All Turbines (Monthly averages,
downtime-filtered)\n"
                 f"Marker size ∝ samples (Hours). Period: {start.date()} to
{end.date()} — Reference: Nordex N90/2500",
                 fontsize=12)
```

```
    fig.text(0.5, 0.02, xlabel, ha='center'); fig.text(0.04, 0.5, "Loss (%)",
va='center', rotation='vertical')
    fig.savefig(out_dir / fname, **SAVE_KW); plt.close(fig)

# ---------------------- Figures: per-turbine ----------------------
def make_per_turbine_time(joined: pd.DataFrame, out_dir: Path, start, end):
    j = joined.copy(); j['Period'] = _period_col(j)
    out_dir.mkdir(parents=True, exist_ok=True)
    for tid, sub in j.groupby('Turbine'):
        sub = sub.sort_values('Period')
        fig, ax = plt.subplots(figsize=(9.6, 5.1), constrained_layout=True)
        ax.plot(sub['Period'], sub['E_loss_%'])
        _draw_gbx_vline(ax, tid, start, end)
        ax.set_ylim(0, YMAX_LOSS)
        medN = int(np.nanmedian(sub['Hours'])) if len(sub) else 0
        ax.set_title(f"{tid} — Monthly Energy Loss (%)\nMonthly averages
(medN={medN}), downtime-filtered | Period: {start.date()} → {end.date()}")
        ax.set_xlabel("Month"); ax.set_ylabel("Loss (%)"); ax.grid(alpha=0.3)
        fig.savefig(out_dir / f"{tid}_eloss_timeseries.png", **SAVE_KW);
plt.close(fig)

def make_per_turbine_scatter_all(joined: pd.DataFrame, out_dir: Path, start,
end):
    j = joined.copy(); j['Period'] = _period_col(j)
    out_dir.mkdir(parents=True, exist_ok=True)
    vars_and_labels = [
        ('Temp_mean', 'Temperature (°C)', 'Temp_mean'),
        ('RH_mean', 'Relative Humidity (%)', 'RH_mean'),
        ('GustRatio_mean', 'Gust ratio (gust/mean)', 'GustRatio_mean'),
        ('Rain_total', 'Monthly rainfall (mm)', 'Rain_total'),
    ]
    for tid, sub in j.groupby('Turbine'):
        medN = int(np.nanmedian(sub['Hours'])) if len(sub) else 0
        for xcol, xlabel, tag in vars_and_labels:
            g = sub.dropna(subset=[xcol, 'E_loss_%']).copy()
            if len(g) == 0:
                continue
            denom = float(g['Hours'].max()) if g['Hours'].max() and
np.isfinite(g['Hours'].max()) else 1.0
            sizes = np.clip((g['Hours'] / denom)*120, 20, 120)
            fig, ax = plt.subplots(figsize=(7.6, 5.2),
constrained_layout=True)
            ax.scatter(g[xcol], g['E_loss_%'], s=sizes, alpha=0.75)
            ax.set_ylim(0, YMAX_LOSS)
            ax.set_title(f"{tid} — Loss (%) vs {xlabel}\nMonthly averages
(medN={medN}), downtime-filtered | Period: {start.date()} → {end.date()}")
            ax.set_xlabel(xlabel); ax.set_ylabel("Loss (%)");
ax.grid(alpha=0.25)
```

```
            fig.savefig(out_dir / f"{tid}_eloss_vs_{tag}.png", **SAVE_KW);
plt.close(fig)

def make_per_turbine_temp_trend(joined: pd.DataFrame, out_dir: Path, start,
end):
    j = joined.copy(); j['Period'] = _period_col(j)
    out_dir.mkdir(parents=True, exist_ok=True)
    for tid, sub in j.groupby('Turbine'):
        x = sub['Temp_mean'].to_numpy(dtype=float)
        y = sub['E_loss_%'].to_numpy(dtype=float)
        mask = np.isfinite(x) & np.isfinite(y)
        x, y = x[mask], y[mask]
        if len(x) < 3:
            continue
        # Fit
        A = np.vstack([x, np.ones_like(x)]).T
        try:
            a, b = np.linalg.lstsq(A, y, rcond=None)[0]
        except Exception:
            continue
        x_line = np.linspace(x.min(), x.max(), 100)
        y_line = a*x_line + b
        # 95% CI
        y_hat = a*x + b
        dof = max(len(x) - 2, 1)
        s2 = np.sum((y - y_hat)**2) / dof
        x_mean = np.mean(x)
        Sxx = np.sum((x - x_mean)**2)
        t = 1.96
        se_line = np.sqrt(s2 * (1/len(x) + (x_line - x_mean)**2 / Sxx)) if
Sxx > 0 else np.full_like(x_line, np.nan)
        y_low = y_line - t*se_line
        y_high = y_line + t*se_line

