The application of Normal Behavior Model (NBM) condition monitoring for optimal wind farm maintenance

Author:
Ming-Chen Kao
DTU Wind-M-959
June 2025

Author:
Ming-Chen Kao

DTU Wind-M-0959
June 2025

Title:
The application of Normal Behavior Model (NBM) condition monitoring for optimal wind farm maintenance

ECTS: 30

Education: Master of Science

Supervisors:
Michael Kenneth McWilliam
Julian Antony Quick
DTU Wind & Energy Systems

Remarks:
This report is submitted as partial fulfillment of the requirements for graduation in the above education at the Technical University of Denmark.

DTU Wind & Energy Systems is a department of the Technical University of Denmark with a unique integration of research, education, innovation and public/private sector consulting in the field of wind energy. Our activities develop new opportunities and technology for the global and Danish exploitation of wind energy. Research focuses on key technical-scientific fields, which are central for the development, innovation and use of wind energy and provides the basis for advanced education at the education.

Technical University of Denmark Department of Wind and Energy Systems Frederiksborgvej 399 4000 Roskilde Denmark
www.wind.dtu.dk

Abstract

With wind energy capacity exceeding 1000 GW globally, operation and maintenance costs constitute 25-30% of total life cycle expenses, driving demand for intelligent predictive maintenance strategies. This research investigates the application of normal behavior models (NBMs) for early-stage bearing fault detection in wind turbines to optimize maintenance costs and improve system reliability.

This study develops and evaluates machine learning-based condition monitoring systems combining Long Short-Term Memory (LSTM) networks and XGBoost models with CUSUM anomaly detection for wind turbine bearing temperature prediction. Using real operational data from five wind turbines, systematic optimization was performed across model architectures, feature engineering approaches, and detection parameters to identify optimal configurations for generator and gearbox bearing monitoring.

The research achieved effective early-stage fault detection performance through component-specific optimization, with XGBoost consistently outperforming LSTM by 10-15% in prediction accuracy. Optimized CUSUM parameters achieved substantial false alarm reduction—reducing false alarms to six events for generator bearings and three for gearbox bearings—while maintaining reliable detection capabilities up to 60 days before component failure.

The study demonstrates that properly configured NBM systems can effectively detect early-stage bearing faults while reducing maintenance costs through minimized false alarms. However, successful deployment requires component-specific optimization, systematic bias handling, and careful preprocessing selection due to fundamental challenges including turbine baseline differences and cross-component interference. This research provides systematic evaluation frameworks and bias correction methodologies that offer practical tools and critical insights for NBM system design in predictive maintenance applications.

Acknowledgements

First and foremost, I would like to thank my supervisors, Michael Kenneth McWilliam and Julian Antony Quick, for the guidance of this research and being such wonderful supervisors who always support me through the challenges. This work would not be completed without your help and insightful advices.

I would also like to thank my family, especially my mother. Thank you for all the support you've given me in completing this degree.

Finally, I appreciate the assistance of AI-based tools in refining the wording of this thesis.

Contents

1 Introduction

Renewable energy has grown significantly in recent decades. The goal of combating climate change urges massive development in the wind energy sector. Wind energy deployment has expanded worldwide, with global installed capacity reaching 1046 GW by 2023 [**2024WWEA2023**]. This growing trajectory is expected to continue in the coming years, driven by new policies which have started to take effect in many countries around the world in response to the escalating climate crisis and fossil fuel crises [**2024WWEA2023**].

As wind farms proliferate and age, maintenance has emerged as a critical factor affecting both economic viability and energy production reliability. Operation and Maintenance (O&M) costs typically constitute 25-30% of the total life cycle costs of offshore wind farms [**Rockmann2017OperationSea**], highlighting the economic impact of the O&M strategy. Thus, the need for intelligent maintenance optimization becomes critical.

The evolution from traditional maintenance approaches toward more sophisticated data-driven strategies has been enabled by advances in sensor technology, data analytics, and machine learning. Modern wind turbines are equipped with extensive sensor networks that continuously monitor critical components including bearings, gearboxes, generators, and control systems. This wealth of operational data presents an opportunity to develop predictive maintenance systems that can identify early signs of component degradation and optimize maintenance scheduling to reduce costs while maintaining reliability.

Normal Behavior Models (NBMs) represent a promising approach to this challenge by learning the typical operational patterns of healthy wind turbine components and detecting deviations that may indicate emerging failures. Unlike traditional threshold-based monitoring systems that rely on fixed alarm limits, NBMs can adapt to the unique operational characteristics of individual turbines and environmental conditions. By establishing baseline models of normal component behavior using machine learning techniques, these systems can identify subtle changes in operational patterns that precede component failures, potentially providing advanced warning for maintenance planning.

This research addresses the fundamental question: How can normal behavior models effectively detect early-stage faults in wind turbine bearings while reducing maintenance costs? Specifically, this study investigates the application of machine learning-based normal behavior models combined with CUSUM anomaly detection for wind turbine bearing condition monitoring, with a focus on optimizing both detection performance and practical deployment considerations.

The primary objectives of this research are to: (1) develop and evaluate machine learning approaches for modeling normal bearing behavior in wind turbines, comparing model architectures across different temporal configurations and feature sets; (2) systematically optimize CUSUM parameters for bearing failure detection to minimize false alarms while maintaining early detection capability; (3) investigate the impact of preprocessing strategies, feature engineering choices, and sensor selection on detection performance and practical deployment requirements; and (4) assess the trade-offs between prediction accuracy and anomaly detection sensitivity to identify optimal configurations for maintenance cost reduction.

The structure of this thesis is organized as follows: Chapter 2 reviews relevant literature on maintenance strategy, normal behavior modeling, and wind turbine condition monitor-

ing approaches; Chapter 3 describes the experimental methodology, dataset characteristics, and evaluation framework; Chapter 4 presents comprehensive results from cross-validation analysis, systematic optimization studies, and practical deployment considerations; Chapter 5 discusses the implications of key findings, implementation challenges, and future research directions; and Chapter 6 provides conclusions and recommendations for NBM-based wind turbine maintenance optimization.

2 Background

2.1 Maintenance Approaches

Wind farm maintenance strategies, scheduling, and asset management are interconnected topics that have been extensively studied in the past decades. As wind farms increase in size while moving to more challenging environments, the maintenance approach has transitioned from a simple reactive strategy to a more sophisticated approach. A comprehensive study on the classification of maintenance policies is provided in [**WangASystems**]. However, generally, a simple classification can put them into three main groups:

- Corrective maintenance
- Periodic maintenance
- Predictive (Condition-based) maintenance

Corrective maintenance, also known as run-to-failure or reactive maintenance, involves repairing or replacing components only after they have failed. Although seemingly cost-effective due to the absence of preventive inspection costs, this strategy often leads to extended downtime, especially for offshore wind farms where weather conditions and maintenance vessel availability can significantly delay maintenance schedules. In addition, the reactive approach is particularly problematic for critical components such as gearboxes and generators, where failures typically result in downtimes of around one week per failure [**Dao2019WindEnergy**]. This extended downtime can cause substantial financial losses, as wind turbines increase in size and power capacity. Despite these drawbacks, corrective maintenance is unavoidable in many scenarios, especially when failure due to (i) unpredictable natural events such as storms, icing, or earthquakes; (ii) condition monitoring systems fail to produce an alarm for developing faults; (iii) economic considerations that favor reactive approaches for non-critical components. According to Nachimuthu et al. [**Nachimuthu2019AUncertainties**], natural events alone account for approximately 60% of offshore turbine failures, making corrective maintenance an essential part of any comprehensive maintenance strategy.

Periodic maintenance (PM) strategies implement scheduled interventions based on elapsed time at predefined intervals. This approach aims to replace components before they fail, thus avoiding unexpected downtime and emergency repairs. In general, components' failure behavior are modeled using Weibull distributions, which provide flexibility in representing various failure patterns through their shape and scale parameters. Wind farm operators schedule maintenance according to the reliability level, balancing the risk of failure against maintenance cost. However, frequent PM interventions can significantly increase unnecessary repair costs. The challenge lies in determining the optimal maintenance frequency that minimizes total costs. The study by Su et al. [**Su2021OptimizationModel**] found that approximately 70% of onshore wind farms primarily rely on periodic maintenance strategies, with maintenance intervals averaging 3-4 months for critical components such as gearboxes and generators.

Both corrective and periodic maintenance can be regarded as traditional maintenance approaches. A combination of periodic and reactive maintenance can improve the reliability, availability, and maintainability of wind turbines while simultaneously reducing the maintenance cost [**Tchakoua2014WindChallenges**]. Usually, corrective strategies are applied for assets with low criticality, and periodic strategies for assets with a well-known and

consistent failure–time correlation. For the most critical assets, condition-based strategies are adopted [**RinaldiCurrentReview**].

The condition-based maintenance refers to the process of using condition monitoring (CM) techniques to identify the current state of components. This approach helps asset managers determine when maintenance is actually needed based on the equipment's condition rather than fixed time intervals, avoiding unnecessary repairs on healthy components. Within the broader category in [**RinaldiCurrentReview**], a more detailed classification distinguishes between predictive and proactive maintenance approaches. Predictive maintenance is the data-driven approach utilizing condition monitoring data combined with analytical models. It analyzes historical and real-time data patterns to forecast remaining useful life, allowing maintenance to be scheduled just before failure is likely to occur.

Proactive maintenance goes a step further by identifying and addressing the root causes of potential failures. This approach aims at continuous monitoring to detect the early signs of failure [**Fox2022AMaintenance**]. Both approaches represent the implementations of condition-based principles, but they differ in their intervention strategies.

2.2 Overview of Condition Monitoring Technologies

Over the past decades, various monitoring technologies have been developed and implemented in wind turbines to detect and diagnose different types of faults. Several non-destructive testing (NDT) techniques have been deployed to monitor both operational and environmental parameters [**HameedConditionReview**]. These technologies include but are not limited to:

- Vibration analysis
- Oil analysis
- Thermography
- Acoustic monitoring
- Strain measurement
- Ultrasonic testing
- Visual inspection

Vibration Analysis

Vibration analysis remains one of the most established condition monitoring techniques, especially for rotating machinery such as the gearbox, generator, and main shaft bearing system [**HameedConditionReview**]. This method typically involves installing accelerometers and piezoelectric or micro electromechanical systems to capture frequency spectra that can reveal characteristic fault signatures [**RinaldiCurrentReview**]. The review by Teng [**TengVibrationInvestigation**] indicates that vibration analysis performs superiorly to other monitoring techniques in fault location and hardware cost. Future trends include the standardization of vibration data acquisition systems and the improvement of communication for massive vibration data.

Oil analysis

Oil analysis monitors the lubrication and hydraulic oil status of mechanical components, such as gearboxes. This technique involves analyzing the lubricant samples to detect contaminants and degradation that indicate the health of the components [**CoronadoMonitoringMethods**]. The implementation of oil analysis in wind turbine maintenance can be periodic offline

sampling to the online monitoring systems. Online oil condition monitoring systems utilizing sensors that measure parameters such as dielectric constant, conductivity, and particle count provide real-time insights into gearbox health [**Coronado2018MonitoringMethods**].

Thermography
Thermography is a condition monitoring technique that measures the temperature difference on the surface of the material, which can indicate areas of damage or defects. This non-contact method uses infrared cameras to detect thermal patterns, making it particularly valuable for monitoring electrical components and mechanical systems in wind turbines [**Yang2013TestingSurvey**]. Thermographic inspection can be classified into passive and active thermography. The former allows thermal changes of a material without an external thermal source to be observed, whereas the latter requires an external thermal excitation source [**Kong2023ProgressReview**]. Aminzadeh's study [**Aminzadeh2023Non-Contact4.0**] found that thermography is particularly effective for detecting subsurface defects such as delamination, debonding, and moisture ingress without requiring turbine disassembly. When deployed via drone-mounted systems, thermographic inspection can reduce maintenance costs by up to 30% compared to traditional rope-access methods [**Aminzadeh2023Non-Contact4.0**].

Acoustic Monitoring
Acoustic Emission monitoring detects transient elastic waves generated by the rapid release of energy from localized sources within materials. In wind turbine condition monitoring, acoustic monitoring technology detects incipient damage in various components, such as blades, bearings, and gearboxes [**Ding2023Acoustic-Signal-BasedReview**][**Chacon2016AnD**
For wind turbine applications, acoustic emission monitoring can detect defects like matrix cracking, delamination, fiber breakage in composite blades, and bearing faults in rotating components [**Ding2023Acoustic-Signal-BasedReview**]. Unlike vibration analysis, acoustic monitoring technology is relatively effective for monitoring low-speed components such as main bearings in direct-drive wind turbines, where traditional vibration techniques often struggle to detect early-stage failures [**Ma2023AnEmission**].

Strain Measurement
Strain measurement technique quantifies material deformation under applied forces. Fiber Bragg Grating sensors and interferometric optical sensors have largely replaced traditional strain gauges due to their superior durability and electromagnetic immunity in the harsh wind turbine environment [**Weijtjens2017VibrationTurbines**]. These strain sensors are typically positioned at critical, locations such as blade roots, tower sections, and foundation interfaces to monitor structural integrity [**He2019StructuralSensors**]. The measurement from these sensors enables load monitoring and fatigue life assessment, with studies demonstrating that blade root strain data can be used to derive comprehensive force and moment profiles across the entire blade structure [**Moynihan2022EstimationVerification**]

Ultrasonic Testing
Ultrasonic testing detects faults using high-frequency sound energy transmitted through the material and received on the opposite surface. This technique is extensively used for investigating inner structure damage, such as delamination and debonding for blades [**GarciaMarquez2012ConditionMethods**]. The reflected waves from internal defects create unique patterns in amplitude, frequency, and time-of-flight that can be used to characterize damage type, size, and location of the damage [**Kong2023ProgressReview**]. Conventional ultrasonic systems typically operate at frequencies between 1-50 MHz, with higher frequencies offering better resolution but less penetration depth [**Du2020DamageReview**].

Visual Inspection
In addition to the above NDT technologies, visual inspections remain fundamental to wind turbine condition monitoring despite their relative simplicity. Visual inspection methods

have evolved significantly with the integration of advanced imaging technologies and autonomous platforms. Remote visual inspection using high-resolution cameras mounted on drones has improved wind turbine blade assessment by providing detailed imagery of surface defects such as cracks, erosion, and lightning damage [**Memari2024ReviewDetection**]. Besides, these inspections can be performed periodically or in response to alarm signals from other monitoring systems, offering a cost-effective first-level assessment [**Shihavuddin2019WindAnalysi**. Multi-modal imaging approaches combining RGB cameras with infrared thermography have further improved defect detection capabilities. This fusion enables the identification of both surface and subsurface defects, as thermal imaging can reveal internal structural issues not visible to standard cameras [**Zhou2023WindFusion**].

2.3 SCADA-based Monitoring

Although various condition monitoring techniques mentioned in section 2.2 have been deployed in wind farm maintenance for a long time, not all of these techniques are widely applied in wind farms. The main reason is that installing additional equipment and sensors could significantly increase operational costs [**Liu2022AData**]. Recent developments in computational capabilities have opened opportunities for integrated and in-depth condition monitoring analytics where different types of data can be used to facilitate informed, reliable, cost-effective, and robust decision-making [**Stetco2019MachineReview**].

Supervisory Control and Data Acquisition (SCADA) systems, originally designed for turbine control and performance monitoring, have become valuable sources of condition monitoring data. Unlike condition monitoring equipment, SCADA systems are standard in modern wind turbines and collect numerous operational parameters, typically at intervals of 10 minutes. These systems typically monitor [**Liu2022AData**]:

- Environmental parameters such as wind speed/direction and ambient temperature
- Electrical parameters such as phase voltage/current and power output
- Temperature of key components such as bearing and winding of the generator
- Control variables such as pitch angle and converter torque

The use of these SCADA data for condition monitoring presents a cost-effective approach since it requires no installation of additional equipment. However, challenges for SCADA-based monitoring remains in (i) the low sampling rate, where 5-10 minutes sampling rate is too slow for most rotating machine fault diagnoses. (ii) time-varying working conditions, where the wide range over operational conditions make it difficult to detect an incipient fault. (iii) lack of historical fault data due to commercial confidentiality [**Tchakoua2014WindChallenges**, **Tautz-Weinert2017UsingReview**, **Pandit2023SCADATrends**].

Despite the challenges, researchers have developed various methods to utilize SCADA data for condition monitoring and fault diagnosis. For instance, Liu et al. [**Liu2022AData**] demonstrated covariate-adjusted preprocessing to account for various working conditions for fault isolation. Chesterman et al. [**Chesterman2023OverviewFarms**] provided an overview of NBMs, which models are trained on healthy turbine data to detect deviations indicating potential failures. Astolfi et al. [**Astolfi2022DiscussionAnalysis**] demonstrated the use of SCADA data to detect performance anomalies in wind turbines by analyzing pitch system measurements.

More specifically, a classification of methods using SCADA data can be found in Black et al. [**Black2021ConditionManagement**], where distinction is made between (1) trending,

(2) clustering, (3) NBMs, (4) damage modeling, (5) alarm assessment, and (6) performance monitoring. However, no particular method has yet been established as being optimal, with Maldonado-Correa et al. demonstrating that the field continues to explore multiple approaches without clear consensus on a superior technique [**Maldonado-Correa2020UsingReview**

2.4 Normal Behavior Model

The current methods for detecting anomalies based on SCADA data can be divided into two main categories, model-based and data-driven approaches. Model-based methods rely on explicit knowledge for the physical and mathematical representations of the components of the wind turbine, which are not often available. In contrast, with the development of artificial intelligence and big data analysis technologies, data-driven methods to explore hidden information in SCADA data have become feasible [**Zhang2022AnomalyXGBoost**].

NBMs operate on the principle of establishing a reference model that represents the expected normal operating behavior of the turbine or its components. The models are trained on healthy data to represent the normal state. Subsequently, deviations between model output and the measured sensor values can be processed and evaluated to identify anomalies. The typical workflow of NBMs is demonstrated in Figure 3.1.

2.4.1 Preprocessing Techniques

Data preprocessing is a critical step in developing effective NBMs, as SCADA data often contains inconsistencies and noise that can impact model performance. The preprocessing stage typically involves techniques such as:

Data Cleaning

The SCADA data in actual wind farm usually faces the problem of missing values and outliers due to sensor malfunctions, maintenance activities, and communication errors. Therefore, one simple method is to omit the missing data from the model and continue the analysis with the available data. However, this might lead to erroneous or biased results if the errors are not ”missing completely at random” [**Rinaldi2021CurrentReview**].

In contrast, Tawn et al. [**Tawn2020MissingForecasts**] conducted a comprehensive analysis of missing data patterns in wind farm time series and found that wind power data is typically ”Missing Not At Random”, where the probability of missing data is related to the value it would have taken. This study demonstrated that simply omitting rows with missing values degrades forecast performance, with up to 19% increase in error rates. Instead, they recommend multiple imputation techniques, such as using a statistical forest model to handle missing training data. This technique almost mitigated the negative impact on the model performance. Additionally, the large gap in the time series can be a problem since the imputation can become meaningless. This can result in the pollution of the relation between multiple signals. On the other hand, if the number of missing data is fairly limited, an aggregate like the mean and median can be used as a proxy [**Chesterman2023OverviewFarms**].

Another factor influencing NBM training is outliers. In most of the work, outliers are simply removed, while a few references treated outliers as missing values and tried to recover them [**Zou2020AMeasurements**]. The detection of outliers can be done using techniques such as the interquartile range method [**Campoverde2022SCADAAnalysis**], a fleet-based normalization approach [**ChestermanConditionModels**], or the 5σ rule [**Miele2022DeepSeries**]. The challenge lies in distinguishing between the true outliers that should be removed from analysis and the false outliers that appear anomalous but represent valid operational states that should be retained. This balance requires a good

understanding of wind turbine operations and the careful consideration of the specific monitoring objectives.

Dimensionality Reduction
SCADA systems typically record numerous parameters, many of which may be redundant or irrelevant for specific monitoring tasks. If the variables are less relevant to the condition of the wind turbine or have a low correlation between each other, the deep learning model will learn indifferent features, which will lead to underfitting and worsen the performance of fault detection [**Kong2020ConditionUnits**]. Secondly, including all available SCADA parameters could lead to overfitting and false alarms, while a carefully selected subset provided more reliable fault indicators [**Wang2017WindNetworks**]. Last but not least, dimensionality reduction could facilitate interpretability. Since models with fewer features reduce complexity, this allows domain experts to more easily understand the relationships between input variables and model outputs [**MolnarInterpretableExplainable**].

Therefore, several approaches have been developed to address the dimensionality challenge in wind turbine condition monitoring. According to [**Stetco2019MachineReview**, **Chesterman2023OverviewFarms**], these methods can be broadly categorized into

- **Domain-Based Methods:**
  - Expert knowledge/physics-informed selection
- **Statistical Methods:**
  - Correlation-based methods (Pearson, etc.)
  - Variance-based methods (PCA, etc.)
- **Machine Learning-Based Methods:**
  - Filter methods (evaluate features independently)
  - Wrapper methods (evaluate feature subsets using the target algorithm)
  - Hybrid approaches (combining multiple techniques)

Some papers focus on selecting a small subset of features based on expert knowledge. The advantage of this method is that the number of signals used for training is limited, which reduces the computational burden of the training process. The disadvantage is however that for cases in which the expert knowledge is incomplete, important signals might be missed [**Chesterman2023OverviewFarms**].

Correlation-based techniques identify variables with strong relationships to key operational parameters. Kong et al. [**Kong2020ConditionUnits**] employed Pearson correlation coefficient analysis to select features for wind turbine condition monitoring based on their correlation with critical performance indicators. This approach helps identify features with linear relationships to target variables but may miss non-linear dependencies.

Variance-based methods leverage mathematical transformations to reduce dimensionality. For example, Wang et al. [**Wang2018WindSelection**] used Principal Component Analysis (PCA) to transform the original feature space into a lower-dimensional representation capturing maximum variance. Similarly, Fu et al. [**Fu2019ConditionModel**] applied adaptive elastic networks that combined feature selection with regularization.

Filter-based methods evaluate features independently based on their intrinsic properties and the relationship with the target variable. Commonly used evaluation criteria include the distance measure, information measure, and correlation measure, as detailed by

Liu [**Liu2023AGearbox**]. A distance measure includes Analysis of Variance (ANOVA), which selects features based on their F-statistic values between different class groups [**Ashkarkalaei2025OptimumBlade**]. Other method, such as Relief-based algorithms operates by calculating weights for each feature to represent their correlation with the target classes [**Wu2020FaultBoosting**].

To detect non-linear relationships, wrapper methods evaluate feature subsets by training models with different combinations of features and selecting the subset that maximizes performance. Unlike filter methods, wrappers incorporate the learning algorithm as part of the feature selection process. For example, Fahim et al. [**Fahim2022AnPrediction**] implemented a genetic algorithm-based wrapper approach for wind turbine power prediction, demonstrating how wrapper methods can reduce dimensionality while preserving predictive power. Sa et al. [**20202020IWSSIP**] implemented a more sophisticated approach using the Non-dominated Sorting Genetic Algorithm II (NSGA-II) to simultaneously optimize feature selection and hyperparameter tuning.

Furthermore, some researchers have developed hybrid approaches combining the efficiency of filters with the effectiveness of wrappers. These methods typically employ a two-stage process where filter methods are used for initial feature screening to reduce dimensionality, followed by wrapper methods for fine-tuning the selection. Li et al. [**Li2021AnMonitoring**] proposed a Feature Simplification Random Forest (FS_RF) algorithm that first removes features with low weights using Euclidean distances (filter stage) and then applies random forest to measure feature importance (wrapper stage). Dhibi et al. [**Dhibi2023ASystems**] developed a fault diagnosis system for wind energy conversion systems. Their approach combines Neighborhood Component Analysis (NCA) as a filter method with the Equilibrium Optimizer (EO) as a wrapper method, followed by a Random Forest classifier.

Beyond traditional filter, wrapper, and hybrid approaches, deep learning-based methods have emerged as alternatives for feature learning and dimensionality reduction in wind turbine condition monitoring. Autoencoders represent an advanced approach where feature extraction and dimensionality reduction occur simultaneously through manifold learning. For example, Gbashi et al. [**Gbashi2024ExploringGearbox**] proposed an autoencoder framework for wind turbine gearbox fault diagnosis that learns latent space representations directly from raw vibration signals. This approach eliminates the need for manual feature engineering while preserving complex non-linear relationships in the data. Similarly, Zhang et al. [**Zhang2022AnomalyXGBoost**] employed long short-term memory-based stacked denoising autoencoders for anomaly detection in wind turbines, enabling the detection of deviations in operational patterns. These advanced approaches suggest a paradigm shift from explicit feature selection to implicit feature learning, potentially offering more robust solutions to capture the signature in high-dimensional data.

Normalization and Scaling

SCADA data collected from wind turbine systems typically contain features that vary significantly in scale and units. These disparate scales can adversely affect the performance of machine learning algorithms, particularly those that use distance metrics or gradient-based optimization. Normalization is therefore a critical preprocessing step for wind turbine condition monitoring.

Several normalization techniques are commonly applied. For example, min-max scaling [**Zhang2022AnomalyXGBoost**, **Miele2022DeepSeries**], and z-score standardization [**Udo2021Data-DrivenData**] methods help transform features to comparable ranges, improving model convergence and performance. In addition, Chesterman et. al [**Chesterman2023Overv**

proposed a fleet median normalization approach where the median value of a signal across all turbines in a wind farm is subtracted from individual turbine signals. This technique removes farm-wide effects like seasonal variations and weather conditions, isolating turbine-specific behavior that may indicate faults.

Liu et al. [**Liu2022AData**] introduced a covariate-adjusted preprocessing approach that addresses the time-varying working conditions of wind turbines. Their method employs spline-based nonparametric regression to eliminate the effects of operational covariates on temperature measurements. Zhang et al. [**Zhang2025StatisticalConditions**] also proposed an amplitude normalization strategy for wind turbine generator bearing condition monitoring under various speed conditions.

Finally, improper normalization can lead to increased false alarms as normal operational variations are misinterpreted as fault conditions [**Liu2022AData**]. Therefore, normalization is a critical element of wind turbine condition monitoring systems.

Lag Feature Implementation
When modeling NBM in SCADA systems, incorporating temporal dependencies have demonstrated improving the modeling performance due to the sequential nature of operational data [**Chesterman2023OverviewFarms**]. The implementation approach typically involves creating time-shifted versions of input variables to capture the system's dynamic behavior across multiple time steps. For a given input variable X, a set of lagged features can be generated by $\{X_t, X_{t-1}, X_{t-2}, ..., X_{t-n}\}$, where n represents the lag order. This allows the model to learn patterns not only from current measurements but also from the recent history of the system. For example, Dimitrov and Göçmen [**Dimitrov2022VirtualModels**] demonstrated that for wind turbine applications, including lagged inputs in feedforward neural networks can achieve similar or sometimes superior performance compared to LSTM networks.

Data Splitting
The first step in building an NBM is to split the data into training and testing datasets. The splitting is done prior to other preprocessing steps to avoid information leakage. Unlike random splitting, which is common in many machine learning applications, wind turbine data often requires splitting the data in a temporal sequence due to the lag step inclusion [**Chesterman2023OverviewFarms**]. Common splitting ratios include 80%-20% or 70%-30%. Thus, the model is trained on historical data and tested on more recent data, reflecting the real-world scenario where models are trained on past data to predict future behavior.

Several considerations should be made when performing this temporal split. First, the training data should represent the normal operational conditions that the turbine may experience, including seasonal variations and different wind regimes [**Jin2021ConditionAnalysis**]. In this case, at least one year of data should be used for training to capture seasonal effects adequately. Second, when dealing with multi-turbine datasets, cross-turbine validation approaches may be employed, where models are trained on data from specific turbines and tested on others to evaluate model applicability across different turbines within a wind farm [**Meyer2021Multi-targetMonitoring**]. This could be beneficial when developing fleet-wide monitoring systems, as it assesses whether patterns learned from one turbine can be applied to others of the same type. Lastly, the splitting strategy needs to account for imbalanced data distribution, as strategically selecting split points in time-dependent SCADA data has been demonstrated to significantly enhance model performance in fault detection tasks [**Velandia-Cardenas2021WindData**].

In conclusion, data splitting strategies for wind turbine NBMs should be carefully designed

with consideration for temporal dependencies, seasonal variations, and the monitoring objectives. A proper temporal split ensures that the model evaluation reflects the actual deployment conditions where models must generalize to future, unseen operational data.

2.4.2 Normal Behavior Modeling Algorithms

Among the diverse ecosystem of algorithms used for NBM in wind turbine condition monitoring, these algorithms can be broadly categorized into: (1) statistical models, (2) traditional machine learning models, and (3) deep learning models [**Chesterman2023OverviewFarms**]. While these categories provide a structured way to classify different methodologies, many studies combine techniques from multiple categories to enhance predictive performance and robustness.

Statistical models

Statistical techniques represent the simplest yet foundational approaches to NBM in wind turbine monitoring. These models establish mathematical relationships between input variables and target signals through statistical principles. Linear regression and Partial Least Squares (PLS) regression are two commonly used methods in this category. For example, Wang et al. [**Hao2021AAnalysis**] demonstrated the effectiveness of PLS regression for wind turbine gearbox fault prediction by analyzing relationships between oil parameters (such as abrasive particle concentration, moisture, viscosity) and gearbox wear state. The advantage of statistical models is that they offer high interpretability, computational efficiency, and require relatively less data compared to more complex approaches. Their transparent nature allows engineers to understand the relationships between variables, making them suitable for initial diagnostics and establishing baseline performance metrics. However, these models are limited in capturing complex, non-linear relationships. Additionally, these models assume static relationships within features, which may not hold under varying operational conditions. Whether this limitation is problematic depends on the case being addressed, but in general, turbine behavior involves complex and non-linear dynamics, often requiring more sophisticated modeling algorithms.

Traditional machine learning models

Traditional machine learning approaches offer greater flexibility than statistical models, particularly in capturing non-linear relationships in SCADA data. These models can learn patterns from historical data and generalize them for fault detection and predictive maintenance. Commonly used machine learning algorithms in normal behavior modeling (NBM) include:

- Support Vector Machines (SVM)
- K-Nearest Neighbors (k-NN)
- Random Forests (RF)
- Gradient boosting trees

For instance, Leahy et al. [**Leahy2018DiagnosingMachines**] demonstrated that SVMs can effectively predict wind turbine faults using SCADA data. Similarly, Random Forest algorithms have been shown to handle high-dimensional SCADA data while maintaining accuracy even with noisy measurements [**Li2021AnMonitoring**, **Dhibi2023ASystems**]. Other machine learning techniques, such as [**Tang2023ApplicationsDamage**] and gradient boosting trees [**Fahim2022AnPrediction**], have also been successfully applied to wind turbine condition monitoring, demonstrating strong performance in detecting abnormal behavior patterns. The choice between these algorithms typically depends on the monitoring task, available computational resources, size of the dataset, and interpretability requirements. While SVMs and k-NN might be preferable for smaller datasets where

interpretability is less critical [**Stetco2019MachineReview**], ensemble methods like Random Forests and gradient boosting trees generally provide higher accuracy for large-scale wind farm monitoring applications [**Black2021ConditionManagement**]. SVMs have the advantage of finding global minima during optimization and have intuitive graphical interpretation, but their training on large datasets remains challenging with time complexity [**Stetco2019MachineReview**]. Conversely, neural networks and tree-based methods have been shown to achieve better results with larger volumes of operational data [**Black2021ConditionManagement**].

