Multi-Channel Swin Transformer Framework for Rolling Bearing Remaining Useful Life Prediction

by

Ali Mohajerzarrinkelk
B.Sc., Sharif University of Technology, 2023

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Applied Science

in the Department of Mechanical Engineering

We acknowledge and respect the Lək̓ʷəŋen (Songhees and Xʷsepsəm/Esquimalt) Peoples on whose territory the university stands, and the Lək̓ʷəŋen and W̱SÁNEĆ Peoples whose historical relationships with the land continue to this day.

Multi-Channel Swin Transformer Framework for Rolling Bearing Remaining Useful Life Prediction

by

Ali Mohajerzarrinkelk
B.Sc., Sharif University of Technology, 2023

Supervisory Committee

Dr. Homayoun Najjaran, Supervisor
(Department of Mechanical Engineering)

Dr. Flavio Firmani, Departmental Member
(Department of Mechanical Engineering)

ABSTRACT

Accurate Remaining Useful Life (RUL) prediction of rotating machinery is a central challenge in predictive maintenance, where timely interventions can significantly reduce operational downtime and prevent catastrophic failures. This thesis introduces a Multi-Channel Swin Transformer (MCSFormer) framework designed to predict the RUL of rolling bearings using dual-sensor vibration data under variable operating conditions. The proposed approach emphasizes a structured preprocessing pipeline that begins with signal denoising through a sequence of low-pass filtering, wavelet-based denoising, and Savitzky-Golay smoothing to suppress noise and preserve relevant signal structure. Vibration signals are then segmented into fixed-length windows using sliding window segmentation method and transformed into time-frequency representations using Wavelet Packet Decomposition (WPD), which enables the extraction of rich degradation features at multiple resolution levels.

To process the resulting data, a convolutional neural network is first applied separately to the WPD-based images of horizontal and vertical signals. These CNN-extracted features are then concatenated and passed to a shared Swin Transformer architecture, enabling the model to jointly capture local and global patterns associated with the progression of degradation. Additionally, a safety-aware loss function is introduced to prioritize safety by penalizing late predictions more heavily than early ones, aligning the learning process with the asymmetric risk profile of industrial failure scenarios.

The framework is evaluated on the PRONOSTIA and XJTU-SY bearing datasets under both intra-condition and cross-condition settings. Comparative experiments against multiple state-of-the-art baselines demonstrate that MCSFormer achieves superior performance in both Mean Absolute Error and a scoring metric designed to assess safety-related prediction quality. Ablation studies further highlight the importance of each component, including the denoising pipeline, segmentation strategy, and custom loss function. The results affirm the proposed framework as a robust and generalizable solution for real-world prognostic systems, offering high predictive accuracy, enhanced interpretability, and risk-aware behavior suitable for deployment in industrial environments.

Contents

List of Tables

List of Figures

LIST OF ACRONYMS

AI	Artificial Intelligence
CNN	Convolutional Neural Network
CWT	Continuous Wavelet Transform
DL	Deep Learning
DWT	Discrete Wavelet Transform
FPT	First Prediction Time
KNN	k-Nearest Neighbors
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MCSFormer	Multi-Channel Swin Transformer
ML	Machine Learning
MSE	Mean Squared Error
PHM	Prognostics and Health Management
PM	Predictive Maintenance
RF	Random Forest
RUL	Remaining Useful Life
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
ViT	Vision Transformer
WPD	Wavelet Packet Decomposition

ACKNOWLEDGEMENTS

I would like to thank:

Dr. Homayoun Najjaran, for his invaluable mentorship, guidance, and continued support throughout my research journey. His insights and encouragement were instrumental in the development and completion of this work.

My family, for their unwavering love, patience, and belief in me. Their support gave me the strength to persevere through every challenge.

Mr. Hootan Mahmoodiyan, my best friend and labmate, for his constant support, technical insights, and the genuine friendship that helped me stay motivated and grounded throughout this journey

My friends and colleagues at the ACIS lab, for their camaraderie, constructive feedback, and the shared experiences that made this journey more meaningful and enjoyable.

DEDICATION

I dedicate this work to my father, mother, and sister in Iran.
Though oceans separate us, your love has never felt distant.
Your strength carried me through the hardest days,
and your unwavering belief gave me the courage to keep going.
This thesis is for you—with all my heart.

Chapter 1

Introduction

1.1 Motivation

Modern industrial systems rely increasingly on proactive strategies to maintain operational reliability and avoid costly downtimes. Among these strategies, predictive maintenance has gained significant traction as a data-driven approach that aims to detect early signs of degradation and anticipate equipment failure before it occurs. Instead of relying on fixed maintenance schedules or waiting for visible signs of malfunction, predictive maintenance continuously monitors system conditions, such as vibration, temperature, and pressure, through embedded sensors. The collected data is then analyzed to assess the health of components and to forecast their remaining service life. This approach allows maintenance activities to be planned with greater precision, reducing unnecessary interventions while preventing unexpected breakdowns.

One of the central tasks in predictive maintenance is the estimation of remaining useful life (RUL), which refers to the time span a component is expected to operate reliably before reaching a failure threshold. Accurate RUL prediction supports informed decision-making and contributes to extending asset life, improving safety, and lowering maintenance costs. In rotating machinery, rolling bearings are especially important in this context. These components support shafts, enable smooth rotational motion, and are critical to the function of motors, turbines, and gearboxes. However, due to their exposure to mechanical stress and dynamic loads, bearings are prone to gradual wear and fatigue. A bearing failure can halt entire production lines and damage adjacent components. As a result, developing accurate and robust methods

for bearing RUL prediction has become a key research area in industrial prognostics and health management.

Given their central role in industrial reliability, bearings have increasingly been used as the reference component for advancing methods in prognostics and RUL prediction. Insights gained from studying their degradation are not only valuable for safeguarding rotating machinery but also serve as a foundation for developing approaches that can be extended to other critical assets. The discussion that follows therefore considers RUL prediction techniques in the general sense, but with bearings as the primary application domain throughout this work.

Early approaches to RUL prediction were primarily based on physics-informed modeling and classical statistical techniques. Physics-based models attempt to simulate the degradation process of a component using analytical formulations derived from material fatigue laws, wear models, or stress-strain relationships. These models offer interpretability and are often grounded in domain expertise, making them attractive in applications where physical understanding is crucial. However, they rely on simplifying assumptions and require accurate knowledge of system parameters, such as material properties, loading conditions, and environmental factors. In complex industrial environments, where such information is often unknown or hard to quantify, the reliability of these models is significantly reduced [17].

In response to these limitations, statistical methods such as autoregressive models, hidden Markov models, and proportional hazard models emerged as data-driven alternatives. These techniques use historical degradation data to estimate future behavior and infer the probability of failure over time. While less dependent on domain-specific physical knowledge, statistical models typically assume stationarity and linearity in the data, which limits their effectiveness in capturing nonlinear and time-varying degradation patterns. As industrial systems become more complex and sensor-rich, these assumptions often fail, revealing the need for more flexible and adaptive RUL prediction approaches [45].

The limitations of traditional RUL prediction methods have led to a growing interest in machine learning (ML) and deep learning (DL) techniques, which offer greater flexibility in modeling complex, nonlinear, and time-dependent degradation behaviors. Unlike physics-based or statistical models, ML and DL approaches learn patterns directly from sensor data without requiring handcrafted assumptions about the underlying system dynamics. This data-driven capability makes them particularly well-suited for industrial applications, where operational conditions are often variable

and difficult to model analytically.

In ML-based RUL prediction, models such as support vector machines, decision trees, and random forests have been widely used for regression tasks. These models are relatively lightweight and interpretable, and they can provide solid performance when appropriate features are extracted from the raw sensor signals. However, their effectiveness often depends on carefully crafted features, which may not generalize well across different machines or fault modes. To address this, DL models have been introduced to automatically learn hierarchical representations from raw or minimally processed signals. Architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), including long short-term memory (LSTM) networks, have shown strong potential in capturing temporal and spatial patterns in degradation data. Their ability to extract meaningful features without extensive manual preprocessing has contributed to significant advances in RUL prediction performance across varying datasets and conditions [65].

While conventional DL models such as LSTMs and CNNs have achieved significant success in RUL prediction, they typically operate on raw or preprocessed 1D time series data. This representation, although effective to a degree, often limits the model's ability to capture localized frequency-domain patterns or multi-scale degradation features that are essential in machinery fault diagnosis. To address these limitations, recent studies have explored the transformation of 1D vibration signals into 2D image representations, such as spectrograms, recurrence plots, and wavelet-based scalograms, enabling the use of computer vision techniques for condition monitoring and RUL estimation. These image-based methods provide a richer encoding of temporal and spectral features, making them particularly well-suited for capturing subtle degradation trends that may not be apparent in raw time-domain signals.

The shift toward vision-based approaches has naturally led to the application of CNNs for learning features from 2D signal representations. However, CNNs have inherent limitations in capturing long-range dependencies and global context due to their localized receptive fields and fixed kernel structures. To overcome these challenges, transformer-based architectures, originally developed for natural language processing, have been adapted for vision tasks. Vision Transformers (ViTs), in particular, have shown promising results in a variety of applications by leveraging self-attention mechanisms to model global relationships across image patches. This capability makes them a strong candidate for RUL prediction using 2D signal images, where capturing both local and global degradation patterns is crucial for accurate estimation [26, 9].

Although recent advancements in deep learning and transformer-based models have shown promising results in prognostics, several challenges remain specific to the domain of bearing RUL prediction. One major issue is the presence of noise in vibration signals, especially in real-world applications, making it difficult for models to extract early indicators of degradation. Furthermore, inconsistencies in degradation behavior across different bearings, which may operate under varied loads, speeds, and environmental conditions, reduce the generalization capability of models trained under fixed assumptions. Many existing methods still rely on handcrafted features or shallow architectures that underutilize the rich temporal and spectral information embedded in raw signals. Even among deep learning models, standard 1D approaches often neglect the benefits of time-frequency analysis, while vision-based frameworks typically process only single-channel inputs, ignoring the complementary insights available from multi-directional sensors. Moreover, only a few studies incorporate effective denoising strategies, despite the fact that signal quality plays a crucial role in improving early failure detection.

In addition to these challenges, many prior works optimize solely for average accuracy and report aggregate error metrics like MAE, without distinguishing between early and late predictions. This is problematic in safety-critical settings, where late predictions are particularly risky, leading to unexpected failures and costly downtime. An effective RUL prediction framework must therefore go beyond accuracy, it must prioritize timely predictions that allow sufficient time for intervention, even if it means tolerating slightly premature alerts.

This work addresses these limitations by proposing a multi-channel Swin Transformer framework for bearing RUL prediction, leveraging time-frequency representations derived from Wavelet Packet Decomposition (WPD) of horizontal and vertical vibration signals. The pipeline begins with a three-stage denoising process applied to raw signals to suppress noise and enhance signal clarity. Next, WPD is used to extract detailed energy distributions across both low- and high-frequency components, capturing rich temporal-spectral features. The resulting energy maps from both sensor directions are stacked into multi-channel 2D representations and passed through a convolutional neural network for low- and mid-level feature extraction. These features are then fused and processed by a Swin Transformer, which employs hierarchical self-attention to capture both localized degradation patterns and long-range temporal dependencies. The framework is trained and evaluated on two benchmark datasets, PRONOSTIA and XJTU-SY, under both intra- and cross-condition settings. To bet-

ter align the model with the practical degradation process, First Prediction Time (FPT) is used to label the onset of failure for each bearing. Furthermore, a custom safety-aware loss function is introduced to prioritize timely predictions by penalizing late estimates more heavily than early ones. Altogether, the proposed approach aims to improve prediction accuracy, robustness to noise and condition variability, and operational safety in real-world predictive maintenance scenarios.

1.2 Contributions

The primary goal of this work is to improve the robustness, timeliness, and safety of remaining useful life (RUL) prediction for rolling bearings operating under variable and challenging conditions. To achieve this, a deep learning framework, MCSFormer, is developed, incorporating methodological innovations across preprocessing, model architecture, and loss formulation. The main contributions are summarized as follows:

- **Time-frequency-aware preprocessing pipeline:** A comprehensive preprocessing pipeline is proposed to enhance signal clarity and preserve degradation-related features across time and frequency domains. The raw vibration signals are first denoised through a multi-stage process involving low-pass filtering, wavelet-based soft thresholding, and Savitzky–Golay smoothing. This reduces high-frequency noise and measurement artifacts that can obscure subtle degradation patterns. A sliding window segmentation strategy is then applied to generate overlapping fixed-length samples, improving training stability, enabling faster processing, and decoupling the model from full run-to-failure histories. Each segment is transformed using Wavelet Packet Decomposition (WPD), which captures localized spectral characteristics across multiple frequency bands. The resulting time–frequency images serve as rich inputs for deep models, preserving both transient features and long-term degradation cues essential for accurate RUL prediction.

- **Dual-Sensor prediction using a multi-channel swin transformer framework:** A novel hybrid architecture, MCSFormer, is introduced, which leverages both horizontal and vertical vibration channels through parallel CNN branches. The resulting feature maps are fused and passed into a Swin Transformer backbone with shifted-window self-attention, allowing the model to capture both directional fault patterns and long-range temporal degradation trends. This

dual-path structure significantly improves representation capacity and cross-condition generalization.

- **Safety-aware loss function for conservative and timely RUL prediction:** A custom loss function is designed to reflect real-world maintenance priorities by penalizing late predictions more heavily than early ones. This asymmetric penalty guides the model toward conservative yet accurate RUL estimates, minimizing the risk of unexpected failures while maintaining strong overall performance.

1.3 Thesis Outline

This thesis is organized into seven chapters, each addressing a critical aspect of the research, from foundational concepts to model development, evaluation, and future directions:

- **Chapter 1 – Introduction:** Establishes the context of RUL prediction within predictive maintenance. It presents the motivation behind the work, highlights key challenges in the field, introduces the proposed approach, and summarizes the main contributions of the thesis.
- **Chapter 2 – Background and Literature Review:** Provides a detailed overview of existing RUL prediction methodologies, including traditional statistical models, machine learning approaches, and modern deep learning architectures. It also explores the role of time-frequency analysis, segmentation techniques, and transformer-based models in improving bearing prognostics.
- **Chapter 3 – Preliminaries and Theoretical Background:** Defines fundamental concepts such as RUL, FPT, and evaluation metrics. It discusses domain-specific challenges and presents the theoretical basis for the signal processing, feature extraction, and modeling techniques employed in the proposed framework.
- **Chapter 4 – Datasets and Preprocessing Pipeline:** Introduces the two benchmark datasets used in this study, PRONOSTIA and XJTU-SY, and explains the preprocessing pipeline in detail. This includes denoising techniques,

sliding window segmentation, WPD, normalization, and RUL label generation using FPT.

- **Chapter 5 – MCSFormer Framework:** Describes the architecture of the proposed multi-channel Swin Transformer framework. It details the dual-branch CNN-based feature extractor, the hierarchical Swin Transformer backbone, the fusion strategy for multi-directional data, and the custom safety-aware loss function designed to penalize late predictions.
- **Chapter 6 – Experimental Analysis and Results:** Presents the experimental design, including intra- and cross-condition evaluations, and analyzes model performance using MAE and Score metrics. It includes comprehensive ablation studies to assess the contributions of denoising, segmentation strategy, and loss function design.
- **Chapter 7 – Conclusions and Future Work:** Summarizes the findings and contributions of the research. It discusses the practical implications of the results, acknowledges limitations, and outlines promising directions for future improvements and extensions of the proposed framework.

Chapter 2

Background and Literature Review

2.1 Predictive Maintenance and RUL Prediction

Modern industrial systems rely on the continuous operation of critical machinery across diverse sectors, including transportation, manufacturing, energy, and aerospace. Components such as bearings, turbines, motors, and gearboxes are essential to sustaining production efficiency, operational safety, and equipment longevity. Unanticipated failures in such components can result in costly downtime, unsafe working conditions, emergency repairs, and substantial financial losses. As a result, maintenance planning has become a vital part of operational strategy for ensuring equipment reliability and minimizing disruptions.

Over the years, maintenance strategies have evolved significantly. Traditional approaches such as corrective maintenance respond to failures after they occur, which often leads to unplanned shutdowns and high recovery costs. Preventive maintenance aims to mitigate these risks by performing service at fixed intervals, regardless of actual component condition. While this strategy reduces failure rates, it may also result in unnecessary maintenance, increased labor, and wasted resources. In contrast, predictive maintenance represents a proactive, data-driven paradigm that uses sensor data, such as vibration, temperature, and acoustic measurements, to assess the real-time health of components and anticipate failures before they happen. This approach not only optimizes maintenance schedules and resource allocation but also aligns with broader goals of Industry 4.0 by enabling intelligent, autonomous decision-making in industrial systems.

A central objective of predictive maintenance is the estimation of the RUL of

critical components. Accurate RUL predictions enable maintenance activities to be scheduled proactively, thereby minimizing unplanned downtime, optimizing resource allocation, and extending the operational lifespan of equipment.

The estimation of RUL involves analyzing various data sources, including historical performance records, sensor measurements, and environmental conditions. Techniques for RUL prediction have evolved over time, encompassing:

- **Model-based approaches**: These utilize physical and degradation models to simulate the behavior of components. For instance, physics-based models can estimate state and damage progression parameters, predicting the end of life by propagating these estimates [11].
- **Data-driven methods**: Leveraging statistical and machine learning algorithms, these methods learn patterns from historical data without explicit physical models. Deep learning techniques, such as RNNs and CNNs, have been applied to RUL prediction tasks [53].
- **Hybrid strategies**: Combining elements of both model-based and data-driven methods, hybrid approaches aim to capitalize on the strengths of each. For example, integrating physics-based performance models with deep learning algorithms can enhance prognostic accuracy in complex systems [5].

The selection of an appropriate methodology depends on factors such as the availability of data, the complexity of the system, and the specific requirements of the maintenance strategy.

2.2 Traditional Model-Based Approaches

Before the advent of data-driven techniques, traditional RUL prediction methods predominantly relied on model-based approaches grounded in physical principles and mathematical formulations. These methods aim to describe the degradation behavior of components using deterministic or probabilistic models derived from first principles, domain knowledge, or empirical observations.

A notable category includes physics-based models that simulate damage accumulation mechanisms, such as fatigue, wear, or crack growth, based on stress, load, and environmental conditions. These models offer interpretability and generalizability across systems with well-understood failure mechanisms. For instance, Paris' law

is widely used to model fatigue crack growth by relating the crack growth rate to the range of stress intensity factors experienced by the material [41]. Such models have been extensively reviewed in the literature, highlighting their applications and limitations in various engineering contexts [11].

Another class involves stochastic process models, such as the Wiener and Gamma processes, which model degradation as a random progression over time. These probabilistic models estimate RUL by learning the parameters of degradation trajectories and predicting the remaining time until a defined failure threshold is reached. Lu and Meeker [32] introduced a general path model that utilizes degradation measures to estimate time-to-failure distributions, providing a framework for incorporating stochastic degradation paths into reliability analysis. Recent advancements have incorporated adaptive mechanisms to account for multi-source variability in degradation processes. For instance, Zheng et al. [63] proposed an adaptive Wiener process-based method that considers the variability arising from uneven measurement intervals and inconsistent measurement frequencies, enhancing the accuracy of RUL predictions in complex systems.

Despite their usefulness, model-based approaches typically require extensive domain expertise, precise material properties, and controlled conditions, which may not be feasible in real-world industrial environments where variability and noise are common. Challenges such as model validation, computational complexity, and the need for comprehensive failure data have been identified as significant barriers to the widespread adoption of these methods in complex systems [61].