        fig, ax = plt.subplots(figsize=(7.6, 5.2), constrained_layout=True)
        ax.scatter(x, y, s=25, alpha=0.8)
        ax.plot(x_line, y_line, linewidth=2)
        if np.all(np.isfinite(y_low)):
            ax.fill_between(x_line, y_low, y_high, alpha=0.15)
        ax.set_ylim(0, YMAX_LOSS)
        ax.set_title(f"{tid} — Loss (%) vs Temperature (°C)\nLinear fit + 95%
CI | Monthly averages, {start.date()} → {end.date()}")
        ax.set_xlabel("Temperature (°C)"); ax.set_ylabel("Loss (%)");
ax.grid(alpha=0.25)
        fig.savefig(out_dir / f"{tid}_eloss_vs_Temp_mean_trend.png",
**SAVE_KW); plt.close(fig)

# ---------------------- SUMMARY table ----------------------
def build_summary(eloss_m: pd.DataFrame) -> pd.DataFrame:
```

```
    """Create per-turbine summary: samples, medN, months, loss stats, gearbox
date."""
    # Aggregate per turbine
    agg = eloss_m.groupby('Turbine').agg(
        Samples_hours=('Hours','sum'),
        medN_month=('Hours','median'),
        N_months=('E_loss_%','count'),
        Loss_mean_pct=('E_loss_%','mean'),
        Loss_min_pct=('E_loss_%','min'),
        Loss_max_pct=('E_loss_%','max'),
    ).reset_index()

    # Attach gearbox date if any
    agg['Gearbox_date'] = agg['Turbine'].map(lambda t: GBX.get(t, pd.NaT))
    # Round for neatness
    for c in ['Loss_mean_pct','Loss_min_pct','Loss_max_pct','medN_month']:
        agg[c] = agg[c].round(2)
    agg['Samples_hours'] = agg['Samples_hours'].astype(int)
    # Nice ordering
    cols =
['Turbine','Samples_hours','medN_month','N_months','Loss_mean_pct','Loss_min_
pct','Loss_max_pct','Gearbox_date']
    return agg[cols].sort_values('Turbine')

# ---------------------- Main ----------------------
def main():
    here = Path(__file__).resolve().parent
    plots_dir = here / PLOTS_DIR
    scada_path = here / SCADA_FILENAME
    out_xlsx = here / OUTPUT_XLSX

    if not scada_path.exists():
        raise FileNotFoundError(f"SCADA Excel not found: {scada_path}")

    # Load data
    scada = pd.read_excel(scada_path)
    scada['DateTime'] = pd.to_datetime(scada['DateTime'])
    start, end = scada['DateTime'].min(), scada['DateTime'].max()

    dfw = fetch_weather(LAT, LON, TIMEZONE, start, end)
    weather_m = monthly_weather(dfw)

    scada_long = melt_scada(scada)
    eloss_m = monthly_eloss(scada_long)

    joined = eloss_m.merge(weather_m, on=['Year','Month'], how='left')

    # SUMMARY table
    summary_df = build_summary(eloss_m)
```

```
    # also write a CSV next to the Excel
    summary_csv = here / "summary_table.csv"
    summary_df.to_csv(summary_csv, index=False)

    # Save tables (add SUMMARY sheet)
    with pd.ExcelWriter(out_xlsx, engine='openpyxl') as xw:
        eloss_m.to_excel(xw, sheet_name='ELOSS', index=False)
        weather_m.to_excel(xw, sheet_name='WEATHER', index=False)
        joined.to_excel(xw, sheet_name='JOINED', index=False)
        summary_df.to_excel(xw, sheet_name='SUMMARY', index=False)

    # Figures (unchanged)
    save_power_curve_figure(plots_dir)
    make_fleet_panel_time(joined, plots_dir, start, end)
    _fleet_scatter(joined, plots_dir, 'Temp_mean',       'Temperature
(°C)',      'fleet_panel_loss_vs_temp.png', start, end)
    _fleet_scatter(joined, plots_dir, 'RH_mean',         'Relative Humidity
(%)', 'fleet_panel_loss_vs_rh.png',   start, end)
    _fleet_scatter(joined, plots_dir, 'GustRatio_mean',  'Gust ratio
(gust/mean)','fleet_panel_loss_vs_gust.png', start, end)
    _fleet_scatter(joined, plots_dir, 'Rain_total',      'Monthly rainfall
(mm)', 'fleet_panel_loss_vs_rain.png', start, end)

    make_per_turbine_time(joined, plots_dir, start, end)
    make_per_turbine_scatter_all(joined, plots_dir, start, end)
    make_per_turbine_temp_trend(joined, plots_dir, start, end)

    print(f"Done. Wrote: {out_xlsx.name}, {summary_csv.name} and plots in
./{PLOTS_DIR}/")

if __name__ == "__main__":
    main()
```

F OIL SCORE AND ENERGY LOSS ANALYSIS PYTHON CODE