Deep learning models

In recent years, deep learning has emerged as a popular and powerful paradigm for wind turbine condition monitoring, particularly for handling complex, high-dimensional SCADA data. Unlike traditional machine learning approaches, deep neural networks can extract hierarchical representations hidden from raw data, enabling them even better to capture subtle patterns and non-linear relationships. Several deep learning architectures have demonstrated promising results in wind turbine applications, such as:

- Artificial Neural Networks (ANNs)
- Convolutional Neural Networks (CNNs)
- Long Short-Term Memory (LSTM) networks
- Autoencoders

ANNs are computational models inspired by the structure and functioning of biological neural networks in the human brain. They consist of interconnected nodes or ”neurons” organized in layers. Each connection between neurons is associated with a weight, which is adjusted during the learning process. Owolabi et al. [**Owolabi2023FEMReview**] demonstrated that ANNs can achieve high accuracy in gearbox fault diagnosis, while Kavaz et al. [**Kavaz2023AFaults**] effectively applied ANNs to detect generator heating faults.

CNNs represent another powerful model that has shown success in wind turbine condition monitoring. Unlike standard ANNs, CNNs utilize convolutional layers that can automatically extract spatial features from input data. Advanced CNN-based approaches have been developed for gearbox bearing monitoring, achieving significantly higher accuracy than traditional methods [**Fu2019ConditionModel**]. Furthermore, spatio-temporal fusion models combining CNNs with recurrent architectures have shown superior performance in capturing both spatial relationships between different SCADA parameters and their temporal evolution [**Kong2020ConditionUnits**]. This ability to simultaneously extract and fuse multi-dimensional features makes CNN-based approaches particularly effective for processing the complex patterns in the SCADA data.

LSTM networks have also been popular in wind turbine condition monitoring due to their ability to model long-term dependencies in sequential data. Unlike other neural networks, LSTMs contain memory cells with gating mechanisms that allow them to selectively remember or forget information over extended time periods. This architectural advantage makes LSTMs particularly well-suited for analyzing SCADA time-series data, where patterns may develop gradually over a long time. LSTM-based approaches have demonstrated superior performance in forecasting potential failures, and detecting anomalies in operational parameters [**Udo2021Data-DrivenData**, **Wu2022ATurbines**]. Hybrid architectures combining LSTM with other neural network types have shown particular promise, with integrated RNN-LSTM approaches demonstrating better performance over conventional methods like XGBoost and Random Forest Regressor for short-term fault prediction [**Fu2019ConditionModel**, **Rama2024Short-TermRNN-LSTM**].

Finally, autoencoders represent another deep learning approach for wind turbine condition monitoring, particularly for anomaly detection. These neural networks learn to compress input data into a lower-dimensional latent space and then reconstruct it, with the reconstruction error serving as an anomaly indicator. LSTM-based autoencoders have proven successful for time-series anomaly detection in wind turbines, combining the temporal modeling capabilities of LSTMs with the dimensionality reduction of autoencoders [**Rama2024Short-TermRNN-LSTM**]. Additionally, Graph Convolutional Autoencoders for Multivariate Time series (MTGCAE), which model sensor networks as dynamical functional graphs, also demonstrated achieving anomaly detection performance with fewer false alarms [**Miele2022DeepSeries**].

Despite the promising performance of deep learning models, several practical challenges remain. First, these models require substantial amounts of high-quality operational data that adequately capture the normal behavior patterns. Second, the hardware infrastructure of existing wind farm SCADA systems is typically designed for data acquisition and basic monitoring rather than advanced analytics. Training and running complex models like LSTMs and autoencoders requires computational resources that might exceed the limit in the systems [**ChoeWeiChang2023RecentReview**]. Third, the black-box nature of deep learning models can make it difficult to interpret their predictions, potentially reducing trust from maintenance personnel. Finally, these models require careful hyperparameter tuning and architecture design to achieve optimal performance, necessitating considerable expertise in both deep learning and wind turbine engineering domains [**Stetco2019MachineReview**].

2.4.3 Analysis of Prediction Error

After establishing a baseline model that represents healthy turbine behavior, the final step of the NBM methodology is analyzing the prediction error. The goal of this analysis is to detect patterns in the residuals that might indicate abnormal conditions or incoming failures. The prediction error is defined as the difference between the observed signal values and those predicted by the NBM. Under normal operating conditions, these errors typically follow specific statistical patterns. When a component begins to degrade or malfunction, these patterns change, manifesting as abnormal deviations or systematic trends in the prediction errors [**Li2023DeepChallenges**].

Prediction error analysis methods can be classified along multiple dimensions. From a domain perspective, categories can be broadly divided into statistical methods, Statistical Process Control (SPC) techniques, and machine learning approaches [**Chesterman2023OverviewFarm**
From an analytical scope perspective, they can be divided into univariate methods that analyze individual signals independently and multivariate methods that consider relationships between multiple signals simultaneously.

Statistical Methods

Statistics-based methods operate under the assumption that prediction errors during healthy and unhealthy operations follow different distributions. The approaches in this category involve setting fixed or adaptive thresholds for the prediction errors. One simple approach in setting thresholds can be based on statistical properties of the training data and adjusted with domain knowledge about physical component limitations [**Schlechtingen2014WindExamples**]. Similarly, another straightforward approach uses multiples standard deviation of the prediction error to establish the anomaly thresholds [**Kusiak2012AnalyzingApproach**]. While simple to implement, these methods may struggle with non-Gaussian error distributions and seasonal variations. Therefore, other approaches employ adaptive thresholds that change based on operating conditions. For example, Zhao et al. [**Zhao2018AnomalyNetwork**] use extreme value to model the distribution of prediction errors under varying wind speeds,

calculating a threshold that adjusts to operational conditions.

For multivariate analysis, Mahalanobis distance (MD) has gained popularity in wind turbine condition monitoring due to its ability to account for correlations between variables in multidimensional spaces. Thresholds for MD-based detection are typically established by calculating the average MD value from known healthy operation periods [**Liu2022WindData**], or by using chi-square distribution percentiles when the data are approximately multivariate normal [**Weil2022AutoencoderTurbine.**]. A key advantage of MD-based methods is their ability to handle high-dimensional data while remaining computationally efficient, making them suitable for real-time monitoring applications. However, these methods assume that normal data follows an approximately elliptical distribution, which may not always hold true in nonlinear systems such as wind turbine wind turbine operational states. To address this limitation, research has explored the combinations with deep learning techniques to improve performance on non-Gaussian data distributions [**Miele2022DeepSeries**].

For handling the non-linearity and complex dynamics of wind turbine systems, Kernel Density Estimation has gained traction as a non-parametric approach that can model arbitrary distributions of prediction errors without assuming Gaussian behavior. By estimating the probability density function directly from the data, KDE establishes flexible thresholds that adapt to the underlying error distribution's shape rather than imposing parametric assumptions [**Choi2024MultivariateBandwidth**]. However, computational efficiency remains a challenge for KDE-based methods, particularly for high-dimensional data common in wind turbine monitoring. Therefore, dimensionality reduction techniques like Principal Component Analysis and t-distributed Stochastic Neighbor Embedding are often applied as preprocessing steps before KDE [**Choi2024MultivariateBandwidth**].

Statistical Process Control

Statistical Process Control (SPC) represents a specialized subset of statistical methods that have been adapted from manufacturing quality control to wind turbine condition monitoring. While sharing the fundamental statistical principles used in threshold-based anomaly detection, SPC provides a formalized framework with standardized tools and specific philosophical underpinnings.

The traditional SPC workflow involves establishing baseline behavior, setting control limits, and continuously monitoring for violations of these limits using control charts. When applied to wind turbines, these techniques monitor time-series data for deviations that may indicate incipient faults. Different monitoring scenarios require specialized control chart variants, each optimized for detecting specific patterns of deviation. Control charts remain the cornerstone of SPC, with several variants adapted for different monitoring needs. Shewhart charts track individual measurements for sudden shifts, while Cumulative Sum (CUSUM) and Exponentially Weighted Moving Average (EWMA) charts offer increased sensitivity to small, persistent changes in process parameters.

The Shewhart control chart represents a fundamental tool in SPC for monitoring wind turbine condition. For example, Udo and Muhammad [**Udo2021Data-DrivenData**] demonstrated the effectiveness of Shewhart charts by establishing control limits to detect residuals from the prediction. The fault thresholds were defined as Upper Control Limit (UCL) and Lower Control Limit (LCL), setting at $\pm 3\sigma$ of the deviation's distribution. Similarly, Sabani et al. [**Sabani2024StatisticalSystems**] applied Shewhart control charts to bearings in turbines, showing how they effectively identify deviations in vibration and speed parameters during operation. Both studies highlight how Shewhart charts enable early fault detection within a predictive maintenance framework, even without prior knowledge of failure patterns.

Beyond traditional Shewhart charts, CUSUM control charts offer increased sensitivity for detecting small but persistent shifts in system parameters. CUSUM charts accumulate deviations from a target value over time, making them effective at identifying subtle and gradual changes. Chesterman et al. [**ChestermanConditionModels**, **Chesterman2022TheDetection**] demonstrated this advantage in their approach for detecting generator bearing failures. Their implementation involved setting reference values to determine detectable anomaly sizes and decision thresholds that triggered alerts when cumulative deviations exceeded them. Furthermore, CUSUM implementations can be computationally optimized for online condition monitoring in wind turbines. These methods maintain running calculations that update with each new measurement, making them ideal for real-time applications [**Dao2021ATurbines**, **Shaheen2023PerformanceStrategy**]. For multivariate frameworks, Latiffianti et al. [**Latiffianti2022WindData**] adopted a hybrid approach combining CUSUM with Local Minimum Spanning Tree (LoMST) anomaly detection for wind turbine gearbox failures. This CUSUM-LoMST method addresses challenges in using CUSUM for complex turbine data by determining which anomaly signals to accumulate, setting appropriate offset values, establishing optimal accumulation window sizes, and defining appropriate control limits.

The Exponentially Weighted Moving Average (EWMA) chart offers another approach for wind turbine condition monitoring, particularly when sensitivity to gradual degradation is required. Unlike CUSUM, which equally weights all deviations within the accumulation window, EWMA assigns exponentially decreasing weights to past observations, giving more emphasis to recent measurements while still retaining the influence of historical data. EWMA formulates a statistic with the smoothing parameter that controls the weighting balance between current and past data, and has been applied in multiple studies [**Harrou2023SensorChart**, **Encalada-Davila2022EarlyData**]. The selection of the smoothing parameter provides an additional tuning mechanism that CUSUM lacks - smaller values (typically 0.1-0.3) enhance sensitivity to minor, gradual changes in turbine behavior, while larger values improve response to more substantial shifts [**Encalada-Davila2022EarlyDa**
Furthermore, modern implementations of EWMA incorporate adaptive threshold determination techniques such as Kernel Density Estimation, which increases flexibility compared to traditional fixed-threshold implementations [**Harrou2023SensorChart**].

Machine Learning Approaches

Machine learning techniques have emerged as alternatives to traditional statistical methods for anomaly detection. The advantage of ML models is that it can automatically learn complex patterns and relationships directly from historical operational data without requiring explicit physical modeling or manual threshold setting. These approaches can broadly be categorized into supervised, semi-supervised, and unsupervised methods.

Supervised learning approaches for anomaly detection typically frame the problem as a binary classification task, distinguishing between normal operation and fault conditions using labeled historical data. Support vector machines and neural networks have shown promising results in identifying abnormal instances [**Chen2021AnomalyNetwork**, **Dhiman2021WindMachines**]. Their primary advantage is high accuracy when sufficient labeled data is available, though acquiring fault data remains challenging.

Semi-supervised approaches on the other hand, require only normal operation data for training, learning the characteristics of normal behavior and identifying deviations as potential anomalies. These methods are particularly valuable when fault data is scarce. Deep autoencoders have been successfully applied to capture normal operational patterns and use reconstruction error to detect anomalies [**Liu2023WindInstances**, **ConradiHoffmann2021**

Unsupervised methods also detect anomalies without requiring labeled data, typically identifying points that significantly deviate from majority patterns. Clustering algorithms like K-means and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) have been have shown promise in this domain. K-means clustering partitions SCADA data into distinct groups, identifying observations that fall far from cluster centroids as potential anomalies [**Rodriguez2023ExploratoryPurposes**]. This technique effectively identifies outliers by analyzing distance metrics, enabling the detection of abnormal behaviors without pre-labeled examples. Density-based approaches like DBSCAN identify anomalies as points located in low-density regions, as demonstrated by Cao et al. [**Cao2025OptimizingAlgorithm**], offering resilience to noise and variable cluster shapes found in turbine operational data. In addition, PCA combined with clustering approaches has also achieved high classification accuracy in recent implementations [**Khan2024DetectingTurbines**]. A key advantage of these unsupervised techniques is their adaptability to evolving operational conditions and ability to discover previously unknown fault patterns that might be missed by predefined models.

3 Methodology

This chapter presents a comprehensive framework for developing Normal Behavior Models (NBMs) for wind turbine condition monitoring, with a focus on detecting generator and gearbox bearing failures. In this approach, the NBM is trained on healthy operating data so that it learns the expected relationships between variables under normal conditions. Deviations between predicted and measured behavior during operation indicate a potential fault, allowing early detection of abnormal bearing behavior.

The methodology is structured across five sections: The input data is first introduced in section 3.1, followed by a comprehensive data analysis in section 3.2 to understand feature relationships and temporal dependencies. The pre-processing pipeline is described in section 3.3, encompassing healthy data selection, outlier removal, and signal filtering techniques. Normal behavior modeling approaches using XGBoost and LSTM are detailed in section 3.4, including feature engineering and cross-validation strategies. Finally, the CUSUM-based anomaly detection framework is presented in section 3.5, with parameter optimization for detecting bearing degradation while minimizing false alarms.

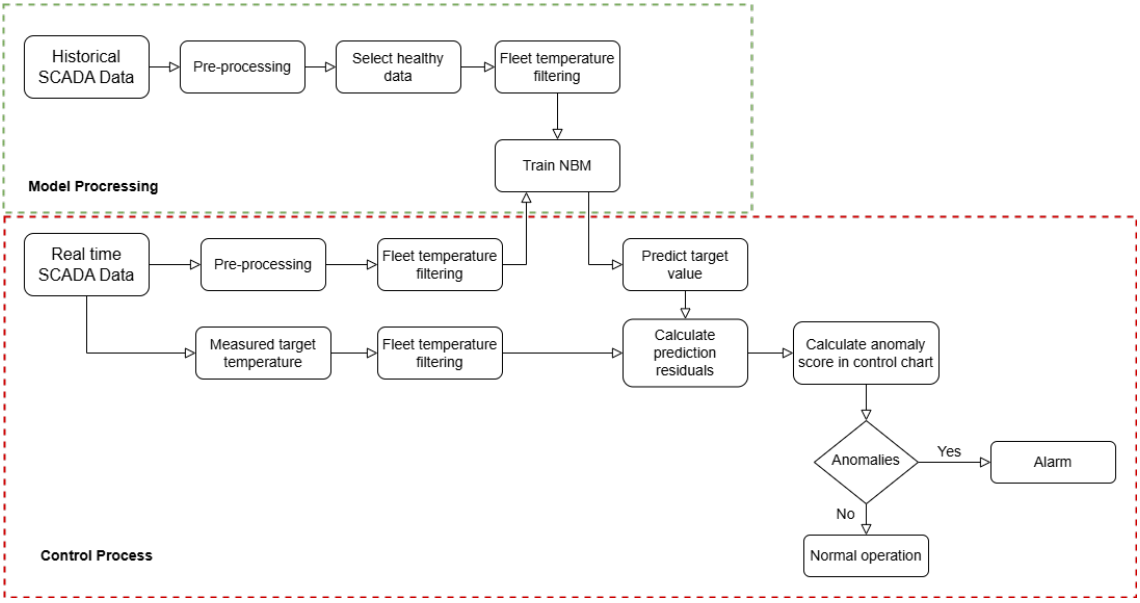

Figure 3.1: Schematic overview of NBM framework, adapted from [**Udo2021Data-DrivenData**]

3.1 Input Data

The data used for this research is the open SCADA Data provided by Energias de Portugal (EDP). This dataset contains operational data from 5 wind turbines (T01, T06, T07, T09, T11) located in Spain, collected over a 2-year period (January 2016 to December 2017). It should be noted that data for turbine T09 has been removed from the EDP open data portal, though it was available and utilized during this research. The data were acquired through the standard SCADA system which records 80 operational parameters at 10 minutes intervals. These operational parameters encompass categories such as:

- Environmental conditions (e.g. ambient temperature, wind speed, wind direction)

- Power generation (e.g. active power, reactive power)
- Thermal conditions (e.g. generator bearing temperature, hydraulic oil temp)
- Mechanical status (e.g. rotor speed, pitch angles)

The notation and description of all variables are listed in Table A.1 in the Appendix.

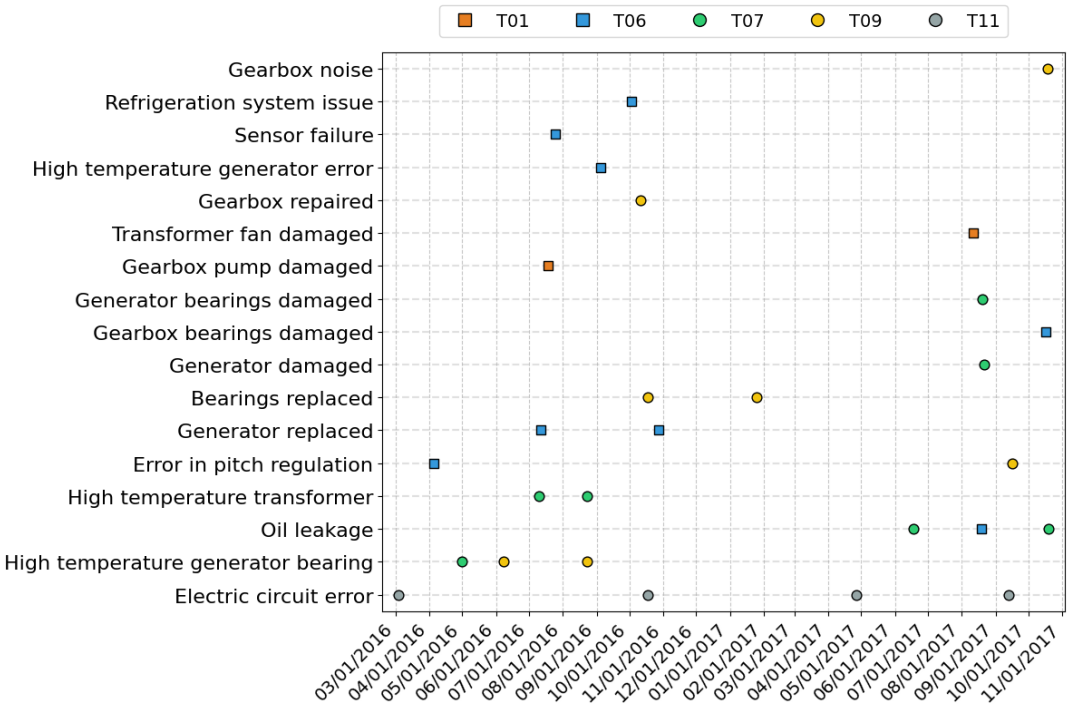

Figure 3.2: Failure remarks contained in the EDP failure log

In addition to the SCADA dataset, EDP also provided nearby mast data, signal log books, and a list of manual annotations of major failures or component replacements. A detail of failure remarks is demonstrated by Figure 3.2.

The wind turbines in the wind farm are all the same type with a rated power of 2 MW, which Figure 3.3 shows the power curve using turbine T01 as a representative. Figure 3.4 presents the distribution of wind speeds measured in the same turbine, and the wind rose across 5 turbines are illustrated in Figure 3.5.

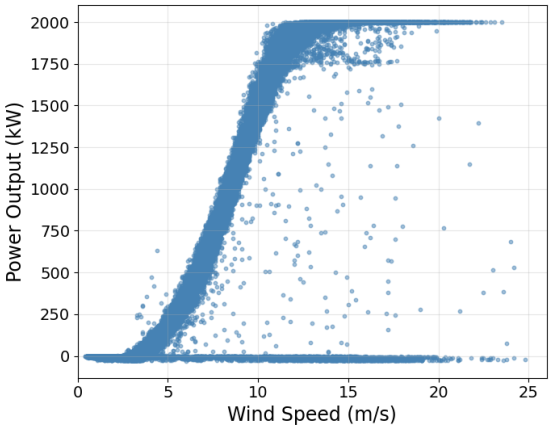

Figure 3.3: Power curve of turbine T01

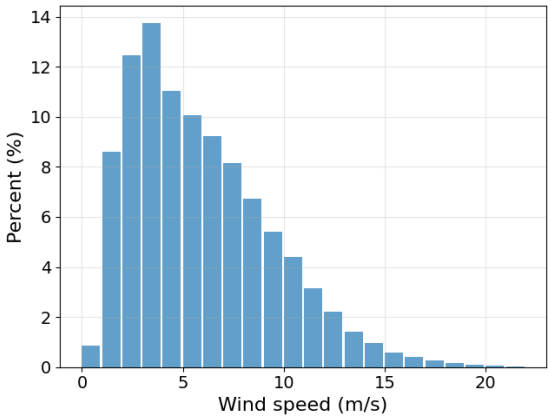

Figure 3.4: Wind speed distribution T01

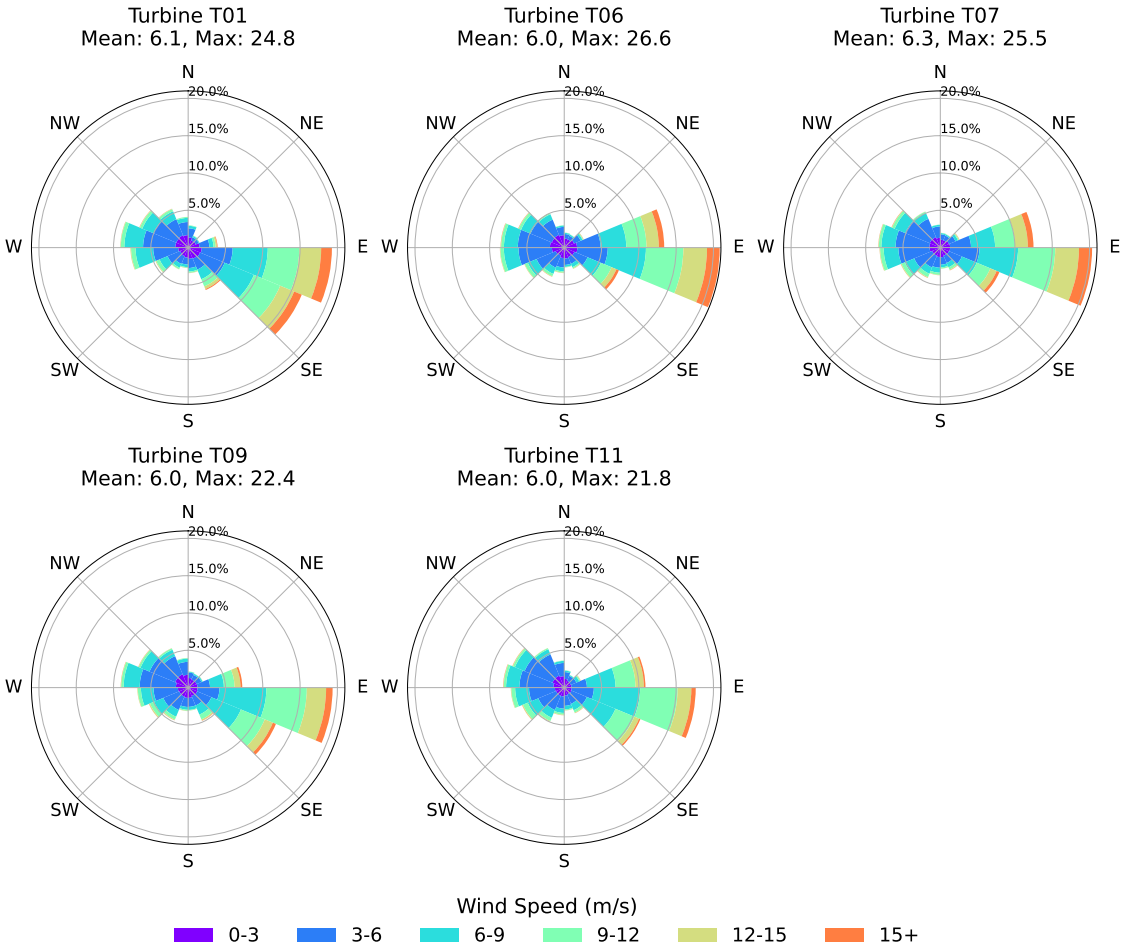

Figure 3.5: Wind rose diagram across all five turbines

Furthermore, Figure 3.6 presents the power curves for the five turbines. These curves were generated by binning the wind speed data at 1 m/s intervals and calculating the mean power output within each bin. With the same binning wind speed, the fitted Weibull probability distributions for five turbines are shown in Figure 3.7.

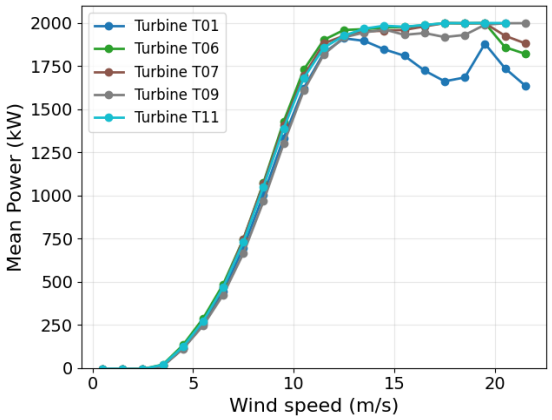

Figure 3.6: Mean power comparison for 5 turbines

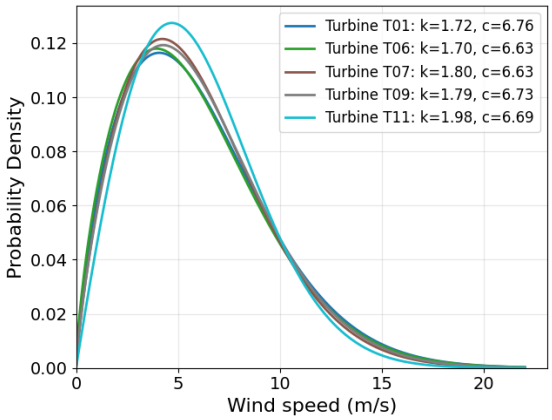

Figure 3.7: Fitted Weibull distribution for 5 turbines

The observed drop in mean power output at high wind speeds across turbines T01, as shown in Figure 3.6, highlights a critical consideration for NBM development. These abnormal operation typically resulting from turbine cut-out mechanisms for a faulty state, suggest that the faulty state needs to be properly identified and removed during the NBM training process. Failing to account for these operational anomalies could lead to NBM also learn the abnormal condition.

3.1.1 Target Feature

Generator bearing temperature is selected as the target feature for detecting generator bearing faults, as bearing degradation due to wear or lubrication failure typically leads to increased friction and gradual temperature rise [**Pandit2023ATurbines**]. Prior studies, such as [**Tautz-Weinert2017UsingReview**], have shown that temperature monitoring offers reliable early indicators of bearing health in wind turbines. In this work, both the drive-end and non-drive-end generator bearing temperature signals were available and demonstrated strong correlation with each other. To determine which is the more suitable target for the normal behavior model, both signals were evaluated and compared with a test.

Similarly, gearbox bearing temperature is used as the target feature for detecting gearbox bearing faults, as this parameter is also sensitive to internal wear and has been used effectively in earlier works [**Guo2018ConditionModel**].

3.2 Data Analysis

The SCADA data described above serves as the foundation for developing the NBMs. However, before implementing preprocessing steps, several analyses were conducted to improve the understanding of the dataset and the relationships between features in the SCADA data. This analysis is meant to determine the methods applicable for NBM training.

3.2.1 Correlation Analysis

To gain insight into feature relationships and temporal dependencies, both autocorrelation and cross-correlation analyses were performed on the SCADA dataset.

Autocorrelation analysis was conducted for all features to identify their temporal dependency patterns. This analysis reveals how long a signal remains correlated with itself,

which helps determine the appropriate time window for modeling and the minimum separation needed between cross validation data to ensure independence. In this study, the autocorrelation was calculated using the Pearson correlation coefficient between the original time series and its lagged versions. The correlation coefficient was computed as:

$$\rho_{XX}(\tau) = \frac{K_{XX}(\tau)}{\sigma^2} = \frac{E[(X_{t+\tau} - \mu)(X_t - \mu)]}{\sigma^2} \tag{3.1}$$

where $\rho_{XX}(\tau)$ is the autocorrelation at lag τ, $K_{XX}(\tau)$ is the autocovariance at time τ, σ^2 is the variance of the time series and μ is the mean of time series.

Two important temporal metrics were derived from the autocorrelation analysis: the decorrelation time and the integral time scale. The decorrelation time is defined as the lag at which the autocorrelation coefficient falls below 0.05. This threshold indicates when observations can be considered sufficiently independent for cross-validation purposes. As shown in Figure 3.8 and Figure 3.9, the generator bearing temperature exhibits a decorrelation time of 80 hours while the average power output of approximately 59 hours.

The integral time scale, calculated as $T_I = \int_0^{\tau_0} \rho_{XX}(\tau) d\tau$ where τ_0 is the first zero-crossing of the autocorrelation function, represents the characteristic time over which the process "remembers" its previous state. It provides a measure of the average time span over which the signal remains correlated with itself. For the generator bearing temperature, the integral time scale was found to be 36 hours. This is significantly shorter than the decorrelation time because it captures the average correlation across all lags rather than the time to reach a specific threshold. Nonetheless, both these two temporal metrics offer insight into characteristics of how features decay and the sufficient time scale for cross validation. The histograms show the distribution of the decorrelation time and integral time scale for all features provided in Figure A.1 and A.2.