2.3 Data-Driven Approaches

The increasing availability of sensor data in industrial systems has led to a major shift from traditional model-based approaches to data-driven techniques for RUL prediction. Model-based methods, such as those built on physical degradation modeling or stochastic processes like the Wiener and Gamma distributions, require prior knowledge of system dynamics, material properties, and failure mechanisms. Although such models can offer accurate results when assumptions hold, they are often limited in practical scenarios due to the difficulty of capturing complex degradation behaviors, environmental variations, and system-specific nonlinearities. In contrast, data-driven approaches infer the mapping between condition monitoring data and component health without requiring detailed physics. These methods are better suited to handle

noisy, multi-source data and have demonstrated adaptability across a range of systems and operating conditions. Particularly in rotating machinery such as bearings, where degradation processes are affected by multiple latent factors and external influences, data-driven models have shown superior flexibility and predictive capability over traditional physics-based approaches [46, 48].

Classical machine learning techniques have significantly contributed to the advancement of data-driven RUL prediction, particularly in industrial systems where degradation behavior is too complex for analytical modeling. These algorithms leverage features extracted from sensor signals, such as time-domain statistics, frequency components, or engineered health indicators, to learn mappings from condition data to estimated life. ML models are typically faster to train and evaluate than deep learning counterparts and remain attractive for many real-world applications, especially when datasets are limited or interpretability is crucial.

Notable early ML approaches include Support Vector Regression (SVR), k-Nearest Neighbors (k-NN), and Random Forests (RF). These techniques are particularly well-suited for structured degradation prediction and have been widely explored in the literature for bearing health monitoring and RUL estimation.

SVR has been successfully applied to predict the RUL of mechanical components, including bearings, due to its ability to model nonlinear degradation trends while maintaining strong generalization performance. For example, Yan et al. [58] proposed a hybrid degradation tracking model using SVR for bearing RUL prediction, demonstrating improved accuracy over traditional methods. Similarly, Huang et al. [20] reviewed the application of SVM-based methods in RUL estimation, highlighting their effectiveness in handling small sample sizes and multi-dimensional data.

The k-NN algorithm offers a non-parametric alternative by inferring RUL based on historical instances with similar degradation patterns. Despite its simplicity, it has been shown to provide reliable estimates when degradation progresses gradually and sufficient failure data is available. Bhandare et al. [4] discussed the use of k-NN for bearing fault diagnosis and RUL prediction, emphasizing its robustness and non-parametric nature. Additionally, a novel method utilizing k-NN for classifying rolling element bearing faults demonstrated high accuracy in RUL prediction within 95% confidence limits [37].

RFs construct multiple decision trees on bootstrapped subsets of the data and aggregate their predictions to reduce overfitting and enhance robustness. Alfarizi et al. [1] proposed an optimized RF model for bearing RUL prediction, integrating

empirical mode decomposition and Bayesian optimization to improve forecasting performance. Furthermore, Palaniappan [40] conducted a comparative analysis of SVR, RF, and k-NN classifiers for predicting RUL of rolling bearings, concluding that SVR achieved the highest mean classification accuracy, followed closely by RF and k-NN.

Comparative analyses in the literature have shown that while SVR often yields high predictive accuracy, RFs are easier to tune and interpret. In contrast, k-NN's performance heavily depends on proper distance metrics and data quality. The selection of an appropriate algorithm is therefore influenced by task complexity, feature design, model transparency, and data volume.

2.3.1 Deep Learning Approaches

In recent years, deep learning techniques have emerged as powerful tools for RUL prediction, owing to their ability to automatically learn hierarchical representations from raw or minimally processed data. Unlike classical machine learning models, which often rely on handcrafted features and shallow architectures, deep neural networks can extract both local and global patterns from complex sensor signals without extensive manual intervention. This has positioned DL as a natural fit for prognostics tasks, especially in the context of vibration-based health monitoring, where data is typically high-dimensional and nonlinear. The growing availability of run-to-failure datasets, has further accelerated the adoption of DL models for bearing degradation modeling and prediction.

Convolutional Neural Networks have become a widely adopted architecture for RUL prediction, particularly in bearing health monitoring, due to their effectiveness in capturing spatial patterns and local degradation features directly from raw or preprocessed sensor signals. Unlike traditional machine learning methods that depend on manual feature engineering, CNNs learn hierarchical representations automatically through stacked convolutional layers. This makes them particularly suitable for analyzing high-resolution time-series or time-frequency inputs derived from vibration signals. In bearing prognostics, CNNs are often used to process either raw waveforms or transformed 2D representations (such as spectrograms or wavelet packet decomposition maps), enabling the model to detect subtle yet informative degradation patterns, as demonstrated in the work of Cheng et al. [6].

CNNs have proven especially powerful when applied to time-frequency representations of sensor signals, such as spectrograms, continuous wavelet transforms, or

wavelet packet decomposition (WPD) maps. These image-like inputs allow CNNs to capture localized degradation patterns that are often difficult to identify in raw time-series data. Li et al. [26] developed a multiscale CNN framework for RUL estimation using spectrogram-based representations of bearing data, showing improved prediction accuracy and robustness. Similarly, Liu et al. [28] introduced the SAL-CNN model, which leverages short-time Fourier transform (STFT) maps and spatial attention layers to guide the model's focus toward informative frequency regions, achieving competitive performance on benchmark datasets. In another example, Wang et al. [51] proposed a recurrent convolutional neural network (RCNN) that integrates temporal dependencies into CNN architectures, enhancing the model's ability to capture sequential degradation patterns in machinery. These studies confirm that CNNs, when combined with appropriate signal transformations and architectural enhancements, are highly effective for modeling the complex degradation behaviors present in bearing systems.

Recent advancements in CNN architectures also have significantly enhanced their capability to predict the RUL of machinery components. One notable development is the integration of residual connections, which facilitate the training of deeper networks by mitigating issues like vanishing gradients. For instance, Lui and Xie [33] proposed a Deep Residual Network (DRN) that effectively captures complex degradation patterns in turbofan engines, demonstrating superior performance on the NASA C-MAPSS dataset.

Attention mechanisms have also been incorporated into CNNs to enhance their focus on critical features within the data. Li et al. [25] introduced an enhanced CNN-LSTM model augmented with a Convolutional Block Attention Module (CBAM), which assigns adaptive weights to spatial and channel features, leading to improved RUL prediction accuracy for aircraft engines.

Moreover, the utilization of multi-channel CNNs has proven beneficial in handling data from multiple sensors or different signal modalities. He et al. [16] proposed a hybrid model combining a multichannel 1D CNN with a Bidirectional LSTM and a self-attention mechanism, which significantly enhanced the prediction accuracy of aero-engine RUL by effectively extracting and integrating features from various sensor channels.

These architectural innovations underscore the evolving landscape of CNN-based models in RUL prediction, highlighting the trend toward more sophisticated and hybrid approaches that leverage the strengths of various deep learning components

to achieve higher accuracy and robustness.

While CNNs have demonstrated strong capabilities in extracting local features from static or transformed representations of sensor signals, they are inherently limited in modeling temporal dependencies across time steps. This poses a challenge in applications like RUL prediction, where degradation evolves gradually and understanding the sequence of states is crucial. To overcome this limitation, Recurrent Neural Networks were introduced as a class of architectures specifically designed to handle sequential data. By maintaining hidden states across time, RNNs allow the model to incorporate historical information into current predictions.

Several studies have explored the application of RNNs for RUL prediction across different domains. Zhang et al. [62] developed a bidirectional GRU model enhanced with a temporal self-attention mechanism for aircraft engine RUL prediction using the C-MAPSS dataset. Their model adaptively emphasized important time steps and achieved substantial improvements in prediction accuracy and robustness compared to standard RNNs. In the energy domain, Jafari and Byun [22] proposed a CNN-GRU framework that utilizes charging profiles of lithium-ion batteries to forecast RUL. Their model integrated spatial degradation features through CNN and temporal dependencies through GRU, producing accurate predictions across various charging conditions. In the bearing systems domain, Ma et al. [35] introduced a hybrid model combining a multiscale efficient channel attention CNN with a bidirectional GRU (BiGRU), effectively capturing both spatial and temporal features. Evaluated on vibration data from rolling bearings, their approach achieved high accuracy and robustness across multiple degradation patterns.

Despite their ability to model sequential data, traditional RNNs encounter significant challenges when dealing with long-term dependencies due to issues like vanishing and exploding gradients. These problems hinder the network's capacity to retain information over extended sequences, which is critical in accurately predicting the RUL of machinery components. To address these limitations, Hochreiter and Schmidhuber [18] introduced the Long Short-Term Memory (LSTM) network, which incorporates memory cells and gating mechanisms to better capture and maintain long-term dependencies in data sequences.

LSTMs have demonstrated considerable success in various RUL prediction tasks. For instance, Zheng et al. [64] applied an LSTM-based approach to predict the RUL of aircraft engines using the C-MAPSS dataset, achieving improved accuracy over traditional RNN models. In the energy domain, Park et al. [42] developed a frame-

work that utilizes LSTM networks and multi-channel charging profiles for lithium-ion battery RUL prediction, showing that integrating rich temporal information from voltage and current signals can lead to more accurate capacity estimation. In the context of bearing systems, Liu et al. [30] introduced a hybrid TCN-LSTM model that combined the strength of temporal convolutions with gated memory, leading to improved RUL prediction under complex operating conditions.

Recent advancements in bearing RUL prediction have seen the emergence of hybrid models that combine LSTM networks with other deep learning architectures to address the complex degradation patterns inherent in rolling bearings. These hybrid approaches aim to leverage the strengths of individual components to improve prediction accuracy and robustness. For instance, Wang et al. [52] proposed a framework that integrates a multistage convolutional autoencoder (MSCAE) with a bias-corrected LSTM (BCM-LSTM) network. The MSCAE effectively captures features across different degradation stages, while the BCM-LSTM mitigates prediction biases, leading to enhanced RUL estimation accuracy. Similarly, Huang et al. [21] developed a hybrid model combining CNNs, LSTM, and an attention mechanism. In this architecture, CNNs extract spatial features from vibration signals, LSTMs model temporal dependencies, and the attention mechanism emphasizes critical information, collectively improving the model's predictive performance.

Further extending this paradigm, Bao et al. [3] introduced a dual-attention mechanism within an LSTM framework. This model applies attention mechanisms to both time steps and feature dimensions, allowing the network to focus on salient temporal and spatial features, thereby enhancing RUL prediction accuracy.

These hybrid models underscore the growing recognition that no single deep learning architecture is sufficient to capture the full complexity of machinery degradation processes. By integrating LSTM networks with complementary modules, such as convolutional layers for spatial feature extraction, autoencoders for dimensionality reduction and denoising, and attention mechanisms for dynamic relevance weighting, researchers have developed increasingly powerful models for RUL prediction. These hybrid approaches have been widely adopted not only in bearing health monitoring but also in domains such as aircraft engines, lithium-ion batteries, and industrial gearboxes, where degradation patterns are nonlinear, nonstationary, and difficult to model with traditional techniques.

2.3.2 Transformer-Based Approaches

Despite the success of CNNs and LSTM networks in modeling degradation patterns for RUL prediction, their inherent limitations have motivated the exploration of attention-based architectures. CNNs, while powerful in extracting localized spatial features from vibration signals and their transformations, often fail to capture long-range temporal dependencies effectively. LSTMs and GRUs partially address this by introducing memory mechanisms, but their sequential nature limits parallelization and makes them computationally inefficient for long time series. Additionally, both architectures can suffer from difficulty attending to the most informative segments of input sequences, especially when degradation evolves subtly or irregularly, a common challenge in bearing health monitoring where early fault indicators may be weak and distributed across time.

Transformer architectures, originally introduced for natural language processing tasks by Vaswani et al. [49], offer a compelling alternative by utilizing self-attention mechanisms to model global dependencies across entire input sequences. Unlike RNNs, Transformers process all time steps simultaneously and learn to assign adaptive weights to different positions in the sequence. This is particularly advantageous in bearing prognostics, where informative features may appear at variable and distant time intervals, and capturing such dependencies is crucial for accurate and early RUL estimation. Furthermore, the growing availability of large run-to-failure datasets such as PRONOSTIA [38], XJTU-SY [50], and others has made it feasible to train deeper attention-based architectures that generalize well across operating conditions and bearing types. These benefits have positioned Transformers as a natural progression from LSTM-based models for capturing complex temporal patterns in RUL tasks.

For time-series data, the Transformer encoder, comprising stacked self-attention layers and feed-forward blocks, can be directly applied to the raw or preprocessed sequences. Recent adaptations include modifications to positional encoding, normalization, and attention scaling to better suit the characteristics of time-domain or frequency-domain inputs. For instance, specialized Transformer variants such as Informer [66], Autoformer [54], and Linear Transformer [23] have been developed to handle long-sequence dependencies and seasonal-trend decomposition, offering improved efficiency and accuracy in time-series RUL prediction tasks. Compared to CNNs and LSTMs, Transformer-based models offer greater flexibility in handling nonstationary

signals, long degradation cycles, and varying failure patterns, which are commonly observed in industrial machinery. As a result, researchers have increasingly adopted Transformer variants to better exploit temporal correlations and dynamic feature relevance across long operating sequences. For instance, Liu et al. [29] proposed a double attention-based data-driven framework for aircraft engine RUL prognostics, leveraging both channel and temporal attention mechanisms to enhance prediction accuracy. Their model outperformed traditional deep learning approaches on the C-MAPSS dataset, highlighting the effectiveness of attention mechanisms in RUL tasks.

Moreover, Kim et al. [24] introduced a Transformer-based framework for predicting the RUL of lubricants in rolling bearings. By integrating a harmonic frequency transformer with vibration signal analysis, their approach effectively captured degradation patterns, leading to improved prediction performance under varying operating conditions. Similarly, Zhan et al. [60] developed a two-stage framework combining a temporal convolutional network and a CNN–Transformer model to predict the RUL of bearings. Their method demonstrated superior accuracy compared to traditional neural networks, particularly in scenarios with complex degradation behaviors.

While traditional Transformer models have shown considerable potential in capturing long-range dependencies in time-series data, their direct application to RUL prediction, especially for rolling bearings, has revealed a few limitations. These models often struggle to effectively capture fine-grained local spatial patterns that are critical in vibration-based prognostics. This shortcoming becomes particularly pronounced in scenarios where degradation signals exhibit nonstationary behavior, noise contamination, or early subtle changes.

To address these issues, researchers have turned to Vision Transformers (ViTs), which divide input data into sequences of image-like patches, enabling the simultaneous modeling of both local spatial features and long-range dependencies. In the context of bearing health monitoring, Hu et al. [19] proposed a ViT-based RUL prediction framework that transforms time-series sensor signals into image representations suitable for vision-based processing. These representations, such as spectrograms and wavelet packet decompositions, preserve both temporal and frequency-domain features, enabling ViTs to attend to localized degradation patterns that may be missed in raw sequences. Their model demonstrated improved performance over standard CNNs by more effectively capturing the spatial structure of degradation patterns in rolling bearings. Similarly, Fan et al. [10] introduced PerFormer, a permutation-

based ViT designed to handle multivariate time-series data. By converting these sequences into ViT-compatible inputs, the model achieved state-of-the-art results on the C-MAPSS dataset, underscoring the potential of vision-based architectures in prognostic modeling.

Building upon the strengths of ViTs, the Swin Transformer has emerged as a particularly promising architecture for RUL prediction. Its hierarchical structure and shifted window mechanism allow it to model both local and global dependencies with high efficiency, making it well-suited for processing time-frequency representations such as spectrograms or wavelet-based transforms. Unlike standard ViTs, which use global attention and fixed-size patch tokens, the Swin Transformer's hierarchical design and shifted windows allow for better modeling of local patterns, which are critical in machinery health monitoring applications. Hao et al. [15] proposed a bi-channel hierarchical Vision Transformer (BCHViT) for bearing RUL prediction, which extracts degradation features at multiple scales through dual-channel fusion. Their model achieved superior accuracy on the PRONOSTIA dataset, demonstrating the Swin Transformer's capacity to extract meaningful spatial-temporal patterns from complex inputs. Further advancing this approach, Xie et al. [57] designed a Swin Transformer framework that integrates CNNs, Bi-LSTM modules, and Gramian Angular Field (GAF) representations, along with domain adaptation techniques. Their model addressed cross-domain variability in bearing degradation data and delivered strong generalization performance.

Despite notable advancements in deep learning for prognostics, standalone ViT models and their derivatives, such as Swin Transformer, continue to face challenges when applied to raw, noise-prone vibration signals. These signals often contain early-stage degradation patterns that are weak and transient, making them difficult to capture using purely global attention mechanisms. To address these limitations, recent research has turned toward hybrid architectures that fuse the localized feature extraction strengths of CNNs with the contextual modeling capabilities of Transformer-based backbones. Such hybrids improve robustness to operational noise and variability while benefiting from structured attention mechanisms for better interpretability. However, they often incur increased computational cost, demanding more memory and GPU resources. These trade-offs, while non-trivial, are frequently justified by significant performance improvements in RUL prediction tasks.

2.3.3 Data Preparation

One emerging design consideration in recent RUL prediction work is the segmentation strategy used to generate training samples from continuous sensor streams. Methods such as sliding windows, which segment data into overlapping fixed-length intervals, and expanding windows, which grow the segment over time, have been proposed to enhance sample diversity and capture progressive degradation trends more effectively. These strategies not only help the model generalize better by exposing it to a broader range of failure progression patterns but also enable earlier fault detection by increasing temporal coverage. For instance, Ogunfowora and Najjaran [39] demonstrated the influence of segmentation on RUL prediction performance using a Transformer-based architecture, highlighting how varying window strategies affect model convergence and accuracy. They showed that using expanding windows in a Transformer-based RUL prediction model led to over 20% improvement in MAE compared to fixed-length sliding windows. This highlights the importance of treating segmentation not just as a formatting step, but as a crucial design decision in prognostics pipelines.

Nevertheless, segmentation remains an underexplored but critical factor in many studies. Despite its potential to influence model generalization and robustness, many recent transformer-based RUL frameworks overlook its effect or treat it as a fixed preprocessing step without further analysis.

Equally important is the role of signal preprocessing, which serves as a critical enabler for reliable feature extraction and effective learning. Vibration signals in machinery health monitoring are characteristically non-stationary and often obscured by high-frequency noise, making raw input signals suboptimal for training deep models. Early fault indicators, especially in bearings, tend to appear as localized transients that may be masked by irrelevant fluctuations or measurement noise. As a result, signal processing techniques have become an integral part of modern predictive maintenance pipelines.

Wavelet-based approaches have gained significant traction for this purpose, particularly due to their ability to perform time-frequency analysis with high temporal resolution. Unlike Fourier-based methods that provide global spectral summaries, wavelet transforms enable localized, multi-scale analysis that is well suited for capturing short-term transients indicative of incipient faults [2]. Among these, Wavelet Packet Decomposition (WPD) has emerged as a powerful technique, extending the traditional Discrete Wavelet Transform (DWT) by recursively decomposing both ap-

proximation and detail coefficients. This results in a richer, more granular representation of signal energy across the full frequency spectrum [13].

The effectiveness of WPD, however, hinges on the quality of the input signal. High-frequency noise can obscure critical fault features or introduce artifacts into the resulting time-frequency maps. To counteract this, wavelet-based denoising techniques, notably the adaptive soft thresholding approach introduced by Donoho [8], are widely employed. These methods selectively attenuate small-magnitude coefficients that are statistically likely to represent noise, preserving the structural integrity of meaningful signal components. By integrating such denoising procedures prior to decomposition, WPD-based representations become more robust and informative for downstream learning.