```
"""
Build OilScore from oil lab Excel (includes turbine_id), aggregate monthly,
optionally merge with monthly losses + weather
(monthly_eloss_weather_V3.xlsx),
and mark gearbox replacement months with vertical dashed lines (hard-coded in
script).
Outputs (auto-created next to MONTHLY_XLS if provided, else next to
OIL_FILE):
- oilscore_monthly_by_turbine_<TS>.xlsx/.csv
- oilscore_monthly_farm_<TS>.xlsx/.csv
- If MONTHLY_XLS is provided (and has JOINED sheet):
    - merged_farm_oilscore_losses_weather_<TS>.xlsx/.csv
    - fig_farm_loss_vs_oilscore.png
    - fig_farm_oilscore_timeseries.png   (with red dashed replacement
markers)
    - fig_farm_oilscore_by_bins.png
    - fig_fleet_oilscore_smallmultiples.png   <-- NEW
    - per_turbine_plots/
        - <TURBINE>_loss_vs_oilscore.png
        - <TURBINE>_oilscore_timeseries.png  (with dashed markers)
        - <TURBINE>_loss_timeseries.png      (optional; with dashed markers)
"""

import os, sys
from math import ceil
from datetime import datetime
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# ========= EDIT THESE PATHS =========
OIL_FILE     = r"C:\Users\Hazem\OIL LAB
ANALYSIS\Gearboxes\oil_parsed_20250928_010539.xlsx"     # <-- oil Excel (must
contain turbine_id)
MONTHLY_XLS  = r"C:\Users\Hazem\E_loss
scripts\monthly_eloss_weather_V3.xlsx"                  # <-- optional ("" 
to skip merge/plots)
# ====================================
```

```
# ========= HARD-CODED GEARBOX REPLACEMENTS =========
REPLACEMENTS = {
    "WT08": ["2021-04-30"],   # G008
    "WT12": ["2022-05-23"],   # G012
    "WT16": ["2022-06-20"],   # G016
    "WT06": ["2023-03-17"],   # G006
    "WT11": ["2023-04-10"],   # G011
    "WT04": ["2023-04-20"],   # G004
    "WT15": ["2024-06-14"],   # G015
    "WT03": ["2024-11-04"],   # G003
}
MAKE_TURBINE_LOSS_TS = True
# ================================================

# ---------- Helpers ----------
def ensure_dir(p: str):
    os.makedirs(p, exist_ok=True)

def pick_col(df: pd.DataFrame, candidates):
    d = {c.lower(): c for c in df.columns}
    for c in candidates:
        if c.lower() in d:
            return d[c.lower()]
    return None

def month_floor(s: pd.Series) -> pd.Series:
    return pd.to_datetime(s,
errors="coerce").dt.to_period("M").dt.to_timestamp()

def to_month_ts(x):
    """Convert a date string (YYYY-MM or YYYY-MM-DD) to a month-start
Timestamp."""
    return pd.to_datetime(str(x),
errors="coerce").to_period("M").to_timestamp()

def zscore(x: pd.Series) -> pd.Series:
    x = pd.to_numeric(x, errors="coerce")
    mu = np.nanmean(x); sd = np.nanstd(x, ddof=0)
    return (x - mu) / (sd if sd > 0 else 1.0)

def percent_visc_dev(visc_cSt: pd.Series, nominal=320.0) -> pd.Series:
    x = pd.to_numeric(visc_cSt, errors="coerce")
    return np.abs((x - nominal) / nominal) * 100.0

def month_from_year_month(year, month):
    s = pd.to_datetime(dict(year=year.astype(int), month=month.astype(int),
day=1))
    return s.dt.to_period("M").dt.to_timestamp()
```

```
def annotate_hot_humid(df_month: pd.DataFrame) -> pd.DataFrame:
    out = df_month.copy()
    if "Temp_mean" in out:
        t_thr = out["Temp_mean"].quantile(0.75)
        out["is_hot"] = (out["Temp_mean"] >= t_thr).astype(int)
    else:
        out["is_hot"] = 0
    if "RH_mean" in out:
        h_thr = out["RH_mean"].quantile(0.75)
        out["is_humid"] = (out["RH_mean"] >= h_thr).astype(int)
    else:
        out["is_humid"] = 0

    def lbl(r):
        tags = []
        if r.get("is_hot", 0) == 1: tags.append("hot")
        if r.get("is_humid", 0) == 1: tags.append("humid")
        return "+".join(tags) if tags else "normal"

    out["wx_bin"] = out.apply(lbl, axis=1)
    return out

def load_any(path: str) -> pd.DataFrame:
    if not path or not os.path.exists(path):
        return pd.DataFrame()
    ext = os.path.splitext(path.lower())[1]
    if ext == ".csv":
        return pd.read_csv(path)
    elif ext in (".xlsx", ".xls"):
        return pd.read_excel(path)
    else:
        raise ValueError(f"Unsupported file type: {ext}")