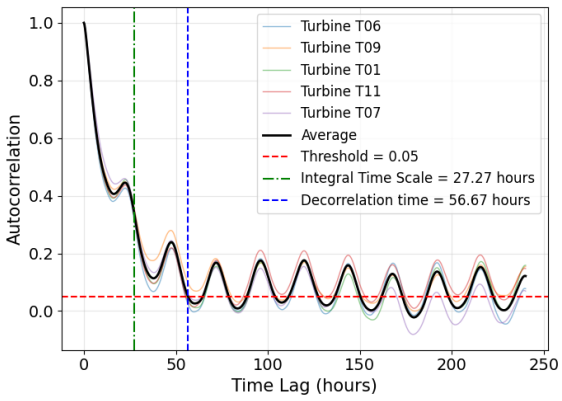

Figure 3.8: Autocorrelation for generator bearing temperature

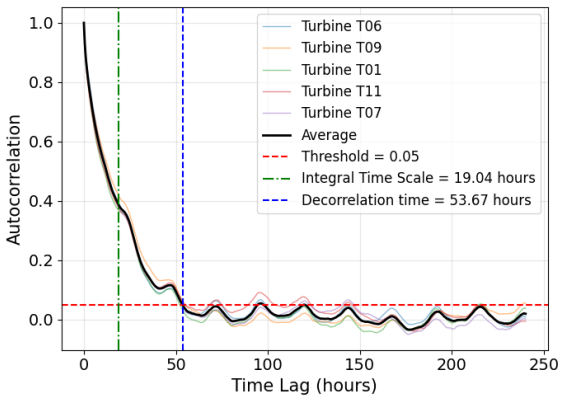

Figure 3.9: Autocorrelation for average power

As illustrated in Figure 3.8 and 3.10, the autocorrelation analysis revealed distinct cyclic patterns across multiple SCADA variables. This diurnal oscillation is characterized by peaks that recur approximately every 24 hours in the autocorrelation function, reflecting the daily temperature cycle that affects both the operational environment and the thermal condition of turbine components.

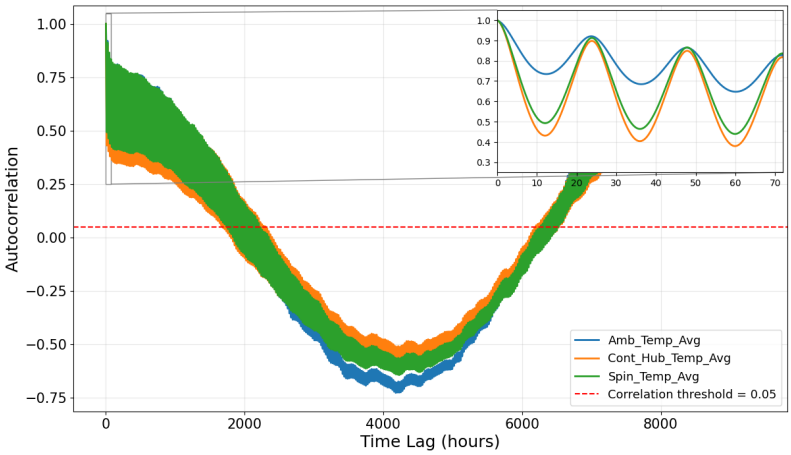

Figure 3.10: Autocorrelation of temperature related features

Beyond the diurnal cycle, longer-term seasonal patterns were also evident in the autocorrelation structure, as demonstrated in Figure 3.10. The analysis revealed a gradual decay followed by negative correlation values around 2000-6000 hours (approximately 3-5 months), which then returned to positive values at approximately 8000 hours (nearly one year). This yearly oscillation pattern aligns with seasonal climate variations that influence wind patterns, ambient conditions, and consequently, turbine operation and thermal behavior.

Similarly, a seasonal variation can also be observed in generator bearing temperatures, as provided in Figure 3.11. This long-term variation reflects the influences of environmental conditions on turbine thermal characteristics. For example, during warmer seasons, higher ambient temperatures affect the cooling efficiency of the generator, potentially leading to slightly elevated bearing temperatures. Conversely, colder seasons may facilitate more efficient cooling. Additionally, seasonal changes in wind patterns and power production also contribute to this cyclical behavior, as higher power output generates more heat in the bearings.

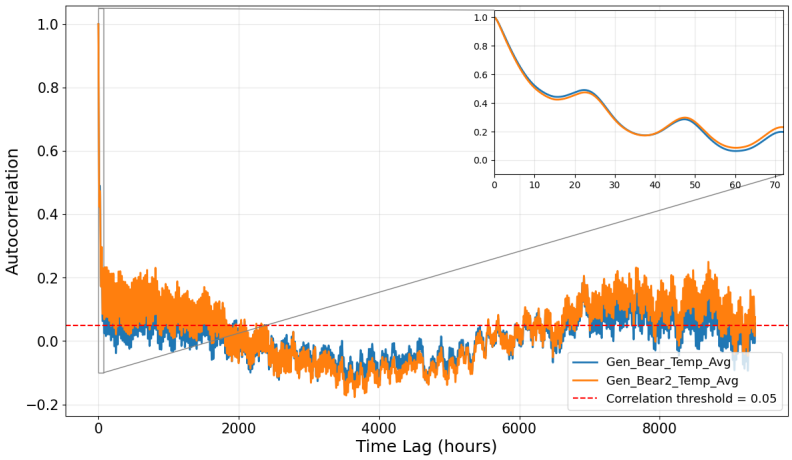

Figure 3.11: Autocorrelation of generator bearing temperatures

The presence of these seasonal patterns in such variables has important implications for NBM training. First, the model must include appropriate features that capture long-term

seasonal trends to predict these variations. Second, the training dataset should better include one complete annual cycle to adequately capture these long-term patterns. Without sufficient temporal coverage, the model may misinterpret seasonal variations as anomalies, leading to false positive detections during normal seasonal transitions. Otherwise, the detrending method could be applied to remove the seasonal trend.

Besides autocorrelation analysis, cross-correlation analysis was performed to examine the relationships between the target feature and other SCADA variables. This analysis helps identify which variables are related to the target feature and at what time lags these relationships are the strongest. The cross-correlation function was calculated as:

$$\rho_{XY}(\tau) = \frac{E[(X_{t+\tau} - \mu_X)(Y_t - \mu_Y)]}{\sigma_X \sigma_Y} \tag{3.2}$$

where $\rho_{XY}(\tau)$ is the cross-correlation at lag τ, μ_X and μ_Y are the means, and σ_X and σ_Y are the standard deviations of time series X and Y, respectively.

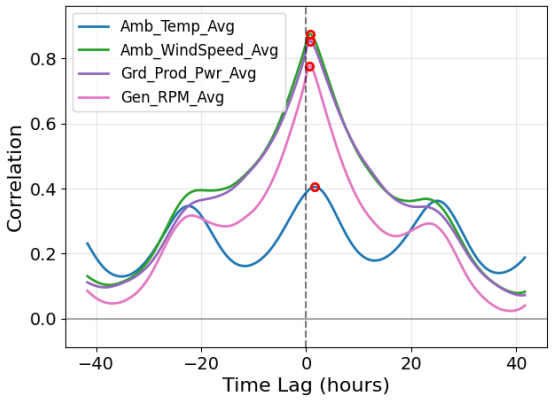

Figure 3.12: Cross-correlation between generator bearing temperature and operational parameters

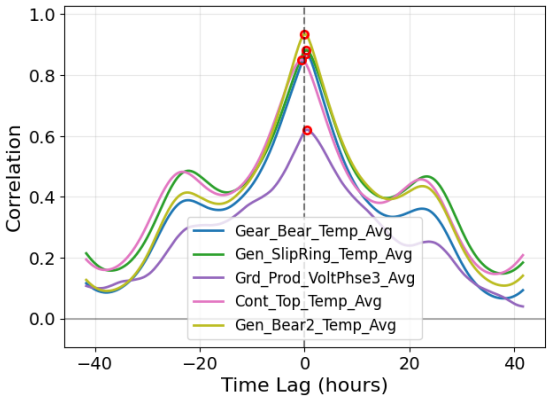

Figure 3.13: Cross-correlation between generator bearing temperature and other thermal parameters

The cross-correlation analysis revealed the relationships between the target variable and various operational and thermal parameters. Figure 3.12 demonstrates that operational variables such as ambient wind speed, power output, and generator RPM exhibit strong positive correlations with generator bearing temperature, with maximum correlation values of approximately 0.8 at a lag of 4. This relationship indicates that the operational parameters have a leading impact on bearing temperature, which is physically consistent with the thermal behavior of the higher wind speed leading to increased generator speed and power output, which subsequently generates more heat in the bearings. These thermal influences become pronounced approximately 40 minutes after the operational changes occur. A histogram counting the number of features with their highest lag steps is presented in Figure A.3.

Figure 3.13 shows the cross-correlation between generator bearing temperature and 5 of the most influential parameters in model training in Figure 3.28. These correlation values peak at lag 2 and 3, suggesting a slightly lagged thermal response throughout the drivetrain components relative to the generator bearing. Interestingly, the nacelle controller temperature peaks at a negative lag of 3, indicating that it responds faster to operational parameter variations compared to generator bearing temperature. This finding suggests

that including lagged features with negative optimal lags would be unnecessary in the NBM training, thereby conserving computational resources.

Notably, both figures reveal a distinct diurnal pattern in the cross-correlations, with secondary peaks occurring at approximately 24 hours. This confirms that the daily temperature cycle influences the thermal state of the entire turbine system, consistent with the findings from the autocorrelation analysis. These cross-correlation findings have implications for feature selection in the NBM training, supporting that including appropriately lagged features could help capture the leading indicators of generator bearing temperature changes.

3.2.2 Principal Component Analysis

To gain insight into the underlying structure of the SCADA data and visualize the distribution of operational states, Principal Component Analysis (PCA) was performed on the dataset. PCA is particularly useful for reducing the high-dimensional operational parameters to a manageable number of principal components while preserving the maximum variance in the data. The principal components are created by linear combinations of the original features, which can be represented as:

$$PC_i = \sum_{j=1}^{n} w_{ij} \cdot X_j \tag{3.3}$$

where PC_i is the i-th principal component, Xj is the j-th original feature, and w_{ij} is the corresponding weight coefficient.

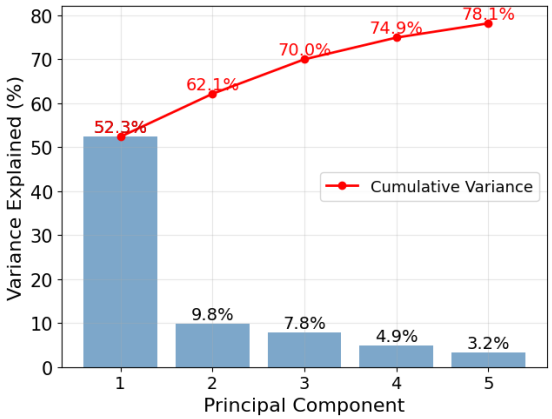

Figure 3.14: Explained variance ratio for each principal component when using all 81 SCADA variables

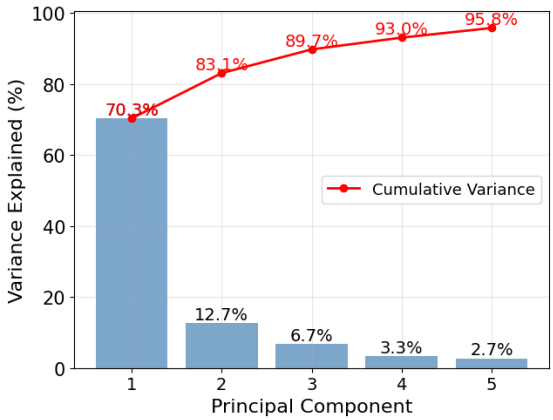

Figure 3.15: Explained variance ratio for each principal component when using only selected temperature variables

An initial PCA test was performed using all 80 variables in the SCADA dataset, with the explained variance presented in Figure 3.14. The relatively low explained variance is attributed to the presence of noise, redundant variables, and potentially non-linear relationships among the original features. To increase the explained variance, a subset of features was selected based on the feature importance analysis described in Section 3.3.4.

Figure 3.15 illustrates the PCA results using only the temperature variables identified as top features. In this case, the principal components captured significantly more variation.

The cumulative explained variance increased from 62.1% to 83.1% for the first two principal components (PC1 and PC2), demonstrating that the selected temperature features better represent the system's underlying structure.

With the optimized principal components, the data were plotted in a PC1/PC2 scatter plot, as shown in Figure 3.16. Since plotting the entire dataset would involve more than 500,000 data points, a stratified random sampling approach was adopted to better visualize the scatter distribution while retaining important clustering behavior. A total of 10,000 points representing normal operation were randomly sampled from the dataset, providing a clear visualization of the operational state distribution.

Furthermore, to identify potential clusters of failure states, data points within the exclusion zones defined in Section 3.3.4 were added to the plot. These points, representing data collected within half a day prior to documented failures in the failure log file, are colored by turbine to highlight turbine-specific patterns. Notably, Turbine T06 shows a distinct cluster in the exclusion zone that deviates from the main operational cluster, confirming that the exclusion methodology successfully excludes pre-failure conditions.

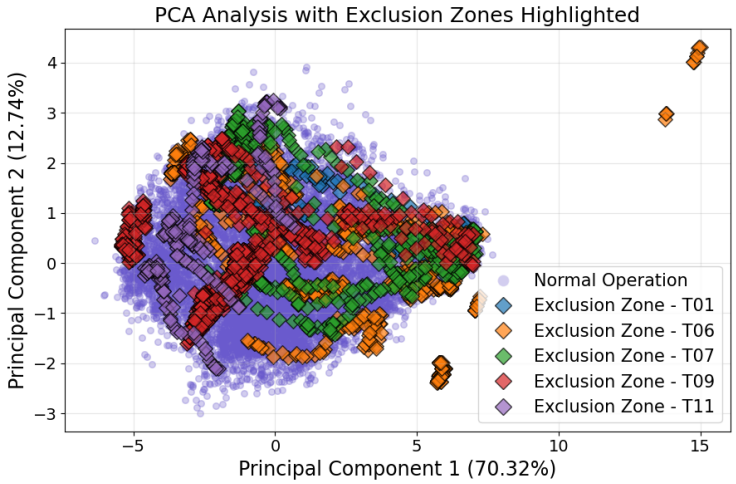

Figure 3.16: PCA scatter plot showing normal operation points and exclusion zones

The PCA feature vectors and turbine centroids are displayed in Figure 3.17. The arrows represent the direction and magnitude of each original feature's contribution to the first two principal components. Longer arrows indicate features with stronger influence on the variance in the data. Temperature variables, including generator bearing temperature, gear bearing temperature, and hydraulic oil temperature, show significant contributions to both PC1 and PC2, confirming their importance in characterizing the system's operational states.

For the turbine centroids, the mean values of PC1 and PC2 for normal operation data from each turbine were calculated. Most turbines cluster relatively close to each other in the PCA space, suggesting consistent operational profiles. However, Turbine T09 is positioned noticeably farther from the center compared to the other four turbines, indicating potential systematic differences in its operational parameters. This observation supports the treat Turbine T09 separately in subsequent modeling steps, as its baseline behavior appears to differ from the fleet average.

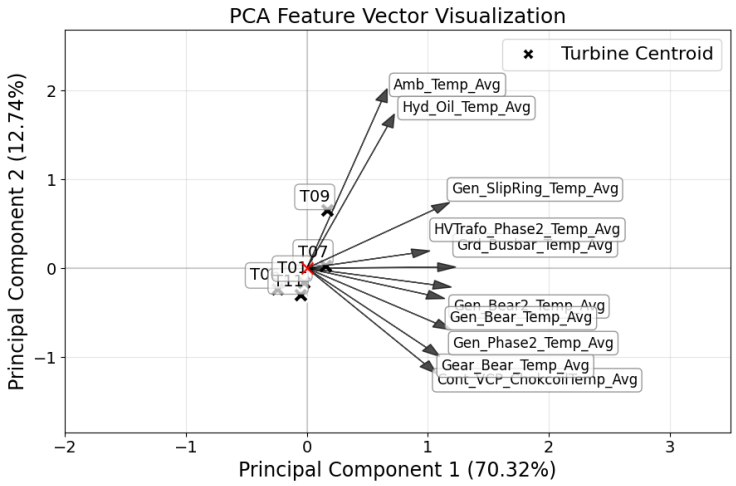

Figure 3.17: PCA biplot showing feature vectors and turbine centroids

3.3 Preprocessing

This section presents an overview of the preprocessing steps employed in this study. Effective preprocessing is essential for NBMs to ensure data quality and establish a healthy state for NBM training.

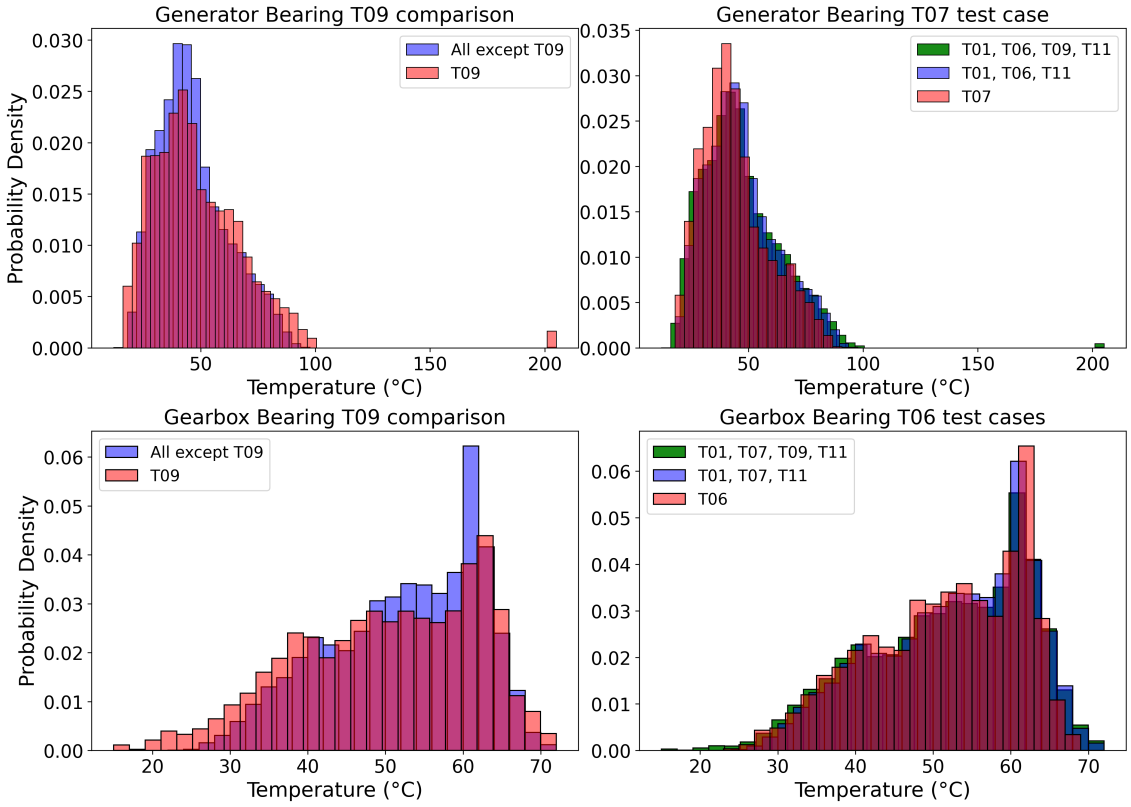

Figure 3.18: Distribution of generator bearing temperature and gearbox bearing temperature across turbines, highlighting the effect of excluding turbine T09

3.3.1 Data Splitting

The SCADA dataset was split into training and testing sets to facilitate model evaluation. As discussed in subsubsection 2.4.1, a conventional approach would involve selecting the last few months of data as the test set. However, this temporal splitting strategy was not optimal for the wind farm dataset since not all turbines contain failures that occurred toward the end of the period. Furthermore, since identifying component degradation patterns constitutes a primary objective of this study, it was crucial to include sufficiently long periods preceding failure events.

To effectively evaluate the model's ability to detect anomalies, a cross-turbine validation approach was implemented. Turbines T06 and T07 were specifically selected as test cases due to their documented failure histories, as detail presented in Table 3.2. For each validation scenario, these turbines were designated as the testing set, while the remaining turbines, with the exception of T09, constituted the training set. The exclusion of turbine T09 was justified by baseline differences identified in the PCA clustering in the data analysis. The distribution differences after T09's exclusion for both test cases are illustrated in Figure 3.18, which shows the distributions of generator bearing temperature and gearbox bearing temperature across turbines.

This cross-turbine validation strategy resulted in an approximate 75%-25% split ratio between the training and test data, providing sufficient data for both model development and validation.

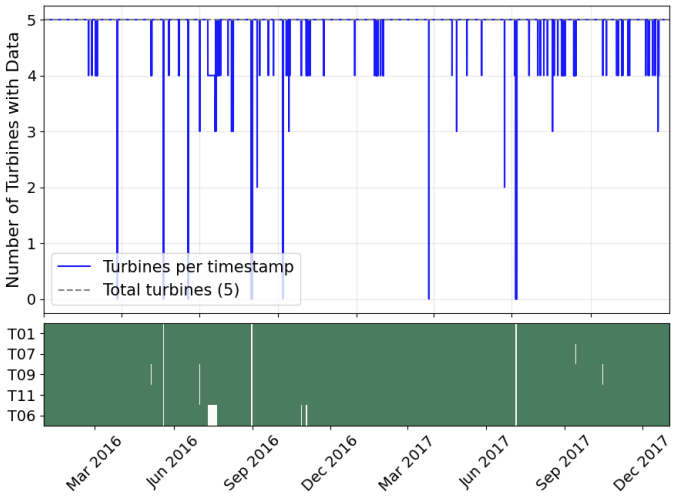

Figure 3.19: Data availability across five wind turbines over a two-year period

3.3.2 Data Cleaning

The dataset exhibited several missing values, including 7 NaN values in the generator bearing temperature (Gen_Bear_Temp_Avg) and 7 NaN values in the average power factor (Grd_Prod_CosPhi_Avg). Additionally, each turbine had timestamps where no data was recorded. Figure 3.19 illustrates the data availability across all turbines in the dataset, showing both periods of complete data collection and intermittent gaps.

As shown in the figure, most turbines maintained high data availability throughout the collection period (97.1% coverage of complete data), with occasional synchronized outages affecting all turbines. Since the NaN values were limited in number and randomly distributed throughout the dataset, timestamps containing these 7 NaN values were simply omitted from the analysis rather than applying imputation techniques.

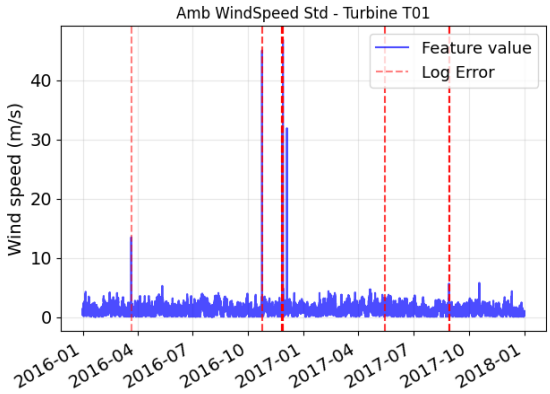

Figure 3.20: Ambient wind speed standard deviation for turbine T01 with log error annotations

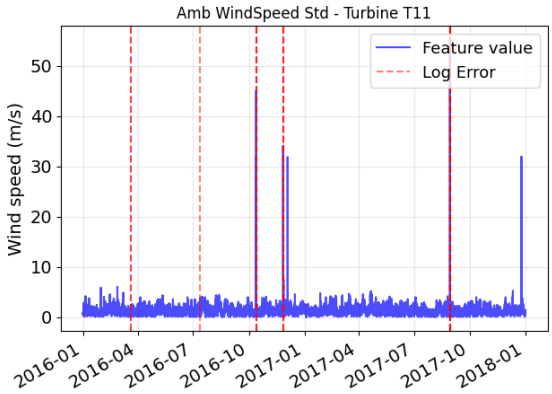

Figure 3.21: Ambient wind speed standard deviation for turbine T11 with log error annotations

Another critical cleaning procedure was outlier removal. Careful treatment of outliers was required, as false outliers might appear but represent valid operational states. Specifically, as shown in Figure 3.25, non-physical extreme values were observed in features including both drive and non-drive end of Gen_Bear_Temp_Avg, and Amb_WindSpeed_Std. Figures 3.20 and 3.21 illustrate the time series of Amb_WindSpeed_Std using turbines T01 and T11 as examples. The signal log remark ”Error on all wind sensors” was plotted as the dashed line in the plots. From these figures, it is evident that not all extreme values could be captured by the signal error remarks. Therefore, an outlier removal approach was needed to handle these extreme values.

For features identified as containing non-physical values, a standard deviation-based thresholding approach was applied. Thresholds were established for each selected feature using the feature's mean and standard deviation (σ). Data points were flagged as outliers when they fell outside a range defined by $\mu \pm n\sigma$, where μ is the mean, σ is the standard deviation, and n is a feature-specific multiplier.

Specifically, the standard deviation threshold method was applied to generator bearing temperatures (both drive and non-drive end) and ambient wind speed standard deviation using a 5σ multiplier. This process successfully identified and removed generator bearing temperatures exceeding around 120 degree, while preserving valid operational data points.

3.3.3 Directional Feature Transformation

The SCADA data contains three directional features: absolute wind direction, relative wind direction, and nacelle position, all measured in degrees (0-360°). These circular variables present a challenge for machine learning algorithms that assume linear relationships between features. For instance, 359° and 1° represent nearly identical wind directions but would be treated differently by standard algorithms.

To address this issue, each directional feature was transformed using a trigonometric decomposition, converting the single angular value into its sine and cosine components. For each directional feature θ , two new features were created by:

$$\begin{aligned} X_{sin} &= \sin(\theta \frac{\pi}{180}) \\ X_{sin} &= \cos(\theta \frac{\pi}{180}) \end{aligned} \tag{3.4}$$

This transformation preserves the circular nature of the data, ensuring similar representations for adjacent angles that cross the 0°/360° boundary. The directional features 'Amb_WindDir_Abs_Avg', 'Amb_WindDir_Relative_Avg', and 'Nac_Direction_Avg' were each converted into their respective sine and cosine components for use in subsequent modeling steps.

Table 3.1 presents the cross-correlation between the original directional features and their trigonometric components with four temperature-based target features. The results demonstrate that the directional decomposition can significantly improve correlation values.

Table 3.1: Cross-correlation between directional features and temperature features

Temperature Feature	**Absolute Wind Direction**			**Nacelle Direction**			**Relative Wind Direction**		
	Original	Cosine	Sine	Original	Cosine	Sine	Original	Cosine	Sine
Generator bearing (non-drive end)	-0.381	-0.208	**0.443**	-0.415	-0.246	**0.463**	-0.029	**0.323**	-0.018
Generator bearing (drive end)	-0.290	-0.238	**0.364**	-0.309	-0.284	**0.381**	-0.023	**0.298**	-0.020
Gearbox bearing	-0.276	-0.227	**0.347**	-0.300	-0.284	**0.369**	-0.025	**0.441**	-0.029
Gearbox oil	-0.256	-0.220	**0.320**	-0.279	-0.275	**0.340**	-0.025	**0.397**	-0.030

3.3.4 Healthy Data Selection

After removing missing values and outliers, a critical challenge remains: identifying and excluding unhealthy observations from the training data. As described in Section 2.2, component failure typically occurs through gradual degradation rather than as a sudden event. To create an effective normal behavior model, clear criteria for healthy data selection were established to prevent the model from learning abnormal behavior patterns.

Previous researchers provide guidance on determining suitable temporal boundaries between healthy and unhealthy data for different components. For gearbox failures, a 4-month exclusion period before known failure dates has been recommended [**Verma2022WindData**], while for main bearing failures, approximately 4-6 months of data preceding the failure should be excluded from training datasets [**Encalada-Davila2021WindData**]. Other studies suggest a 6-month exclusion window before bearing failures when using principal component analysis [**Campoverde2022SCADAAnalysis**]. These exclusion windows provide a sufficient temporal margin to capture early fault symptoms while maintaining adequate healthy data for model training.

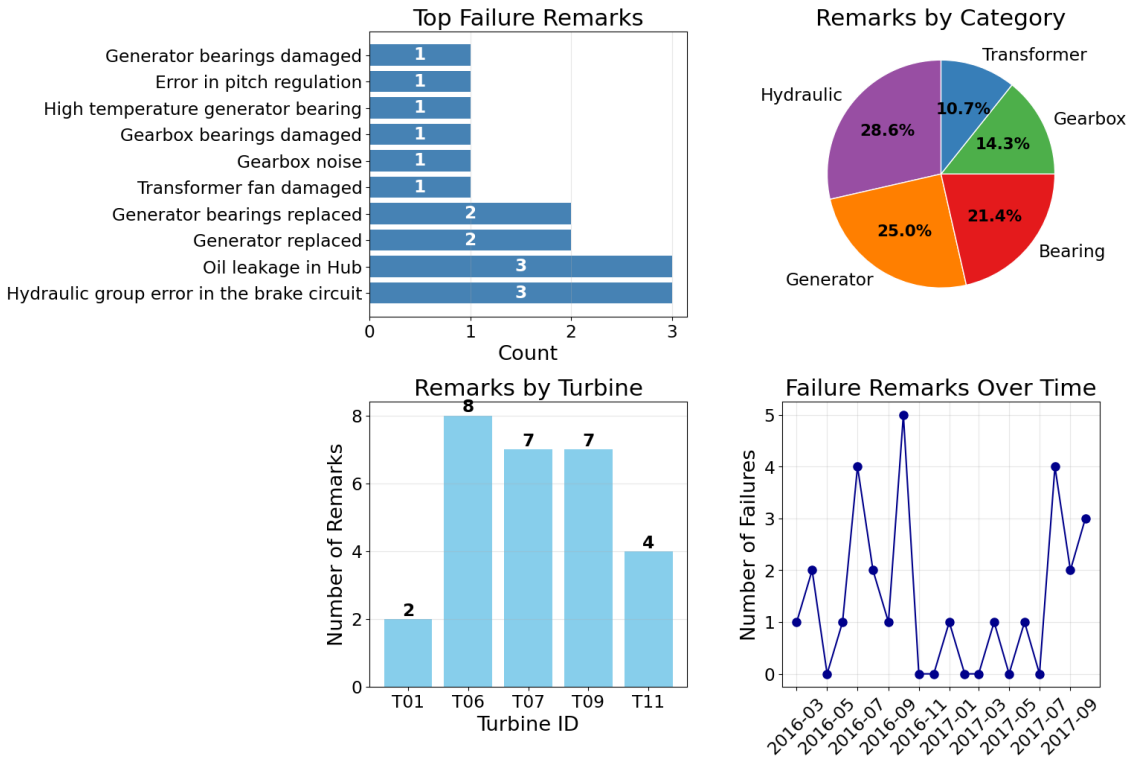

Figure 3.22: Analysis of wind turbine failure log: (a) Top-left: Frequency of top 10 failure remarks; (b) Top-right: Distribution of remarks by component category; (c) Bottom-left: Number of remarks per turbine; (d) Bottom-right: Temporal distribution of failures

A detailed examination of the failure log file was conducted to distinguish between actual component failures versus warnings or routine maintenance annotations. As shown in the top-left subplot of Figure 3.22, certain remarks like ”Hydraulic group error in the brake circuit” occurred multiple times but appeared to be annotations rather than actual failures. Similarly, for generator bearing issues, entries such as ”High temperature generator bearing” were classified as warnings rather than catastrophic failures. Each remark in the failure log was carefully assessed to identify genuine failure events, with comprehensive results presented in Table 3.2.