Thus, the combination of denoising and WPD transforms the raw signal into a structured input that aligns well with vision-oriented deep learning models. When used in conjunction with CNNs and Transformer variants, these preprocessing steps not only enhance interpretability but also substantially improve the model's capacity to capture degradation progression and deliver reliable RUL estimates.

To this end, an earlier version of the framework developed in this thesis, called the Multi-Channel Swin Transformer (MCSFormer), was introduced by the author in a prior publication [36]. That initial design demonstrated the viability of using wavelet-based denoising and WPD for time-frequency transformation, combined with CNN layers for local feature extraction and a Swin Transformer backbone for capturing spatial-temporal dependencies across horizontal and vertical vibration signals. It also included a custom loss function designed to prioritize timely predictions in line with practical maintenance needs. Evaluated on the PRONOSTIA dataset, the original MCSFormer showed strong performance in terms of MAE and generalization across conditions.

In this thesis, the framework has been significantly extended and refined with a more comprehensive preprocessing pipeline, improved segmentation strategy, and deeper ablation analysis across multiple datasets. Table 2.1 compares this prior version with other representative Transformer-based models for RUL prediction, highlighting differences in signal processing, architectural choices, and attention mechanisms.

Table 2.1: Comparison of Transformer-based models for RUL prediction across signal processing and architectural features.

Model	Denoising	Feature Prep.	Spatial Modeling	Temporal Modeling	Attention Mechanism	Dataset
Double-Attn Transformer [29]	✗	Raw Time-Series	✗	✓	Temporal + Channel Attention	C-MAPSS
Informer [66]	✗	Raw Time-Series	✗	✓	ProbSparse Self-Attention	Electricity, Weather
Autoformer [54]	✗	Raw Time-Series	✗	✓	Decomposition-based Self-Attention	Electricity, Weather
ViT [19]	✗	GAF Images	✓	✗	Vision Transformer (Global Self-Attention)	PRONOSTIA
PerFormer [10]	✗	Time-Series to Patches	✗	✓	Permutation-based Self-Attention	C-MAPSS
BCHViT [15]	✗	Dual-Channel Frequency Maps	✓	✓	Hierarchical Vision Transformer Attention	PRONOSTIA
Swin-CNN-BiLSTM [57]	✓	GAF + CNN + BiLSTM	✓	✓	Shifted Window-based Self-Attention	PRONOSTIA
MCSFormer [36]	✓	WPD + CNN	✓	✓	Hierarchical Swin Transformer Self-Attention	PRONOSTIA

In summary, the field of RUL prediction has undergone a notable transformation, from traditional shallow models reliant on handcrafted features to advanced deep learning architectures capable of extracting hierarchical representations from complex sensor data. CNNs laid the foundation by enabling spatial feature extraction from raw or transformed vibration signals. RNNs and LSTMs extended this by introducing mechanisms to model temporal dependencies. Vision-based transformers, especially Swin Transformers, have further pushed the boundaries by offering joint modeling of spatial and temporal structures through attention mechanisms. Hybrid models that combine CNNs, transformers, and advanced preprocessing pipelines now represent the forefront of RUL prediction research.

This chapter has surveyed the key developments and representative models that have shaped the trajectory of data-driven prognostics, particularly in the context of rolling bearings. The insights gained from this literature inform the design choices behind the proposed MCSFormer framework. In the following chapters, we delve into the theoretical background, datasets, and experimental methodology that underpin the model's development and validation.

Chapter 3

Preliminaries and Theoretical Background

This chapter provides the theoretical foundations and notational definitions required to understand the proposed framework for RUL prediction of rolling bearings. It begins by formalizing essential terms such as RUL, First Prediction Time (FPT), and normalized degradation labels. Next, it outlines the motivation and mathematical background behind the transformation of raw vibration signals into time-frequency representations, with a focus on wavelet-based denoising and Wavelet Packet Decomposition (WPD). The chapter also reviews key deep learning building blocks used in this work, including CNNs, ViTs, and the Swin Transformer. Finally, it presents the evaluation metrics used for model performance.

3.1 Key Definitions and Notation

3.1.1 Remaining Useful Life (RUL)

Remaining Useful Life (RUL) refers to the expected duration a mechanical component will continue to operate within acceptable performance limits before reaching the end of its service life. In predictive maintenance, RUL is the central target of estimation, enabling maintenance decisions that prevent failure while avoiding unnecessary part replacements. For a component under observation at time t, RUL is mathematically defined as shown in Equation (3.1):

$$\mathrm{RUL}(t) = t_f - t \tag{3.1}$$

where t_f is the time of failure. Since failure time t_f is unknown during operation, predictive models estimate RUL based on degradation signals such as vibration measurements. In this work, to standardize across different bearings and lifespans, the RUL is normalized to the range $[0, 1]$, where a value of 1 corresponds to the beginning of the degradation phase, and 0 represents failure. This normalization facilitates training and comparison across bearings with different life durations.

3.1.2 First Prediction Time (FPT)

In practical predictive maintenance systems, raw sensor data includes both healthy and degrading stages of a component's life. However, only the degradation phase contributes useful information for training models to estimate RUL. The First Prediction Time (FPT) defines the transition point from the healthy state to the onset of degradation and is used to filter out non-informative early data. Failing to exclude the healthy-stage data can mislead the model, leading to poor convergence and degraded performance.

In this work, the FPT is identified using the statistical property of kurtosis, a fourth-moment measure that is sensitive to outliers and changes in distribution shape. During a bearing's healthy operation, kurtosis values calculated from short time segments tend to stay within a stable range. As damage initiates, the vibration signal becomes more impulsive, resulting in an abrupt deviation in kurtosis.

To detect this deviation robustly, we compute the sample mean μ_K and standard deviation σ_K of kurtosis over the initial (assumed healthy) portion of the signal. The FPT is then determined as the first time point t at which the kurtosis K_t exits the 3-standard-deviation confidence interval, defined in Equation (3.2), for three consecutive measurements:

$$K_t \notin [\mu_K - 3\sigma_K,\ \mu_K + 3\sigma_K] \tag{3.2}$$

This condition minimizes the influence of transient spikes due to noise and ensures a reliable estimate of degradation onset. All samples prior to the detected FPT are excluded from training and evaluation, and the RUL labeling process begins at FPT, with the corresponding RUL label set to 1 and decreasing linearly thereafter.

3.1.3 Normalized RUL Labeling

Once the FPT is identified, the remaining samples in the bearing's life are considered part of the degradation phase and are used for training and evaluation. For effective learning and generalization across bearings with different total lifetimes, the RUL is normalized to the range $[0, 1]$. This normalization not only stabilizes model training but also allows direct comparison between bearings under different operating conditions.

Let N_d denote the number of samples after the FPT for a given bearing, and let $i \in \{0, 1, \ldots, N_d - 1\}$ index each post-FPT sample. The normalized RUL label y_i assigned to the i-th sample is computed using a simple linear decay function, as defined in Equation (3.3).

$$y_i = 1 - \frac{i}{N_d - 1} \tag{3.3}$$

Under this formulation, the first sample after FPT (i.e., $i = 0$) receives a label of 1, while the final sample before failure (i.e., $i = N_d - 1$) receives a label of 0. This approach ensures smooth label decay aligned with degradation progression and implicitly encourages early prediction behavior in the learning objective. These labels are used as continuous regression targets in the training process.

The definitions introduced in this section establish the basis for the data representation and modeling strategy adopted in this thesis. RUL estimation is treated as a continuous regression problem, where each bearing's degradation is modeled from its FPT onward. By assigning normalized labels between 1 and 0 to the post-FPT samples, the task becomes consistent across varying bearing lifespans and operating conditions. Figure 3.1 illustrates a representative vibration signal from a bearing undergoing gradual degradation. The lower plot shows the normalized RUL label, which remains constant at 1 prior to the FPT and then decreases linearly to 0 as the bearing approaches failure.

3.2 Challenges in RUL Prediction

3.2.1 Degradation Complexity and Signal Nonstationarity

One of the primary challenges in RUL prediction is the inherent complexity and variability of degradation processes. In real-world applications, mechanical components

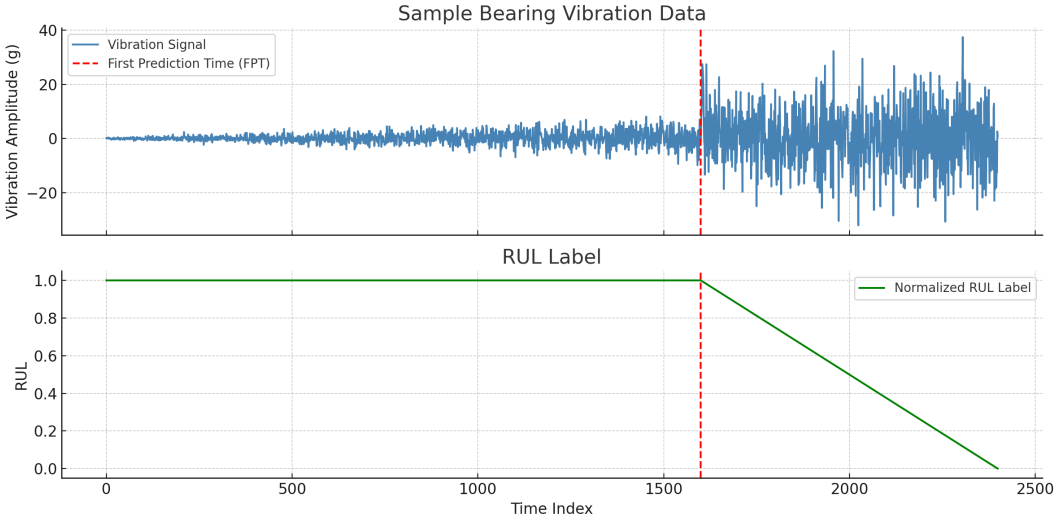

Figure 3.1: Sample bearing vibration data with the assigned RUL plot.

such as rolling bearings degrade under dynamic and often unpredictable conditions. The degradation patterns are rarely linear and may be influenced by multiple interacting factors including speed, load, lubrication, temperature, and operating history. These factors introduce stochastic behavior into the signal, resulting in patterns that are not easily generalized across different machines or even across different instances of the same machine.

Moreover, the vibration signals used for prognostics are fundamentally nonstationary, meaning their statistical properties evolve over time. In the early stages of a component's lifecycle, vibration signals may appear stable and low in amplitude, but as wear progresses, these signals exhibit increased variance, higher-order statistical changes (such as kurtosis), and transient bursts associated with developing faults. Capturing these evolving features reliably requires signal processing techniques that can adapt to or explicitly model nonstationarity, such as time-frequency transformations or data-driven temporal modeling.

3.2.2 Early vs. Late Prediction Trade-Off

A critical consideration in RUL modeling is the trade-off between early and late prediction accuracy. An early prediction occurs when the model estimates a smaller RUL than the actual remaining life, whereas a late prediction occurs when the predicted RUL is longer than the true remaining time. While both types of error are undesirable, late predictions are particularly problematic in safety-critical systems, as they

suggest a false sense of reliability and may lead to unexpected failures without timely intervention.

In contrast, early predictions, though more conservative, allow maintenance teams to act in advance and prevent potential damage or downtime. However, excessive early predictions can result in unnecessary maintenance and increased operational cost. Therefore, the goal is not simply to predict earlier but to maintain accuracy across the entire degradation trajectory. In practice, slight early predictions are often preferable to late ones due to their safer implications.

3.2.3 Variability and Generalization Across Domains

Another major challenge lies in the generalization of RUL models across varying operating conditions, machine types, and sensor configurations. A model trained on data from one set of bearings, operating under controlled conditions, may perform poorly when deployed on bearings subject to different speeds, loads, or fault types. This is particularly problematic in industrial settings where labeled failure data is limited, and different techniques may be required to bridge the gap between training and deployment domains.

In the case of the bearing datasets used in this thesis, both PRONOSTIA and XJTU-SY, bearings are tested under three distinct operating conditions with different rotational speeds and loads. A successful model must learn not only the temporal degradation signatures of each bearing but also remain invariant to domain shifts across operating settings. Addressing this issue is critical to building reliable and generalizable prognostic systems that can function in real-world environments where diversity and unpredictability are the norms.

3.3 Signal Denoising for Vibration-Based Prognostics

Accurate RUL prediction relies heavily on the quality of input signals. In real-world industrial environments, vibration signals recorded from sensors are often contaminated with various types of noise that obscure underlying degradation patterns. These sources of noise include electrical interference, mechanical backlash, environmental vibrations, and imperfections in sensor calibration. Particularly during the early stages of bearing wear, when failure indicators are faint and irregular, noise can significantly

reduce the detectability of meaningful features, distort key statistical indicators, and hinder the performance of machine learning models.

To ensure that critical degradation signatures are preserved while irrelevant fluctuations are suppressed, denoising is considered an essential preprocessing step in vibration-based prognostics. Various filtering techniques have been developed to address this challenge, each offering distinct advantages depending on the nature of the signal and the type of noise present. Commonly used methods include low-pass filtering for attenuating high-frequency interference, Savitzky–Golay filtering for smoothing without distorting local signal trends, and wavelet-based denoising for multi-scale and nonstationary signal decomposition. These approaches contribute in different ways to enhancing signal quality, facilitating more reliable feature extraction and model training. The following subsections present the theoretical background and typical use cases for each technique in the context of bearing health monitoring.

3.3.1 Low-Pass Filtering

Low-pass filtering is a fundamental technique in signal processing used to suppress high-frequency components of a signal while preserving lower-frequency trends. In the context of vibration analysis for machinery health monitoring, high-frequency noise often arises from electrical interference, environmental vibrations, or sensor instability, and may not carry relevant information about the degradation state of the component. Low-pass filters can therefore improve the interpretability of the signal by reducing such noise and enhancing the visibility of underlying patterns such as amplitude modulation or periodic impulses.

Finite impulse response (FIR) and infinite impulse response (IIR) filters are commonly used implementations, with design parameters such as cutoff frequency and filter order selected based on the signal sampling rate and the expected frequency range of meaningful content. A discrete low-pass FIR filter can be mathematically described using the convolution formula shown in Equation (3.4), where the input signal $x[n]$ is convolved with a finite set of filter coefficients $h[k]$:

$$y[n] = \sum_{k=0}^{M-1} h[k] \cdot x[n-k] \tag{3.4}$$

In this equation, $y[n]$ is the output (filtered signal), $x[n]$ is the original (raw) vibration signal, $h[k]$ are the filter coefficients defining the impulse response, and M

is the number of coefficients, which also determines the filter length or order. The filter coefficients $h[k]$ are designed to pass frequencies below a certain cutoff value f_c, while attenuating those above it. When applied to a vibration signal, this operation effectively smooths the waveform by averaging local values in a weighted manner, reducing short-duration high-frequency noise while preserving slower-varying, structurally relevant features.

The choice of filter design, such as window type, transition band sharpness, and gain, affects how aggressively the filter removes noise and how well it maintains critical fault signatures. While low-pass filtering is computationally efficient and effective for broad-spectrum noise suppression, it lacks selectivity and may inadvertently attenuate high-frequency components that are diagnostically significant, especially in the early detection of localized bearing faults. Consequently, low-pass filters are often used as a coarse denoising step in conjunction with more adaptive techniques [56].

3.3.2 Savitzky–Golay Filtering

The Savitzky–Golay (SG) filter is a smoothing technique that preserves the shape and features of a signal more effectively than traditional moving average filters. Unlike filters that apply uniform or low-pass weighting, the SG filter performs a local polynomial regression on a sliding window of the signal. This approach smooths out high-frequency noise while maintaining the local structure of the signal, making it particularly suitable for vibration signals where fault-related transients must be preserved.

Mathematically, for each point $x[n]$ in the signal, a polynomial of degree d is fitted to a window of $2k + 1$ points centered at $x[n]$. The central point is then replaced by the value of the fitted polynomial at that position. This process can be expressed compactly as a convolution with a set of precalculated filter coefficients h_i, derived from the least-squares fitting procedure, as defined in Equation (3.5):

$$y[n] = \sum_{i=-k}^{k} h_i \cdot x[n+i] \tag{3.5}$$

Here, $y[n]$ is the smoothed output signal, $x[n]$ is the original input, and h_i are the convolution weights based on the polynomial fit. The parameters k (half window size) and d (polynomial order) govern the degree of smoothing and the filter's ability to follow local variations.

A key advantage of the SG filter is its ability to reduce noise without significantly distorting signal peaks or short transients, which are essential for diagnosing early-stage bearing faults. Because it preserves higher-order signal characteristics such as slope and curvature, it is particularly effective for applications that require maintaining subtle trends and localized impulses in nonstationary time-series data [34].

3.3.3 Wavelet-Based Denoising

Wavelet-based denoising is a widely used technique in signal processing, especially effective for analyzing nonstationary signals that contain short-duration transients. Unlike traditional filters that apply uniform smoothing across time or frequency domains, wavelet-based methods enable multi-resolution analysis, allowing localized control over the denoising process in both time and frequency.

The denoising procedure typically follows three steps: (1) Discrete Wavelet Transform (DWT) decomposition, (2) thresholding of the detail coefficients, and (3) signal reconstruction. During the decomposition phase, the input signal $x[n]$ is passed through a series of high-pass and low-pass filters to obtain approximation coefficients $a_j[n]$ and detail coefficients $d_j[n]$ at different levels j, effectively splitting the signal into frequency bands with time-localized content.

In the thresholding step, noise is attenuated by suppressing small-magnitude detail coefficients. A commonly used method is soft thresholding, defined in Equation (3.6):

$$\mathcal{T}_\lambda(d_i) = \begin{cases} \text{sign}(d_i)(|d_i| - \lambda), & \text{if } |d_i| > \lambda \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

where d_i is a detail coefficient and λ is the threshold. This operation reduces noise without introducing discontinuities in the reconstructed signal. The threshold value can be selected using the universal threshold, which is defined in Equation (3.7):

$$\lambda = \sigma\sqrt{2\log n} \tag{3.7}$$

where σ is the estimated noise standard deviation and n is the number of samples [7].

The final stage involves reconstructing the signal using the inverse DWT, combining the approximation coefficients with the thresholded detail coefficients. This

process effectively reduces noise while preserving important transient features and local structures, making it highly suitable for machinery condition monitoring and fault detection tasks.

The choice of wavelet basis affects the quality of the denoising process. Daubechies wavelets are often favored due to their compact support and ability to capture localized signal variations. In particular, the Daubechies-5 (`db5`) wavelet is commonly used in mechanical diagnostics for its balance between smoothness and time localization.

Wavelet-based denoising has demonstrated strong effectiveness in vibration signal preprocessing for fault diagnosis tasks, particularly in scenarios involving impulse-like or burst-type features [12]. Figure 3.2 visually summarizes the full denoising process, highlighting each key step from a sample synthetic noisy input to final reconstruction and comparison.