# ---------- Core ----------
def build_oilscore(oil_path: str):
    if not os.path.exists(oil_path):
        print(f"ERROR: OIL_FILE not found:\n{oil_path}", file=sys.stderr);
sys.exit(1)

    oil = pd.read_excel(oil_path)

    # Identify required columns
    id_col   = pick_col(oil, ["turbine_id", "turbine", "id_key"])
    date_col = pick_col(oil, ["date_sample_taken", "date_tested", "date"])
    if not id_col or not date_col:
        print("ERROR: oil file must have 'turbine_id' and a date column
('date_sample_taken' or 'date_tested').",
              file=sys.stderr); sys.exit(1)
```

```
    oil.rename(columns={id_col: "turbine_id"}, inplace=True)
    oil["month"] = month_floor(oil[date_col])

    # Component columns
    fe_col   = pick_col(oil, ["Fe_mgkg", "fe", "iron"])
    pq_col   = pick_col(oil, ["PQ_index", "pq"])
    h2o_col  = pick_col(oil, ["Water_ppm", "water"])
    visc_col = pick_col(oil, ["Viscosity_40C_cSt", "viscosity_40c",
"visc40"])
    ox_col   = pick_col(oil, ["Oxidation", "oxidation"])

    comps = {}
    if fe_col:          comps["z_Fe"]   = zscore(oil[fe_col])
    if pq_col:          comps["z_PQ"]   = zscore(oil[pq_col])
    if h2o_col:         comps["z_H2O"]  = zscore(oil[h2o_col])
    if visc_col:        comps["z_VISC"] =
zscore(percent_visc_dev(oil[visc_col], nominal=320.0))
    if ox_col:          comps["z_OXID"] = zscore(oil[ox_col])

    comp_df = pd.DataFrame(comps)
    oil = pd.concat([oil, comp_df], axis=1)

    # OilScore = mean of available z-scores (skip missing)
    oil["OilScore"] = comp_df.mean(axis=1, skipna=True)

    # Monthly aggregation
    # Per-turbine: conservative MAX of OilScore in that month (if multiple
samples)
    oil_turb_month = oil.groupby(["turbine_id", "month"],
as_index=False).agg({"OilScore": "max"})
    # Farm: median across turbines per month
    oil_farm_month = oil_turb_month.groupby("month",
as_index=False).agg({"OilScore": "median"}) \
                                   .rename(columns={"OilScore":
"OilScore_farm"})

    return oil, oil_turb_month, oil_farm_month

def merge_with_monthly(oil_turb_month: pd.DataFrame, oil_farm_month:
pd.DataFrame, monthly_xls: str):
    if not monthly_xls or not os.path.exists(monthly_xls):
        return None, None, None

    xls = pd.ExcelFile(monthly_xls)
    sheets = {s.lower(): s for s in xls.sheet_names}
    if "joined" not in sheets:
        print("ERROR: Expected a 'JOINED' sheet in monthly workbook.",
file=sys.stderr); sys.exit(1)
```

```
    joined = pd.read_excel(monthly_xls, sheet_name=sheets["joined"])
    for c in ["Year", "Month"]:
        if c not in joined.columns:
            print(f"ERROR: JOINED sheet missing '{c}' column.",
file=sys.stderr); sys.exit(1)

    joined["month"] = month_from_year_month(joined["Year"], joined["Month"])

    # Farm losses (energy-weighted %)
    if {"E_act_MWh", "E_theo_MWh"}.issubset(joined.columns):
        farm = joined.groupby("month", as_index=False).agg({
            "E_act_MWh": "sum",
            "E_theo_MWh": "sum",
            "Temp_mean": "mean" if "Temp_mean" in joined.columns else
"first",
            "RH_mean": "mean" if "RH_mean" in joined.columns else "first"
        })
        farm["Loss_pct"] = ((farm["E_theo_MWh"] - farm["E_act_MWh"]) /
farm["E_theo_MWh"]) * 100.0
    else:
        if "E_loss_%" not in joined.columns:
            print("ERROR: JOINED needs E_act_MWh & E_theo_MWh or E_loss_%",
file=sys.stderr); sys.exit(1)
        farm = joined.groupby("month", as_index=False).agg({
            "E_loss_%": "mean",
            "Temp_mean": "mean" if "Temp_mean" in joined.columns else
"first",
            "RH_mean": "mean" if "RH_mean" in joined.columns else "first"
        }).rename(columns={"E_loss_%": "Loss_pct"})