Based on this analysis, 28 total failures across 5 component types were identified. Following component characteristics and failure progression rates, different exclusion periods were implemented:

- **Four-month exclusion window:** Applied to four cases of slow-deteriorating component failures, including damaged gearbox pumps (T01), damaged gearbox bearings (T06), damaged generators (T07), and damaged generator bearings (T07). This longer window accounts for the gradual degradation that might occur prior to physical damage being detected.
- **One-month exclusion window:** Applied to 24 cases, including component replacements, hydraulic oil leakages, transformer issues, and cases where the remarks indicated less severe issues. Examples include generator replacements (T06), generator bearing replacements (T09), and oil leakage in hubs (T06, T07, T09).

For all failure events, a post-failure period of 2 days was also excluded to ensure complete coverage of the potential repair period and to prevent any transitional states from being

included in the training data.

Table 3.2: Wind Turbine Component Failures

Turbine	Component	Timestamp	Remarks
	Physical Damage		
T01	TRANSFORMER	2017-08-11	Transformer fan damaged
T01	GEARBOX	2016-07-18	Gearbox pump damaged
T06	GEARBOX	2017-10-17	Gearbox bearings damaged
T07	GENERATOR	2017-08-21	Generator damaged
T07	GENERATOR_BEARING	2017-08-20	Generator bearings damaged
	Component Replacements		
T06	GENERATOR	2016-10-27	Generator replaced
T06	GENERATOR	2016-07-11	Generator replaced
T09	GENERATOR_BEARING	2016-10-17	Generator bearings replaced
T09	GENERATOR_BEARING	2017-01-25	Generator bearings replaced
T09	GEARBOX	2016-10-11	Gearbox repaired
	Oil Leakage Issues		
T06	HYDRAULIC_GROUP	2017-08-19	Oil leakage in Hub
T07	HYDRAULIC_GROUP	2017-06-17	Oil leakage in Hub
T07	HYDRAULIC_GROUP	2017-10-19	Oil leakage in Hub

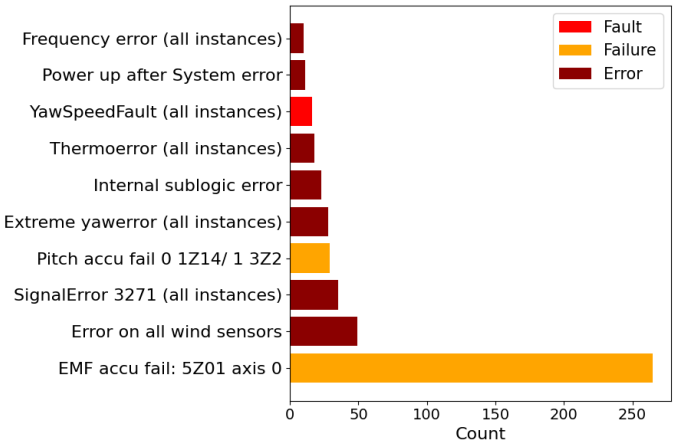

Figure 3.23: Distribution of the top 10 error-related signals from the signal log file

In addition to component failures, sensor errors and turbine operation anomalies were also addressed. As illustrated in Figure 3.23, the signal log contained numerous error entries including ”Error on all wind sensors” (49 entries) and ”Extreme yawerror” (28 entries). While most signals represented sensor errors rather than component failures, they indicated abnormal turbine operation states that could affect data quality. These errors were distributed across all turbines, with T01 having the highest number of wind sensor errors and T06 showing the most extreme yaw errors. Time periods corresponding to these events were precisely excluded based on their recorded start and end timestamps in the signal log file.

To further detect and exclude any residual outliers that are not captured in previous criteria, an Isolation Forest–based anomaly detection method was applied. A consideration of varying contamination levels by power state is implemented based on the power curve in Figure 3.3. Specifically, contamination levels were set at 10% for idle state ($<$100 kW), 5% for low power (100-500 kW), 3% for medium power (500-1000 kW), 2% for high power

(1000-1500 kW), and 1% for full power operation (>1500 kW). This method is considered as conservative approach which the cluster at idle stage were mostly excluded. The percentage choice is based on the assumption that outliers that are attributed to failures would tend to result in low or no power production.

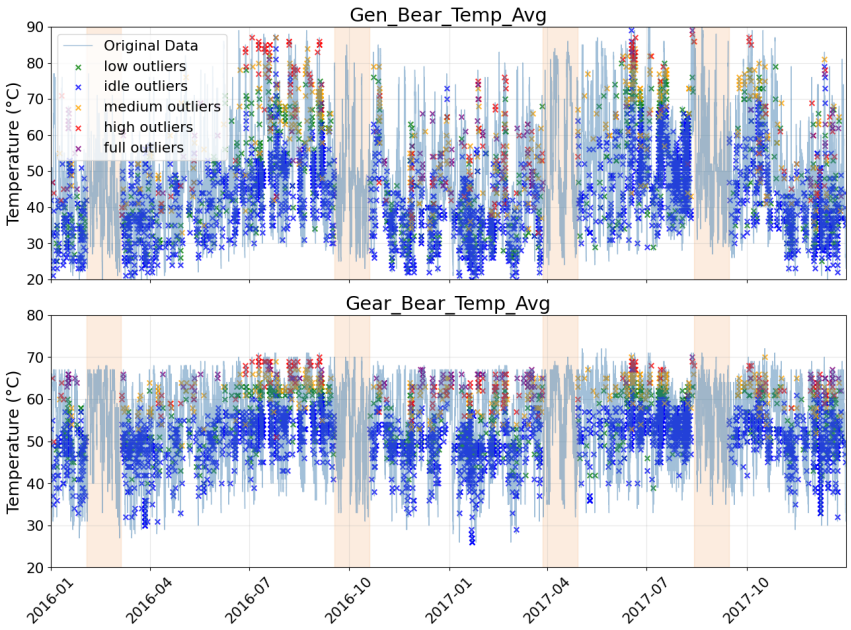

Figure 3.24: Data filtering visualization using T11 as an example

Figure 3.24 illustrates this outlier detection process applied to turbine T11. The Isolation Forest identified numerous anomaly clusters across different operational states. For T11, this included 3,274 anomalies in the idle state, 1,069 in the low state, 393 in the medium state, 173 in the high state, and 106 in the full power state. When dense clusters of anomalies (defined as 75 or more anomalies within a 12-hour window) were detected, the entire period was excluded from the training data. This clustering approach is a balance of not losing too many continuous sequential data while removing potential outliers that deviate from the normal states.

The effectiveness of the healthy data selection approach is summarized in Table 3.3, which shows the data retention statistics for each turbine after applying all exclusion criteria. Overall, 68.1% of the original data points were retained after cleaning, providing a substantial dataset of normal operation for developing the models. The turbines with the most failures naturally showed the lowest retention rates, with T06 retaining only 58.5% of its original data points due to having the highest number of failure events (8 total, including 1 with a 4-month exclusion period). Conversely, T11 had the highest retention rate at 77.7%, corresponding to its fewer failure events.

Table 3.3: Data Retention Statistics After Healthy Data Selection

Turbine	Original Points	Retained Points	Retention (%)	4-Month Exclusions	1-Month Exclusions
T01	104,682	77,686	74.2%	1	1
T06	102,920	60,219	58.5%	1	7
T07	104,738	64,574	61.7%	2	5
T09	104,640	71,710	68.5%	0	7
T11	104,797	81,382	77.7%	0	4
Overall	521,784	355,571	68.1%	4	24

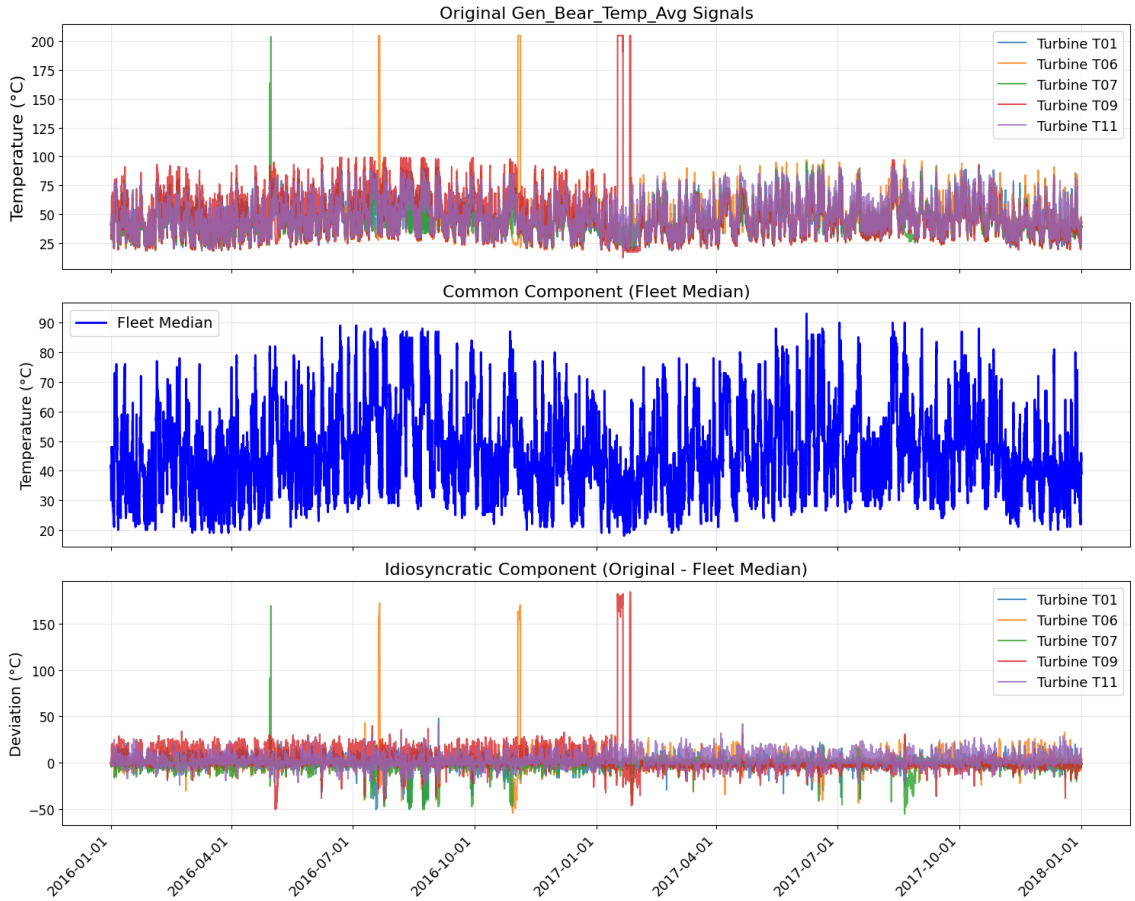

Figure 3.25: Decomposition of generator bearing temperature using fleet median filtering on original dataset. Top: Original temperature signals from five turbines. Middle: Common component extracted via fleet median. Bottom: Idiosyncratic component showing turbine-specific deviations from fleet behavior.

3.3.5 Signal Filtering

Wind turbine SCADA signals are influenced by numerous factors, including environmental conditions that affect all turbines within a farm simultaneously. For example, the higher ambient temperature can lead to overall higher temperature measurement in the temperature-related features. To isolate turbine-specific behaviors from these farm-wide effects, a fleet median filtering approach was implemented similar to that described by Chesterman et.al. [**Chesterman2023OverviewFarms**, **ChestermanConditionModels**].

The method decomposes each signal into two components:

1. A “common component” representing farm-wide effects (calculated as the median value across all turbines at each timestamp)
2. An “idiosyncratic component” representing turbine-specific behaviors (calculated by subtracting the common component from the original signal)

This decomposition is illustrated in Figure 3.25, which shows the generator bearing temperature for all five turbines in the wind farm. The top panel displays the original temperature signals, which contain turbine-specific behaviors. The middle panel shows the common component extracted through the fleet median. This component captures environmental factors affecting all turbines, such as ambient temperature fluctuations and

shared operational conditions. The bottom panel displays the idiosyncratic component for each turbine, highlighting individual differences that may indicate potential anomalies or operational characteristics.

The fleet median serves as an implicit normal behavior model without requiring predictor selection or model training. By removing the common component, the modeling task was simplified by eliminating seasonal fluctuations and shared transient behaviors, allowing the analysis to focus on turbine-specific deviations.

To ensure the reliability of this method, particularly in a small wind farm with only five turbines, a safeguard rule was implemented: timestamps where fewer than four turbines have valid data are excluded from the analysis. This constraint helps maintain the representativeness of the fleet median, as suggested by Chesterman et al. [**ChestermanConditionModels**].

3.4 Normal Behavior Modeling

3.4.1 Standardization

Z-score standardization was applied to all features using scikit-learn's StandardScaler to normalize the data before model training. For each feature f, the standardized values were computed by:

$$Z_f = \frac{X_f - \mu_f}{\sigma_f} \tag{3.5}$$

where μ_f and σ_f are the mean and standard deviation of feature f calculated from the healthy training data during the preprocessing stage. This transformation ensures all features in the training data have zero mean and unit variance. This step is specifically important for neural network models.

Z-score standardization was selected over alternative normalization methods, particularly min-max scaling, due to the sensitivity of min-max scaling to extreme values. This characteristic is crucial for the study, as the test data from the faulty turbine exhibits several extreme values that could extend the distribution in the testing data. Min-max scaling would constrain all values to a fixed [0,1] range, potentially limiting the model's ability to detect such anomalous behavior.

3.4.2 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) was selected as the first machine learning algorithm for normal behavior modeling due to the advantages suited to this application. As an ensemble method, XGBoost constructs multiple decision trees sequentially, where each tree corrects the errors of its predecessors trees, making it effective at capturing non-linear relationships between SCADA features [**Chen2016XGBoost:System**]. The algorithm's built-in regularization mechanisms help prevent overfitting, which is crucial when working with high-dimensional datasets that include multiple lagged variables. The successful implementation of XGBoost in NBM condition monitoring includes [**Udo2021Data-DrivenData**]. Figure 3.26 demonstrates an example of the decision tree visualization from the XGBoost ensemble, showing the splitting process and the temperature range in the leaf.

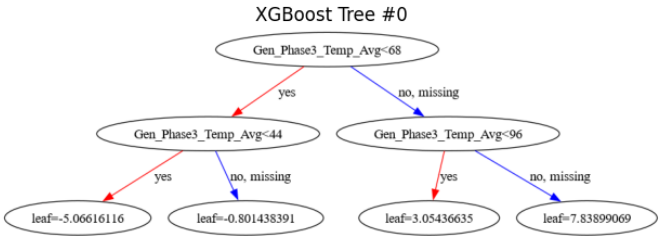

Figure 3.26: Demonstration of XGBoost decision tree structure for generator bearing temperature prediction showing feature splits and leaf values.

Key hyperparameters were configured as follows: learning rate of 0.05 to ensure stable convergence, maximum tree depth of 5 to balance model complexity and generalization, and early stopping with 30 rounds patience to prevent overfitting. The number of estimators was dynamically adjusted based on the feature count using the rule of 10 estimators per feature.

To understand the model's decision-making process and identify the most influential features, SHAP (SHapley Additive exPlanations) values were computed for feature importance analysis. SHAP provides both global feature importance rankings and local explanations for individual predictions, enabling interpretable insights into which operational parameters most strongly influence both bearing temperature predictions.

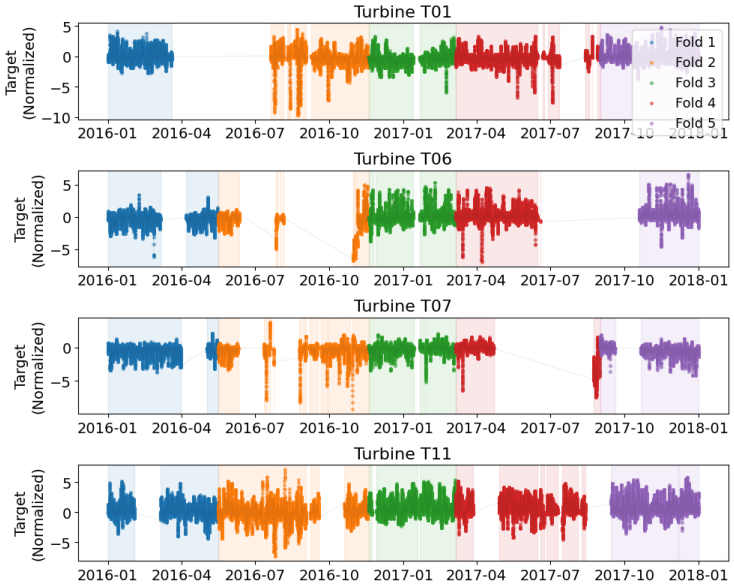

Figure 3.27: Temporal k-fold cross-validation of normalized gearbox bearing temperature, showing five consecutive folds of equal size.

3.4.3 Cross Validation

To evaluate the generalization performance of the NBM, a k-fold cross-validation procedure was implemented. Specifically, the dataset was partitioned into $k = 5$ equal-sized folds. For each iteration, four folds were used for training the model, while the remaining fold was used for validation. This process was repeated five times, ensuring that all folds served as a validation set once. Crucially, data shuffling was disabled during the partitioning process to maintain temporal dependencies inherent in the time series data. A

visualization of the k-fold partition is demonstrated in Figure 3.27, where the data in each fold is maintained in the same amount of data. Figure 3.27 illustrates the temporal partitioning strategy, where each fold contains an equal number of consecutive data points. This cross-validation framework served as the primary method for evaluating how hyperparameter modifications influence model performance and for selecting optimal model configurations prior to final testing on the holdout turbine data.

3.4.4 Feature Engineering and Selection

Lagged Feature

The correlation analysis in Section 3.2.1 revealed that current bearing temperatures are influenced by previous operational conditions, indicating temporal dependencies in the data. To capture these temporal relationships, lagged features were created from the original SCADA signals. For each feature f in the dataset, lagged versions $f_{t-1}, f_{t-2}, ..., f_{t-L}$ were generated by shifting the data with different time lags, where L represents the maximum lag step and t denotes the current time step. These lagged features were additionally included as the training inputs for the XGBoost model.

To determine the number of lag steps L to be included, a systematic evaluation was conducted using cross-validation performance metrics. Lag steps ranging from 1 to 8 were tested, which covers most of the highest correlation lags distribution shown in Figure A.3. Model performance was measured using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as evaluation metrics, as demonstrated in Figure 4.1.

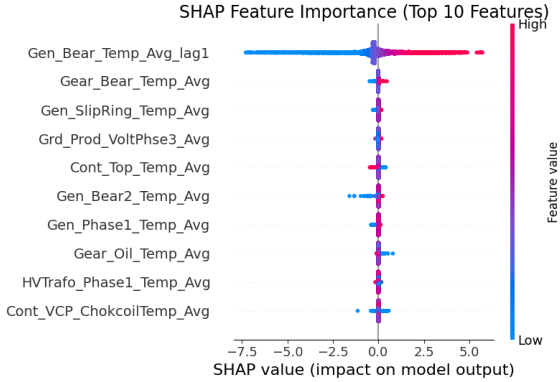

Figure 3.28: SHAP feature importance ranking for generator bearing temperature prediction, ordered by mean absolute SHAP values.

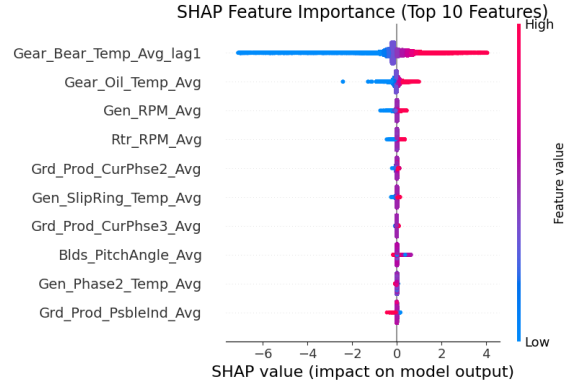

Figure 3.29: SHAP feature importance ranking for gearbox bearing temperature prediction, ordered by mean absolute SHAP values.

SHAP-based Feature Selection

Following lag inputs optimization, feature importance analysis was conducted using SHAP values to identify the most influential features for bearing temperature prediction. SHAP values quantify each feature's contribution to individual predictions by calculating the marginal contribution of each feature across all possible feature combinations, providing an efficiency framework for feature importance [**Lundberg2017APredictions**]. Additionally, SHAP provides both global feature importance rankings and individual prediction explanations, enabling interpretable insights into which operational parameters most strongly influence the target variable. The global feature importance was visualized using a summary plot ranking features by their mean absolute SHAP values, as demonstrated in Figure 3.28 and 3.29 for the top 10 most important features.

To determine the optimal number of features for model training, a wrapper-based feature selection approach was implemented. Features were incrementally added to the model in order of their SHAP importance ranking, and cross-validation performance was evaluated at each step. This systematic evaluation, shown in Figure 4.2, tested up to 20 features to identify the point of diminishing returns where additional features provide minimal performance improvement.

3.4.5 Long Short-Term Memory Neural Network

While XGBoost effectively captures complex relationships through explicit lagged features, LSTM neural networks offer an alternative approach to modeling temporal dependencies in wind turbine SCADA data. LSTMs are a specialized type of recurrent neural network designed to learn long-term dependencies in sequential data, making them particularly suitable for time series prediction tasks.

The key advantage of LSTM networks lies in their ability to automatically learn temporal patterns from sequences of data without requiring explicit lag feature engineering. Instead of manually creating lagged features as done for XGBoost, LSTM models process sequences of consecutive time steps, allowing the network to learn which historical information is most relevant for predicting current bearing temperatures.

The LSTM model architecture employed in this study consists of a single LSTM layer with 64 hidden units, followed by a dropout layer (rate=0.2) for regularization, a dense layer with 32 units and ReLU activation, and a final output layer producing a single predicted value. This relatively simple architecture was chosen to balance model performance with computational efficiency, avoiding the complexity of deeper networks that would require significantly more training time and resources. The input to the LSTM model has dimensions (N, L, F), where N is the number of sequences, L is the sequence length (number of time steps), and F is the number of features per time step. Feature selection for the LSTM model was based on SHAP importance rankings derived from the XGBoost model trained with non-lagged features and lagged target variables.

A significant challenge in applying LSTM models to wind turbine data is ensuring temporal continuity within input sequences. The input healthy state data after pre-processing creates gaps within the time series. To address this, a continuity-aware sequence creation process was implemented that only creates sequences where all time steps are in 10 minutes interval, ensuring that the LSTM learns from continuous temporal patterns.

The LSTM model was trained using the Adam optimizer with a learning rate of 0.001, mean squared error as the loss function, and mean absolute error as an additional monitoring metric. Early stopping was implemented with a patience of 10 epochs to prevent overfitting, and the model was trained for a maximum of 50 epochs. The same standardized data used for XGBoost training was employed to ensure fair comparison between the two approaches.

3.5 Anomaly Detection Framework

Once the normal behavior models are trained, they form the basis for anomaly detection in wind turbine operations. This study employs a CUSUM control chart approach to detect deviations from normal bearing temperature patterns that may indicate failures.

The traditional CUSUM algorithm monitors the cumulative sum of deviations from expected values, making it suitable for detecting small but persistent shifts in bearing temperature patterns. The algorithm maintains two cumulative sums for detecting both posi-

tive and negative deviations:

$$\begin{aligned} S_t^+ &= \max(0,\ S_{t-1}^+ + (X_t - \mu) - k) \\ S_t^- &= \max(0,\ S_{t-1}^- - (X_t - \mu) - k) \end{aligned} \tag{3.6}$$

where S_t^+ and S_t^- are the positive and negative CUSUM scores at time t, X_t represents the prediction error, μ is the expected mean of prediction errors, and k is the reference value controlling the algorithm's sensitivity to small changes. An alarm is triggered when either the CUSUM statistic exceeds the decision threshold h:

$$\text{Alarm} = S_t^+ > h \text{ or } S_t^- > h \tag{3.7}$$

The expected mean μ is set to zero, assuming unbiased predictions from the trained models.

3.5.1 Optimization of Parameters

The CUSUM parameters k and h require optimization to achieve optimal detection performance. The sensitivity parameter k controls the algorithm's responsiveness to small deviations and is typically set between 0.5-2.0 times the standard deviation of the monitoring statistic. The threshold parameter h determines when an alarm is triggered and is optimized to balance detection capability with false alarm rates.

This study employs grid search optimization using known failure events to determine optimal parameter combinations. A suitable searching range was manually determined on the basis of the testing cases and the models. The optimization objective seeks to minimize the false alarm rates while the constraint setting as successfully detection. Additional information, such as the lead time of the successful detection, were also calculated for the understanding of a potential future implementation.

Detection Success Criteria

A detection is considered successful if an alarm occurs within a predefined detection window prior to the actual failure event. This study employs a detection window of 60 days before the failure occurrence, which provides sufficient lead time for maintenance planning while ensuring the detected anomalies are genuinely related to the impending failure.

False Alarm Rate Calculation

False alarm rate is calculated as the frequency of distinct alarm events per day during operation periods, excluding the detection windows around known failures. To avoid inflating false alarm counts due to closely occurring alarms from the same underlying cause, a consolidation logic is implemented by setting multiple alarms occurring within a 24-hour window to be consolidated into a single alarm event. This prevents short-term alarm clusters from artificially increasing the false alarm rate while reflecting practical operational scenarios where maintenance teams would treat closely spaced alarms as a single actionable event.

The false alarm rate is then calculated as:

$$\text{False Alarm Rate} = \frac{\text{Number of Distinct Alarm Events}}{\text{Total Normal Operation Days}} \tag{3.8}$$

where normal operation days exclude periods within 60 days of any known failure event.

Lead Time Calculation

Lead time quantifies the advance warning provided by the anomaly detection system before an actual failure occurs. This metric is crucial for assessing the practical value of the detection system in enabling proactive maintenance interventions.

For each successfully detected failure, the lead time is calculated as the temporal difference between the alarm occurrences within the detection window and the actual failure timestamp:

$$\text{Lead Time} = t_{\text{failure}} - t_{\text{alarm}} \tag{3.9}$$

where t_{failure} is the recorded failure time and t_{alarm} is the timestamp of the CUSUM alarms within the 60-day detection window preceding the failure. When multiple alarms occur within the detection window, the earliest alarm is used to indicate the lead time, providing the maximum possible time for maintenance decisions.

4 Results

This chapter presents comprehensive evaluation results for normal behavior models applied to wind turbine bearing fault detection, systematically examining model performance and detection capabilities under various configurations and practical constraints. The results are structured across two main sections:

Baseline model performance is first established in section 4.1, encompassing cross-validation analysis of temporal dependencies and feature selection to identify optimal configurations for both LSTM and XGBoost architectures. Complete anomaly detection performance is then evaluated in section 4.2, examining feature count impact on detection capability, temporal dependencies influences, systematic CUSUM parameter optimization, target feature engineering effects, preprocessing strategy evaluation, bias correction methodologies, and detection lead time analysis across different model-component combinations.

4.1 Model Performance Evaluation

4.1.1 Temporal Dependency Analysis

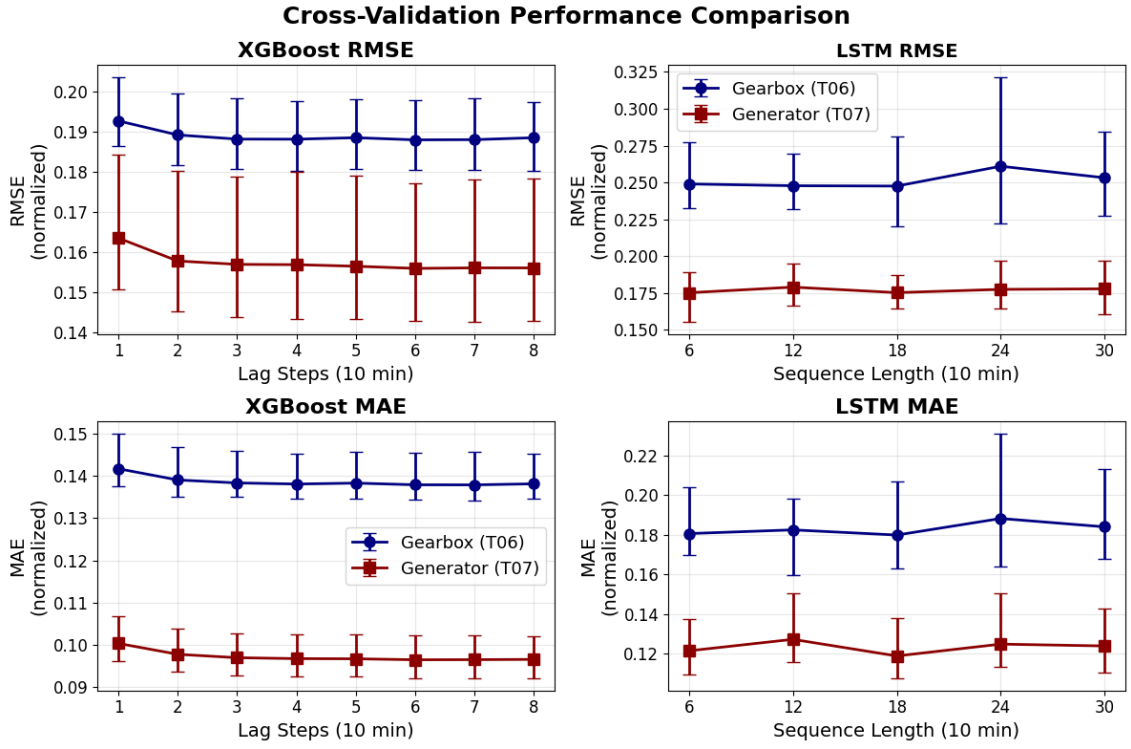

Figure 4.1: Cross-validation performance metrics (RMSE, MAE) versus lag steps for XGBoost model training and sequence length for LSTM model training. All features included.