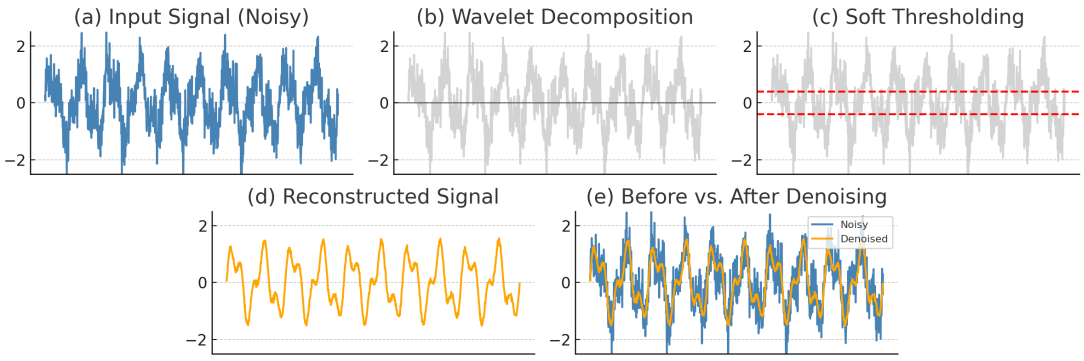

Figure 3.2: Step-by-step visualization of wavelet-based denoising: (a) The input synthetic signal containing high-frequency noise and transients; (b) Decomposition of the signal using the DWT into multi-scale approximation and detail coefficients; (c) Soft thresholding applied to suppress small-magnitude noise components; (d) Reconstructed signal using inverse DWT that retains structural features while reducing noise; (e) Comparison of the noisy and denoised signals, clearly showing improved clarity and signal preservation.

3.4 Time-Frequency Representation

As mentioned in the previous sections, vibration signals captured from rotating machinery such as bearings are inherently nonstationary and often exhibit localized transients associated with fault initiation and progression. These transients, though sparse in time, carry vital diagnostic information about a component's health state.

Traditional time-domain analysis techniques are often inadequate for capturing these frequency-varying characteristics, and purely frequency-domain methods such as the Fast Fourier Transform (FFT) ignore the temporal evolution of signal features. Consequently, time-frequency representations (TFRs) have become essential tools in machinery condition monitoring and RUL prediction tasks.

By transforming the raw vibration signal into a two-dimensional time-frequency domain, it becomes possible to visualize how the spectral content of the signal evolves over time. This facilitates the detection of early-stage degradation patterns and transient fault signatures, which may be difficult to distinguish in either domain alone. Effective time-frequency transformation not only improves the interpretability of degradation signals but also enables deep learning models to exploit both temporal and spectral dependencies during feature extraction.

Several time-frequency transformation techniques have been developed to handle nonstationary signals, each with unique advantages and limitations. The Short-Time Fourier Transform (STFT) is one of the earliest and most commonly used approaches [44]. It divides the signal into fixed-length overlapping windows and applies the Fourier Transform within each window. While this enables localized frequency analysis, it suffers from a fixed resolution trade-off: smaller windows yield better time resolution but poorer frequency resolution, and vice versa. This makes STFT suboptimal for analyzing signals that contain both slow-varying trends and fast transients.

The Continuous Wavelet Transform (CWT) improves upon this by using scalable wavelets that adapt their width to the frequency content. High-frequency components are analyzed with narrow wavelets for fine temporal resolution, while low-frequency trends are captured with wider wavelets. This multiscale flexibility allows CWT to better capture transient patterns in nonstationary signals [55]. However, CWT is computationally expensive and often produces redundant representations, which may limit its practicality for large-scale or real-time applications.

The Discrete Wavelet Transform (DWT) addresses some of these computational challenges by decomposing the signal into hierarchical approximation and detail coefficients using dyadic scales [43]. It produces a compact representation with lower redundancy and good localization. However, DWT performs an asymmetric decomposition: only the approximation coefficients are further decomposed at each level, while the detail branches are discarded. This can result in the loss of important high-frequency information, especially in applications where transients appear across the full spectral range.

3.4.1 Wavelet Packet Decomposition (WPD)

Wavelet Packet Decomposition (WPD) is an advanced extension of the DWT that provides a more detailed and symmetric time-frequency analysis. While DWT recursively decomposes only the low-frequency (approximation) components of a signal, WPD extends this process by decomposing both the approximation and high-frequency (detail) components at each level. This recursive decomposition results in a complete binary tree of subbands, enabling uniform frequency resolution across the entire spectrum. Such granularity is especially beneficial in vibration signal analysis, where transient fault signatures may be dispersed over high and low frequencies simultaneously.

The WPD process begins with a discrete signal $x[n]$ of length N, and decomposes it using a pair of quadrature mirror filters: a low-pass filter $h[m]$ and a high-pass filter $g[m]$, where m indexes the filter coefficients. At each level j, every signal segment or node $s_j^{(k)}[n]$, where n is the discrete time index and k is the node index, is filtered and downsampled to generate two subbands:

$$\begin{aligned} s_{j+1}^{(2k)}[n] &= \sum_m h[m] \cdot s_j^{(k)}[2n-m] \\ s_{j+1}^{(2k+1)}[n] &= \sum_m g[m] \cdot s_j^{(k)}[2n-m] \end{aligned} \tag{3.8}$$

Here, $s_j^{(k)}[n]$ is the signal at node k of level j, and $s_{j+1}^{(2k)}[n]$, $s_{j+1}^{(2k+1)}[n]$ are the subbands produced by low-pass and high-pass filtering respectively, followed by downsampling. The filter index m runs over the length of the wavelet filters, and the term $2n-m$ represents convolution with downsampling by 2. This decomposition continues until a predefined level is reached, producing 2^j subbands at level j.

The energy of each subband provides a useful feature for characterizing signal behavior in the time-frequency domain. It is computed as:

$$E^{(k)} = \sum_n \left(s_j^{(k)}[n]\right)^2 \tag{3.9}$$

where $E^{(k)}$ is the energy of the k-th subband at level j, and n indexes the discrete time samples. This energy quantifies the localized power in each frequency band and is often used to detect anomalies or assess component health.

The outputs of WPD can be reformatted into structured two-dimensional matrices, commonly referred to as WPD images. In this representation, each row corre-

sponds to a specific frequency subband, while each column represents a sequential time segment or window of the signal. For each subband, features such as energy, RMS, or kurtosis may be extracted, resulting in a dense time-frequency matrix. This layout preserves both the temporal progression and spectral content of the vibration signal in a format analogous to an image. The structured output facilitates the application of deep neural networks, such as CNNs and Vision Transformers [27], which can effectively extract hierarchical features from these image-like inputs by leveraging both spectral and temporal correlations. An illustrative example of such a time-frequency representation is shown in Figure 3.3.

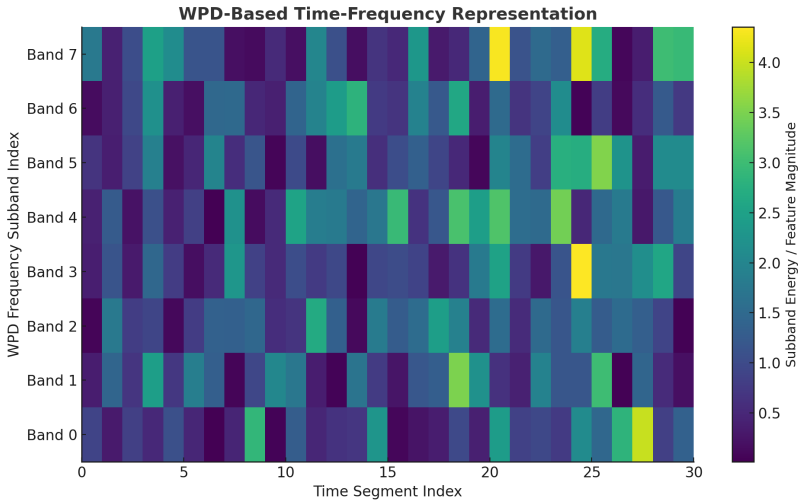

Figure 3.3: Sample WPD-based time-frequency image (synthetic data).

Compared to other time-frequency techniques, WPD offers several key advantages. The STFT applies fixed-size windows to the signal, resulting in a fundamental trade-off between time and frequency resolution that limits its effectiveness in capturing both slow degradation patterns and abrupt transient events. The CWT improves upon this by offering adaptive resolution across scales; however, it introduces substantial redundancy and is computationally expensive, making it less efficient for real-time or large-scale analysis. The DWT provides a more compact representation through hierarchical decomposition, yet it asymmetrically processes the signal, decomposing only low-frequency components and potentially discarding critical high-frequency information.

In contrast, WPD applies a balanced decomposition to both approximation and detail coefficients at each level, preserving complete spectral information across uni-

formly distributed frequency bands. This symmetrical structure enhances time-frequency localization and yields interpretable subband representations that can be reshaped into structured inputs for deep learning models. As a result, WPD is particularly effective for analyzing nonstationary signals in diagnostic and prognostic tasks, where multiscale fault signatures and transient patterns must be captured consistently. Its balance of computational efficiency, flexibility, and interpretability makes it a strong foundation for modern data-driven approaches in machinery condition monitoring.

3.5 Data Segmentation

In data-driven RUL prediction tasks, especially those involving vibration signals from rotating machinery, data segmentation plays a critical role in shaping the model's input and determining its ability to capture degradation dynamics. Since the raw signal length varies across different bearings and operating cycles, segmentation is essential to generate uniformly sized samples for supervised training.

Segmentation not only enables fixed-length model inputs but also determines the temporal context available to the model. The choice of segmentation strategy can impact the diversity, temporal resolution, and interpretability of the training data. Two widely used strategies in RUL modeling are sliding windows and expanding windows, both of which are explored and compared in this section.

3.5.1 Expanding Window Segmentation

The expanding window strategy incrementally increases the sequence length over time. Starting from a minimum window size, new samples are created by appending each new time step to the previous window. This produces a series of growing subsequences that retain full temporal context from the onset of degradation. It is especially suitable for architectures such as Transformers that can process variable-length sequences and benefit from long-term context modeling.

Figure 3.4 illustrates how the expanding window includes an increasing number of observations over time. This method yields fewer training samples but ensures that each one captures the full evolution of degradation up to that point.

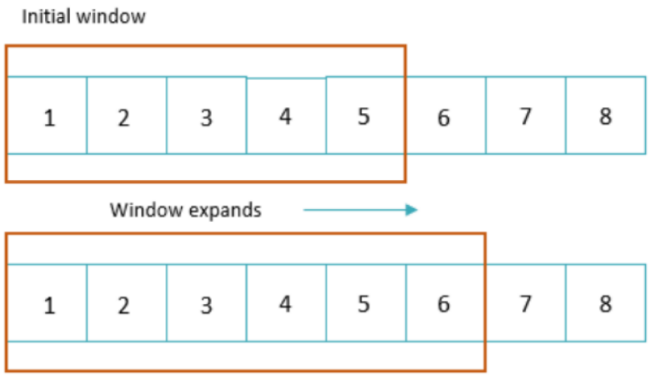

Figure 3.4: Expanding window segmentation, adapted from [39].

3.5.2 Sliding Window Segmentation

The sliding window strategy, on the other hand, extracts fixed-length segments from the time-series signal using a constant window size T and a defined stride. At each step, a window of length T is moved forward by the stride amount to create overlapping input sequences. This method is effective for learning localized trends and allows the generation of many more training samples, especially for short-to-medium-range modeling tasks.

Figure 3.5 shows an example of how a sliding window moves across the sequence. Although this method offers higher training sample density, it may discard longer-term dependencies unless the window size is sufficiently large.

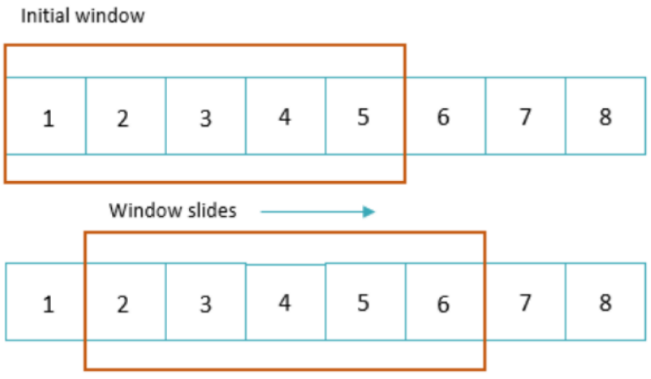

Figure 3.5: Sliding window segmentation, adapted from [39].

3.5.3 Sliding Window vs. Expanding Window

The choice between sliding and expanding window strategies hinges on the trade-off between training sample density and temporal context richness. Sliding windows generate a large number of overlapping samples, offering dense supervision and enhancing data diversity, particularly valuable when working with limited datasets. However, because each sample spans only a fixed time interval, this method may fail to capture long-range temporal dependencies or full degradation trajectories, especially when faults evolve gradually over time.

In contrast, expanding windows retain the complete historical context up to each time step, enabling the model to learn from the full evolution of the degradation process. This extended temporal view can improve the model's ability to detect subtle early-stage patterns and understand the sequential nature of fault development. However, this comes at the cost of fewer training samples and potentially increased computational overhead, particularly for models with high sequence-length sensitivity.

Ultimately, the segmentation strategy should align with the architectural properties of the model and the temporal characteristics of the degradation signals. For instance, CNNs may benefit from the uniformity and quantity of sliding segments, while Transformer-based models may be better suited to the contextual continuity offered by expanding windows.

3.6 CNNs for Feature Extraction

CNNs are a widely used class of deep learning models that excel at processing grid-like structured data, such as images or time-frequency representations. Originally developed for visual recognition tasks, CNNs have been successfully adapted to a variety of signal processing problems, including fault diagnosis and RUL prediction. Their architectural design is grounded in the use of convolutional filters that extract local spatial features, along with pooling operations that provide translational invariance and dimensionality reduction.

In a standard 2D convolutional layer, a learnable kernel $K \in \mathbb{R}^{k \times k}$ is applied to an input feature map $X \in \mathbb{R}^{M \times N}$, producing an output $Y \in \mathbb{R}^{M' \times N'}$ via local weighted summations:

$$Y(i,j) = \sum_{u=0}^{k-1} \sum_{v=0}^{k-1} X(i+u, j+v) \cdot K(u,v) \tag{3.10}$$

This operation captures spatial dependencies in the input, such as localized bursts or degradation signatures in time-frequency maps. Stacking multiple convolutional layers enables the model to learn hierarchical abstractions, with earlier layers detecting edges or impulses and deeper layers capturing more complex fault patterns or degradation trends.

CNNs are especially suitable for processing WPD-based time-frequency representations derived from vibration signals, where relevant information is often embedded in localized spectral bands or short-duration transients. Their strong inductive bias toward local feature extraction, combined with efficient parameter sharing and adaptability, makes them well-suited for learning degradation features across a wide range of machinery operating conditions.

In modern prognostic frameworks, CNNs are commonly used as backbone encoders or embedding modules, often serving as the first stage in hybrid pipelines that feed into temporal models or attention-based networks for final RUL estimation.

3.7 Transformer Architectures for Structured Feature Modeling

3.7.1 Self-Attention and Transformer Foundations

Originally introduced for natural language processing tasks, Transformers emerged as a versatile and powerful architecture for a wide range of domains, including computer vision, time-series forecasting, and machinery prognostics. Their strength lies in the self-attention mechanism, which enables the model to dynamically weigh the relevance of different input positions when forming contextualized representations. Unlike traditional recurrent or convolutional models that operate with fixed local receptive fields, Transformers capture long-range dependencies in a highly parallel and data-driven manner. This makes them particularly effective for signals or representations, such as time-frequency maps, where interactions across distant regions may contain critical degradation patterns or evolving fault features.

At the heart of the Transformer architecture is the self-attention mechanism, which enables each input element to attend to every other element in the sequence. Given an input matrix $X \in \mathbb{R}^{n \times d}$, where n is the number of input tokens and d is the feature dimension, the model first projects the inputs into three separate matrices: queries

Q, keys K, and values V, each of dimension $\mathbb{R}^{n \times d_k}$. The scaled dot-product attention is then computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{3.11}$$

Here, the dot product between queries and keys measures similarity between input tokens, and the softmax operation produces attention weights that sum to one across each row. These weights are used to aggregate the values V, producing a new representation for each token that incorporates information from all other positions in a content-aware manner. The division by $\sqrt{d_k}$ stabilizes gradients and prevents the dot products from growing too large. This mechanism allows the model to learn dynamic interactions and dependencies across different parts of the input, regardless of their position or distance.

To enhance the model's ability to capture diverse relational patterns, Transformers employ multi-head attention, where multiple self-attention operations are performed in parallel, each with different learned projections of Q, K, and V. The outputs of all heads are concatenated and linearly transformed to produce the final attention output. This allows the model to attend to information at different semantic or spatial levels simultaneously, enriching its representational capacity.

Since the attention mechanism is inherently permutation-invariant, positional encodings are added to the input embeddings to inject information about the sequence order or spatial layout. In the original Transformer design, sinusoidal positional encodings were used, computed as:

$$\text{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right), \quad \text{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right) \tag{3.12}$$

where pos is the position index and i indexes the embedding dimension. These encodings are added to the input tokens to provide the model with a sense of relative or absolute position, which is critical for modeling sequential or spatial dependencies.

3.7.2 Vision Transformers (ViT)

Vision Transformers (ViTs) extend the standard Transformer architecture to image data by treating an image as a sequence of non-overlapping patches. Instead of using convolutional filters to extract spatial features, ViTs flatten each image patch into a vector and feed the sequence of these vectors into a Transformer encoder. This

formulation enables the model to learn spatial dependencies globally across the entire image using self-attention mechanisms.

Formally, given an input image of size $H \times W \times C$, it is divided into N patches of size $P \times P$, where $N = HW/P^2$. Each patch is flattened into a vector $x_i \in \mathbb{R}^{P^2 \cdot C}$, and linearly projected into a latent dimension D through a learnable embedding layer. These patch embeddings are then combined with learnable position embeddings and passed into a standard Transformer encoder composed of multi-head self-attention and feedforward layers. A special classification token is optionally prepended to the sequence to aggregate global context.

ViTs are highly flexible and have demonstrated strong performance in tasks involving structured visual or spectral data, particularly when sufficient training data is available. Their ability to model global interactions makes them attractive for applications like bearing RUL prediction using time-frequency representations such as WPD images. However, their reliance on full self-attention across all patches can lead to high computational cost and a lack of localized inductive bias, motivating the development of more efficient variants like the Swin Transformer.

Figure 3.6 illustrates the ViT architecture reproduced from Dosovitskiy et al. [9].

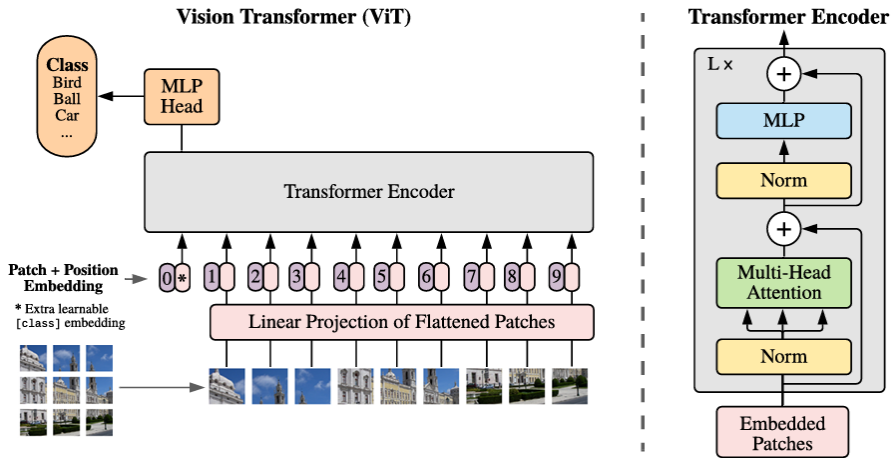

Figure 3.6: Illustration of the ViT architecture, reproduced from [9].