    # Annotate hot/humid bins
    farm = annotate_hot_humid(farm)

    # Merge farm OilScore with farm losses+weather
    merged_farm = pd.merge(oil_farm_month, farm, on="month", how="left")

    # Per-turbine merge (if JOINED has turbine names matching oil
'turbine_id')
    t_col = pick_col(joined, ["Turbine", "turbine", "turbine_id"])
    if t_col is not None:
        joined = joined.rename(columns={t_col: "turbine_id"})
        if {"E_act_MWh", "E_theo_MWh"}.issubset(joined.columns):
            jt = joined.groupby(["turbine_id", "month"],
as_index=False).agg({
                "E_act_MWh": "sum",
                "E_theo_MWh": "sum",
                "Temp_mean": "mean" if "Temp_mean" in joined.columns else
"first",
```

```
                "RH_mean": "mean" if "RH_mean" in joined.columns else "first"
            })
            jt["Loss_pct"] = ((jt["E_theo_MWh"] - jt["E_act_MWh"]) /
jt["E_theo_MWh"]) * 100.0
        else:
            jt = joined.groupby(["turbine_id", "month"],
as_index=False).agg({
                "E_loss_%": "mean",
                "Temp_mean": "mean" if "Temp_mean" in joined.columns else
"first",
                "RH_mean": "mean" if "RH_mean" in joined.columns else "first"
            }).rename(columns={"E_loss_%": "Loss_pct"})

        jt = jt.merge(farm[["month", "is_hot", "is_humid", "wx_bin"]],
on="month", how="left")
        merged_turb = oil_turb_month.merge(jt, on=["turbine_id", "month"],
how="left")
    else:
        merged_turb = None

    return merged_farm, merged_turb, farm

# ---------- Plot helpers for replacement markers ----------
def get_replacement_months_for_farm():
    months = []
    for dates in REPLACEMENTS.values():
        for d in dates:
            ts = to_month_ts(d)
            if pd.notna(ts):
                months.append(ts)
    if not months:
        return []
    return sorted(pd.Series(months).dropna().unique())

def get_replacement_months_for_turbine(tid: str):
    ds = REPLACEMENTS.get(tid, [])
    months = [to_month_ts(d) for d in ds]
    months = [m for m in months if pd.notna(m)]
    return sorted(pd.Series(months).dropna().unique()) if months else []

def draw_replacement_lines(ax, months, color="red", linestyle="--",
alpha=0.8, lw=1.5, label="Gearbox replacement"):
    if not months:
        return
    labeled = False
    for m in months:
        ax.axvline(m, color=color, linestyle=linestyle, alpha=alpha,
linewidth=lw,
                   label=(label if not labeled else None))
```

```
        labeled = True

# ---------- Save outputs & plots ----------
def save_tables_and_plots(oil_turb_month, oil_farm_month, merged_farm,
merged_turb, farm, out_dir):
    ensure_dir(out_dir)
    ts = datetime.now().strftime("%Y%m%d_%H%M%S")

    # Save OilScore tables regardless of MONTHLY_XLS
    oil_turb_csv  = os.path.join(out_dir,
f"oilscore_monthly_by_turbine_{ts}.csv")
    oil_turb_xlsx = os.path.join(out_dir,
f"oilscore_monthly_by_turbine_{ts}.xlsx")
    oil_turb_month.to_csv(oil_turb_csv, index=False)
    with pd.ExcelWriter(oil_turb_xlsx, engine="xlsxwriter") as wr:
        oil_turb_month.to_excel(wr, sheet_name="by_turbine", index=False)

    oil_farm_csv  = os.path.join(out_dir, f"oilscore_monthly_farm_{ts}.csv")
    oil_farm_xlsx = os.path.join(out_dir, f"oilscore_monthly_farm_{ts}.xlsx")
    oil_farm_month.to_csv(oil_farm_csv, index=False)
    with pd.ExcelWriter(oil_farm_xlsx, engine="xlsxwriter") as wr:
        oil_farm_month.to_excel(wr, sheet_name="farm", index=False)

    if merged_farm is None:
        print("\nNo monthly workbook provided. Saved OilScore tables only.")
        print(f"- {oil_turb_xlsx}")
        print(f"- {oil_farm_xlsx}")
        return

    # Save merged farm
    merged_farm_csv  = os.path.join(out_dir,
f"merged_farm_oilscore_losses_weather_{ts}.csv")
    merged_farm_xlsx = os.path.join(out_dir,
f"merged_farm_oilscore_losses_weather_{ts}.xlsx")
    merged_farm.to_csv(merged_farm_csv, index=False)
    with pd.ExcelWriter(merged_farm_xlsx, engine="xlsxwriter") as wr:
        merged_farm.to_excel(wr, sheet_name="farm", index=False)