Figure 4.1 presents the cross-validation performance analysis for both XGBoost lag step optimization and LSTM sequence length optimization across generator and gearbox bearing temperature prediction tasks.

The left column illustrates the relationship between cumulative lag steps and XGBoost model performance. Including additional lag features initially decreases the prediction error, with detectable performance decreasing until a lag step of 3. Beyond this point, diminishing returns are observed, indicating that lag step 3 provides an optimal balance between model performance and computational efficiency. The analysis reveals component-specific performance differences, with generator bearing temperature prediction achieving superior accuracy (RMSE ≈ 0.155) compared to gearbox bearing temperature prediction (RMSE ≈ 0.190). This performance difference suggests that generator bearing temperature exhibits more predictable patterns under the same XGBoost configuration, likely due to differences in the training data characteristics between components.

The right column presents cross-validation results for different sequence lengths using all available features, including a 10-minute lag of the target feature. LSTM performance remains relatively stable across tested sequence lengths, with RMSE and MAE showing minimal variation. Optimal sequence lengths of 6 and 18 time steps achieved the best RMSE and MAE, while longer sequences provided no additional performance improvement. This stability suggests that LSTM effectively captures relevant temporal patterns within shorter sequences, with extended historical context providing limited additional benefit.

The comparative analysis reveals that XGBoost consistently outperforms LSTM across all tested configurations. The optimized XGBoost model achieves approximately 10-15% better performance in both RMSE and MAE metrics compared to the optimized LSTM model. This performance gap indicates that for this specific application, the explicit lag features used by XGBoost may be more effective than LSTM's implicit temporal modeling, possibly due to the challenge of optimizing LSTM hyperparameters for this domain.

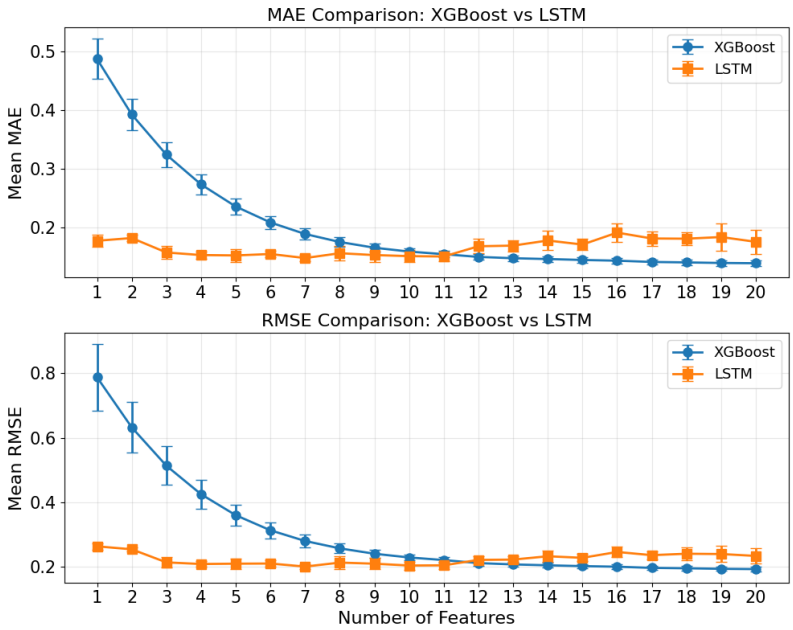

Figure 4.2: Feature selection analysis demonstrating XGBoost performance improvement with additional SHAP-ranked features versus LSTM stability across feature counts for gearbox bearing temperature prediction. LSTM sequence:6, XGBoost lag: 3.

4.1.2 Feature Selection Analysis

Figure 4.2 presents cross-validation performance as features were incrementally added according to SHAP importance rankings. The XGBoost model used SHAP rankings generated with a maximum lag step of 3, while the LSTM model followed the baseline SHAP ranking in Figure 3.29 to ensure fair comparison between temporal modeling approaches.

The XGBoost model demonstrates clear diminishing returns with increasing feature count. Performance improves substantially with initial features, showing MAE reduction from 0.48 (1 feature) to 0.16 (10 features), representing approximately 67% improvement. Beyond 15-17 features, additional features provide minimal benefit, suggesting that the most informative features are captured within this range. This pattern indicates effective feature ranking by SHAP importance and confirms that XGBoost can efficiently leverage the most relevant predictive signals while avoiding overfitting from excessive feature inclusion.

In contrast, the LSTM model exhibits stable performance across different feature counts, maintaining MAE around 0.16-0.19 regardless of the number of features used. This stability suggests that LSTM's temporal modeling capability allows effective utilization of patterns even with limited feature sets.

Based on cross-validation results, component-specific optimal configurations were established. For XGBoost, optimal performance is achieved with lag step 3 and approximately 15-17 features, balancing predictive accuracy with computational efficiency. For LSTM, optimal configuration uses sequence length 6 time steps and maintains stable performance with as few as 7 features, demonstrating efficiency in feature utilization. These configurations provide baseline parameters for subsequent end-to-end testing while avoiding overfitting from excessive parameter complexity.

4.2 Anomaly Detection Performance

This section evaluates complete anomaly detection systems, examining the integration of prediction models with CUSUM analysis under various feature engineering, preprocessing, and optimization strategies to identify optimal configurations for bearing failure detection.

4.2.1 Feature Count Impact

In the feature count impact analysis, CUSUM control charts were implemented with fixed parameters k=0.5 and h=2, where k represents the sensitivity to shifts of 0.5 times the standard deviation and h=2 sets the decision threshold. The choice of parameters k=0.5 and h=2 represents a conservative detection threshold, where k=0.5 provides sensitivity to moderate shifts and h=2 enables early detection with high sensitivity. This baseline setting also serves as a general test case, providing insight into the system's typical reaction and sensitivity under standard operating conditions before fine-tuning these parameters for specific component behavior.

Generator Bearing Temperature

The influence of feature count on anomaly detection performance was evaluated using the generator bearing temperature prediction case (T07), with results presented in Figure 4.3. The analysis encompassed feature counts ranging from 7 to 84, representing the full spectrum from optimal cross-validation configurations to maximum available LSTM features. The feature selection methodology follows the approach described in subsubsection 3.4.4. For LSTM models, features were selected according to the SHAP importance ranking shown in Figure 3.28, which includes the original sensor features plus lagged target features, as the LSTM architecture inherently captures temporal dependencies through its sequence processing mechanism. In contrast, XGBoost models utilized

an expanded feature set that incorporated cumulative lag steps up to 3 time steps for each input feature.

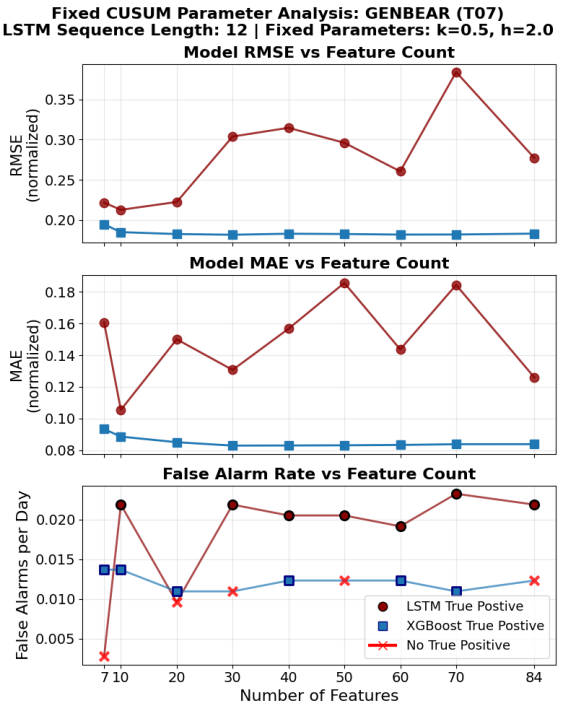

Figure 4.3: Feature count impact on generator bearing temperature. Fixed CUSUM parameters: k=0.5, h=2.

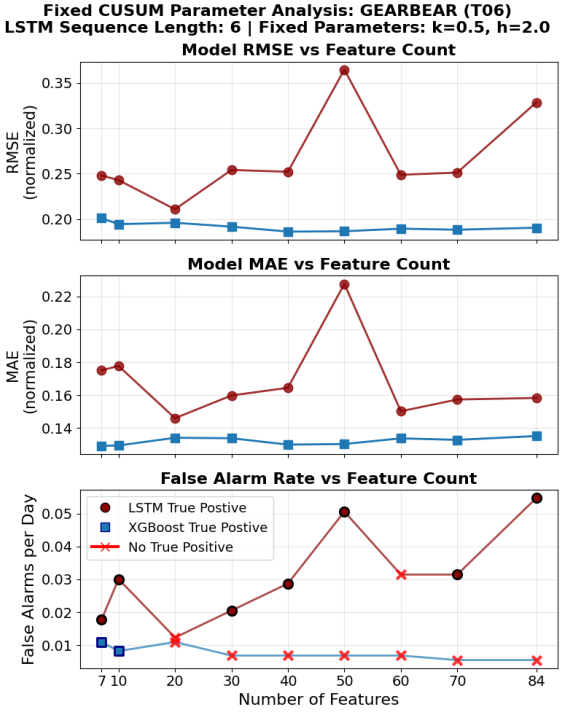

Figure 4.4: Feature count impact on gearbox bearing temperature. Fixed CUSUM parameters: k=0.5, h=2.

Based on Figure 4.3, the LSTM model exhibits generally increasing prediction errors (RMSE and MAE) as the feature count increases, suggesting potential overfitting with higher-dimensional inputs. In contrast, the XGBoost model maintains relatively stable prediction performance across all tested feature counts, demonstrating diminishing returns in prediction improvement.

More importantly, the false alarm rate in the third panel reveals different performance patterns between models. Since there is only one failure event per test component, the successful failure detections occurring within the predefined 60-day detection window are indicated by filled markers (circles for LSTM, squares for XGBoost), while failed detections are marked with red crosses. The XGBoost model demonstrates detection capability in multiple feature configurations with the lower false alarm rates (0.010-0.013 alarms per day). In contrast, the LSTM model shows higher false alarm rates (0.018-0.023 alarms per day) across different numbers of features.

To illustrate the detection mechanisms, Figure 4.5 compares LSTM model performance using two representative configurations: 10 features (lowest prediction residuals) and 60 features (lowest false alarm rate). The gray shaded region indicates the 60-day pre-failure detection window, during which anomaly detection is considered successful. The analysis focuses on a 3-month period to demonstrate CUSUM score accumulation patterns leading to failure detection. Both feature configurations successfully detect the target generator bearing failure, with CUSUM values exceeding the threshold (h=2) within the detection window. The failure detection manifests as sudden spikes in prediction errors, resulting

in rapid CUSUM accumulation that triggers anomaly alerts. Notably, both models exhibit distinct negative prediction residuals following the actual generator bearing failure occurrence, confirming the models' sensitivity to generator degradation after failure. However, this behavior is not ideal for a warning system and will be counted as one false alarm.

The time series reveals additional CUSUM threshold exceedances corresponding to hydraulic group oil leakage events occurring outside the detection window. While these represent genuine system anomalies, they are classified as false alarms for bearing failure prediction purposes, highlighting the challenge of classifying between different failure modes.

Similarly, XGBoost model detection performance is evaluated in Figure 4.6, comparing configurations with 40 features (lowest prediction residuals) and 70 features (lowest false alarm rate). The XGBoost models demonstrate lower prediction errors compared to LSTM, resulting in slower CUSUM accumulation at the same sensitivity level (k=0.5). Nonetheless, both feature configurations successfully detect the bearing failure within the target window, triggered by the sudden increase in prediction error observed around 2017-08-10.

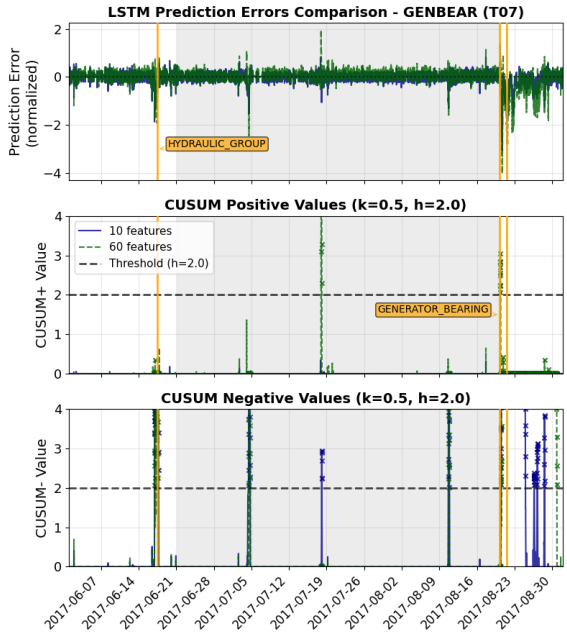

Figure 4.5: LSTM model anomaly detection time series comparison for generator bearing temperature.

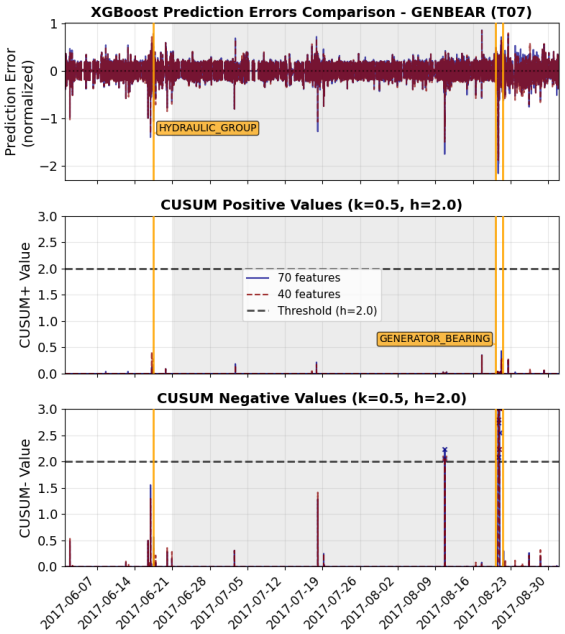

Figure 4.6: XGBoost model anomaly detection time series comparison for generator bearing temperature.

Gearbox Bearing Temperature

The same feature count impact analysis was performed with the gearbox bearing temperature prediction case (T06), with results illustrated in Figure 4.4. The analysis reveals similarities to the generator bearing case, where the LSTM model exhibits generally higher prediction errors compared to XGBoost across most feature configurations. For LSTM predictions, the model with 20 features achieves optimal performance with the lowest RMSE of 0.21 and MAE of 0.14, while the 50-feature configuration shows the most degraded performance with significantly higher error metrics (RMSE: 0.37, MAE: 0.23). The XGBoost model maintains stable performance across all tested feature counts, with RMSE values consistently around 0.19-0.20 and MAE values around 0.13-0.14.

The false alarm panel reveals challenging detection conditions for gearbox bearing failures. The XGBoost model demonstrates limited detection capability, with successful detection occurring only at 7 and 10 feature configurations, while most other feature counts fail to detect target failure within the 60-day window. The LSTM model shows successful detection in most feature configurations except for 20 and 60 features, but with consistently higher false alarm rates compared to XGBoost models.

The gearbox bearing case presents more challenges in failure detection than the generator bearing, with both the LSTM and XGBoost models showing higher false alarm rates. This suggests that the gearbox bearing may require another CUSUM parameter adjustment for different parameters.

Figure 4.7 illustrates LSTM detection performance using 7 features (lowest false alarm rate among successful configurations) and 20 features (lowest prediction errors). The analysis reveals that even though the 20-feature configuration achieves better prediction performance, the CUSUM values fail to reach the threshold. However, the CUSUM+ value around 2017-08-20 nearly reaches the threshold, demonstrating that the CUSUM parameter tuning can influence the detection results. In addition, the results show that lower prediction errors are not always optimal for anomaly detection, as overfitted models may predict degradation patterns that should instead manifest as accumulating CUSUM scores for failure detection.

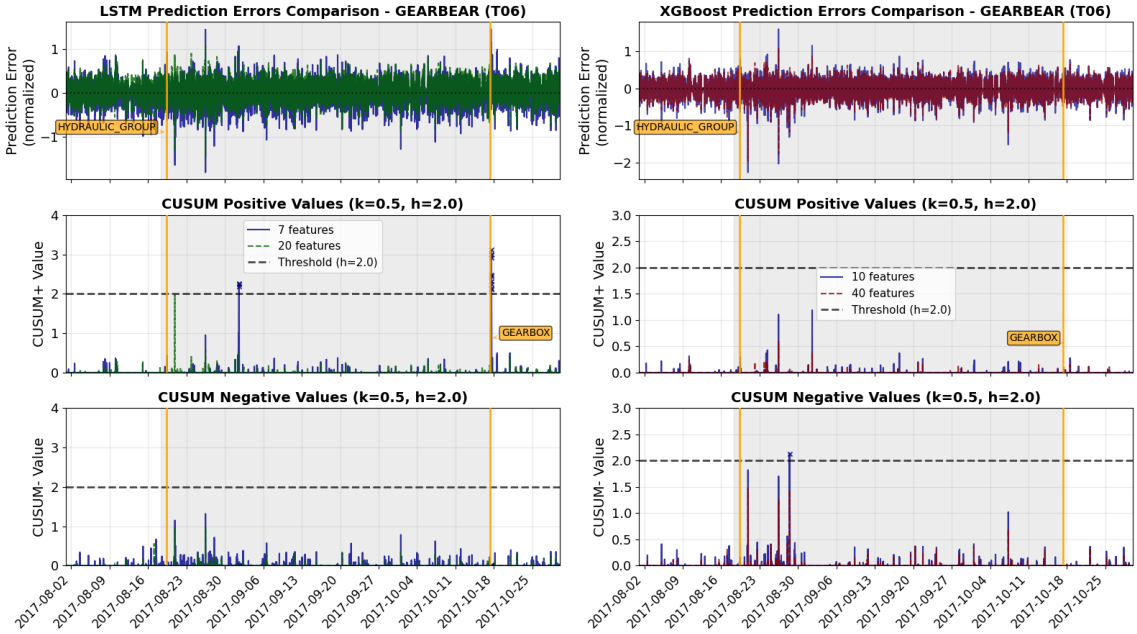

Figure 4.7: LSTM model anomaly detection time series comparison for gearbox bearing temperature.

Figure 4.8: XGBoost model anomaly detection time series comparison for gearbox bearing temperature.

Figure 4.8 presents XGBoost results for 10 features (lowest false alarm rate among successful configurations) and 40 features (lowest prediction errors). The 10-feature configuration successfully identifies the gearbox bearing failure through rapid error accumulation, while the 40-feature model fails to accumulate sufficient CUSUM values within the detection window. This again demonstrates the importance of finding the right balance between the prediction accuracy while maintain the ability to not over predict the degradation behavior.

Overall, the results for feature count impact indicate that optimal feature counts are component-specific: generator bearing monitoring performs well with 10-70 features for LSTM and 20-84 features for XGBoost, while gearbox bearing detection requires careful feature selection, with successful configurations limited to 7-10 features for XGBoost and specific configurations for LSTM. The analysis also reveals that gearbox bearing failures may require alternative CUSUM parameter optimization, as evidenced by near-threshold CUSUM accumulation in several failed detection cases.

The optimal feature counts were selected based on minimum false alarm rates among successful detection configurations, as summarized in Table 4.1.

Table 4.1: Optimal feature counts based on minimum false alarm rates for feature count testing. CUSUM parameters: k=0.5, h=2

Component	Model	Optimal Features	Lead Time (hrs)	False Alarms
Generator Bearing	LSTM	60	1122.1	14
	XGBoost	20	228.0	8
Gearbox Bearing	LSTM	7	1099.6	13
	XGBoost	10	1197.6	6

4.2.2 Model Temporal Effect

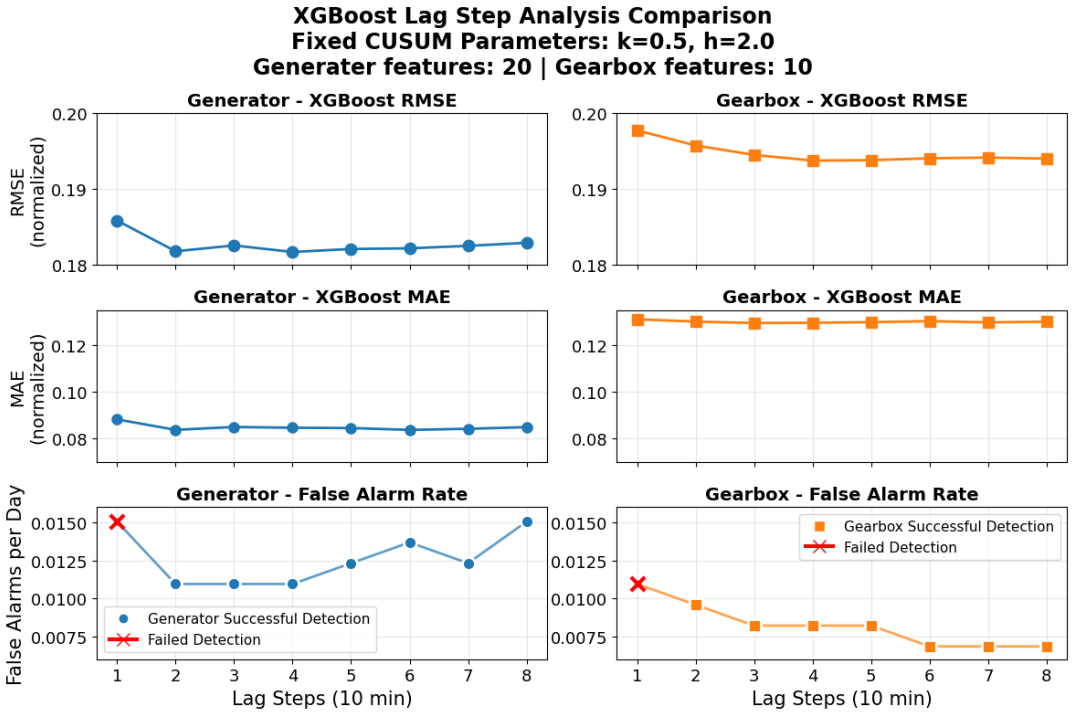

Figure 4.9: XGBoost lag step analysis for generator (20 features) and gearbox (10 features) bearing temperature prediction. Fixed CUSUM parameters: k=0.5, h=2.

Using the optimal feature counts identified in the previous analysis, the temporal influence of model configurations was further evaluated to determine the optimal lag steps for XGBoost models and sequence lengths for LSTM models.

The detection performance of XGBoost models was evaluated using the optimal feature

counts from Table 4.1, with results illustrated in Figure 4.9. To maintain consistent feature dimensionality across different lag configurations, separate SHAP importance rankings were generated for each lag step, and the optimal number of features was selected based on these newly generated feature ranking lists.

The RMSE and MAE evaluated on the test turbine demonstrate similar performance across all lag steps for both generator and gearbox bearing predictions, with values slightly decreasing as more lag steps are included. However, both test cases failed to detect failures when using a lag step of 1, indicating insufficient temporal context for effective anomaly detection. The false alarm rate patterns differ between components: the generator bearing shows an upward trend in false alarms as lag steps increase, while the gearbox bearing demonstrates decreasing false alarm rates with increased lag steps.

Figure 4.10 and 4.11 illustrate how different lag steps influence CUSUM detection performance. The differences in false alarm counts can be observed when CUSUM values approach the threshold, such as the event around 2017-07-01 for the generator bearing case. Therefore, an optimal lag step should be achieved to reduce unwanted alarms that occur near the threshold. Based on these comparative results, the optimal lag steps for XGBoost models are 3 and 6 for the generator and gearbox test cases, respectively.

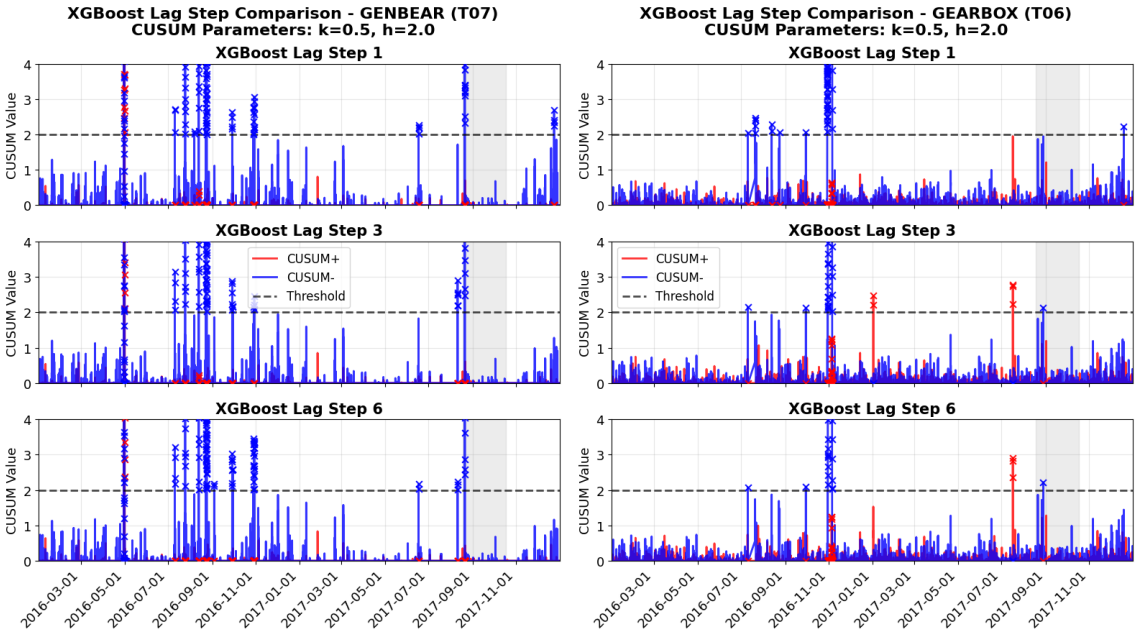

Figure 4.10: Generator bearing XGBoost model CUSUM detection comparison across different lag steps.

Figure 4.11: Gearbox bearing XGBoost model CUSUM detection comparison across different lag steps.

For LSTM models, the influence of sequence length on detection performance was evaluated using the optimal feature counts, with results presented in Figure 4.12. Both RMSE and MAE show no clear trend in response to different sequence lengths, and the false alarm rate similarly shows no obvious relationship with sequence length. For the generator bearing case, all tested sequence lengths successfully detect failures within the 60-day detection window, as demonstrated in Figure 4.13. Several rapid accumulations of CUSUM values can be observed throughout the time series, indicating that CUSUM parameters could be further optimized to reduce false alarms. In contrast, the gearbox bearing case shows that successful detection capability was lost as sequence length

increased, leading to CUSUM values remaining below the threshold, as illustrated in Figure 4.14. This result demonstrates the significant influence of sequence length on CUSUM-based detection performance.

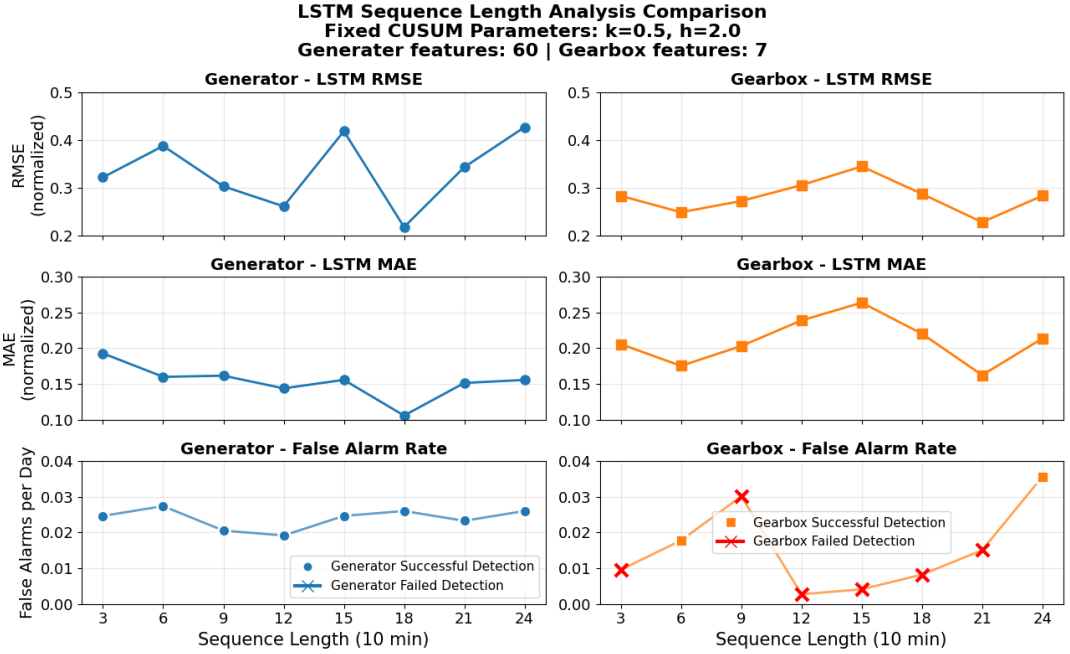

Figure 4.12: LSTM sequence length analysis for generator (60 features) and gearbox (7 features) bearing temperature prediction. Fixed CUSUM parameters: k=0.5, h=2.

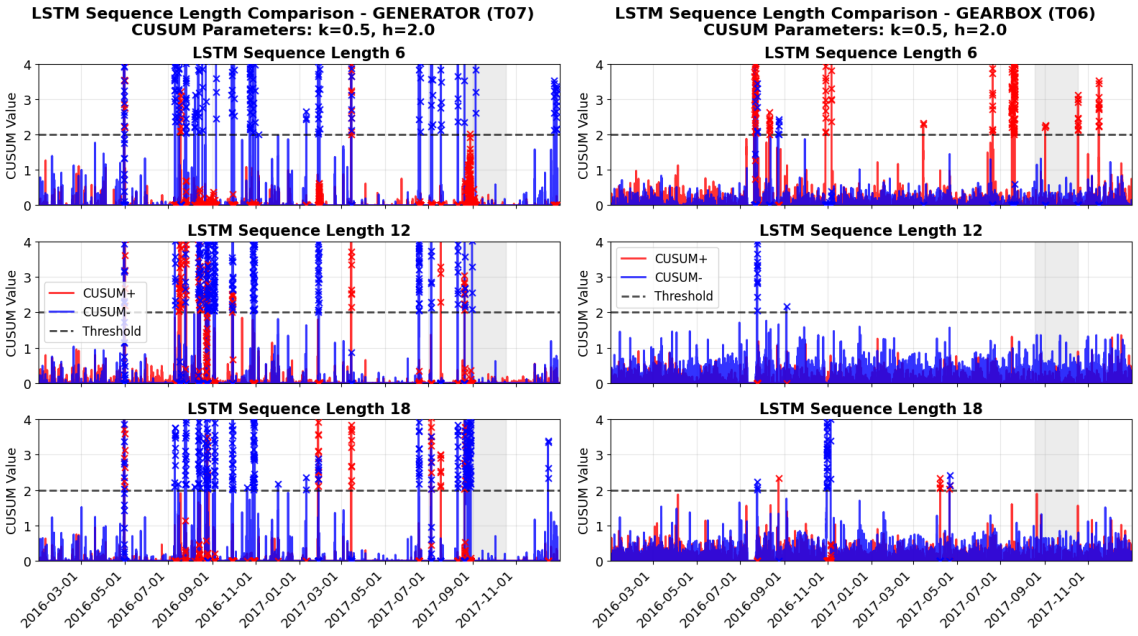

Figure 4.13: Generator bearing LSTM model CUSUM detection comparison across different sequence lengths.