3.7.3 Swin Transformer

The Swin Transformer is a hierarchical vision architecture that improves upon ViT by introducing localized self-attention and multi-resolution feature learning. While ViTs

compute global self-attention across all image patches, this design leads to quadratic computational complexity and lacks the inductive bias of locality found in convolutional models. The Swin Transformer addresses these limitations by restricting attention computation to non-overlapping local windows, significantly reducing computational cost and enabling scalability to high-resolution inputs.

To preserve cross-window information flow, the Swin Transformer employs a technique known as shifted window attention. In alternating layers, the window partitioning is shifted such that tokens that were previously in different windows are now grouped together. This mechanism enables inter-window communication without incurring the cost of global attention. Additionally, Swin introduces a hierarchical representation by merging patches in deeper layers, allowing the model to capture both fine-grained and high-level features. These properties make the Swin Transformer particularly effective for structured prediction tasks, including those involving time-frequency representations like WPD images.

An overview of the Swin Transformer architecture, including window-based attention, shifted windows, and patch merging across hierarchical stages, is shown in Figure 3.7, reproduced from Liu et al. [31].

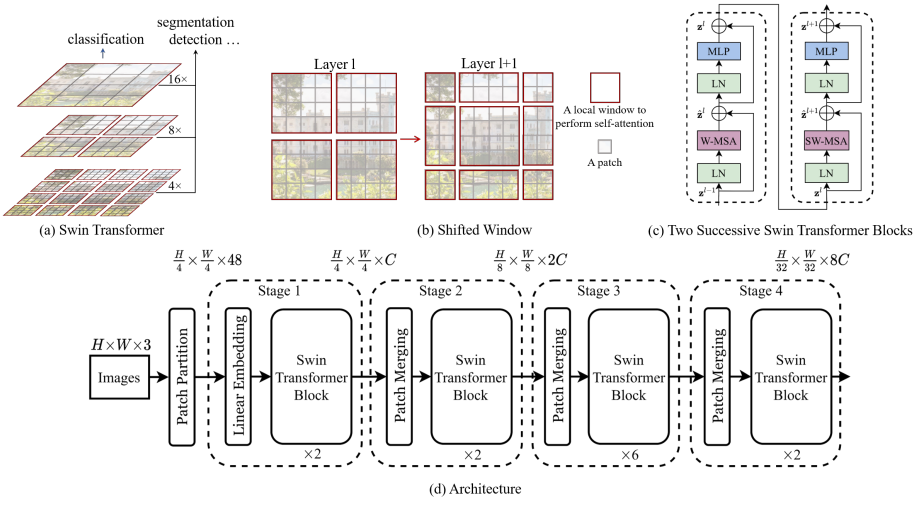

Figure 3.7: Architecture of the Swin Transformer, including hierarchical stage-wise processing, patch merging, and alternating window-based and shifted-window attention blocks, reproduced from [31].

Each stage of the Swin Transformer is composed of multiple Swin Transformer blocks, where each block consists of two main components: a self-attention mechanism and a feedforward multilayer perceptron (MLP), both preceded by Layer Normaliza-

tion and followed by residual connections. The self-attention layers alternate between regular window-based multi-head self-attention (W-MSA) and shifted window-based self-attention (SW-MSA), allowing both local and cross-window dependencies to be captured without incurring the high cost of global attention. This alternating pattern enables effective feature interaction across windows while maintaining linear complexity with respect to input size.

Between stages, spatial resolution is reduced through a patch merging operation, which merges every 2×2 neighboring patches. If each input patch has a feature dimension of C, the resulting concatenated vector will have a dimension of $4C$, which is then projected to a lower dimension through a learnable linear layer. This transformation is formally defined in Equation (3.13):

$$\hat{x}_i = W_m \cdot \text{Concat}(x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}) \tag{3.13}$$

where $x_{i,j} \in \mathbb{R}^C$ are the feature vectors of four spatially adjacent patches, and $W_m \in \mathbb{R}^{4C \times C'}$ is a trainable weight matrix projecting the combined features into a new embedding dimension C'.

In summary, the Swin Transformer builds on the strengths of both convolutional and transformer-based models by combining localized attention, multiscale feature learning, and computational efficiency. Its hierarchical, window-based architecture makes it particularly effective for structured prediction tasks involving time-frequency representations, where both fine-grained transient features and broader degradation trends must be captured.

3.8 Evaluation Metrics

To evaluate the accuracy of regression models in RUL prediction tasks, appropriate performance metrics must capture both average prediction error and the criticality of early or late forecasts. In this thesis, two metrics are used to assess model performance: Mean Absolute Error (MAE), and a domain-specific scoring function adapted from the PRONOSTIA dataset manual. While MAE quantifies average prediction deviations from ground truth, the scoring function assigns asymmetric penalties, with greater cost assigned to late predictions due to their potential safety implications. Together, these metrics provide a balanced assessment of both statistical accuracy and practical risk.

3.8.1 Mean Absolute Error (MAE)

MAE is a widely used regression metric that measures the average absolute difference between predicted and true values. In the context of RUL prediction, MAE reflects the average deviation of the predicted RUL values from their actual counterparts.

Formally, for a test set containing N samples, the MAE is computed as:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{3.14}$$

Where $\hat{y}_i$ is the predicted RUL for the i^{th} sample, and y_i is the corresponding ground truth RUL. The absolute error for each sample is computed and averaged across all predictions to provide an overall measure of prediction accuracy.

3.8.2 Scoring Metric from the PRONOSTIA Dataset

The IEEE PHM 2012 Prognostic Challenge introduced a custom scoring function for evaluating RUL predictions, designed to reflect the real-world consequences of early versus late prognostic decisions [38]. In industrial settings, early predictions may lead to premature maintenance but are generally safer, while late predictions risk catastrophic failures. The PRONOSTIA scoring function incorporates this asymmetry by reducing the score assigned to late predictions more severely than early ones.

Given the predicted RUL $\hat{RUL}_i$ and true RUL RUL_i for the i-th time sample, the percentage error is defined as:

$$\%E_{r_i} = 100 \times \frac{RUL_i - \hat{RUL}_i}{RUL_i} \tag{3.15}$$

The individual score A_i for each sample is then computed using an exponential penalty based on this percent error:

$$A_i = \begin{cases} \exp\left(-\ln(0.5)\cdot\frac{\%E_{r_i}}{5}\right), & \text{if } \%E_{r_i} \leq 0 \quad \text{(late prediction)} \\ \exp\left(+\ln(0.5)\cdot\frac{\%E_{r_i}}{20}\right), & \text{if } \%E_{r_i} > 0 \quad \text{(early prediction)} \end{cases} \tag{3.16}$$

This design ensures that a perfect prediction ($\%E_{r_i} = 0$) yields the maximum score of 1, while the score decays exponentially for inaccurate predictions, faster for late than early estimates. The final score over the entire set of N samples is computed as the mean of individual scores:

$$\text{Score} = \frac{1}{N}\sum_{i=1}^{N} A_i \tag{3.17}$$

Figure 3.8 visualizes this asymmetric penalty function, illustrating the steeper drop in score associated with late predictions compared to early ones.

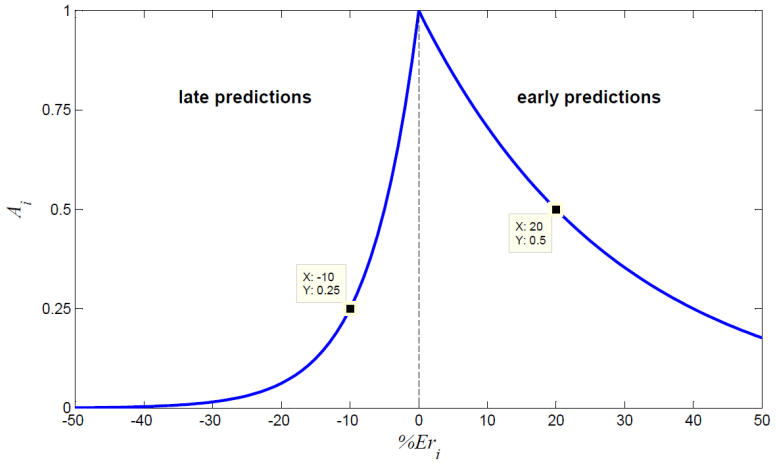

Figure 3.8: Scoring function Plot based on the prediction error, reproduced from [38].

This scoring function effectively aligns model evaluation with the safety-critical nature of predictive maintenance, rewarding cautious forecasting while discouraging overly optimistic RUL estimates.

In summary, this chapter has established the theoretical foundation and key concepts that underpin the remainder of this thesis. Beginning with formal definitions of RUL, FPT, and normalized degradation labeling, we explored the major challenges inherent in prognostics, including the effects of nonstationarity, noise, and the critical balance between early and late predictions. We then introduced signal denoising methods and time-frequency representations, with a particular emphasis on Wavelet Packet Decomposition (WPD) as a flexible and information-rich transformation technique. The chapter also reviewed the fundamental deep learning architectures employed throughout this work, including CNNs, ViTs, and the Swin Transformer, each offering unique capabilities for capturing spatiotemporal patterns in transformed vibration signals. Finally, relevant evaluation metrics were presented to assess model performance from both statistical and application-specific perspectives. Together, these preliminaries form the backbone of the methodology described in the following chapters.

Chapter 4

Datasets and Preprocessing Pipeline

This chapter details the datasets, preprocessing techniques, and signal transformations used in the development and evaluation of the proposed MCSFormer framework. Two benchmark run-to-failure datasets, PRONOSTIA and XJTU-SY, are used to assess the generalizability of the model under varying operating conditions. The chapter begins by describing the datasets and their acquisition setups, followed by the preprocessing steps including denoising, segmentation strategies, time-frequency transformation, and labeling methodology that shape the input pipeline for the following framework.

4.1 Dataset Description

Accurate prediction of bearing degradation requires access to rich datasets, especifically run-to-failure datasets, that capture the dynamics of wear progression under realistic operating conditions. This thesis utilizes two publicly available benchmark datasets, PRONOSTIA and XJTU-SY, both of which provide high-frequency vibration signals recorded from rolling bearings operating until failure. These datasets are widely used in prognostics and health management (PHM) research due to their controlled yet diverse experimental setups, making them suitable for training and evaluating deep learning models for RUL prediction. By incorporating both datasets into this study, the goal was to assess the generalizability of the proposed model across different degradation scenarios, sensor configurations, and operating loads.

4.1.1 PRONOSTIA Dataset

The PRONOSTIA dataset was developed for the IEEE PHM 2012 Prognostics Challenge by the FEMTO-ST Institute in France [38]. It provides high-resolution run-to-failure vibration data from bearings subjected to accelerated degradation under controlled laboratory conditions. The dataset is widely used in PHM community due to its clean design, multiple operating conditions, and precise vibration sampling.

The test bench consists of a rotating part, a loading mechanism, and a measurement system. A radial force is applied via a pneumatic actuator to induce bearing degradation, and vibration signals are captured using two perpendicular accelerometers mounted on the bearing housing. Vibration data are recorded at a sampling rate of 25.6 kHz, saved in CSV files, with each CSV file containing 2560 data points (representing 0.1 seconds of signal) sampled every 10 seconds.

An overview of the PRONOSTIA platform is shown in Figure 4.1, illustrating the main components used for rotation, loading, and signal acquisition.

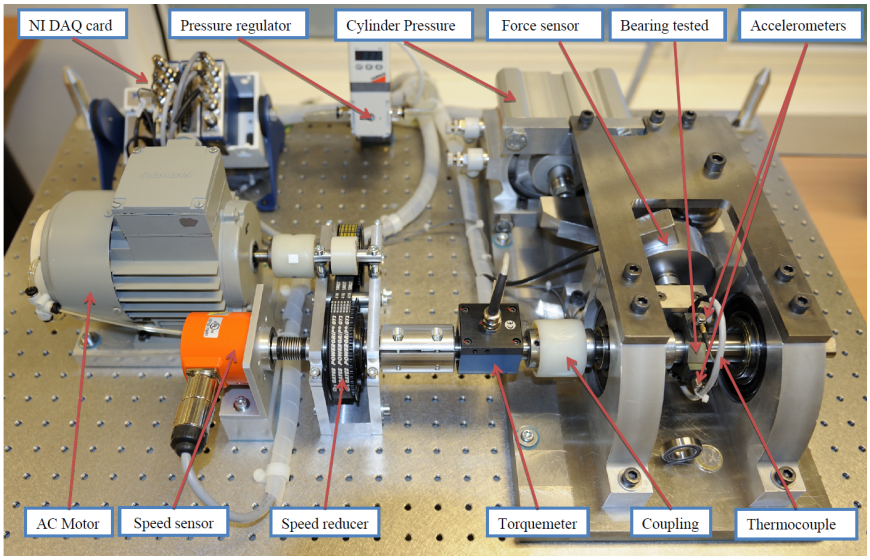

Figure 4.1: Overview of the PRONOSTIA test platform, reproduced from [38].

The experiment includes 17 bearings in total, tested under three distinct constant operating conditions. The operating settings are summarized in Table 4.1.

Table 4.1: Operating conditions in the PRONOSTIA dataset

Condition	**Rotational Speed (rpm)**	**Radial Load (N)**
Condition 1	1800	4000
Condition 2	1650	4200
Condition 3	1500	5000

Each experiment continues until the onset of a terminal fault, defined as the point at which the vibration amplitude exceeds a 20g threshold. The resulting dataset includes detailed degradation trajectories, enabling both within-condition and cross-condition RUL prediction. Only the horizontal and vertical vibration signals are used in this work; temperature measurements are excluded from model inputs.

4.1.2 XJTU-SY Dataset

The XJTU-SY dataset was released by Xi'an Jiaotong University (XJTU) and Sumyoung Technology Co., Ltd. (SY), China, and has become a widely used benchmark for RUL of rolling bearings [50]. It includes complete run-to-failure vibration data collected from 15 bearings operating under three different controlled conditions. The experimental setup and acquisition protocol were designed to simulate accelerated degradation while maintaining realistic signal characteristics across various fault modes.

The testbed comprises an AC induction motor, motor speed controller, support shaft, two heavy-duty support bearings, and a hydraulic loading system. Bearings of type LDK UER204 are used, and vibration signals are captured using two PCB 352C33 accelerometers mounted perpendicularly on the bearing housing. Figure 4.2 shows the schematic layout of the XJTU-SY bearing degradation platform.

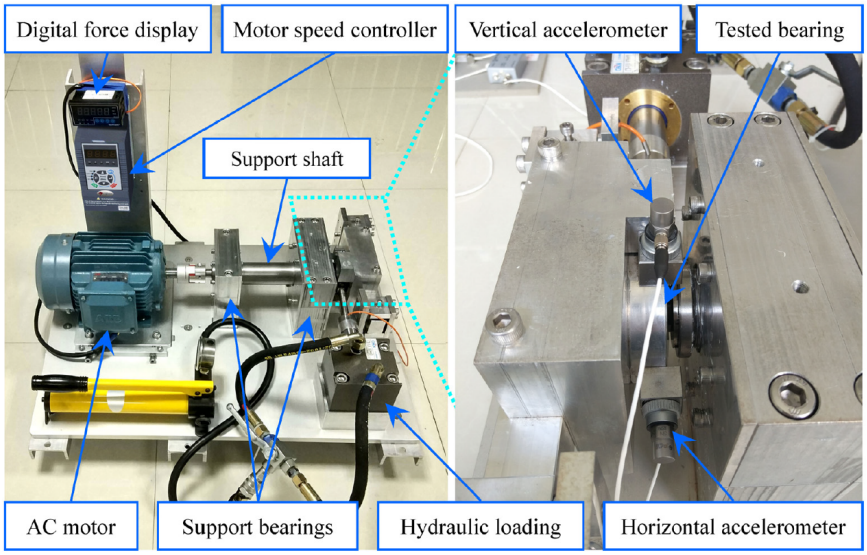

Figure 4.2: Overview of the XJTU-SY bearing degradation testbed, reproduced from [50].

Vibration data are recorded at a sampling rate of 25.6 kHz. For each bearing, a total of 32768 points (approximately 1.28 seconds) are captured every minute throughout the experiment. This results in a high-resolution signal trajectory of the entire bearing lifespan. The degradation tests are terminated when the vibration amplitude exceeds ten times the highest amplitude observed during the bearing's stable phase.

The 15 bearings are divided equally among three operating conditions, each with distinct speed and radial load combinations. Table 4.2 summarizes the settings.

Table 4.2: Operating conditions in the XJTU-SY dataset

Condition	**Rotational Speed (rpm)**	**Radial Load (kN)**
Condition 1	2100	12
Condition 2	2250	11
Condition 3	2400	10

The dataset includes diverse failure modes, including outer race wear, cage fracture, and inner race faults. Each CSV file contains two columns corresponding to the horizontal and vertical vibration signals. Like PRONOSTIA, the XJTU-SY dataset is well-suited for both within-condition and cross-condition RUL prediction and allows comprehensive analysis of model generalizability under different stress scenarios.

4.2 Signal Denoising

The raw vibration signals collected from the PRONOSTIA and XJTU-SY datasets exhibit significant levels of high-frequency noise, amplitude bursts, and structural vibrations. Such artifacts can distort time-frequency representations and negatively affect RUL prediction models. To address this, a multi-stage denoising pipeline was applied, combining low-pass filtering, wavelet-based denoising, and Savitzky-Golay smoothing. Each stage was selected based on its ability to address different characteristics of the noise profile while preserving underlying degradation dynamics.

The first stage involved a low-pass Butterworth filter with a cutoff frequency of 10 kHz, applied to remove high-frequency components outside the mechanical bandwidth of interest. A fourth-order zero-phase design was used to prevent phase distortion, ensuring that temporal alignment of features remained intact.

In the second stage, wavelet-based denoising was performed using the Daubechies 5 (db5) wavelet. A single-level DWT was used to decompose each signal, and soft thresholding was applied to the detail coefficients using an adaptive universal threshold:

$$\lambda = \sigma\sqrt{2\log n} \tag{4.1}$$

where σ is the estimated noise level from the median absolute deviation of the first-level detail coefficients, and n is the signal length. The inverse DWT was then used to reconstruct the denoised signal. This step proved essential for suppressing burst noise and preserving non-stationary transients.

Finally, a Savitzky-Golay filter with a window size of 5 and a polynomial order of 2 was applied to further smooth local fluctuations while retaining curvature information in the mid-frequency range. This configuration provided sufficient smoothing without distorting fault-related oscillations.

These denoising parameters were applied uniformly across both the PRONOSTIA and XJTU-SY datasets to maintain consistency throughout the preprocessing pipeline. To assess the impact of this preprocessing stage, a separate experiment is conducted in Chapter 6 comparing model performance on denoised versus raw vibration signals.

Figure 4.3 illustrates the result of the denoising pipeline applied to the horizontal signal of Bearing2_1 from the PRONOSTIA dataset. The combined method effectively reduces noise while preserving key degradation trends essential for subsequent analysis.