    # ===== Plots (farm level) =====
    # 1) Loss vs OilScore (color by bins)
    mf = merged_farm.dropna(subset=["OilScore_farm"]).copy()
    fig1 = plt.figure(figsize=(8,6)); ax1 = plt.gca()
    if "Loss_pct" in mf.columns and "wx_bin" in mf.columns:
        for b in ["normal", "hot", "humid", "hot+humid"]:
            mb = mf[mf["wx_bin"] == b]
            if not mb.empty:
                ax1.scatter(mb["OilScore_farm"], mb["Loss_pct"], label=b,
alpha=0.85)
        ax1.legend(title="Month bin")
```

```
        ax1.set_ylabel("Farm Loss (%)")
    else:
        if "Loss_pct" in mf.columns:
            ax1.scatter(mf["OilScore_farm"], mf["Loss_pct"], alpha=0.85)
            ax1.set_ylabel("Farm Loss (%)")
        else:
            ax1.scatter(mf["OilScore_farm"], range(len(mf)), alpha=0.85)
            ax1.set_ylabel("Index")
    ax1.set_xlabel("OilScore (farm, median across turbines)")
    ax1.set_title("Farm Loss (%) vs OilScore (annotated by hot/humid)")
    ax1.grid(True, alpha=0.3)
    fig1_path = os.path.join(out_dir, "fig_farm_loss_vs_oilscore.png")
    plt.tight_layout(); plt.savefig(fig1_path, dpi=150); plt.close(fig1)

    # 2) OilScore over time (farm)  ---- with hard-coded gearbox replacement
markers ----
    fig2 = plt.figure(figsize=(10,5)); ax2 = plt.gca()
    ax2.plot(merged_farm["month"], merged_farm["OilScore_farm"], marker="o",
linewidth=1)
    draw_replacement_lines(ax2, get_replacement_months_for_farm())
    ax2.set_xlabel("Month"); ax2.set_ylabel("OilScore (farm)")
    ax2.set_title("Farm OilScore over time (gearbox replacements marked)")
    ax2.grid(True, alpha=0.3)
    ax2.legend(loc="upper left")
    fig2_path = os.path.join(out_dir, "fig_farm_oilscore_timeseries.png")
    plt.tight_layout(); plt.savefig(fig2_path, dpi=150); plt.close(fig2)

    # 3) Boxplot OilScore by bins
    if "wx_bin" in merged_farm.columns:
        fig3 = plt.figure(figsize=(8,5))
        order = [b for b in ["normal", "hot", "humid", "hot+humid"] if b in
merged_farm["wx_bin"].unique()]
        data = [merged_farm.loc[merged_farm["wx_bin"]==b,
"OilScore_farm"].dropna() for b in order]
        if data:
            plt.boxplot(data, labels=order)
            plt.ylabel("OilScore (farm)")
            plt.title("OilScore by hot/humid bins")
            plt.grid(True, axis="y", alpha=0.3)
            fig3_path = os.path.join(out_dir,
"fig_farm_oilscore_by_bins.png")
            plt.tight_layout(); plt.savefig(fig3_path, dpi=150);
plt.close(fig3)

    # ===== Per-turbine plots & FLEET VIEW =====
    if 'merged_turb' in locals() and merged_turb is not None and not
merged_turb.empty:
        pt_dir = os.path.join(out_dir, "per_turbine_plots")
        ensure_dir(pt_dir)
```

```
        # --- Per-turbine standard plots (unchanged) ---
        for tid, g in merged_turb.sort_values("month").groupby("turbine_id"):
            g = g.copy()

            # Scatter Loss vs OilScore
            fig = plt.figure(figsize=(7,5)); ax = plt.gca()
            if "Loss_pct" in g.columns and "wx_bin" in g.columns:
                for b in ["normal", "hot", "humid", "hot+humid"]:
                    gb = g[g["wx_bin"] == b]
                    if not gb.empty:
                        ax.scatter(gb["OilScore"], gb["Loss_pct"], label=b,
alpha=0.85)
                ax.legend(title="Month bin"); ax.set_ylabel("Loss (%)")
            else:
                if "Loss_pct" in g.columns:
                    ax.scatter(g["OilScore"], g["Loss_pct"], alpha=0.85);
ax.set_ylabel("Loss (%)")
                else:
                    ax.scatter(g["OilScore"], range(len(g)), alpha=0.85);
ax.set_ylabel("Index")
            ax.set_xlabel("OilScore"); ax.set_title(f"{tid} — Loss vs
OilScore"); ax.grid(True, alpha=0.3)
            plt.tight_layout(); plt.savefig(os.path.join(pt_dir,
f"{tid}_loss_vs_oilscore.png"), dpi=130); plt.close(fig)