Figure 4.14: Gearbox bearing LSTM model CUSUM detection comparison across different sequence lengths.

Table 4.2 summarizes the optimal lag steps and sequence lengths based on the feature

counts determined in the previous analysis. The results show that optimal temporal configurations are model-specific: the XGBoost models show variation of the CUSUM value by the change of lag steps, while LSTM models illustrate no clear trend with the sequence length changes. Notably, the XGBoost model for gearbox bearing achieved one fewer false alarm when using a lag step of 6 instead of 3 with the same number of features.

Table 4.2: Optimal temporal configurations based on minimum false alarm rates

Component	Model	Features	Sequence length / Lag Step	Lead Time (hrs)	Total False Alarms
Generator	LSTM	60	12	1122.1	14
Bearing	XGBoost	20	3	228.0	8
Gearbox	LSTM	7	6	1099.6	13
Bearing	XGBoost	10	6	1197.6	5

4.2.3 CUSUM Parameters Optimization

With the optimal number of features and temporal configurations (lag steps/sequence lengths) established, a comprehensive grid search optimization was performed to determine the optimal CUSUM parameters k and h for both XGBoost and LSTM models. The objective was to minimize the false alarm rate (false alarms) while maintaining successful failure detection within the 60-day detection window.

Generator Bearing Parameter Optimization

The grid search results for generator bearing detection are illustrated in Figure 4.15 and 4.16 for LSTM and XGBoost models, respectively. The visualization uses a color-coded approach where red areas indicate parameter combinations that failed to raise alarms within the 60-day detection period, while successful detection configurations are colored according to their false alarm rates. The optimal parameter combinations correspond to the lowest false alarm rates among successful detection configurations.

The time series analysis of the optimized CUSUM parameters, shown in Figure 4.17 and 4.18, reveals distinct characteristics in the prediction error patterns. For the LSTM model, the prediction errors exhibit several sudden high-magnitude peaks throughout the time series. These abrupt error spikes trigger rapid CUSUM accumulation, leading to correspondingly high CUSUM values that exceed the detection threshold.

The optimization process addresses the challenge of distinguishing between failure-related error peaks and noise-induced fluctuations. When prediction errors show sudden peaks within the detection window, the optimization increases the sensitivity parameter k to filter out smaller error fluctuations that might generate false alarms. This approach transforms the detection mechanism to resemble threshold-based detection rather than gradual error accumulation, due to the higher k value. The critical balance lies in setting the threshold h such that the CUSUM peak within the detection window maintains sufficient magnitude to exceed the threshold, while naturally suppressing alarms outside the detection window.

Similarly, the XGBoost model exhibits a pronounced prediction error event occurring near the actual failure. The optimization process adjusts both k and h parameters to target this specific peak, positioning the threshold appropriately so that other false alarm events fall below the detection level.

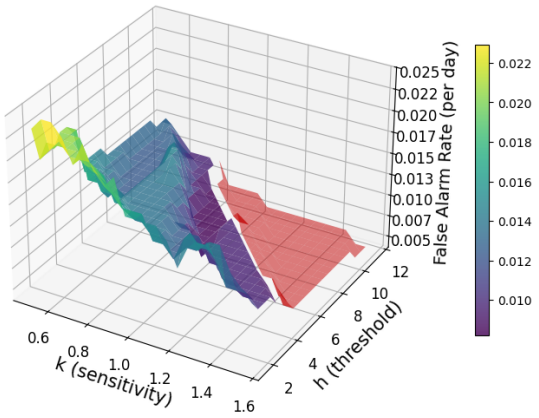

Figure 4.15: LSTM model CUSUM parameter optimization for generator bearing (T07, 60 features, sequence length 12).

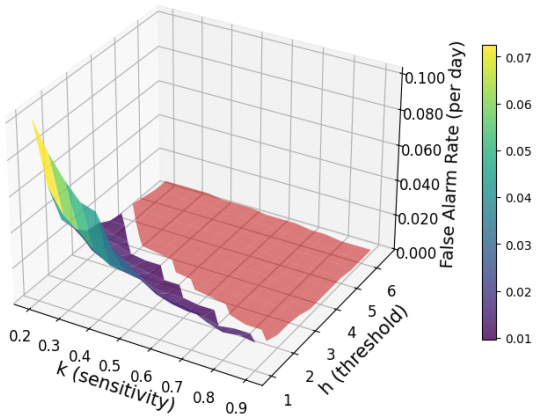

Figure 4.16: XGBoost model CUSUM parameter optimization for generator bearing (T07, 20 features, lag step 3).

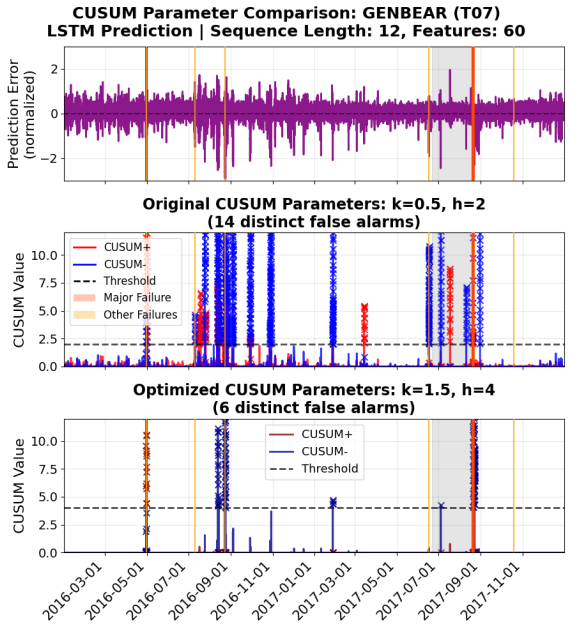

Figure 4.17: Generator bearing LSTM model CUSUM detection with optimized parameters.

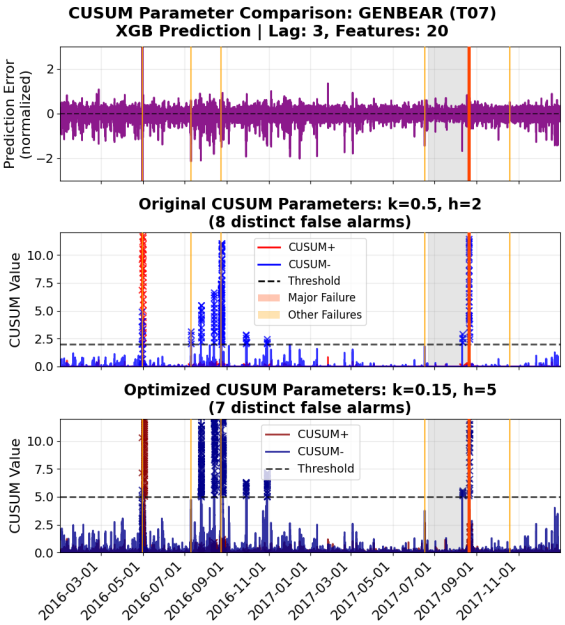

Figure 4.18: Generator bearing XGBoost model CUSUM detection with optimized parameters.

Gearbox Bearing

The same grid search methodology was applied to gearbox bearing detection, with results presented in Figure 4.19 and 4.20. The optimization challenges differ from the generator bearing case. The LSTM model does not exhibit distinct high-magnitude errors within the detection period, as demonstrated in the corresponding time series analysis in Figure 4.19. Consequently, the optimization selects a smaller sensitivity parameter k to facilitate easier error accumulation over time. To balance detection capability with false alarm reduction, the threshold h is increased to a level where the accumulated CUSUM value within the detection window just exceeds the threshold. This fine-tuning resulted in a reduction of one false alarm.

For the XGBoost model, the sensitivity k was similarly decreased to promote more pronounced CUSUM accumulation. The combination of lower k and higher h values successfully reduced false alarms from 5 to 3.

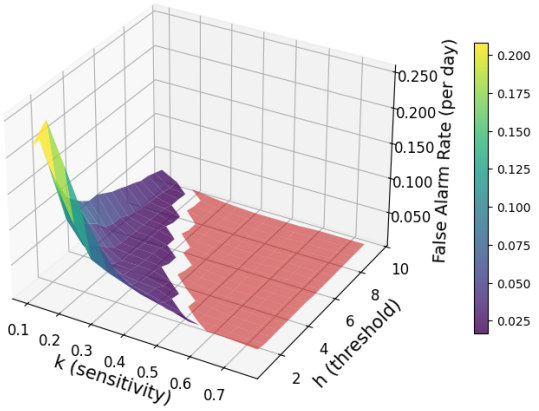

Figure 4.19: LSTM model CUSUM parameter optimization for gearbox bearing (T06, 7 features, sequence length 6).

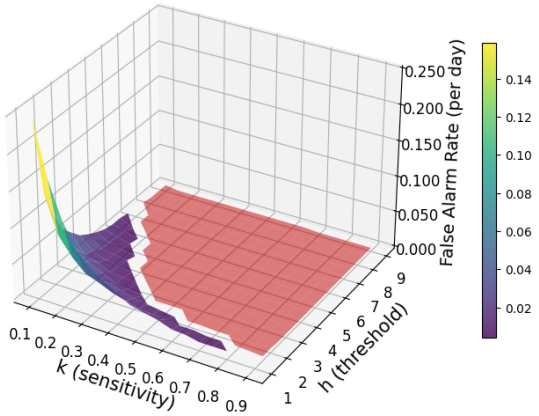

Figure 4.20: XGBoost model CUSUM parameter optimization for gearbox bearing (T06, 10 features, lag step 6).

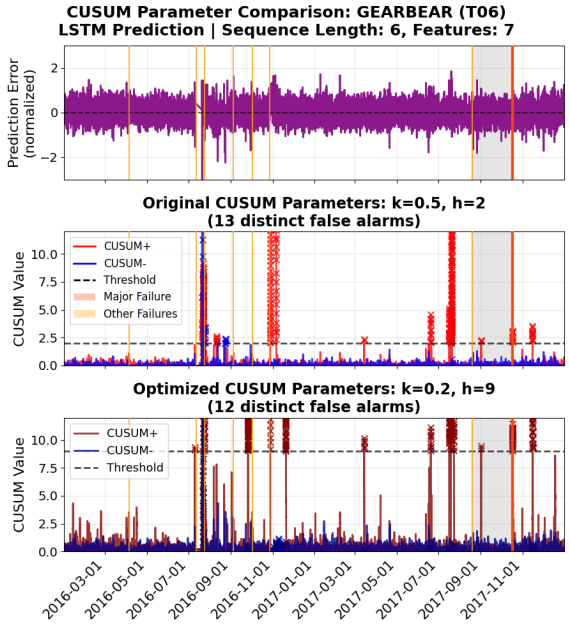

Figure 4.21: Gearbox bearing LSTM model CUSUM detection with optimized parameters.

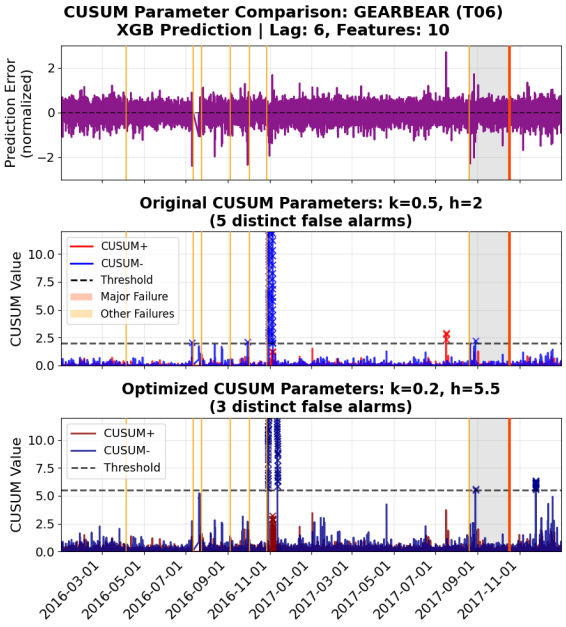

Figure 4.22: Gearbox bearing XGBoost model CUSUM detection with optimized parameters.

Table 3 presents the final optimization results, showing the optimal CUSUM parameters and corresponding false alarm performance. The optimization achieved substantial improvements across all model-component combinations. The LSTM model for generator bearing detection achieved the most significant improvement, reducing false alarms from 14 to 6, while the XGBoost model showed a modest reduction from 8 to 7 alarms. For gearbox bearing detection, the LSTM model decreased false alarms from 13 to 12, while

the XGBoost model demonstrated substantial improvement, reducing false alarms from 5 to 3.

The optimization results reveal model-specific parameter requirements: LSTM model of generator bearing detection benefits from higher sensitivity values to capture sudden failure signatures, while gearbox bearing detection requires lower sensitivity values to accumulate gradual degradation patterns. The threshold values (h) also vary significantly, ranging from 4-5 for generator bearings to 5.5-9 for gearbox bearings, reflecting the different CUSUM accumulation characteristics of each failure mode.

Table 4.3: Optimal CUSUM parameters and performance after grid search optimization

Component	Model	CUSUM k	CUSUM h	Lead Time (hrs)	Total False Alarms
Generator	LSTM	1.5	4	1120.6	6
Bearing	XGBoost	0.15	5	227.3	7
Gearbox	LSTM	0.2	9	1099.0	12
Bearing	XGBoost	0.2	5.5	1195.6	3

4.2.4 Target Feature Analysis

After establishing optimal model configurations, additional experiments were conducted to evaluate the sensitivity of model performance to different input features, particularly the impact of including lagged target variables and different target input features.

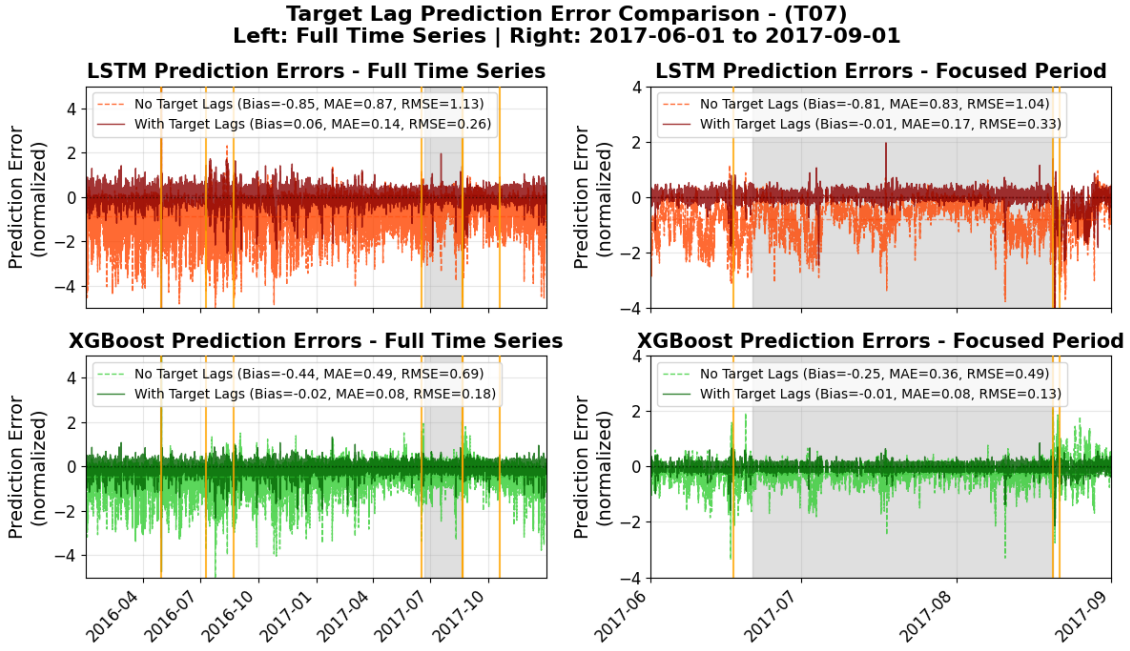

Figure 4.23: Generator bearing temperature prediction comparison with and without lagged target features. The left column shows the full time series, and the right column focuses on the pre-failure period.

Target Lagged Input Feature

The SHAP importance analysis in Figure 3.28 and 3.29 revealed that lagged target variables consistently ranked as the most important features, significantly exceeding the importance of all other input variables. However, this high reliance on lagged target features

raised concerns about the models' ability to detect emerging degradation patterns, as high prediction accuracy might mask early signs of failure.

To investigate this phenomenon, a study was conducted in which models were trained both with and without lagged target features as inputs. New SHAP importance rankings (excluding the target lag feature) were generated for both the LSTM and XGBoost models, as illustrated in Figure A.4 and Figure A.5 in the Appendix.

For generator bearing temperature prediction, excluding lagged target features resulted in substantial performance degradation for both models. As shown in Figure 4.23, the MAE increased dramatically: from 0.14 to 0.87 for LSTM and from 0.09 to 0.49 for XGBoost. RMSE also rose significantly, from 0.26 to 1.13 for LSTM and from 0.18 to 0.69 for XGBoost. More critically, both models exhibited significant systematic bias when lagged target features were excluded. The prediction errors were no longer centered around zero, violating a key assumption of CUSUM-based anomaly detection algorithms. This bias obscured the models' ability to capture degradation patterns, as the large systematic offset dominated the prediction errors.

The pre-failure period plot further highlighted this limitation. The exclusion of target lagged input revealed the underlying challenge: the models struggled to capture subtle degradation patterns due to biased prediction.

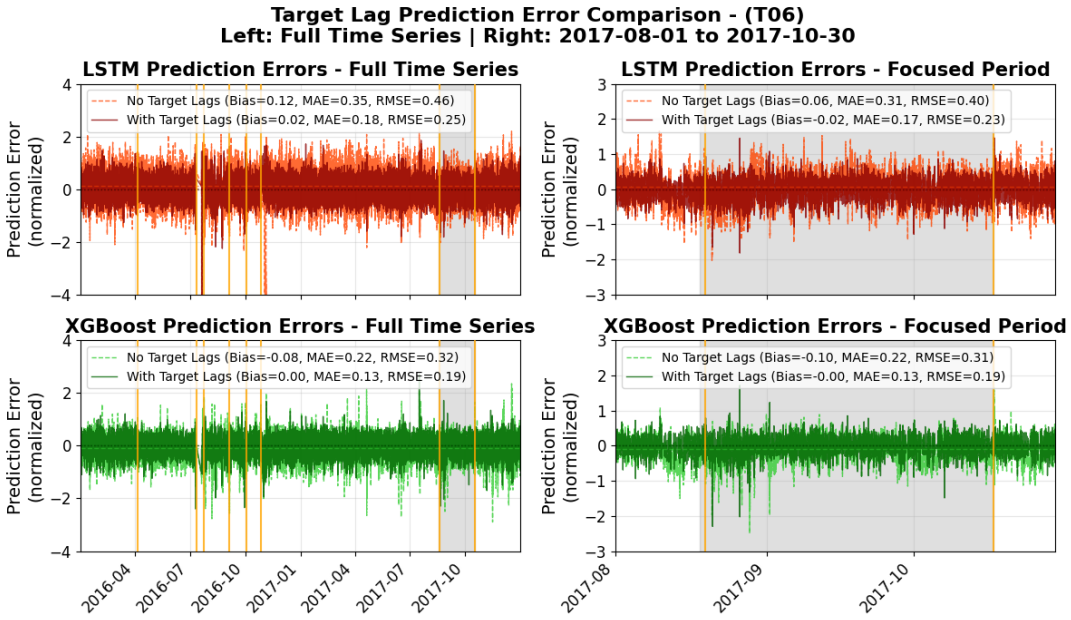

Figure 4.24: Gearbox bearing temperature prediction comparison with and without lagged target features. The left column shows the full time series, and the right column focuses on the pre-failure period.

The gearbox bearing case showed similar but less pronounced effects. MAE increased from 0.18 to 0.35 for LSTM and from 0.13 to 0.22 for XGBoost. RMSE also rose—from 0.25 to 0.46 for LSTM and from 0.19 to 0.32 for XGBoost. Although the performance degradation was less severe than for generator bearings, both models still failed to maintain zero-centered predictions and could not effectively model degradation patterns without lagged target information.

The results, together with the analysis from the previous sections, demonstrate a trade-off

in predictive maintenance applications, where including lagged target features improves prediction accuracy, but may suppress early degradation signals that are crucial for failure prediction. Conversely, excluding lagged target features may introduce systematic bias that violates CUSUM assumptions, requiring bias correction methods to enable effective anomaly detection. Moreover, generator bearings exhibited greater sensitivity to the exclusion of lagged target inputs compared to gearbox bearings, suggesting differences in underlying failure mechanisms or data characteristics between the two components.

These findings suggest that bias-adjustment methods should be implemented to modify prediction errors, ensuring they meet the zero-centered assumption required for effective CUSUM-based failure detection while maintaining the ability to detect emerging degradation patterns.

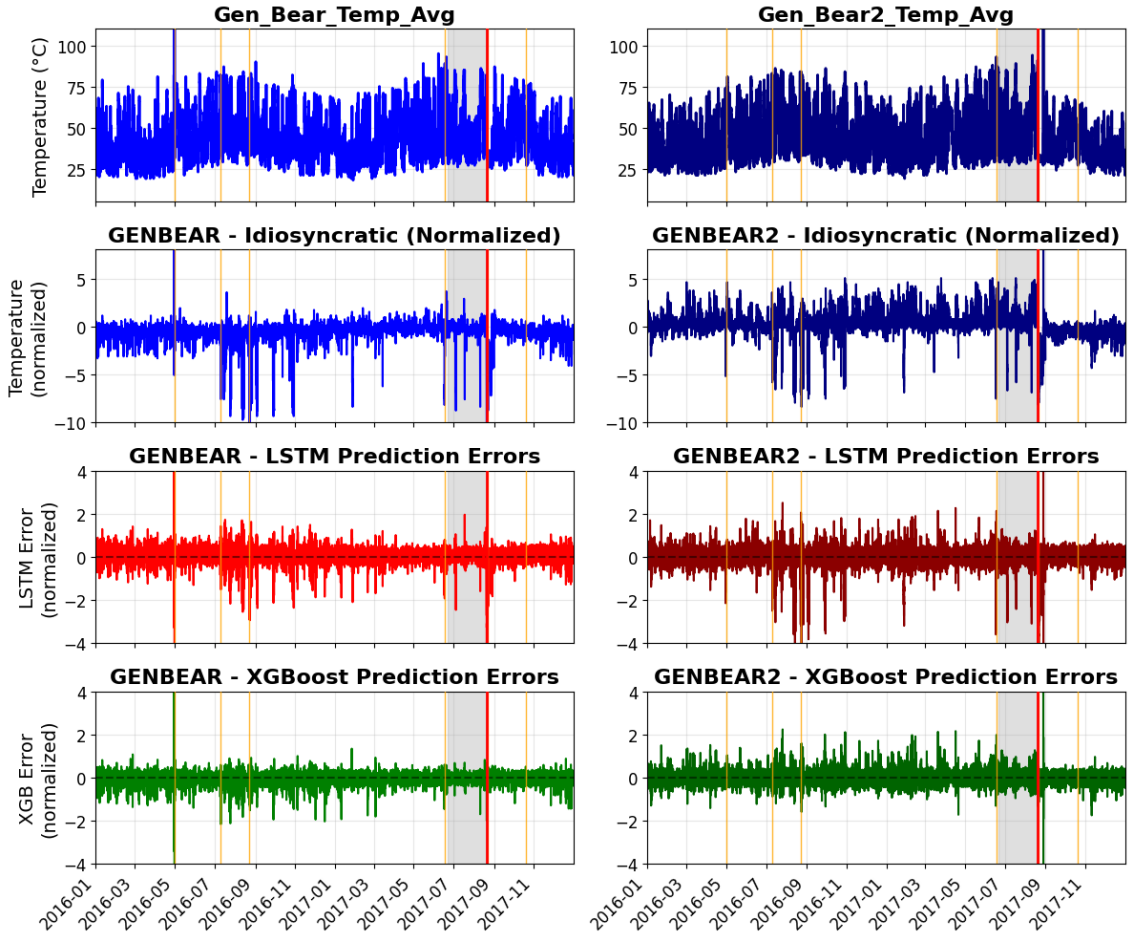

Figure 4.25: Generator bearing temperature prediction comparison between non-drive and drive end measurements. Shows temperature measurements, fleet median normalized values, and prediction errors for both LSTM and XGBoost models. The model configuration follows the optimal result in Table 4.2.

Target Feature Selection Analysis

To evaluate the impact of target feature selection on model performance and failure detection capabilities, a comparative analysis was conducted using different generator bearing temperature measurements. The SCADA dataset includes temperature measurements from both the non-drive end ”Gen_Bear_Temp_Avg” and drive end ”Gen_Bear2_Temp_Avg” of the generator bearing, providing an opportunity to assess how sensor location affects predictive performance. The same optimal model configuration established in previous

sections was applied to both target features to ensure a fair comparison. This analysis examined both prediction accuracy and CUSUM-based failure detection performance when switching from the non-drive end (baseline) to the drive end temperature measurement.

Figure 4.25 presents a comprehensive comparison showing Gen_Bear2_Temp_Avg measurements, fleet median normalized values, and prediction errors for both models. The analysis reveals different characteristics for each temperature sensor location, where both sensors captured significant temperature anomalies at different time periods: Gen_Bear_Temp_Avg exhibited a major outlier during the first generator bearing replacement on April 30, 2016, while Gen_Bear2_Temp_Avg showed a pronounced spike after the generator bearing damage on August 21, 2017. These temperature spikes resulted in substantial deviations from fleet median values, creating high-magnitude prediction errors that appeared as false alarms since they occurred outside the 60-day detection window.

Additionally, switching from non-drive to drive end temperature measurements resulted in significant performance degradation for both models, as shown in Table 4.4. Comparing the prediction error for both models, LSTM struggled to predict sudden, high-magnitude temperature deviations characteristic, while XGBoost demonstrated better capability to handle the normalized idiosyncratic values.

Model	**Metric**	**Gen_Bear_Temp_Avg** [-]	**Gen_Bear2_Temp_Avg** [-]
LSTM	RMSE	0.261	0.534
LSTM	MAE	0.144	0.174
XGBoost	RMSE	0.183	0.433
XGBoost	MAE	0.085	0.110

Table 4.4: Performance metrics for LSTM and XGBoost models. The model configuration follows the optimal result in Table 4.2.

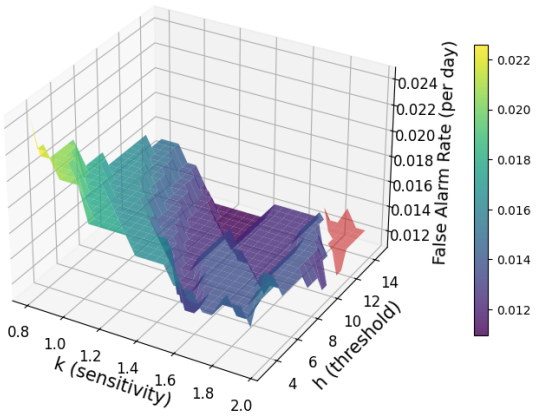

Figure 4.26: LSTM CUSUM parameter optimization grid search for drive end generator bearing temperature.

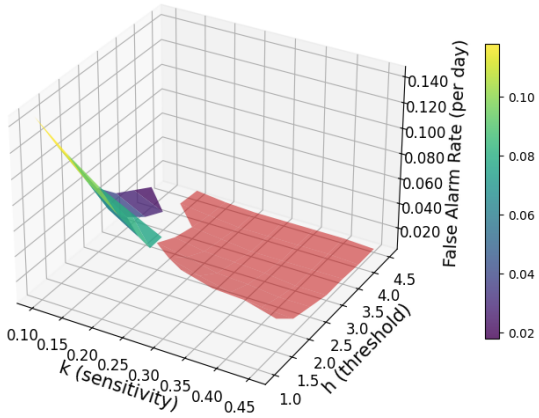

Figure 4.27: XGBoost CUSUM parameter optimization grid search for drive end generator bearing temperature.

Following the prediction performance analysis, CUSUM parameter optimization was performed to achieve optimal failure detection for the drive end temperature target. Figure 4.26 and 4.27 show the grid search optimization results for LSTM and XGBoost models, respectively. The optimization revealed different parameter requirements compared

to the non-drive end analysis. For LSTM detection, the middle panel in Figure 4.28 uses parameters optimized for the non-drive end. The performance generated similar baseline false alarms plus additional alarms around the transformer refrigeration replacement on August 23, 2016. After CUSUM parameter optimization, the false alarms were reduced to 8 events, but still exceeded the 6 false alarms achieved with non-drive end temperature. The additional false alarms at the end of 2016 resulted in inferior detection performance compared to the baseline target feature.

As for the XGBoost detection performance in Figure 4.29, the prediction errors showed relatively consistent variation patterns. The pre-failure degradation patterns become less distinct, requiring threshold adjustments to achieve successful detection within the 60-day window. After parameter optimization, the false alarms increased to 13 events, significantly higher than the 7 false alarms achieved with the non-drive end temperature.