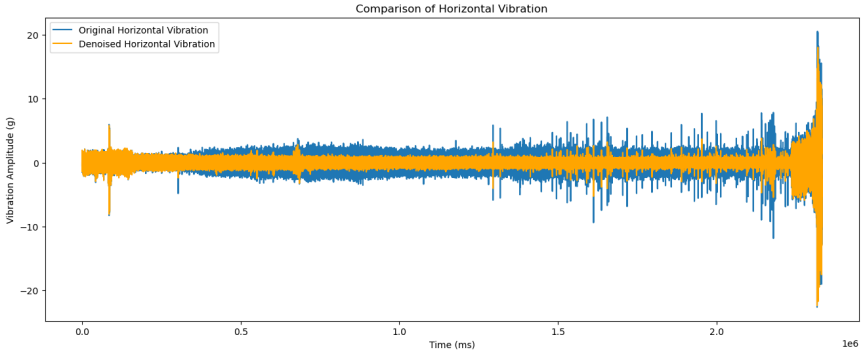

Figure 4.3: Comparison of original and denoised horizontal vibration signal for Bearing2_1 (PRONOSTIA dataset).

4.3 Segmentation

Following denoising, the vibration data is segmented to generate meaningful temporal input samples for the model. Since each raw file represents only a short duration of time, 2560 samples per file in PRONOSTIA and 32,768 samples per file in XJTU-SY, individual files alone are insufficient to capture degradation patterns. Instead, multiple consecutive files are grouped together into larger segments using a windowing strategy.

Let w denote the window size, defined as the number of consecutive denoised files combined into one segment. For example, in the PRONOSTIA dataset, a window size of $w = 10$ leads to a segment containing 10×2560 samples per vibration channel. In the XJTU-SY dataset, where each file contains 32,768 samples, the same window size produces segments of $10 \times 32{,}768$ samples. These full-length 1D signals are then passed to the time-frequency transformation step.

Two segmentation strategies are evaluated: sliding windows and expanding windows. In the sliding approach, a fixed-length window moves forward with a fixed stride s, producing a large number of overlapping segments. This allows dense sampling of the degradation timeline. In contrast, expanding windows start from the beginning of a bearing's life and grow by adding one file at a time, increasing temporal coverage. Both strategies are visually introduced in Figures 3.4 and 3.5 in the Preliminaries chapter.

For all general training and evaluation experiments, the sliding window method is adopted as the default segmentation approach. This decision is based on its lower

computational cost and its ability to include recent historical information while keeping input length fixed. A window size of $w = 10$ and a stride of $s = 5$ are used consistently across both datasets. A comparative analysis of model performance using sliding versus expanding window segmentation is presented in Chapter 6.

4.4 Time-Frequency Representation

Once the denoised vibration signals have been segmented, each resulting time-domain segment is transformed into a time-frequency representation using Wavelet Packet Decomposition (WPD). This transformation enhances the model's ability to capture non-stationary signal characteristics and local frequency shifts, which are key indicators of progressive bearing degradation.

WPD provides a full binary tree decomposition of both approximation and detail coefficients at each level. This enables richer multi-resolution frequency analysis, making WPD particularly well-suited for degradation modeling. The db5 wavelet is selected as the basis function for all experiments, in line with its use in the denoising stage, due to its compact support and capability to capture transient behavior.

For the PRONOSTIA dataset, each segment consists of 25,600 samples (from 10 CSV files, each with 2560 samples), and is decomposed up to level 3 using WPD. This yields $2^3 = 8$ frequency sub-bands, each containing a portion of the energy distribution over time. For the XJTU-SY dataset, where each CSV file contains 32,768 samples, the same window size results in segments of 327,680 samples. To maintain comparable time-frequency granularity across datasets, segments from XJTU-SY are decomposed to level 4, producing $2^4 = 16$ sub-bands. In both cases, the resulting coefficients from all sub-bands are concatenated and reshaped into a fixed-size 64×64 image.

This two-dimensional representation captures both temporal and spectral structures of the input signal, enabling the convolutional and attention-based components of the proposed model to extract spatially structured features. One WPD image is generated per segment for each vibration channel (horizontal and vertical), resulting in two images per time step fed into the dual-branch network.

Figure 4.4 illustrates the output of this transformation for two parts of horizontal vibration signals from Bearing2_3 in the PRONOSTIA dataset. (a) and (c) show the denoised signal at the beginning and end of the bearing's life, while (b) and (d) show the corresponding WPD images.

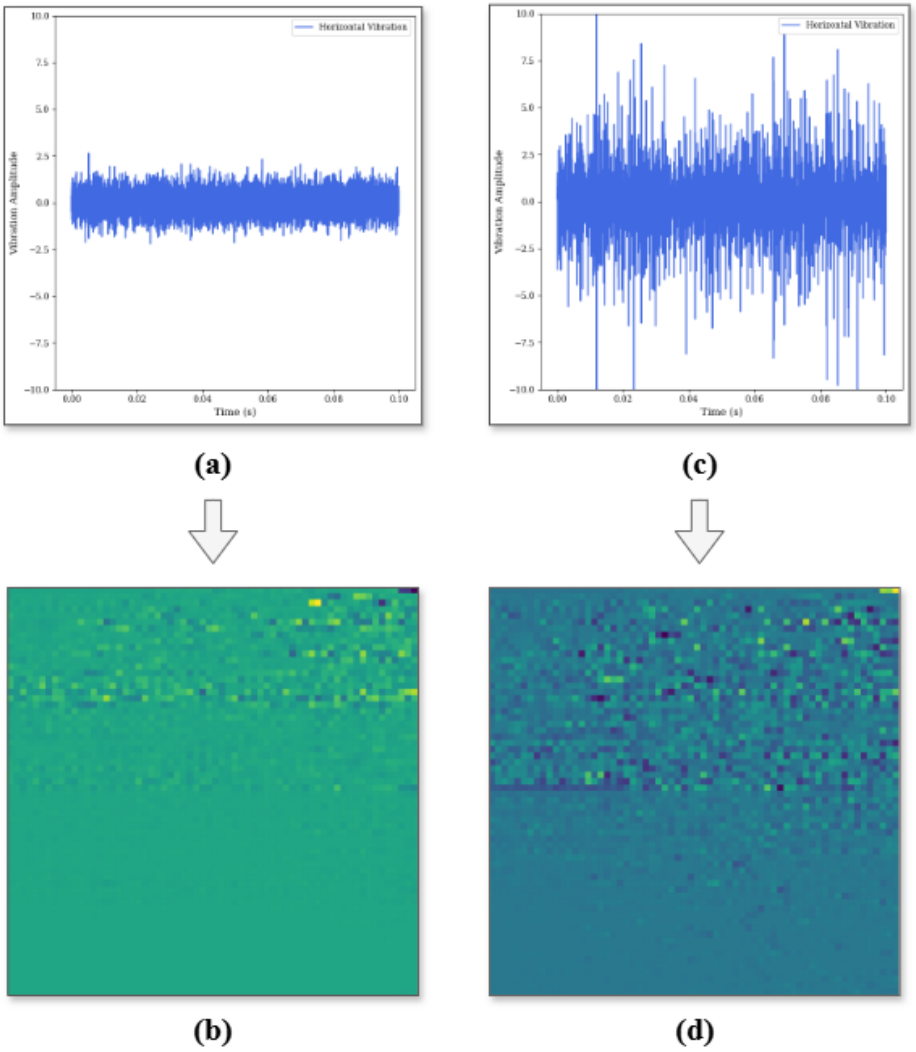

Figure 4.4: Visualization of denoised horizontal vibration signal and WPD image for Bearing2_3 from PRONOSTIA dataset: (a) denoised signal at the beginning of the degradation process; (b) corresponding WPD image; (c) denoised signal at the end of the degradation process; (d) corresponding WPD image.

4.5 Normalization and Labeling

To generate effective training targets for RUL prediction, each segment must be assigned a label that reflects the remaining useful life of the bearing at that point in time. However, raw vibration data contains long periods of healthy operation that do not provide informative degradation signals. As introduced in Chapter 3, a First Pre-

diction Time (FPT) is determined for each bearing to mark the onset of degradation and exclude early-life samples from training.

The FPT is identified using a kurtosis-based statistical thresholding strategy, where a segment is considered to belong to the degradation phase if its kurtosis deviates significantly from the healthy baseline. Specifically, the FPT is defined as the first point where the kurtosis exits a 3-standard-deviation confidence interval (see Equation (3.2)) for three consecutive segments. This method minimizes the effect of noise while providing a robust signal for degradation onset. All samples prior to the FPT are discarded.

From the FPT onward, RUL labels are assigned using a normalized linear decay. For a given bearing, let N_d be the number of segments after FPT, and let $i \in \{0, 1, \ldots, N_d - 1\}$ denote the index of each post-FPT segment. Then, the normalized RUL label y_i is given by Equation (3.3):

$$y_i = 1 - \frac{i}{N_d - 1}.$$

This results in a target value of 1 at FPT and a value of 0 at failure, ensuring consistent label scaling across bearings of different lifespans and operating conditions. These continuous values are used as regression targets in all model training and evaluation tasks.

The same procedure is applied across both the PRONOSTIA and XJTU-SY datasets. While the actual FPT values vary depending on each bearing's signal behavior, the kurtosis-based detection method and normalized labeling remain fixed across datasets. This allows fair comparison and stable learning behavior, while supporting generalization to unseen degradation trajectories.

Chapter 5

MCSFormer Framework

This chapter introduces the proposed multi-channel swin transformer framework for RUL prediction of rolling bearings, referred to as MCSFormer. The framework is designed to jointly model spatial, frequency, and temporal information from multi-channel vibration signals, incorporating preprocessing techniques developed in Chapter 4. MCSFormer leverages wavelet-based time-frequency analysis, dual-branch convolutional feature extraction, and a Swin Transformer backbone to learn both local degradation patterns and global progression trends. The sections that follow describe the architecture and its individual components in detail.

5.1 Architecture Overview

The input to the framework consists of horizontal and vertical vibration signals collected from the testbed. Each signal is first denoised and segmented as described in Chapter 4, and transformed into a two-dimensional time-frequency representation using Wavelet Packet Decomposition (WPD). This results in a pair of 64×64 images, one for each vibration channel, per input segment.

These WPD images are processed through two parallel convolutional branches, one for the horizontal channel and one for the vertical channel. Each branch consists of a series of convolutional and pooling layers designed to extract localized spatial and spectral features while reducing dimensionality. The output feature maps from both branches are then combined using a feature fusion module. This module concatenates the extracted features and applies patch partitioning and linear embedding to prepare the input for the transformer.

The fused feature representation is passed into a Swin Transformer backbone, which consists of four hierarchical stages. Each stage includes window-based multi-head self-attention blocks and patch merging operations. This structure enables scalable modeling of both short- and long-range dependencies, while efficiently capturing the hierarchical evolution of degradation patterns across scales.

Following the Swin Transformer stages, the resulting features are pooled and passed through a regression head composed of fully connected layers. The final output is a scalar value representing the normalized RUL of the corresponding segment.

A high-level overview of the complete pipeline is shown in Figure 5.1. The framework integrates time-frequency transformation, feature extraction, global context modeling, and regression in an end-to-end learnable architecture.

5.2 CNN-Based Feature Extraction

The initial feature extraction stage of the MCSFormer framework consists of two parallel convolutional branches that process the horizontal and vertical vibration signals independently. Each input to this stage is a 64×64 image, representing a time-frequency transformed segment of vibration data generated using Wavelet Packet Decomposition (WPD).

Each CNN branch follows an identical architecture, composed of a single 2D convolution layer followed by an activation and pooling operation. Specifically, the Conv2D layer uses 32 filters with a kernel size of 3×3, followed by a ReLU activation function. A 2×2 max pooling operation is applied to reduce the spatial resolution by half, resulting in a feature map of size $32 \times 32 \times 32$ for each channel.

- **Conv2D** (filters = 32, kernel size = 3×3, stride = 1, padding = 'same')
- **ReLU activation**
- **MaxPooling2D** (pool size = 2×2)

These convolutional branches act as local feature encoders, distilling spatial and spectral characteristics unique to each vibration channel. This design allows each branch to adapt to the structure and noise profile of its respective input without interference from the other. The outputs of these branches are passed forward to a dedicated fusion module described in the next section.

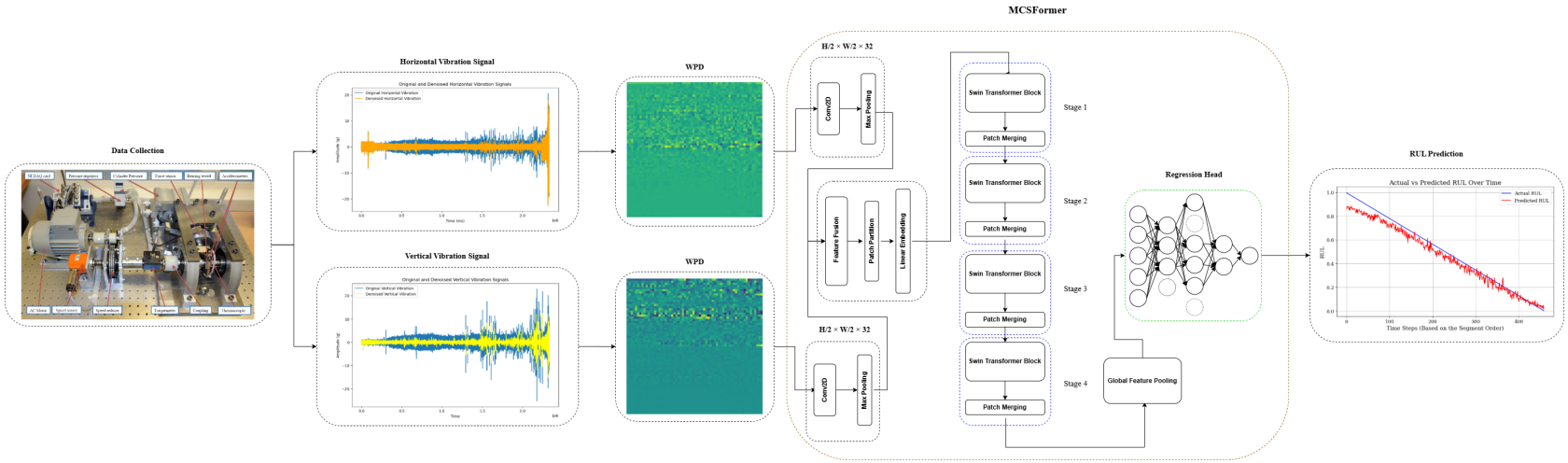

Figure 5.1: Overview of the proposed MCSFormer framework. Vibration signals are denoised and converted to time-frequency representations via WPD. A dual-branch CNN extracts local features, followed by a shared Swin Transformer backbone that models global degradation behavior. A regression head predicts normalized RUL, shown against actual degradation trajectory on the right.

5.3 Multi-Channel Fusion Strategy

Following local feature extraction via the dual CNN branches, the resulting feature maps, each of size $32 \times 32 \times 32$, are concatenated along the channel dimension to form a unified representation of shape $32 \times 32 \times 64$. This mid-level fusion strategy allows the model to combine degradation patterns from both horizontal and vertical vibration channels while preserving the spatial structure of the extracted features.

This fused tensor is then passed through an embedding module designed to prepare the data for transformer-based processing. The module performs the following operations:

1. **Patch Partition:** The $32 \times 32 \times 64$ tensor is divided into non-overlapping patches, typically of size 4×4, resulting in a sequence of spatially organized local tokens.

2. **Linear Embedding:** Each patch is flattened and projected to a vector with fixed dimensions using a fully connected layer. This produces a token sequence that serves as the input to the Swin Transformer.

The choice of mid-level fusion, after CNN feature extraction but before global context modeling, offers multiple advantages:

- It preserves channel-specific learning in early layers.
- It reduces interference between signals of varying frequency content.
- It enables the transformer to model joint degradation evolution using a fused, information-rich representation.

This design ensures that both sensor channels contribute complementary information to the subsequent Swin Transformer stages, facilitating more robust and generalizable RUL predictions.

5.4 Swin Transformer Backbone

The fused feature map produced by the dual-branch CNN and embedding module is passed to a Swin Transformer backbone, which serves as the primary context modeling component in the MCSFormer architecture. The Swin Transformer captures

hierarchical representations through localized self-attention within non-overlapping windows, while progressively expanding the receptive field across multiple stages. This structure enables the model to learn both short-term and long-range degradation patterns across the time-frequency domain.

The input to the Swin Transformer consists of embedded patch tokens derived from the fused $32 \times 32 \times 64$ feature map. These patches are flattened and projected into a fixed-dimensional embedding space using a linear layer. The token sequence is then passed through four hierarchical stages, each composed of Swin Transformer blocks followed by patch merging operations.

Each Swin Transformer block consists of the following:

- **Layer Normalization**
- **Window-based Multi-Head Self-Attention (W-MSA)** using a fixed window size
- **Shifted Window-based Attention (SW-MSA)** to enable cross-window information flow
- **MLP** with GELU activation and residual connections

Patch merging is applied between stages to reduce token resolution while increasing the channel dimension. Specifically, every four neighboring patches are merged, halving the spatial resolution and doubling the channel dimension. This process creates a pyramid structure similar to CNNs, where each stage operates at a different spatial scale.

Let C denote the initial token embedding dimension ($C = 96$ in this case). Then across the four stages, the token dimensions evolve as follows:

- **Stage 1:** token size $= \frac{H}{4} \times \frac{W}{4}$, channels $= C$
- **Stage 2:** token size $= \frac{H}{8} \times \frac{W}{8}$, channels $= 2C$
- **Stage 3:** token size $= \frac{H}{16} \times \frac{W}{16}$, channels $= 4C$
- **Stage 4:** token size $= \frac{H}{32} \times \frac{W}{32}$, channels $= 8C$

The hierarchical structure allows the model to efficiently capture both local frequency anomalies and long-term degradation trends. The final stage outputs are passed to a global pooling layer before being forwarded to the regression head.

The Swin Transformer backbone is designed to balance modeling capacity with computational efficiency. Compared to standard Vision Transformers, the use of window-based attention significantly reduces complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$, where n is the number of tokens. The combination of local and shifted windows ensures that both fine-grained and cross-regional features are captured effectively across the signal's degradation timeline.

At the final stage of the Swin Transformer, the output tokens are globally aggregated using average pooling to form a compact feature vector. This vector captures the learned representation of the segment's degradation characteristics across both time and frequency dimensions. The pooled vector is then passed through a regression head composed of five fully connected layers with ReLU and dropout ($p = 0.3$), followed by a final layer for RUL prediction. This final mapping enables the model to produce continuous RUL estimates suitable for downstream evaluation and maintenance decision-making.

5.5 Loss Function Design

In RUL prediction tasks, accuracy alone is not sufficient, timing is equally critical. Early predictions, where the estimated RUL is shorter than the true RUL, may lead to unnecessary preventive maintenance, increasing operational costs and downtime. In contrast, late predictions, where the model overestimates RUL, pose a much higher risk by delaying essential maintenance. This can result in unexpected system failure or catastrophic damage, especially in safety-critical applications. Despite advances in deep learning models, most standard loss functions do not differentiate between early and late errors, treating them symmetrically and thus overlooking their operational consequences.

To explicitly address this challenge, a custom loss function is designed for the MCSFormer framework. This function adds a penalty term to the standard Mean Squared Error (MSE), focusing specifically on late predictions. It ensures that overestimates, where the predicted RUL exceeds the actual RUL, are penalized more heavily than underestimates. This encourages the model to prioritize safety in its predictions.

The custom loss is defined as:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 + \lambda \cdot \max(0, \hat{y}_i - y_i) \tag{5.1}$$

where:

- N is the number of samples in the training batch,
- $\hat{y}_i$ is the predicted normalized RUL for the i^{th} segment,
- y_i is the corresponding ground truth RUL label, and
- λ is a hyperparameter that controls the penalty for late predictions.