            # OilScore timeseries with markers
            fig = plt.figure(figsize=(7,4)); ax = plt.gca()
            ax.plot(g["month"], g["OilScore"], marker="o", linewidth=1)
            draw_replacement_lines(ax,
get_replacement_months_for_turbine(tid))
            ax.set_xlabel("Month"); ax.set_ylabel("OilScore")
            ax.set_title(f"{tid} — OilScore over time (gearbox replacements
marked)")
            ax.grid(True, alpha=0.3); ax.legend(loc="upper left")
            plt.tight_layout(); plt.savefig(os.path.join(pt_dir,
f"{tid}_oilscore_timeseries.png"), dpi=130); plt.close(fig)

            # Optional: Loss(%) timeseries with markers
            if MAKE_TURBINE_LOSS_TS and "Loss_pct" in g.columns:
                fig = plt.figure(figsize=(9,5)); ax = plt.gca()
                ax.plot(g["month"], g["Loss_pct"], linewidth=1.2)
                draw_replacement_lines(ax,
get_replacement_months_for_turbine(tid))
                if not g["month"].empty:
                    period = f"{g['month'].min().date()} →
{g['month'].max().date()}"
                else:
                    period = ""
```

```
                ax.set_xlabel("Month"); ax.set_ylabel("Loss (%)")
                ax.set_title(f"{tid} — Monthly Energy Loss (%)\nPeriod:
{period}")
                ax.grid(True, alpha=0.3); ax.set_ylim(bottom=0);
ax.legend(loc="upper left")
                plt.tight_layout(); plt.savefig(os.path.join(pt_dir,
f"{tid}_loss_timeseries.png"), dpi=130); plt.close(fig)

        # --- NEW: Fleet view of OilScore for all turbines ---
        turbs = sorted(merged_turb["turbine_id"].dropna().unique())
        n = len(turbs)
        if n > 0:
            ncols = 4
            nrows = ceil(n / ncols)
            fig = plt.figure(figsize=(4.0*ncols, 2.8*nrows))
            for i, tid in enumerate(turbs, 1):
                ax = plt.subplot(nrows, ncols, i)
                g = merged_turb[merged_turb["turbine_id"] ==
tid].sort_values("month")
                ax.plot(g["month"], g["OilScore"], linewidth=1.1)
                # replacement markers in each small panel
                draw_replacement_lines(ax,
get_replacement_months_for_turbine(tid))
                ax.set_title(tid, fontsize=10)
                ax.grid(True, alpha=0.25)
                # cleaner axes
                if i <= (nrows-1)*ncols:
                    ax.set_xticklabels([])
                else:
                    ax.tick_params(axis='x', labelsize=8, rotation=0)
                ax.tick_params(axis='y', labelsize=8)
            # single legend for replacement marker (once)
            handles, labels = ax.get_legend_handles_labels()
            if labels:
                fig.legend(handles, labels, loc="upper center", ncol=3,
frameon=False)
            fig.suptitle("Fleet View — OilScore over time (gearbox
replacements marked)", fontsize=14, y=0.98)
            fig.tight_layout(rect=[0, 0, 1, 0.96])
            fleet_path = os.path.join(out_dir,
"fig_fleet_oilscore_smallmultiples.png")
            plt.savefig(fleet_path, dpi=160); plt.close(fig)

    print("\nDone. Outputs saved in:", out_dir)

# ---------- Run ----------
if __name__ == "__main__":
    # Build OilScore
    oil_raw, oil_turb_month, oil_farm_month = build_oilscore(OIL_FILE)
```

```
    # Decide output folder: next to monthly workbook if provided, else next
to oil file
    base_out = os.path.dirname(MONTHLY_XLS) if MONTHLY_XLS else
os.path.dirname(OIL_FILE)
    if not base_out: base_out = os.getcwd()

    # Merge with monthly losses+weather (optional)
    merged_farm, merged_turb, farm_wx = merge_with_monthly(oil_turb_month,
oil_farm_month, MONTHLY_XLS)