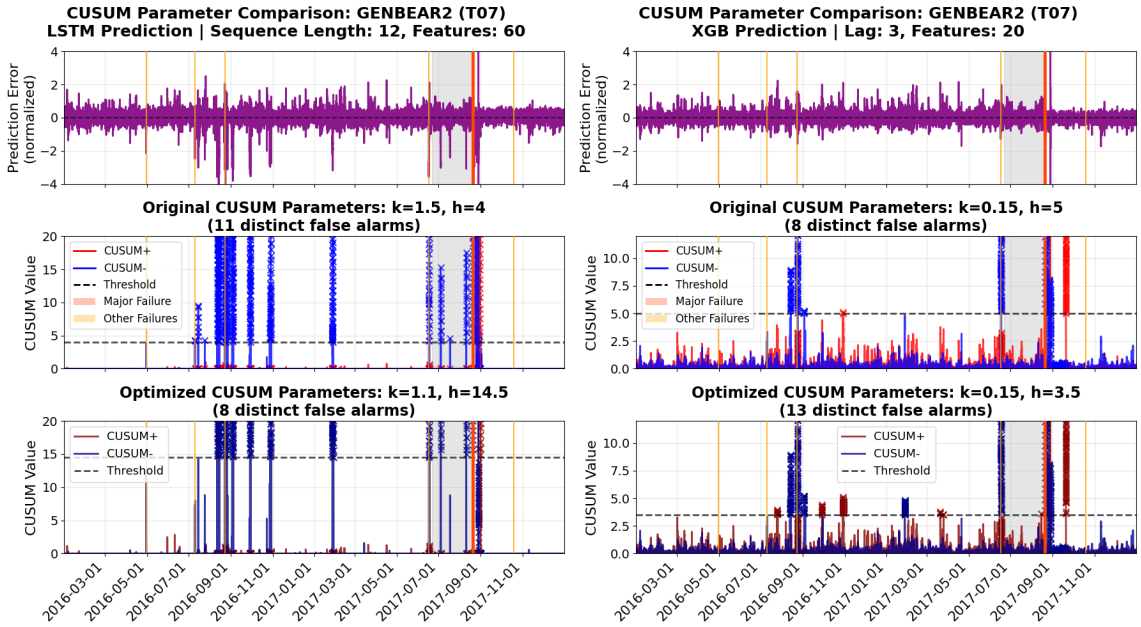

Figure 4.28: LSTM CUSUM detection comparison for drive end temperature with original and optimized parameters.

Figure 4.29: XGBoost CUSUM detection comparison for drive end temperature with original and optimized parameters.

The target feature comparison reveals that the drive end temperature measurements are more challenging to predict, requiring another configuration optimization. Both LSTM and XGBoost showed substantial performance degradation when switching target features from non-drive end to drive end, with RMSE increases exceeding 100% for both models. In addition, while CUSUM parameter optimization could partially compensate for reduced prediction accuracy, it resulted in higher false alarm rates compared to the optimal non-drive end configuration. These results demonstrate that target feature selection significantly impacts both prediction accuracy and anomaly detection performance, emphasizing the importance of careful sensor selection and the potential value of multi-sensor fusion approaches.

4.2.5 Median Filtering Analysis

This section evaluates the impact of using temperature deviation from fleet median values as the target feature, comparing median-filtered against raw temperature measurements.

This preprocessing approach aims to isolate turbine-specific temperature anomalies by removing fleet-wide baseline variations. Two preprocessing approaches were compared:

1. Median filtering: Target feature defined as deviation from fleet median temperature
2. No median filtering: Target feature using raw temperature measurements

Both approaches used identical model configurations and separate Standard Scalers fitted on healthy data for each preprocessing method.

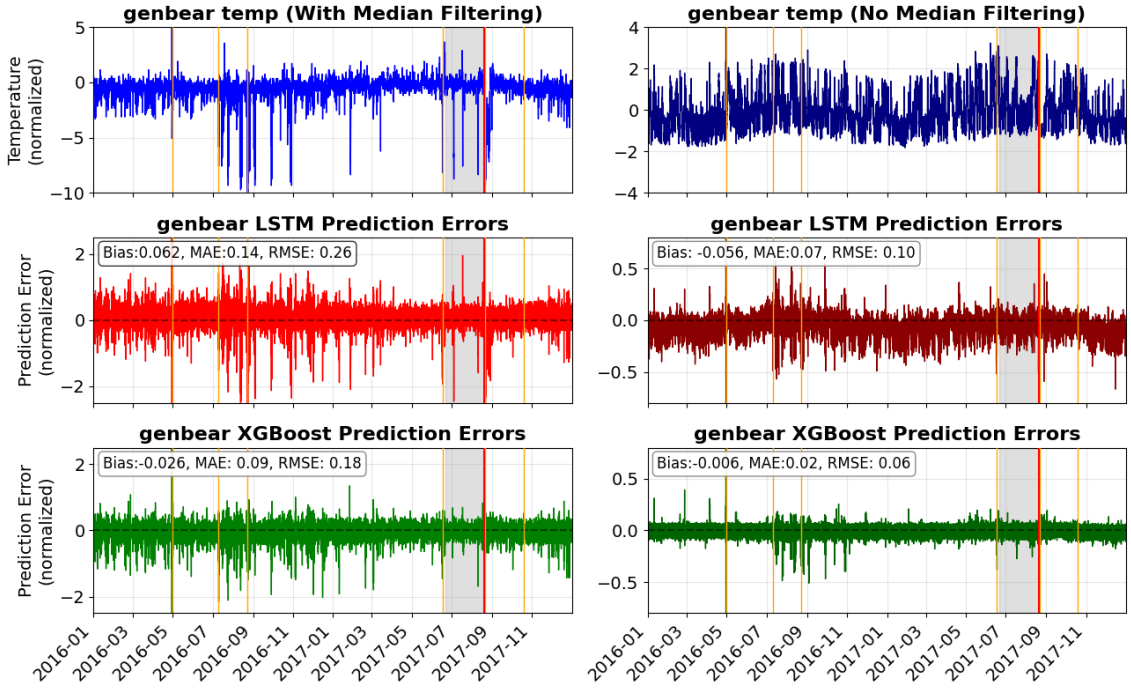

Figure 4.30: Generator bearing temperature prediction comparison between median-filtered (left) and raw temperature (right) configurations.

Figure 4.30 presents a comprehensive comparison for generator bearing temperature prediction, with median-filtered results on the left and raw temperature results on the right. The median-filtered target exhibits several prominent negative temperature spikes, indicating periods when turbine T07 deviated significantly from fleet median behavior. These deviations correlate with various maintenance events, including transformer refrigeration repairs in July and August 2016. However, not all large-magnitude deviations correspond to the target component (generator bearing) failures, introducing potential false alarm sources. In contrast, the raw temperature measurements show more stable baseline behavior but retain inherent turbine-specific offsets that challenge standardization assumptions.

When examining model performance, LSTM demonstrates high-magnitude prediction errors during temperature deviation events in the median-filtered configuration, requiring higher CUSUM sensitivity parameters. Both LSTM predictions exhibit systematic bias due to baseline differences between turbines, violating the zero-mean assumption despite standardization. XGBoost shows the capability to predict large deviation spikes with reasonable accuracy in the median-filtered case, while demonstrating more robust performance with relatively unbiased predictions due to its tree-based algorithm's resilience to input magnitude variations.

The use of healthy data statistics for standardization created distinct challenges for each approach. Median-filtered data exhibited higher variance due to maintained outliers, leading to larger normalized magnitudes in test data. Raw temperature data suffered from baseline differences between turbines that prevented achieving true zero-mean, unit-variance normalization. Additionally, different scalers for each preprocessing method resulted in different prediction error magnitudes, necessitating separate CUSUM parameter optimization.

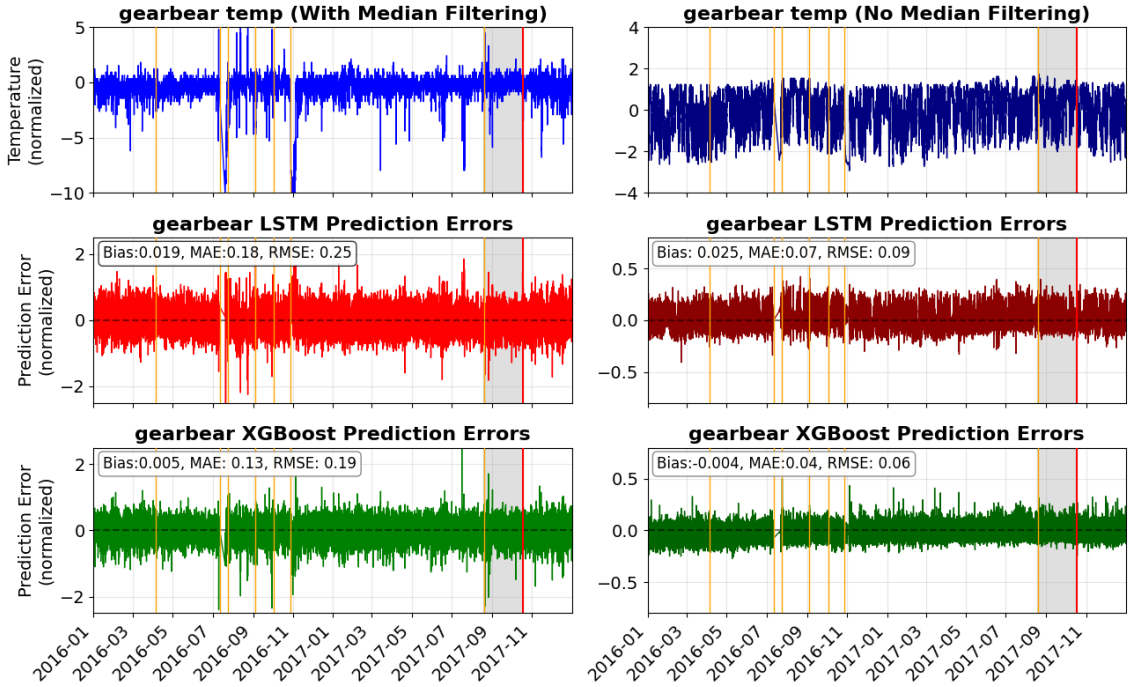

Figure 4.31: Gearbox bearing temperature prediction comparison between median-filtered (left) and raw temperature (right) configurations.

Figure 4.31 shows similar analysis for gearbox bearing temperature, revealing consistent patterns with some notable differences. The median-filtered configuration shows distinct deviations coinciding with the generator replacement events on July 11, 2016, and October 27, 2016. Both LSTM and XGBoost demonstrate improved ability to predict large deviation spikes compared to the generator bearing case, showing better model adaptability to component-specific anomaly patterns. For raw temperature performance, LSTM continues to exhibit systematic bias issues, indicating the need for bias correction methods, while XGBoost maintains relatively stable performance across both preprocessing approaches.

Figure 4.32 demonstrates optimized CUSUM detection results using raw temperature data without median filtering. For generator bearing detection, XGBoost achieved excellent performance with only 1 false alarm occurring during generator bearing sensor replacement on April 30, 2016, which represents a legitimate maintenance event. This demonstrates robust detection capability despite using raw temperature inputs. In contrast, LSTM generated 13 false alarms due to systematic bias in prediction errors, with bias issues compromising CUSUM's zero-centered assumption and highlighting the need for bias correction when using raw temperature data with LSTM.

The gearbox bearing detection results show contrasting patterns. LSTM achieved sur-

prisingly good detection performance with only 2 false alarms, though this performance may be coincidental given persistent mean drift issues, requiring careful interpretation due to underlying bias problems. XGBoost struggled significantly with degradation patterns in pre-failure periods, where consistent negative prediction errors before the July 11, 2016 generator replacement event led to excessive CUSUM accumulation, generating 50 false alarms. This poor performance was attributed to a combination of slightly negatively biased prediction errors and a relatively low CUSUM sensitivity threshold (k = 0.04), making it easier for the CUSUM score to accumulate.

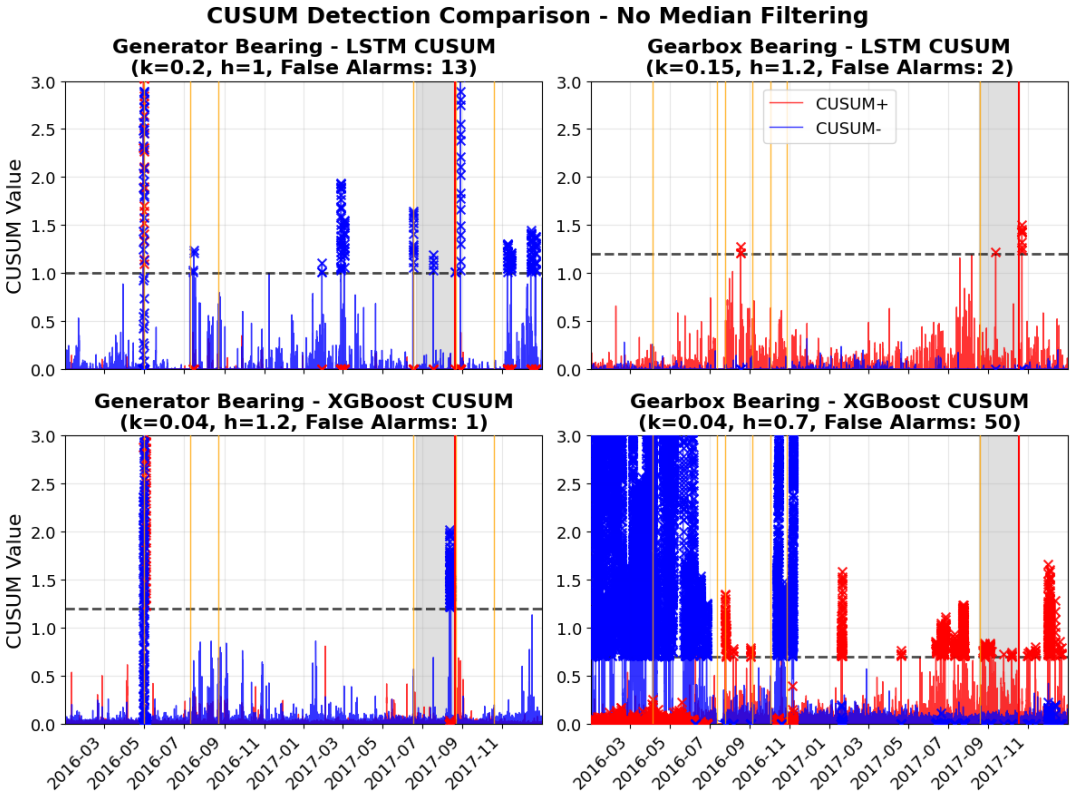

Figure 4.32: CUSUM detection performance using raw temperature data without median filtering. Left column shows generator bearing results, right column shows gearbox bearing results, comparing LSTM and XGBoost detection capabilities with optimized parameters.

The median filtering analysis reveals several critical insights. The preprocessing approach presents trade-offs where median filtering helps isolate component-specific baseline differences but introduces cross-component interference from other maintenance events. LSTM shows higher sensitivity to input preprocessing due to activation function characteristics, while XGBoost demonstrates greater robustness to input variations. Standard scaling on healthy data creates different challenges for each preprocessing approach, requiring method-specific parameter optimization. Detection performance varies significantly, with raw temperature data yielding excellent results for some cases (XGBoost-generator) and poor results for others (XGBoost-gearbox), depending on specific bias patterns. These results emphasize the importance of careful preprocessing selection and the necessity for robust bias handling mechanisms, particularly the need for bias correction methods to meet CUSUM's zero-centered assumption in predictive maintenance systems.

4.2.6 Prediction Error Mean Adjustment Analysis

Since biased prediction errors violate CUSUM assumptions and significantly influence CUSUM score accumulation, systematic bias correction is required. To address this issue, a mean adjustment approach was implemented by subtracting a rolling average from the prediction errors to transform them into a more zero-centered time series. The rolling average was calculated over a moving window of prediction errors, representing the previous measurements' mean bias, which was then subtracted from current prediction errors to remove systematic offset.

A critical consideration is that CUSUM is designed to detect degradation patterns that may themselves manifest as drift characteristics. Therefore, the rolling average window must be sufficiently long to preserve short-term degradation signals while removing long-term bias. A 100-day rolling window was selected for mean adjustment, corresponding to the 60-day detection window assumption and allowing degradation patterns occurring within a 2-month period to be preserved. This window size functions as a high-pass filter, retaining lower-frequency variations in the time series that may represent genuine degradation trends.

Target Lag Prediction Bias Removal

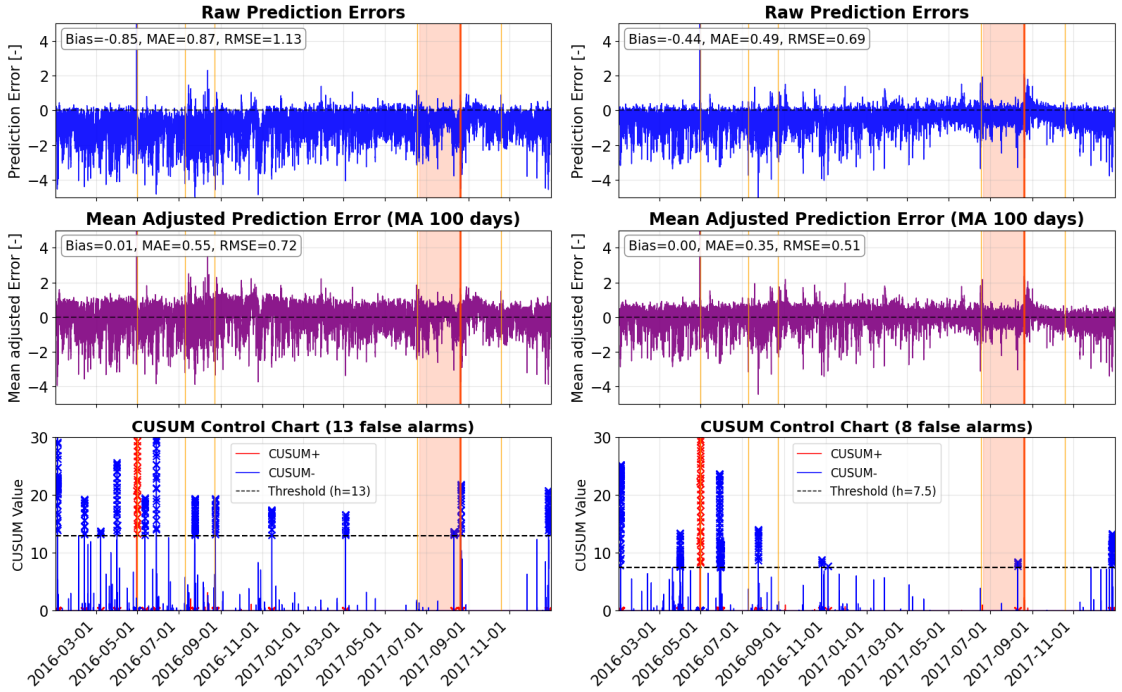

Figure 4.33: LSTM generator bearing failure detection with mean-adjusted prediction errors (no target lag features). (k=1.9, h=13).

Figure 4.34: XGBoost generator bearing failure detection with mean-adjusted prediction errors (no target lag features). (k=1.7, h=7.5).

Figure 4.33 and Figure 4.34 demonstrate model predictions without lagged target features for generator bearing failure detection. After subtracting the 100-day moving average from raw predictions, the time series shows improved prediction errors with reduced systematic bias. However, following CUSUM parameter optimization for the mean-adjusted predictions, detection performance remains challenged in identifying degradation patterns, resulting in more false alarms compared to the optimal configuration with target lags. This performance deficit highlights the fundamental challenge in detecting generator bearing

pre-failure degradation patterns when excluding lagged target features as inputs, indicating that prediction bias is not the only limitation for target lag exclusion cases.

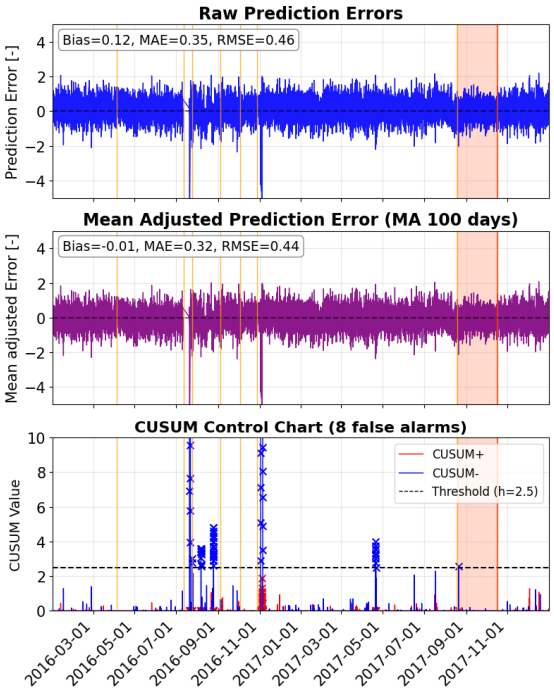

Figure 4.35: LSTM gearbox bearing failure detection with mean-adjusted prediction errors (no target lag features). (k=1.4, h=2.5).

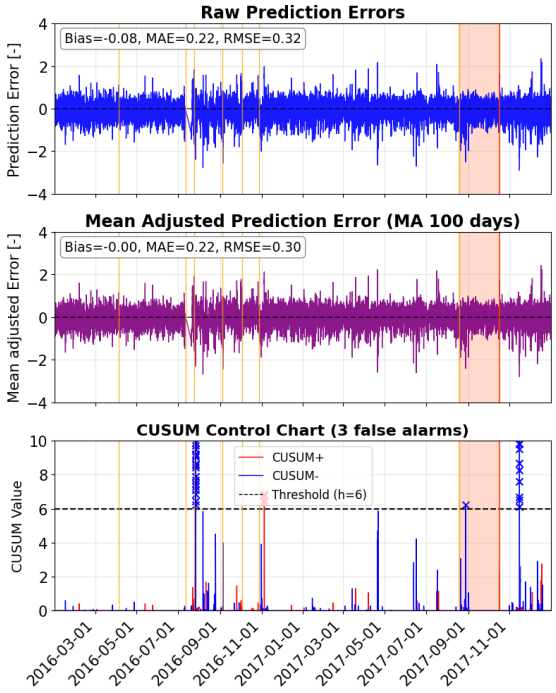

Figure 4.36: XGBoost gearbox bearing failure detection with mean-adjusted prediction errors (no target lag features). (k=1.2, h=6.0).

For gearbox bearing failure detection without target lag inputs, the prediction error bias was less pronounced than in the generator bearing case. Figure 4.35 and Figure 4.36 demonstrate improved failure detection capability using mean-adjusted prediction errors. LSTM prediction reduced false alarms from 12 to 8, while XGBoost maintained the same performance with 3 false alarms. This behavior demonstrates component-specific differences in failure detection characteristics. However, a concern arises for LSTM failure prediction where the successful detection alarm occurs closely after an oil leakage annotation on August 19, 2017, suggesting the alarm may be attributed to hydraulic group issues rather than gearbox bearing degradation.

Table 4.5: Optimal CUSUM parameters and performance for no target lag input and bias corrected prediction

Component	Model	CUSUM k	CUSUM h	Lead Time (hrs)	Total False Alarms
Generator	LSTM	1.9	13	226.5	13
Bearing	XGBoost	1.7	7.5	226.3	8
Gearbox	LSTM	1.4	2.5	1381.1	8
Bearing	XGBoost	1.2	6	1195.5	3

No Median Filtering Bias Correction

Bias removal was also applied to raw temperature data without median filtering. Figure 4.37 illustrates improved LSTM performance with false alarms reduced from 13 to 5.

The pre-failure period analysis in Figure 4.38 shows that sudden negative mean-adjusted errors contributed to CUSUM negative scores exceeding the threshold, generating true positive detection. This result indicates that addressing drift issues is crucial for generator bearing failure detection using LSTM models.

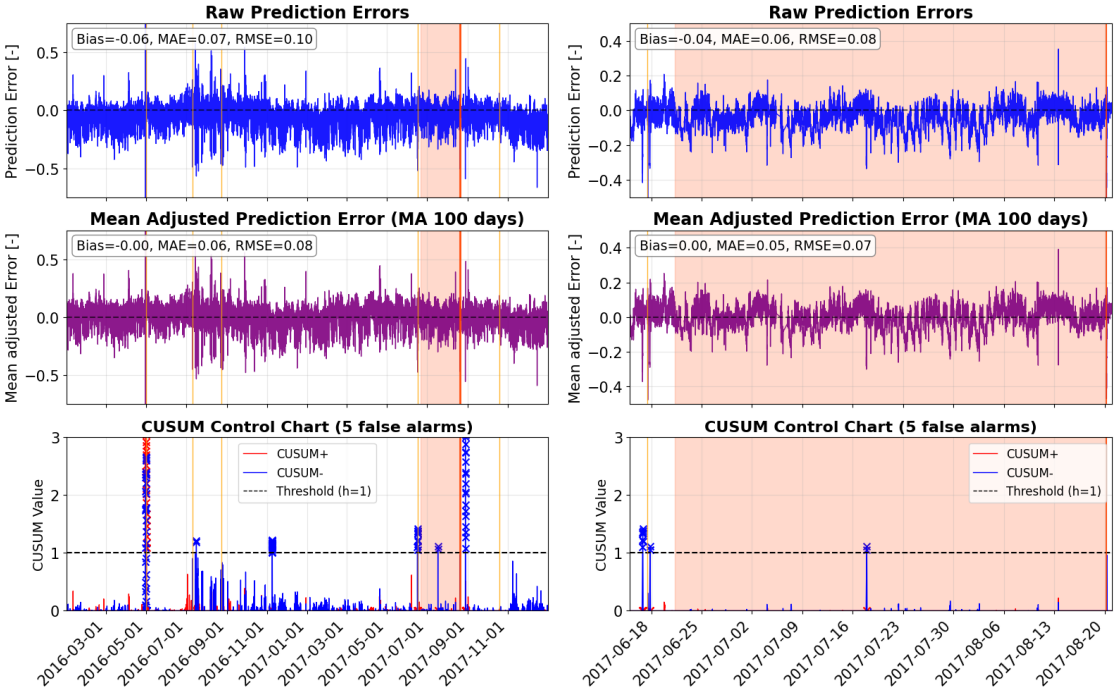

Figure 4.37: LSTM generator bearing failure detection with 100-day rolling mean adjustment. (k=0.17, h=1).

Figure 4.38: LSTM generator bearing pre-failure period with mean-adjusted prediction errors.

XGBoost prediction performance in Figure 4.39 achieved superior detection with only one false alarm attributed to outlier sensor values. This detection exemplifies how CUSUM algorithms are designed to detect small but persistent shifts. The accumulation of negative CUSUM scores reveals that 100-day rolling mean adjustment preserves high-frequency drift patterns without removing genuine degradation signals. Furthermore, the optimized CUSUM parameters achieved balance by avoiding false alarms from previous negative error spikes. This XGBoost test demonstrates the failure detection capability of normal behavior models combined with CUSUM algorithms for condition monitoring, revealing the possibility of detecting failures through small persistent shifts due to component degradation rather than sudden error spikes.

However, a remaining challenge is that these persistent shifts do not continue until failure occurs but return to baseline levels, suggesting difficulty in modeling continuous degradation patterns. The pre-failure analysis in Figure 4.40 shows that while degradation signals are detectable, they exhibit intermittent rather than monotonic characteristics.

Figure 4.41 and Figure 4.42 present failure prediction results for gearbox bearing using LSTM and XGBoost models with bias correction. After adjusting LSTM prediction errors, false alarms increased from 2 to 27, proving that the previous low false alarm count was coincidental due to systematic bias offsetting detection sensitivity. For XGBoost prediction, failure detection remained challenging in identifying degradation patterns, demonstrating the inherent difficulty in modeling gearbox failure behavior even with bias correc-

tion.

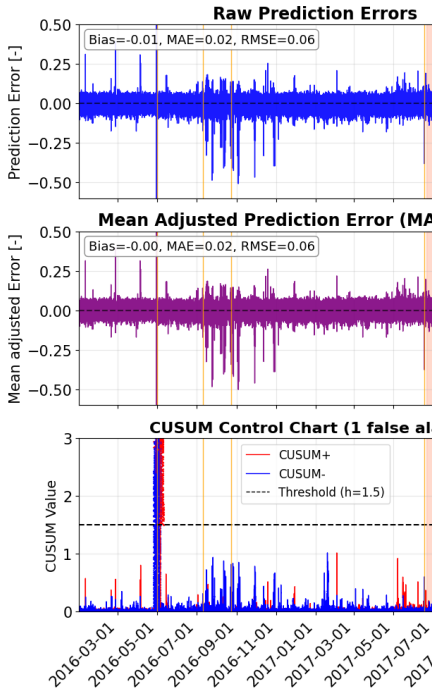

Figure 4.39: XGBoost generator bearing failure detection with 100-day rolling mean adjustment. (k=0.03, h=1.5).

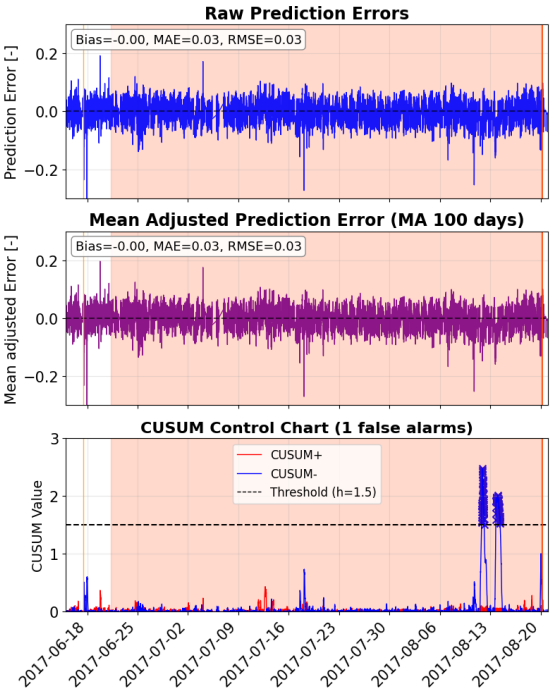

Figure 4.40: XGBoost generator bearing pre-failure period analysis with mean-adjusted prediction errors.

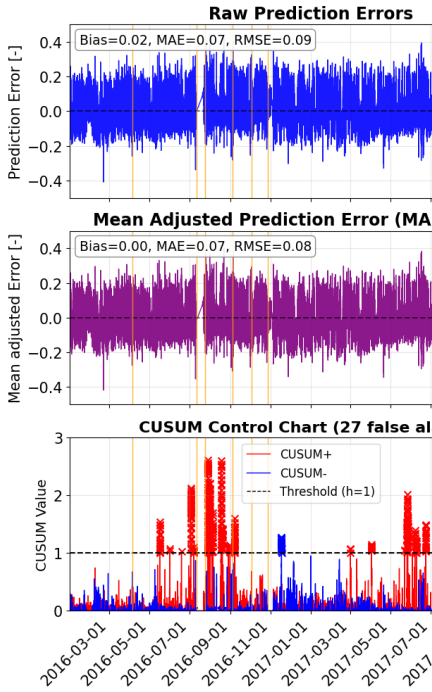

Figure 4.41: LSTM gearbox bearing failure detection with 100-day rolling mean adjustment. Optimized parameters (k=0.11, h=1).