The first term represents the MSE loss, which ensures general prediction accuracy. The second term is activated only when $\hat{y}_i > y_i$, i.e., when the model has overestimated the RUL. The scalar weight λ is introduced to amplify the importance of these late predictions during optimization. A typical value of $\lambda = 5$ is used, focusing on sensitivity to safety-critical scenarios.

This loss function reflects the practical trade-off between early and late predictions. By penalizing overconfidence more aggressively, the model is encouraged to deliver reliable early warnings without excessively triggering premature maintenance.

Comparison with MSE

To assess the impact of the custom loss formulation, a comparative experiment is conducted using the standard Mean Squared Error loss:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{5.2}$$

While MSE provides a balanced view of overall accuracy, it fails to account for the operational cost asymmetry between early and late predictions. In Chapter 6, the effect of using the custom loss function versus plain MSE is explored experimentally. Unless otherwise stated, all training throughout this thesis is conducted using the custom loss defined in Equation (5.1).

Chapter 6

Experimental Analysis and Results

This chapter presents a comprehensive evaluation of the proposed framework for predicting the RUL of rolling bearings. The experiments are designed to assess the model's performance under varying conditions, quantify its generalization capability, and analyze the contributions of individual components through ablation studies. We begin by outlining the experimental setup, including training configuration. Evaluation metrics are then defined, followed by detailed comparisons with competing models in both intra-condition and cross-condition scenarios. Additional experiments investigate the effects of denoising, segmentation strategy, and loss function design.

6.1 Experimental Setup

All experiments were implemented in Python using PyTorch 2.1.0 and executed in Jupyter Notebook on a system equipped with an NVIDIA RTX 4060 GPU and an Intel Core i7-13700H processor, running Windows 11. GPU acceleration was enabled via CUDA 12.6. The training pipeline made use of standard scientific computing libraries, including NumPy, SciPy, Pandas, Matplotlib, Scikit-learn, and PyWavelets. The tqdm library was employed for progress monitoring. Random seeds were fixed to ensure reproducibility.

Two benchmark datasets were used: PRONOSTIA and XJTU-SY. Both datasets contain dual-channel run-to-failure vibration data recorded under varying operating conditions. Horizontal and vertical acceleration signals were used as inputs, and only data after the First Prediction Time (FPT) were considered for training.

As described in Chapter 5, each signal was denoised using a combination of low-

pass filtering, wavelet-based soft thresholding (db5), and Savitzky–Golay smoothing. The cleaned signals were segmented using a sliding window approach with a window size of 10 and a stride of 5 (50% overlap). Each segment was then transformed into a 2D image using Wavelet Packet Decomposition (WPD): level 3 for PRONOSTIA and level 4 for XJTU-SY, resulting in 64×64 time-frequency images per vibration direction. These multi-channel images were passed to the CNN-Swin Transformer architecture.

The model was trained for 100 epochs using the Adam optimizer with a learning rate of 1×10^{-4} and a batch size of 16. A custom loss function was used to penalize late predictions more heavily than early ones. Table 6.1 summarizes the full experimental configuration.

Table 6.1: Summary of experimental configuration and training parameters

Category	**Parameter / Setting**
Hardware	GPU: NVIDIA RTX 4060
	CPU: Intel Core i7-13700H
	OS: Windows 11
Software	Programming Language: Python
	DL Framework: PyTorch 2.1.0
	CUDA Version: 12.6
	Development Environment: Jupyter Notebook
Training	Optimizer: Adam
	Learning Rate: 1×10^{-4}
	Batch Size: 16
	Epochs: 100
	Loss Function: MSE + Late Penalty ($\lambda = 5$)
Preprocessing	Window Size: 10 segments
	Stride: 5 segments (50% overlap)
	Denoising: Low-pass, Wavelet (db5), Savitzky–Golay
	WPD Level: 3 (PRONOSTIA), 4 (XJTU-SY)
	WPD Image Size: 64×64 per channel

6.2 Evaluation Metrics

To assess the performance of the proposed framework and the baseline models, two evaluation metrics are employed: MAE, and a scoring metric adapted from the PRONOSTIA dataset challenge. These metrics are applied consistently across all experiments, including intra-condition, cross-condition, and ablation studies.

MAE provides complementary views of the model's regression accuracy. It quantifies the average absolute deviation between predicted and ground truth RUL values.

Beyond average error, the scoring metric plays a critical role by differentiating between early and late predictions. Late predictions, which suggest the system is healthier than it truly is, can delay necessary maintenance and lead to unexpected failures. Early predictions, while potentially conservative, offer safer intervention opportunities. The scoring function used in this work imposes an asymmetric penalty, assigning lower scores to late predictions relative to early ones, and is aligned with the original formulation in the PRONOSTIA challenge, with adaptations to fit the normalized RUL labeling strategy employed in this study.

Together, these metrics allow for a comprehensive evaluation of the model's predictive accuracy, robustness, and safety-awareness across diverse conditions and degradation patterns, as discussed in Chapter 3.

6.3 Model Evaluation Experiments

6.3.1 Intra-Condition Evaluation

The intra-condition evaluation assesses each model's ability to generalize across bearings operating under the same operating condition. For each condition in both datasets, one bearing is excluded and used as the test set, while the remaining bearings from the same condition are used for training. This process is repeated so that each bearing in a condition serves once as the test set. Performance is evaluated individually for each bearing, and the results are averaged per condition and across all conditions.

This setup is useful for simulating real-world industrial scenarios where only historical failure data from similar operating conditions are available, and the model must be deployed to monitor a new component without retraining. It isolates the model's ability to generalize over unit-to-unit variations within the same environment while

eliminating distribution shift from operating condition changes.

The proposed **MCSFormer** is compared against four competing baselines:

- **CNN-SRU** [59]: A hybrid convolutional and Simple Recurrent Unit (SRU) model designed for local feature extraction followed by temporal modeling.
- **MDAN** [67]: A Multi-Domain Adversarial Network that learns domain-invariant features to improve cross-unit performance.
- **Adaptive Transformer** [47]: A self-attention-based architecture that dynamically adjusts attention scope for different degradation phases.
- **MSG-CNN-TR** [14]: A multi-scale gated convolutional neural network integrated with a Transformer encoder to capture degradation trends at multiple levels.

These models were selected to provide a comprehensive comparison across convolutional, recurrent, attention-based, and hybrid architectures. This evaluation is conducted independently on the PRONOSTIA and XJTU-SY datasets using only MAE metric, computed over normalized RUL values. Focusing on MAE allows consistent and interpretable comparison across models and datasets. Final results are presented in Tables 6.2 and 6.3, categorized by operating condition.

Table 6.2: Intra-condition MAE results on the PRONOSTIA dataset (MAE or Avg MAE ± Std where applicable)

Bearing	**CNN-SRU [59]**	**MDAN [67]**	**Adaptive Transformer [47]**	**MSG-CNN-TR [14]**	**MCSFormer**
			1st Operating Condition		
Bearing1_1	0.1725	0.1714	0.1342	0.0910	**0.0730**
Bearing1_2	0.1869	0.1572	0.1058	0.1038	**0.0640**
Bearing1_3	0.1621	0.1486	0.1273	0.0878	**0.0627**
Bearing1_4	0.1648	0.1461	0.1279	0.1029	**0.0604**
Bearing1_5	0.1707	0.1238	0.1143	0.0890	**0.0562**
Bearing1_6	0.1920	0.1489	0.1063	0.0916	**0.0581**
Bearing1_7	0.1811	0.1436	0.1105	0.0943	**0.0517**
Avg. MAE ± Std	0.1757 ± 0.0095	0.1485 ± 0.0155	0.1180 ± 0.0096	0.0943 ± 0.0057	**0.0609 ± 0.0067**
			2nd Operating Condition		
Bearing2_1	0.2051	0.1649	0.0630	0.0582	**0.0404**
Bearing2_2	0.1927	0.1653	0.0581	0.0643	**0.0412**
Bearing2_3	0.1875	0.1807	0.0516	0.0541	**0.0387**
Bearing2_4	0.2160	0.1755	0.0688	0.0614	**0.0429**
Bearing2_5	0.2044	0.1939	0.0604	0.0558	**0.0395**
Bearing2_6	0.2157	0.1736	0.0649	0.0532	**0.0390**
Bearing2_7	0.2243	0.1835	0.0652	0.0530	**0.0377**
Avg. MAE ± Std	0.2065 ± 0.0124	0.1768 ± 0.0094	0.0617 ± 0.0055	0.0571 ± 0.0040	**0.0400 ± 0.0017**
			3rd Operating Condition		
Bearing3_1	0.2355	0.1840	0.1384	0.1229	**0.0861**
Bearing3_2	0.2221	0.1985	0.1503	0.1206	**0.0914**
Bearing3_3	0.2210	0.1931	0.1274	0.1297	**0.0913**
Avg. MAE ± Std	0.2262 ± 0.0062	0.1919 ± 0.0059	0.1387 ± 0.0093	0.1244 ± 0.0038	**0.0896 ± 0.0023**
Overall Avg. MAE ± Std	0.1895 ± 0.0266	0.1646 ± 0.0217	0.1061 ± 0.0323	0.0913 ± 0.0268	**0.0646 ± 0.0200**

Table 6.3: Intra-condition MAE results on the XJTU-SY dataset (MAE or Avg MAE ± Std where applicable)

Bearing	**CNN-SRU [59]**	**MDAN [67]**	**Adaptive Transformer [47]**	**MSG-CNN-TR [14]**	**MCSFormer**
			1st Operating Condition		
Bearing1_1	0.2110	0.1752	0.0894	0.0772	**0.0546**
Bearing1_2	0.1958	0.1820	0.0867	0.0729	**0.0503**
Bearing1_3	0.2035	0.1685	0.0788	0.0705	**0.0520**
Bearing1_4	0.1897	0.1634	0.0763	0.0690	**0.0484**
Bearing1_5	0.1983	0.1706	0.0809	0.0751	**0.0497**
Avg. MAE ± Std	0.1997 ± 0.0078	0.1719 ± 0.0066	0.0824 ± 0.0050	0.0729 ± 0.0031	**0.0510 ± 0.0022**
			2nd Operating Condition		
Bearing2_1	0.2173	0.1865	0.0921	0.0813	**0.0580**
Bearing2_2	0.2024	0.1799	0.0855	0.0794	**0.0542**
Bearing2_3	0.2158	0.1851	0.0872	0.0775	**0.0569**
Bearing2_4	0.2105	0.1912	0.0937	0.0822	**0.0593**
Bearing2_5	0.2181	0.1940	0.0955	0.0808	**0.0576**
Avg. MAE ± Std	0.2128 ± 0.0062	0.1873 ± 0.0052	0.0908 ± 0.0040	0.0802 ± 0.0018	**0.0572 ± 0.0018**
			3rd Operating Condition		
Bearing3_1	0.2294	0.1931	0.1026	0.0898	**0.0664**
Bearing3_2	0.2156	0.1863	0.0953	0.0857	**0.0625**
Bearing3_3	0.2242	0.1982	0.0980	0.0911	**0.0647**
Bearing3_4	0.2195	0.1897	0.0936	0.0872	**0.0630**
Bearing3_5	0.2316	0.1955	0.1014	0.0905	**0.0651**
Avg. MAE ± Std	0.2241 ± 0.0055	0.1926 ± 0.0043	0.0982 ± 0.0035	0.0889 ± 0.0020	**0.0643 ± 0.0015**
Overall Avg. MAE ± Std	0.2122 ± 0.0104	0.1839 ± 0.0089	0.0904 ± 0.0091	0.0807 ± 0.0070	**0.0575 ± 0.0061**

The intra-condition evaluation results presented in Tables 6.2 and 6.3 show that across both datasets, the proposed MCSFormer consistently outperforms all investigated models, demonstrating its superior ability to model degradation dynamics when operating conditions remain stable.

On the PRONOSTIA dataset, MCSFormer achieves an overall MAE of 0.0646, significantly lower than those of MSG-CNN-TR (0.0913), Adaptive Transformer (0.1061), MDAN (0.1646), and CNN-SRU (0.1895). This performance gain is consistent across all three operating conditions. In the first condition, which exhibits moderate degradation patterns, MCSFormer achieves a mean MAE of 0.0609, clearly outperforming Adaptive Transformer (0.1180) and MSG-CNN-TR (0.0943). The second condition, characterized by sharper degradation curves and clearer failure signatures, shows the lowest average MAE for MCSFormer (0.0400), suggesting that the model is especially effective in scenarios where the degradation behavior is more distinguishable. Even

in the more challenging third condition, where only three bearings are available and the degradation trends are less predictable, MCSFormer maintains its lead with an average MAE of 0.0896, outperforming Adaptive Transformer (0.1387) and MSG-CNN-TR (0.1244). These results confirm that MCSFormer generalizes well even with limited data and under noisier signals.

The XJTU-SY dataset, which presents longer sequences and subtler degradation patterns, provides a more rigorous testbed. Here, also, MCSFormer achieves the best performance with an overall MAE of 0.0575, outperforming MSG-CNN-TR (0.0807), Adaptive Transformer (0.0905), MDAN (0.1839), and CNN-SRU (0.2122). Notably, across all three conditions, MCSFormer consistently achieves the lowest MAEs and exhibits the lowest standard deviations, indicating both high accuracy and robustness. For instance, in the first condition, MCSFormer achieves 0.0510, while MSG-CNN-TR, the next best model, records 0.0729. This trend persists in the second and third conditions as well, underscoring the model's ability to handle long-range dependencies and subtle signal variations more effectively than other approaches.

These findings highlight the strength of MCSFormer's architecture and preprocessing strategy in modeling bearing degradation under stable operational condition. By combining a denoising pipeline and wavelet packet decomposition, the model receives cleaner, structured input features that enhance the learning process. These are then processed by a multi-channel Swin Transformer backbone, which effectively captures both local and global temporal dependencies in the data. This synergy enables MCSFormer to achieve the lowest MAE across all three operating conditions in both PRONOSTIA and XJTU-SY datasets, with reduced standard deviation, indicating robustness and stability. Compared to traditional models like CNN-SRU and domain-adversarial methods like MDAN, as well as more advanced architectures such as the Adaptive Transformer and MSG-CNN-TR, MCSFormer consistently demonstrates superior accuracy and generalization. Notably, even hybrid models like MSG-CNN-TR, which combine convolutional and attention mechanisms, were consistently outperformed by the Vision Transformer-based MCSFormer. These findings suggest that these transformer-based architectures are well-suited for RUL prediction tasks. Their ability to capture long-range dependencies and integrate information across channels makes them a compelling alternative to traditional CNNs, RNNs, and hybrid architectures for modeling multivariate degradation processes. A visual comparison of the overall intra-condition MAE and model robustness across both datasets is presented in Figure 6.1.

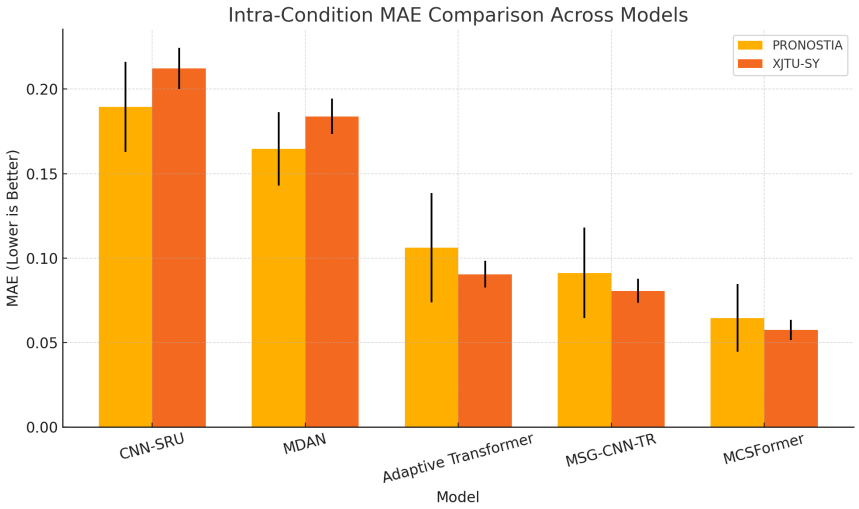

Figure 6.1: Comparison of overall intra-condition MAE across models for PRONOSTIA and XJTU-SY datasets. Error bars indicate standard deviation.

Having established MCSFormer's superiority in stable environments, the next section explores its robustness under varying operating conditions through cross-condition evaluation.

6.3.2 Cross-Condition Evaluation

Cross-condition evaluation assesses a model's ability to generalize under distribution shifts caused by varying operating conditions. Unlike the intra-condition setup, where training and testing bearings are from the same condition, this experiment follows a leave-one-bearing-out strategy across the entire dataset. In each trial, the model is trained on all bearings except one and evaluated on the one that was held out. Since the excluded bearing often originates from a different operating condition, the test distribution introduces a domain shift relative to the training data.

This setup is applied independently to both the PRONOSTIA and XJTU-SY datasets, enabling an assessment of robustness across different sequence lengths, degradation behaviors, and sensor characteristics. It reflects realistic industrial deployment scenarios where a predictive model trained on historical data must infer RUL for a new bearing under novel operating conditions. Evaluation is based on two metrics: MAE and the scoring function (Score), both computed over normalized RUL values for consistency across datasets.

The proposed MCSFormer is compared against three vision-based models:

- **CNN-SRU** [59]: A hybrid convolutional and Simple Recurrent Unit (SRU) model used in the previous section.
- **ProgSViT** [19]: A Vision Transformer-based model designed specifically for RUL prediction. The model uses transformer blocks to capture temporal degradation features directly from reshaped time-series segments, avoiding convolutional preprocessing.
- **Bi-Channel Swin Transformer (BCHVIT)** [15]: A hierarchical Swin Transformer model that processes horizontal and vertical vibration signals through separate branches. This structure allows the model to learn multiscale features with cross-channel attention.

These models were selected to represent a diverse spectrum of transformer-based approaches for time-series degradation modeling. By including ViT, hierarchical Swin, and multi-branch transformer variants, this evaluation provides insight into how different vision-based architectures respond to operating condition shifts. Comparing them against MCSFormer enables a focused assessment of transformer robustness and design trade-offs in the context of cross-condition RUL prediction. Results for this evaluation are presented in Tables 6.4 and 6.5.

Table 6.4: Cross-condition MAE and Score results on the PRONOSTIA dataset per test bearing.

Bearing	**CNN-SRU**		**ProgSViT**		**BCHViT**		**MCSFormer**	
	MAE	Score	MAE	Score	MAE	Score	MAE	Score
Bearing1_1	0.1321	0.7923	0.0881	0.8562	0.0484	0.9588	**0.0455**	**0.9615**
Bearing1_2	0.1302	0.7859	0.0826	0.8532	0.0501	0.9332	**0.0426**	**0.9611**
Bearing1_3	0.1319	0.7794	0.0854	0.8661	**0.0431**	**0.9673**	0.0445	0.9493
Bearing1_4	0.1346	0.8147	0.0811	0.8703	**0.0452**	**0.9600**	0.0472	0.9588
Bearing1_5	0.1293	0.7977	0.0823	0.8693	0.0497	0.9440	**0.0457**	**0.9500**
Bearing2_1	0.1330	0.8040	0.0872	0.8634	**0.0435**	**0.9642**	0.0462	0.9628
Bearing2_2	0.1286	0.7945	0.0804	0.8699	0.0481	0.9501	**0.0450**	**0.9554**
Bearing2_3	0.1357	0.7910	0.0842	0.8586	0.0502	0.9421	**0.0436**	**0.9572**
Bearing2_4	0.1363	0.8021	0.0816	0.8720	0.0463	0.9570	**0.0444**	**0.9585**
Bearing2_5	0.1341	0.7987	0.0835	0.8610	0.0471	0.9480	**0.0456**	**0.9538**
Bearing2_6	0.1308	0.8002	0.0809	0.8597	**0.0442**	**0.9612**	0.0460	0.9596
Bearing2_7	0.1297	0.8061	0.0828	0.8663	0.0455	0.9498	**0.0439**	**0.9563**
Bearing3_1	0.1335	0.7914	0.0850	0.8607	0.0483	0.9455	**0.0455**	**0.9489**
Bearing3_2	0.1311	0.7885	0.0837	0.8641	0.0472	0.9462	**0.0430**	**0.9591**
Bearing3_3	0.1296	0.7990	0.0815	0.8680	0.0459	0.9481	**0.0441**	**0.9524**
Average	0.1320	0.7965	0.0837	0.8636	0.0502	0.9275	**0.0444**	**0.9562**

Table 6.5: Cross-condition MAE and Score results on the XJTU-SY dataset per test bearing.