    # Save everything and draw plots (including Fleet View)
    save_tables_and_plots(oil_turb_month, oil_farm_month, merged_farm,
merged_turb, farm_wx, base_out)
```

G EXAMPLE OF LAB REPORT FOR OIL ANALYSIS

LAB REPORT

OELCHECK since 1991

+49 8034-9047-210

Unit ID 80914

Component Wind turbine - main gear

Current sample number 4053398

OELCHECK GmbH · Kerschelweg 28 · 83098 Brannenburg

Nordex Energy GmbH
Erich-Schlesinger-Straße 50
18059 Rostock

Machine name: **N90**
Gear manufacturer: **Bosch**
Serial number: **1368941**
Oil brand name: **Mobil Mobilgear XMP 320**
Oil quantity in system: **525 l**

Region: PT
Windpark: SERRA DA LOUSA2
Service technician: C.Serpa
Sample related to: SERRA DA LOUSA2

Diagnosis for the current laboratory values
The wear values are in the normal range. The cleanliness class of the oil complies with the requirements. Viscosity and additive levels are in the normal range. I recommend that you send the next sample at the next service interval or at your regular inspection for trend analysis.
Dipl.-Ing. Andy Böhme (MLA II + CLS)

Sample Rating

normal

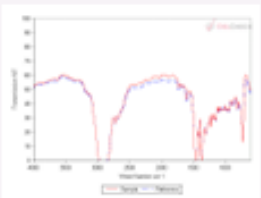
ANALYSIS RESULTS			Current sample	17 previous samples not shown		
LAB NUMBER			4053398	3909552	3750075	3333978
SAMPLE RATING			✓	!	✓	✓
Date tested			20.01.2020	28.03.2019	18.10.2018	18.06.2018
Date of sample taken			10.01.2020	14.03.2019	01.10.2018	11.06.2018
Date of last oil change			16.04.2018	16.04.2018	16.04.2018	16.04.2018
Top-up since change			-	-	20	-
Operating time since change		h	11830	6685	3022	1054
Total operating time		h	74620	68794	65950	63825
Oil changed			no	-	no	yes
WEAR						
Iron	Fe	mg/kg	44	32	21	12
Chrome	Cr	mg/kg	0	0	0	0
Tin	Sn	mg/kg	0	0	0	0
Aluminum	Al	mg/kg	0	0	0	0
Nickel	Ni	mg/kg	0	0	0	0
Copper	Cu	mg/kg	0	0	0	0
Lead	Pb	mg/kg	0	0	0	0
Manganese	Mn	mg/kg	1	0	0	0
PQ index	-		< 25	< 25	< 25	< 25
CONTAMINATION						
Silicon	Si	mg/kg	0	1	1	1
Potassium	K	mg/kg	0	0	1	0
Sodium	Na	mg/kg	0	1	0	0
Silver	Ag	mg/kg	-	-	1	-
Water K. F.	ppm		73	124	142	153
OIL CONDITION						
Viscosity at 40°C	mm²/s		323.98	319.75	322.09	322.21
Viscosity at 100°C	mm²/s		24.10	23.98	23.90	24.04
Viscosity index	-		95	95	94	95
Oxidation	A/cm		2	1	1	1
IR index	-		99.75	99.94	99.94	99.96
ADDITIVES						
Calcium	Ca	mg/kg	1	1	2	2
Magnesium	Mg	mg/kg	0	0	1	0
Boron	B	mg/kg	0	0	0	0
Zinc	Zn	mg/kg	8	8	7	8
Phosphorus	P	mg/kg	202	209	210	199
Barium	Ba	mg/kg	0	0	0	0
Molybdenum	Mo	mg/kg	0	1	7	2
Sulphur	S	% Wt.	1.19	1.12	1.18	1.06

Additional sample details

Total production: **66667000 kWh**
Production since oil change: **10613000 kWh**
Sampling point: **Drain valve**

Bottle and Cap

Infrared Spectrum



LAB REPORT

OELCHECK since 1991

+49 8034-9047-210

Unit ID 80914

Component Wind turbine - main gear

Current sample number 4053398

Nordex Energy GmbH
Erich-Schlesinger-Straße 50
18059 Rostock

Machine name:	**N90**
Gear manufacturer:	**Bosch**
Serial number:	**1368941**
Oil brand name:	**Mobil Mobilgear XMP 320**
Oil quantity in system:	**525 l**

Region: PT
Windpark: SERRA DA LOUSA2
Service technician: C.Serpa
Sample related to: SERRA DA LOUSA2

ANALYSIS RESULTS		Current sample	17 previous samples not shown		
LAB NUMBER		4053398	3909552	3750075	3333978
SAMPLE RATING		✓	!	✓	✓
Date tested		20.01.2020	28.03.2019	18.10.2018	18.06.2018
Date of sample taken		10.01.2020	14.03.2019	01.10.2018	11.06.2018
Date of last oil change		16.04.2018	16.04.2018	16.04.2018	16.04.2018
Top-up since change		-	-	20	-
Operating time since change	h	11830	6685	3022	1054
Total operating time	h	74620	68794	65950	63825
Oil changed		no	-	no	yes
ADDITIONAL TESTS					
AN / NN	mgKOH/g	0.79	0.68	0.72	0.72
Cleanliness class	ISO 4406	21/18/12	24/21/13	23/19/13	21/19/12
A: >4µm = ISO >4µm	Particles/100ml	1142119	9576218	4153801	1419066
B: >6µm = ISO >6µm	Particles/100ml	152704	1142884	413063	264396
C: >14µm = ISO >14µm	Particles/100ml	3161	5339	6832	2337
D: >21µm	Particles/100ml	800	801	1906	406
E: >38µm	Particles/100ml	34	20	80	0
F: >70µm	Particles/100ml	11	20	0	0
Cleanliness class	SAE AS 4059	11A	> 12A	> 12A	11A