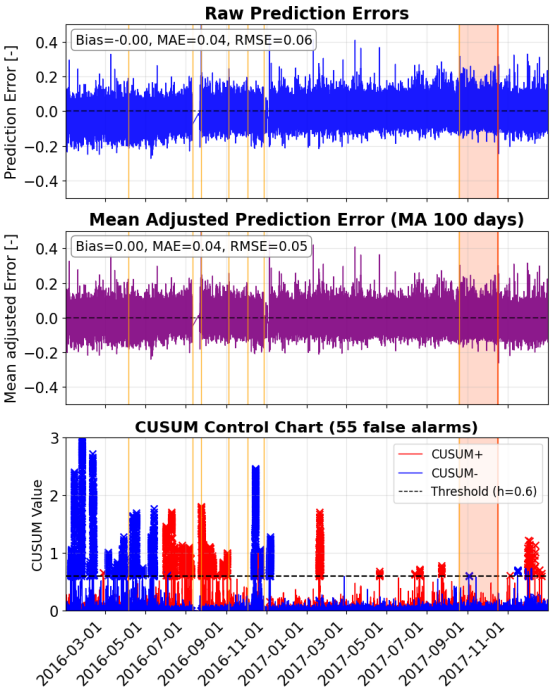

Figure 4.42: XGBoost gearbox bearing failure detection with 100-day rolling mean adjustment. (k=0.04, h=0.6).

The mean adjustment analysis reveals that bias correction can significantly improve detection performance for specific model-component combinations, particularly XGBoost for generator bearing detection. However, the effectiveness varies considerably depending on the underlying degradation characteristics and model behavior. While bias correction addresses systematic offset issues, it does not fundamentally solve the challenge of detecting subtle, intermittent degradation patterns that characterize real-world component failures.

Table 4.6: Optimal CUSUM parameters and performance for no median filtering and bias corrected prediction

Component	Model	CUSUM k	CUSUM h	Lead Time (hrs)	Total False Alarms
Generator	LSTM	0.17	1	799.6	5
Bearing	XGBoost	0.03	1.5	203.5	1
Gearbox	LSTM	0.11	1	836.6	27
Bearing	XGBoost	0.04	0.6	1067.1	55

4.2.7 Detection and Lead Time Analysis

The detection lead time provides critical insights for practical NBM deployment, as it determines the available window for maintenance planning and scheduling. The analysis reveals significant variations in lead times across different model configurations and preprocessing approaches, with important implications for understanding detection mechanisms and practical deployment strategies.

Median Filtering Configuration Lead Times

The CUSUM-optimized results in Table 4.3 show substantial lead time variations for generator bearing monitoring, with LSTM achieving significantly longer lead times (1120.6 hours) compared to XGBoost (227.3 hours). For gearbox bearing detection, lead times are more comparable between models (LSTM: 1099 hours, XGBoost: 1195.6 hours), suggesting different underlying detection mechanisms for each component type.

Analysis of the detection patterns reveals that successful detection alarms in the median filtering configuration correspond primarily to temperature deviation spikes occurring within the detection window. The models demonstrate difficulty predicting sudden, high-magnitude temperature deviations, as evidenced in Figure 4.30, where spiky prediction errors coincide with temperature deviation peaks. This characteristic indicates that detection success depends more on the occurrence of identifiable anomalous events rather than gradual degradation pattern recognition.

However, gearbox bearing prediction in Figure 4.31 demonstrates superior capability for predicting sudden temperature deviations. Notably, sudden temperature variations outside the detection window do not generate significant prediction error spikes, while the XGBoost model actually amplifies prediction errors when high temperature deviations occur within the detection window. This behavior explains the superior performance of XGBoost for gearbox bearing monitoring: the model successfully predicts temperature variations outside the detection window (reducing false alarms) while preserving error amplification for genuine anomalies within the critical pre-failure period.

Target Lag Exclusion with Bias Correction

The lead time results for bias-corrected predictions without target lag features (Table 4.5) reveal distinct component-specific detection characteristics. Generator bearing detection achieved consistent 226-hour lead times for both LSTM and XGBoost models, while

gearbox bearing detection showed much longer lead times (LSTM: 1381 hours, XGBoost: 1196 hours).

The identical lead times for generator bearing detection across both models suggest that median filtering effects dominate the detection mechanism regardless of model architecture or target lag inclusion. For gearbox bearing detection, the extended lead times may be attributed to early median-filtered temperature variations, as demonstrated in Figure 4.31. However, the absence of distinct temperature deviations closer to actual failure highlights a fundamental limitation of median filtering: difficulty detecting small, persistent changes when sudden spikes are absent.

Raw Temperature Data with Bias Correction

The bias-corrected raw temperature analysis (Table 4.6) demonstrates fundamentally different detection mechanisms, with lead times of 800 hours (LSTM) and 204 hours (XGBoost) for generator bearing, and 837 hours (LSTM) and 1067 hours (XGBoost) for gearbox bearing detection.

The dramatic difference in generator bearing lead times compared to median filtering reveals a fundamental shift in detection approach: from identifying sudden temperature deviation spikes to detecting persistent prediction error accumulation. The XGBoost model's 204-hour lead time exemplifies ideal CUSUM operation, where small but persistent drift patterns are detected while temperature variation peaks are mitigated through appropriate sensitivity parameter (k) selection. This demonstrates CUSUM functioning as designed: focusing on persistent shifts rather than transient variations.

For gearbox bearing detection using raw temperature data, results become less reliable due to substantial false alarm increases during testing. This degradation illustrates the inherent difficulty in identifying persistent prediction error shifts for gearbox bearing failures, suggesting that gearbox degradation patterns may be more subtle or intermittent than generator bearing failures.

The lead time analysis reveals two distinct detection mechanisms operating across different configurations. In median filtering set up, success depends on occurrence of identifiable temperature deviation events within detection windows, with lead times determined by when such events occur rather than gradual degradation onset. As in the raw temperature with bias correction, success depends on accumulation of small but consistent prediction errors over time, with lead times reflecting the onset of degradation patterns. Generator bearings show more consistent detectability across configurations, while gearbox bearings exhibit high variability suggesting more complex or subtle degradation characteristics.

These findings have important implications for maintenance planning, as different detection mechanisms provide different types of advance warning. Spike-based detection offers longer potential lead times but depends on occurrence of identifiable events, while persistent drift detection provides more reliable early warning of systematic degradation but could be component specific. The choice between pre-processing approaches should consider these trade-offs based on specific maintenance requirements and component characteristics.

5 Discussion

5.1 Interpretation of Key Findings

This study investigated how NBMs can effectively detect early-stage bearing faults in wind turbines to optimize maintenance strategies and reduce false alarms. The comprehensive analysis revealed critical insights about model performance, feature engineering requirements, and practical implementation challenges for NBM-based condition monitoring systems.

Cross-Validation Baseline Performance

The systematic cross-validation analysis established optimal model configurations and revealed fundamental model characteristics. XGBoost consistently outperformed LSTM by 10-15% across RMSE and MAE metrics, achieving optimal performance with lag step 3 and 15-17 features. LSTM demonstrated greater stability across feature counts but required careful sequence length tuning (optimal: 6 time steps). The deficit between the LSTM and XGBoost models could partly be attributed to the relatively simple architecture of the LSTM. Specifically, the LSTM model employed in this study contains only a single recurrent layer with 64 units, followed by a dense output layer. This lightweight design was chosen for computational efficiency and ease of training, but it may also have limited the model's capacity to capture complex temporal patterns in the data. Exploring deeper or more complex recurrent architectures, such as multi-layer or bidirectional LSTMs, may improve performance and reduce the observed gap with XGBoost.

Additionally, component-specific differences emerged, with generator bearing temperature showing around 18% better predictability than gearbox bearing temperature, indicating varying complexity across components. This may partly stem from differences between the training and testing data distributions; as shown in Figure 3.18, the gearbox bearing temperature exhibits a more similar distribution across datasets, which could influence model generalization.

End-to-End Detection Performance

The systematic optimization process identified component-specific optimal configurations for practical deployment. For generator bearing monitoring, LSTM models achieved best performance with 60 features and 12-step sequences, while the XGBoost model achieved optimal performance using 20 features and 3-step lags. Gearbox bearing detection required different parameters: LSTM with 7 features and 6-step sequences, XGBoost with 10 features and 6-step lags. These findings demonstrate that effective NBM-based condition monitoring requires component-specific customization rather than universal approaches.

CUSUM Parameter Optimization for Early Detection

Grid search optimization revealed significant sensitivity to CUSUM parameters, with substantial impact on early fault detection capability. Optimal parameter tuning achieved significant false alarm reductions while maintaining detection sensitivity. LSTM generator bearing detection improved from 14 to 6 false alarms, and XGBoost gearbox bearing detection reduced from 5 to 3 false alarms. Component-specific parameter requirements emerged, with generator bearings benefiting from higher sensitivity parameters (k=1.5) and gearbox bearings requiring lower sensitivity (k=0.2) for gradual degradation accumulation. This difference stems from distinct target input characteristics: generator bearing

temperature measurements exhibit more pronounced spikes compared to gearbox bearing temperature data. The analysis revealed that CUSUM optimization adapts to model prediction limitations. When models cannot predict sudden temperature spikes, these events manifest as high-magnitude prediction errors. CUSUM optimization responds by increasing sensitivity to preserve spike-related errors as true positives while filtering smaller false alarms. This transforms detection from traditional drift-based to spike-based detection, highlighting how CUSUM parameters inherently reflect the underlying model prediction capabilities.

Target Lag Feature Impact on Detection Sensitivity:
A critical trade-off was identified between prediction accuracy and early fault detection capability. Including lagged target features dramatically improved prediction accuracy (Generator bearing temperature prediction MAE and RMSE decrease around 80%, and gearbox bearing temperature around 45%) but potentially masked early degradation signals crucial for preventive maintenance. This is evident in the generator bearing monitoring, where prediction error variance differs significantly between the detection window and periods outside it.

This finding directly addresses the research question by revealing that: highly accurate NBMs may actually impair early-stage fault detection, while less accurate predictions might provide clearer degradation patterns. This challenges conventional assumptions about prediction accuracy as an indicator of anomaly detection effectiveness and highlights the need to balance model accuracy with detection sensitivity in predictive maintenance applications.

Preprocessing Strategy Effects on Maintenance Implementation:
The median filtering analysis revealed important trade-offs for practical deployment. Median filtering helped isolate component-specific anomalies by removing fleet-wide baseline variations, but introduced cross-component interference from maintenance events on unrelated systems, potentially triggering unnecessary maintenance actions on healthy components. Raw temperature data avoided this interference but required systematic bias correction to meet CUSUM's zero-centered assumptions.

The systematic bias in raw temperature predictions is attributed to normalization challenges: the healthy training data scaler differs from individual testing turbine characteristics, creating prediction offsets that violate CUSUM assumptions. This highlights a fundamental challenge in scaling NBM systems - the difficulty of achieving consistent normalization across diverse turbine baselines when testing data cannot use future statistics for standardization.

These preprocessing decisions directly impact maintenance cost-effectiveness by influencing false alarm rates and detection sensitivity, requiring careful selection based on specific deployment contexts and tolerance for different types of detection errors.

Bias Correction for Reliable Early Detection:
Systematic bias correction using a 100-day rolling mean adjustment proved essential for reliable NBM performance, particularly when using raw sensor data. XGBoost generator bearing detection achieved exceptional performance (1 false alarm) after bias correction, demonstrating the potential for NBM systems to provide reliable early warning while minimizing unnecessary maintenance interventions. However, bias correction effectiveness varied significantly across model-component combinations, and fundamental standardization challenges persist due to turbine baseline differences that cannot be fully corrected through preprocessing.

Sensor Selection Impact on Maintenance Strategies:
The comparison between drive-end and non-drive-end generator bearing sensors revealed that sensor selection critically affects NBM performance and thus maintenance effectiveness. Switching sensors resulted in 100%+ RMSE and 20%+ MAE increases, emphasizing that target sensor decisions directly influence the reliability and cost-effectiveness of condition monitoring systems. Additionally, sensor errors and outlier measurements in testing set contribute to false alarm generation, requiring robust outlier detection methods for practical deployment.

Detection Lead Time:
The analysis revealed two distinct detection mechanisms with different lead time characteristics: spike-based detection (median filtering) providing longer but event-dependent lead times, raw temperature detection offering more reliable early warning based on prediction errors drift. However, the detection depends on component-specific variations, with the generator bearing showing more consistent detectability in raw temperature inputs of around 200 hours.

Data and Generalizability Limitations:
The study revealed important constraints on NBM generalizability, including insufficient failure events for comprehensive testing and a limited fleet size (5 turbines) that may compromise the reliability of fleet-based median calculations. These limitations, combined with preprocessing challenges such as median filtering creating spiky inputs that generate false alarms from cross-component failures, highlight the complexity of deploying NBM systems in real-world environments with diverse operating conditions.

5.2 Challenges and Limitations

Systematic Bias and Standardization Challenges
The analysis revealed persistent systematic bias issues that fundamentally compromise NBM performance. Baseline differences between turbines create non-standardized inputs despite applying Standard Scaler normalization, as each turbine operates with inherent offsets that cannot be completely removed. This standardization particularly degrades LSTM performance because the activation functions are highly sensitive to input magnitude and distribution. The challenge is exacerbated in real-world deployment, where testing turbines cannot know their own future baseline statistics, meaning input features may be difficult to achieve near zero-mean, close unit-variance characteristics assumed by the normalization process.

While a 100-day rolling mean adjustment showed promise for bias correction, it introduced new challenges in preserving genuine degradation signals while removing systematic offset. The XGBoost generator bearing case demonstrated excellent results (1 false alarm) after bias correction, but success was highly dependent on specific model-component combinations, highlighting the lack of generalizable bias correction approaches.

Data Limitations and Generalizability Concerns
The dataset contains insufficient failure events to thoroughly test model generalizability across different failure modes and operating conditions. With only a small number of turbines (5 turbines), the calculated fleet median may not accurately represent true healthy fleet states, undermining the reliability of median filtering approaches. This limitation affects both the training of robust models and the validation of their performance across diverse scenarios, raising questions about deployment scalability to larger wind farms.

Preprocessing-Induced False Alarms
Median filtering, while designed to isolate component-specific anomalies, creates spiky temperature deviations that introduce challenges in predicting these spikes. Tempera-

ture spikes often result from failures in unrelated components (e.g., transformer refrigeration or hydraulic system issues affecting bearing temperature calculations), creating false degradation signals that are difficult to distinguish from genuine bearing problems. These cross-component interactions fundamentally challenge the assumption that individual component monitoring can be isolated from system-wide effects.

Additionally, sensor errors generating outlier measurements contribute to false alarm generation, as the models interpret these outliers as potential degradation signals. The sudden spikes in prediction errors force CUSUM parameter optimization toward higher k values to filter high-frequency noise, but this adaptation compromises sensitivity to genuine degradation patterns that may manifest as gradual rather than sudden changes.

Component-Specific Behavior and Detection Complexity
Generator and gearbox bearings exhibited fundamentally different failure characteristics, with generator bearings showing more pronounced temperature deviations while gearbox bearings exhibited subtler, more variable degradation patterns. This heterogeneity complicates the development of unified monitoring frameworks and suggests that component-specific modeling approaches are necessary, but each requires individual optimization and validation.

Real-World Deployment Scaling Challenges
The systematic optimization requirements identified in this study include component-specific parameter tuning, bias correction, and preprocessing selection. These findings suggest that successful deployment requires substantial customization for each application. This need for individualized optimization may limit the scalability and cost-effectiveness of NBM systems when deployed across large fleets with diverse operating conditions and component characteristics.

5.3 Real-World Deployment Considerations

Several factors complicate the practical deployment of NBM-based condition monitoring systems in operational wind farms, requiring careful consideration of technical and operational constraints.

The preprocessing analysis revealed fundamental trade-offs that directly impact deployment feasibility. Median filtering effectively isolates component-specific anomalies but introduces cross-component interference from maintenance events on unrelated systems, creating false degradation signals that complicate maintenance decision-making. Raw temperature data avoids this interference but suffers from turbine-specific baseline differences that challenge standardization assumptions, particularly affecting LSTM performance due to activation function sensitivity to input distributions. These preprocessing decisions must be made based on specific deployment contexts, available data characteristics, and tolerance for different types of false alarms.

In addition, baseline differences between turbines complicate normalization and require turbine-specific calibration that may not scale efficiently. The current approach of using healthy data statistics for standardization becomes problematic as fleet size grows and the baseline differences become prominent. Successful deployment requires robust scaling methods that can handle baseline differences without requiring individual turbine calibration, as personalized systems can increase implementation complexity.

Furthermore, practical deployment necessitates integration with existing maintenance workflows and decision-making processes. The alarm verification mechanism represents an opportunity for a secondary checking system to verify whether raised alarms actually indicate faulty states of target components could significantly reduce false alarm rates and

improve maintenance efficiency. This double-check system could incorporate additional sensor data, operational context, or maintenance history to validate NBM-generated alerts before triggering maintenance actions.

The lead time analysis provides critical guidance for maintenance planning, as different pre-processing approaches yield fundamentally different warning characteristics. Median filtering configurations may provide 1000+ hour lead times when temperature deviation events occur, while bias-corrected raw temperature approaches offer more reliable 200-800 hour warnings based on systematic degradation patterns.

Last but not least, the component-specific optimization requirements identified in this study suggest that effective deployment requires sophisticated decision support systems that can adapt detection parameters based on component type, operating conditions, and historical performance. The complexity of manual parameter tuning for each component-model combination highlights the need for automated optimization approaches that can handle the heterogeneity of real-world wind farm operations while maintaining detection reliability.

5.4 Future Work

Feature Engineering and Normalization
Future work should prioritize the development of improved normalization methods that can successfully eliminate baseline differences between turbines without compromising degradation signal preservation. This includes exploring physics-informed normalization approaches that account for turbine-specific operating characteristics and environmental conditions.

Additionally, more sophisticated feature engineering approaches are needed that better capture degradation patterns while maintaining CUSUM compatibility, including:

1. Development of degradation-aware features that naturally exhibit drift characteristics aligned with physical failure mechanisms
2. Investigation of multi-scale temporal features that capture both short-term variations and long-term degradation trends
3. Exploration of physics-informed features that incorporate domain knowledge about bearing failure mechanisms and thermal dynamics

Automated Systems
The significant parameter sensitivity observed throughout this study indicates the need for adaptive CUSUM systems that can automatically adjust parameters based on operating conditions, component types, and historical performance. Machine learning-based parameter optimization could potentially improve detection performance while reducing the manual tuning that currently limits scalability. These adaptive systems should incorporate:

1. Real-time parameter adjustment based on changing operating conditions
2. Historical performance feedback to continuously improve detection accuracy

Multi-Sensor Fusion and Uncertainty Quantification
The sensor selection analysis demonstrates the potential for multi-sensor fusion approaches that leverage information from multiple temperature sensors, vibration data, and operational parameters to improve detection robustness and reduce dependence on single-sensor measurements. Future systems should also incorporate uncertainty quantification to provide confidence estimates for predictions and detections, enabling more nuanced

decision-making and helping distinguish between high-confidence detections and uncertain alerts that may require additional verification.

Validation and Generalization

Future research should address the data limitations identified in this study and use larger datasets with more failure events. This includes developing standardized evaluation frameworks for NBM systems that can assess generalization capability across different wind farm environments, turbine types, and failure modes, ensuring that research findings translate effectively to practical deployment scenarios.

6 Conclusion

This research investigated the application of normal behavior models (NBMs) for optimal wind farm maintenance, addressing the fundamental question of how NBMs can effectively detect early-stage faults while reducing false alarms. Through a comprehensive analysis of machine learning approaches combined with CUSUM anomaly detection, this study demonstrates both the significant potential and inherent challenges of NBM-based condition monitoring systems for wind turbine bearing maintenance.

The cross-validation analysis established that XGBoost consistently outperforms LSTM for bearing temperature prediction, achieving 10-15% better performance across RMSE and MAE metrics. However, the end-to-end detection analysis revealed that optimal configurations are highly component-specific, with generator and gearbox bearings requiring fundamentally different model parameters and CUSUM sensitivity settings.

The identification of the target lag analysis reveals a critical insight for NBM development: while including lagged target features dramatically improves prediction accuracy, it potentially masks the early degradation signals essential for effective preventive maintenance. The systematic bias correction methodology using a 100-day rolling mean adjustment provides a practical solution for handling systematic offset issues. When properly applied, this approach enabled exceptional detection performance, with XGBoost achieving generator bearing detection with only 1 false alarm, demonstrating the potential for significant maintenance cost reduction through minimized unnecessary interventions.

The detection lead time analysis revealed that effective NBM systems can provide 200-1200 hours of advance warning depending on configuration, with different preprocessing approaches yielding distinct detection mechanisms suitable for different maintenance planning strategies.

The research also addresses the question of how NBMs can detect early-stage faults. The CUSUM parameter optimization achieved substantial false alarm reductions: reducing false alarms from 14 to 6 for LSTM generator bearing detection, and from 5 to 3 for XGBoost gearbox bearing detection. These improvements translate to reduced unnecessary maintenance actions and associated costs while maintaining early detection capability within 60-day windows before failures.

However, the study also reveals significant implementation challenges that affect practical cost benefits. The systematic bias issues, preprocessing complexity, and component-specific optimization requirements suggest that successful deployment demands ongoing calibration. Baseline differences between turbines create standardization challenges that particularly affect LSTM performance, while insufficient failure events in the dataset limit generalizability assessment.

Returning to the central research question, this study demonstrates that normal behavior models can indeed effectively detect early-stage faults and contribute to maintenance cost savings, but with important caveats. Success depends critically on proper system configuration, component-specific optimization, systematic bias handling, and careful preprocessing selection. The research provides methodologies for achieving these requirements, including the systematic optimization framework, bias correction approach, and component-specific parameter guidelines. Ultimately, this research demonstrates that

while significant challenges remain, normal behavior models represent a promising approach for early fault detection in wind turbines. Through careful attention to these implementation details, NBM systems can evolve from research prototypes to practical tools that meaningfully reduce wind farm maintenance costs while improving system reliability.

A Appendix

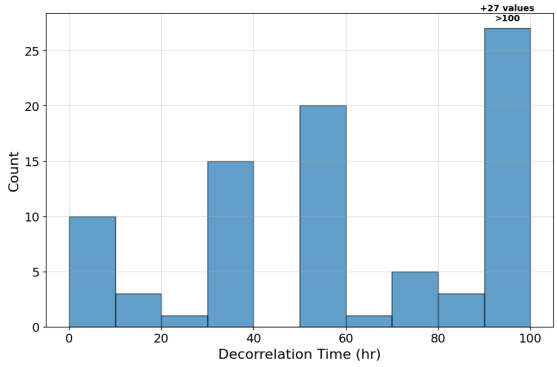

Figure A.1: Distribution of decorrelation time

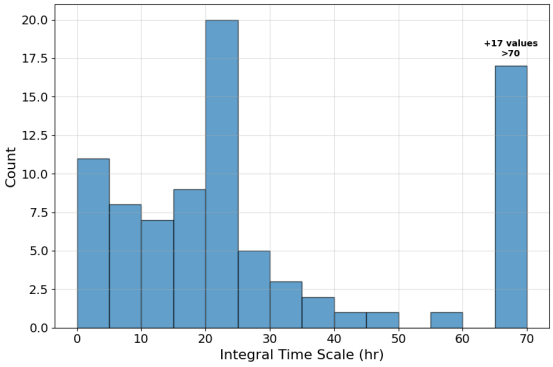

Figure A.2: Distribution of integral time scale

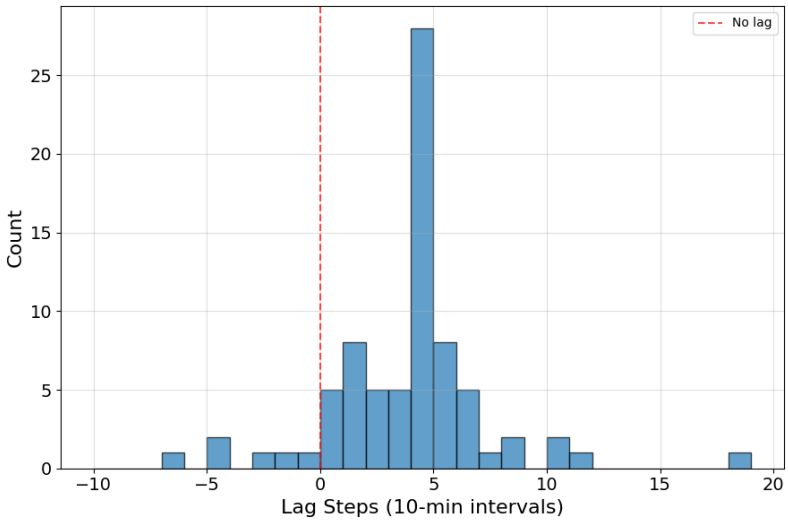

Figure A.3: Distribution of optimal lag steps

Table A.1: Overview of SCADA Dataset Variables by Component Category

Category	Variable	Description
Generator	Gen_Bear_Temp_Avg	Generator bearing temperature, non-drive end [°C]
	Gen_Bear2_Temp_Avg	Generator bearing temperature, drive end [°C]
	Gen_SlipRing_Temp_Avg	Generator slip ring temperature [°C]
	Gen_RPM_{Avg, Max, Min, Std}	Generator RPM statistics [rpm]
	Gen_Phase{1,2,3}_Temp_Avg	Generator phases temperature [°C]
Power Production	Grd_Prod_Pwr_{Avg, Max, Min, Std}	Power output statistics [kW]
	Grd_Prod_ReactPwr_{Avg, Max, Min, Std}	Reactive power statistics [kVAr]
	Grd_Prod_PsblePwr_{Avg, Max, Min, Std}	Possible power statistics [kW]
	Prod_LatestAvg_TotActPwr	Total active power production [Wh]
	Prod_LatestAvg_TotReactPwr	Total reactive power production [Wh]
	Prod_LatestAvg_ActPwrGen0	Active power - generator disconnected (yaw motor, hydraulic motor, etc.) [Wh]
	Prod_LatestAvg_ActPwrGen1	Active power - generator connected in delta [Wh]
	Prod_LatestAvg_ActPwrGen2	Active power - generator connected in star [Wh]
	Prod_LatestAvg_ReactPwrGen{0,1,2}	Reactive power generation per connection configuration [VAr]
	Grd_Prod_CosPhi_Avg	Power factor average [-]
Ambient Conditions	Amb_WindSpeed_{Avg, Max, Min, Std}	Ambient wind speed statistics [m/s]
	Amb_WindSpeed_Est_Avg	Estimated ambient wind speed average [m/s]
	Amb_WindDir_Abs_Avg	Absolute wind direction average [deg]
	Amb_WindDir_Relative_Avg	Relative wind direction to nacelle [deg]
	Amb_Temp_Avg	Ambient temperature [°C]
	Nac_Direction_Avg	Nacelle direction average [deg]
Temperature Sensors	Nac_Temp_Avg	Nacelle temperature [°C]
	Spin_Temp_Avg	Temperature in the nose cone [°C]
	Cont_Hub_Temp_Avg	Temperature in the hub controller [°C]
	Cont_Top_Temp_Avg	Temperature in the nacelle controller [°C]
	Cont_VCP_Temp_Avg	Temperature on VCP-board [°C]
	Cont_VCP_WtrTemp_Avg	Temperature in the VCS cooling water [°C]
	Cont_VCP_ChokcoilTemp_Avg	VCP choke coil temperature [°C]
Drivetrain	Gear_Bear_Temp_Avg	Gearbox bearing temperature [°C]
	Gear_Oil_Temp_Avg	Gearbox oil temperature [°C]
	HVTrafo_Phase{1,2,3}_Temp_Avg	High voltage transformer temperatures [°C]
	Hyd_Oil_Temp_Avg	Hydraulic oil temperature [°C]
	Rtr_RPM_{Avg, Max, Min, Std}	Rotor RPM statistics [rpm]
	Blds_PitchAngle_{Avg, Max, Min, Std}	Blade pitch angle statistics [deg]
Grid Connection	Grd_Busbar_Temp_Avg	Grid busbar temperature [°C]
	Grd_InverterPhase1_Temp_Avg	Grid inverter phase 1 temperature [°C]
	Grd_RtrInvPhase{1,2,3}_Temp_Avg	Rotor inverter phases temperature [°C]
	Grd_Prod_CurPhse{1,2,3}_Avg	Grid production current by phase [A]
	Grd_Prod_VoltPhse{1,2,3}_Avg	Grid production voltage by phase [V]
	Grd_Prod_PsbleCap_{Avg, Max, Min, Std}	Possible capacitive reactive power statistics [kVAr]
	Grd_Prod_PsbleInd_{Avg, Max, Min, Std}	Possible inductive power statistics [kVAr]
	Grd_Prod_Freq_Avg	Grid frequency average [Hz]

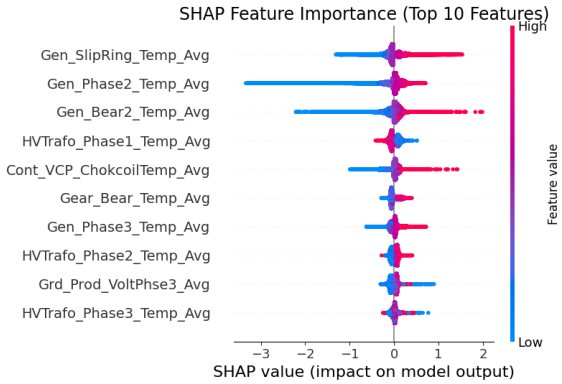

Figure A.4: Generator bearing temperature SHAP importance beeline without lagged generator bearing temperature as inputs. (Predicting Gen bear temp)

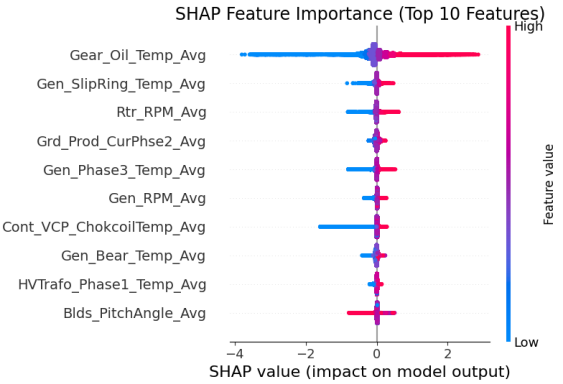

Figure A.5: Gearbox bearing temperature SHAP importance beeline without lagged gearbox bearing temperature as inputs. (Predicting Gear bear temp)

Technical
University of
Denmark

Nils Koppels Alle, Bygning 403
2800 Kgs. Lyngby
Tlf. 4525 1700

www.wind.dtu.dk