Bearing	**CNN-SRU**		**ProgSViT**		**BCHViT**		**MCSFormer**	
	MAE	Score	MAE	Score	MAE	Score	MAE	Score
Bearing1_1	0.1254	0.7920	0.0930	0.8623	0.0526	0.9392	**0.0437**	**0.9576**
Bearing1_2	0.1241	0.7897	0.0908	0.8619	**0.0460**	**0.9535**	0.0468	0.9490
Bearing1_3	0.1263	0.7862	0.0916	0.8577	0.0505	0.9421	**0.0432**	**0.9608**
Bearing1_4	0.1279	0.7941	0.0893	0.8680	0.0480	0.9479	**0.0428**	**0.9622**
Bearing1_5	0.1257	0.7986	0.0901	0.8632	0.0486	0.9455	**0.0441**	**0.9575**
Bearing2_1	0.1284	0.7869	0.0925	0.8611	0.0472	0.9500	**0.0423**	**0.9606**
Bearing2_2	0.1275	0.7842	0.0918	0.8626	**0.0457**	**0.9524**	0.0459	0.9493
Bearing2_3	0.1269	0.7883	0.0905	0.8642	0.0483	0.9468	**0.0442**	**0.9579**
Bearing2_4	0.1251	0.7914	0.0898	0.8651	0.0479	0.9472	**0.0439**	**0.9597**
Bearing2_5	0.1265	0.7890	0.0910	0.8637	0.0485	0.9449	**0.0435**	**0.9584**
Bearing3_1	0.1248	0.7931	0.0902	0.8667	0.0488	0.9464	**0.0431**	**0.9592**
Bearing3_2	0.1259	0.7895	0.0911	0.8639	0.0476	0.9491	**0.0440**	**0.9576**
Bearing3_3	0.1244	0.7953	0.0889	0.8693	0.0482	0.9458	**0.0436**	**0.9587**
Bearing3_4	0.1260	0.7910	0.0906	0.8672	0.0475	0.9487	**0.0429**	**0.9604**
Bearing3_5	0.1252	0.7940	0.0897	0.8681	0.0469	0.9505	**0.0430**	**0.9595**
Average	0.1262	0.7902	0.0905	0.8647	0.0482	0.9475	**0.0439**	**0.9583**

Shown in Tables 6.4 and 6.5, across both datasets, the proposed MCSFormer consistently outperforms the competing models on average. On the PRONOSTIA dataset, MCSFormer achieves the lowest average MAE of 0.0444 and the highest average Score of 0.9562, reflecting both precise and cautious RUL predictions when tested on previously unseen bearings. While BCHViT, another Swin Transformer-based hierarchical model, delivers competitive performance with an MAE of 0.0502 and Score of 0.9275, MCSFormer shows greater consistency, outperforming all other models on 13 out of 17 bearings. ProgSViT, a Vision Transformer architecture, yields moderate

results (MAE: 0.0837, Score: 0.8636), whereas CNN-SRU, the convolutional-recurrent baseline, shows the weakest generalization capacity with an MAE of 0.1320 and Score of 0.7965.

A similar trend appears on the XJTU-SY dataset. MCSFormer again leads with an average MAE of 0.0439 and Score of 0.9583. BCHViT performs comparably well with an MAE of 0.0482, outperforming ProgSViT (MAE: 0.0905) and CNN-SRU (MAE: 0.1262). Although BCHViT achieves better results on 2 of the 15 test bearings, MCSFormer consistently outperforms all others on the remaining 13, highlighting its robustness across varying degradation profiles and sensor dynamics.

The superior performance of MCSFormer can be attributed to several architectural strengths inherent in Swin Transformer-based designs, particularly when applied to RUL prediction under domain shift. Unlike standard Vision Transformers (e.g., ProgSViT), which operate on fixed-size global patches and lack hierarchical representation, Swin Transformers use shifted windows and hierarchical layers that enable localized feature extraction while maintaining global context through cross-window attention. This design is better suited for modeling the subtle and progressive nature of degradation in time-series vibration data. Moreover, MCSFormer enhances this architecture through its multi-branch design, enabling the model to independently process horizontal and vertical signals before merging them, which improves feature discrimination across sensor channels. Compared to CNN-based models like CNN-SRU, which rely on static convolutional filters and struggle to generalize across changing operating conditions, MCSFormer dynamically adapts to temporal and contextual variations, offering stronger generalization and stability across both datasets.

To further illustrate the predictive behavior of each model, Figure 6.2 presents the RUL prediction curves for Bearing1_1 from the PRONOSTIA dataset. MCSFormer's trajectory aligns most closely with the ground-truth RUL, especially in the later stages of degradation where predictive accuracy is most critical. ProgSViT, while generally accurate, exhibits larger fluctuations and overestimations across the portion of sequence after early degradation stage. BCHViT follows the general degradation trend but when compared with MCSFormer, it exhibits larger fluctuations, particularly in early and mid stages, and overestimations, particularly in end-of-life stages. CNN-SRU shows the most unstable behavior, with more significant underestimations and erratic fluctuations throughout the timeline. Although all transformer-based models benefit from the shared preprocessing pipeline, MCSFormer achieves the best alignment overall, demonstrating a stronger ability to capture degradation dynamics

across domains and minimize risky late predictions.

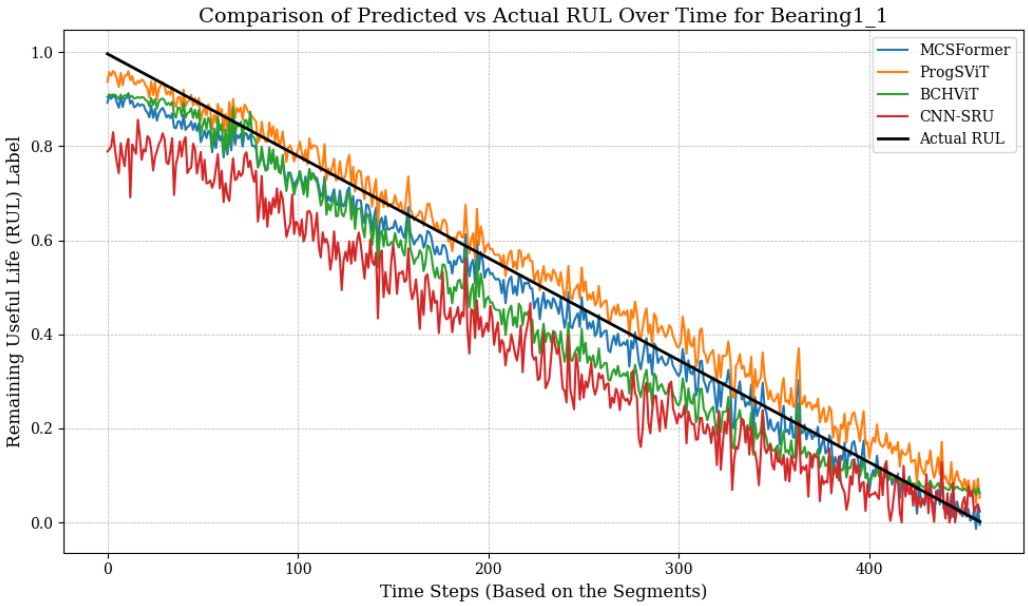

Figure 6.2: Cross-condition RUL prediction trajectories for Bearing1_1 (PRONOSTIA dataset).

In summary, the cross-condition evaluation results demonstrate that, vision-based architectures, particularly hierarchical Swin Transformer variants like MCSFormer and BCHViT, offer strong generalization capabilities across unseen bearings. While conventional ViTs and hybrid convolutional baselines show moderate to desirable performance, MCSFormer stands out as a robust and accurate solution for real-world prognostic tasks. The consistent performance across both datasets validates the efficacy of its multi-branch structure, temporal modeling design, and penalty-aware training.

6.4 Ablation Studies

To assess the individual contributions of the main components in the proposed framework, we conducted a set of ablation studies. These focused on three critical aspects: (1) the denoising pipeline applied to raw vibration signals, (2) the segmentation strategy used to structure input sequences, and (3) the custom loss function designed to penalize late predictions. By systematically removing or replacing each component,

we aim to evaluate its effect on model performance using MAE and the scoring metric. All ablation experiments were carried out on the PRONOSTIA dataset due to its more stable measurement intervals, and larger number of bearings, which allow for controlled and repeatable comparisons across trials.

6.4.1 Effect of Denoising

To evaluate the influence of signal denoising on model performance, an ablation study was conducted by removing the entire denoising pipeline from the framework. In the original configuration, raw vibration signals were sequentially processed through a low-pass filter to suppress high-frequency interference, wavelet-based denoising using db5 wavelets with adaptive thresholding, and a Savitzky-Golay filter for smoothing. This three-stage denoising process is intended to enhance the quality of the time-frequency representations generated by Wavelet Packet Decomposition. The ablation was performed under a cross-condition setup using a randomly selected representative subset of five bearings, Bearing1_1, Bearing1_5, Bearing2_3, Bearing2_6, and Bearing3_2, selected to reflect varying degradation behaviors across the three operating conditions. Table 6.6 presents the results in terms of MAE and the scoring metric, along with the relative performance difference.

Table 6.6: Effect of denoising: MAE and Score with and without the denoising pipeline

Bearing	**MAE (Denoised)**	**MAE (Raw)**	**Score (Denoised)**	**Score (Raw)**	**MAE Δ%**	**Score Δ%**
Bearing1_1	0.0455	0.0495	0.9615	0.9047	+8.8%	–5.9%
Bearing1_5	0.0457	0.0500	0.9500	0.8974	+9.4%	–5.5%
Bearing2_3	0.0436	0.0475	0.9572	0.8892	+8.9%	–7.1%
Bearing2_6	0.0460	0.0495	0.9596	0.9007	+7.6%	–6.1%
Bearing3_2	0.0430	0.0467	0.9591	0.8915	+8.6%	–7.0%
Average	0.0448	0.0486	0.9575	0.8967	+8.7%	–6.3%

Table 6.6 shows that incorporating denoising consistently improved both MAE and Score across all test bearings. The relative improvement ranged from 7% to 10% in MAE, and up to 7.1% in Score. The improvement was particularly notable in Bearing1_5 and Bearing3_2, where noisy signals had a greater influence on time-frequency features. Overall, the results validate the inclusion of the denoising step as a beneficial component for enhancing signal clarity and stability, especially under cross-condition settings.

6.4.2 Effect of Segmentation Strategy

To assess the impact of input structuring on model performance, the sliding window and expanding window segmentation strategies were compared. Both approaches had been integrated and evaluated within the framework using the same preprocessing and modeling pipeline. The goal of this ablation is to determine whether providing more historical context, as in the expanding window, leads to improved RUL prediction, or if the fixed-length sliding window offers a better trade-off between context and data consistency. This comparison is particularly relevant in real-world scenarios where the full operational history of a system may not always be available.

To ensure a fair and controlled comparison, both segmentation strategies were evaluated under a cross-condition setup, where the model was trained on all bearings from different operating conditions except one and tested on the unseen bearing. The same denoised input signals, model architecture, and preprocessing parameters were used for both strategies. Specifically, the sliding window used a fixed length of 10 measurement intervals with a stride of 5, while the expanding window began with a minimum of 10 intervals and grew incrementally with each new segment. A subset of five bearings was randomly selected from the dataset, Bearing1_2, Bearing1_4, Bearing2_1, Bearing2_7, and Bearing3_3, to cover all three operating conditions and a range of degradation behaviors. These bearings include sequences with different lengths, making them suitable for testing how each segmentation method handles early, mid, and late life stages. The results for the sliding window configuration are drawn from the earlier cross-condition experiment (Table 6.4), while the expanding window results were obtained using the same model and hyperparameters. Table 6.7 presents the comparative performance of both approaches.

Table 6.7: Comparison of MAE and Score using sliding and expanding window segmentation strategies

Bearing	**MAE (Sliding)**	**MAE (Expanding)**	**Score (Sliding)**	**Score (Expanding)**	**MAE Δ%**	**Score Δ%**
Bearing1_2	0.0426	0.0459	0.9611	0.9350	+7.7%	–2.7%
Bearing1_4	0.0472	0.0520	0.9588	0.9203	+10.2%	–4.0%
Bearing2_1	0.0462	0.0501	0.9628	0.9306	+8.4%	–3.3%
Bearing2_7	0.0439	0.0480	0.9563	0.9175	+9.3%	–4.1%
Bearing3_3	0.0441	0.0475	0.9524	0.9226	+7.7%	–3.1%
Average	0.0448	0.0487	0.9583	0.9252	+8.7%	–3.4%

The results in Table 6.7 show that the sliding window segmentation consistently

outperforms the expanding window across all test bearings. On average, the MAE increased by 8.7% and the Score dropped by 3.4% when switching from sliding to expanding window. This performance gap can be attributed to several practical limitations of the expanding window approach. First, expanding windows incur variable input lengths, which introduce architectural and computational complexity that can degrade model stability. Second, in early stages, insufficient historical data leads to shorter and less informative input sequences, reducing the benefit of the longer-term accumulation. Moreover, as the window grows, the accumulated signal history may dilute recent degradation patterns with redundant or less relevant early-life information. In contrast, the sliding window provides a consistent and focused view of recent behavior, which appears to be more effective for tracking the progression of faults. This fixed-length segmentation also simplifies training and inference while preserving enough temporal context for the model to make accurate predictions. These findings support the use of the sliding window strategy as the preferred choice for robust and generalizable RUL estimation in the proposed framework.

6.4.3 Effect of Loss Function

To evaluate the practical effect of the custom loss function introduced in Chapter 5, an ablation study was conducted comparing it to the standard MSE loss. Both loss functions were applied under identical training conditions using the cross-condition setup, where the model was trained on 16 bearings and tested on the remaining one. All other components, including architecture, segmentation strategy, and denoising pipeline, remained unchanged to isolate the effect of the loss function. While MSE treats early and late errors symmetrically, the custom loss applies an additional penalty to late predictions, encouraging the model to prioritize safety by avoiding overestimates of remaining useful life.

The experiment was conducted on five randomly selected bearings, Bearing1_4, Bearing1_5, Bearing2_3, Bearing2_6, and Bearing3_2, spanning all three operating conditions. Table 6.8 presents the model's MAE and Score for both the custom loss and standard MSE loss. The values reported for the custom loss are based on the cross-condition results already established in Table 6.4, while the MSE results were obtained from a parallel training run using the same configuration.

The results in Table 6.8 demonstrate that the custom loss consistently improves the scoring metric across all five test bearings, with gains ranging from 4.6% to

Table 6.8: Comparison of MAE and Score using custom loss and standard MSE loss

Bearing	MAE (Custom Loss)	MAE (MSE)	Score (Custom Loss)	Score (MSE)	MAE Δ%	Score Δ%
Bearing1_4	0.0472	0.0460	0.9588	0.9107	+2.6%	+5.3%
Bearing1_5	0.0457	0.0442	0.9500	0.9051	+3.4%	+5.0%
Bearing2_3	0.0436	0.0425	0.9572	0.8936	+2.6%	+7.1%
Bearing2_6	0.0460	0.0443	0.9596	0.9178	+3.8%	+4.6%
Bearing3_2	0.0430	0.0418	0.9591	0.9123	+2.9%	+5.1%
Average	0.0451	0.0438	0.9569	0.9079	+2.9%	+5.5%

7.1%. These improvements are particularly important in practical applications, as the score metric penalizes late predictions more heavily, aligning closely with real-world maintenance priorities. While MAE slightly increases, by an average of just 2.9%, this rise is modest and does not indicate a significant degradation in raw accuracy. For instance, on Bearing2_3, the MAE increased by only 2.6%, yet the score improved by 7.1%, illustrating that the custom loss successfully reorients the model's behavior toward safer and more timely predictions. Figure 6.3 provides a visual comparison of the predicted RUL trajectories for Bearing2_3 under the two loss functions.

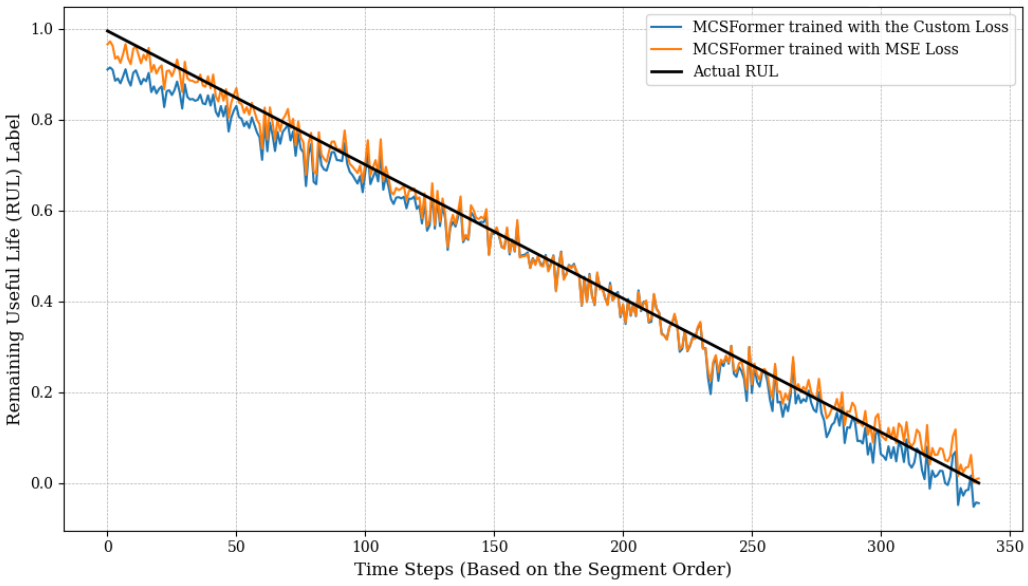

Figure 6.3: Predicted RUL curves for Bearing2_3 (PRONOSTIA dataset) using custom safety-aware loss and standard MSE loss.

As shown in Figure 6.3, both models begin by slightly underestimating RUL in early stages, but their behaviors diverge significantly as the system nears failure. The

model trained with MSE loss increasingly overestimates RUL in the later degradation stages, often remaining above the true RUL curve until the end of life. In contrast, the model trained with the custom safety-aware loss tracks the ground truth with high accuracy, especially in the final one-third of the operational timeline. This more conservative trajectory results in fewer late predictions. As discussed before, late predictions pose a major risk in prognostics, as they delay necessary maintenance and can lead to unplanned downtime or even catastrophic failure. By introducing an asymmetric penalty, the custom loss explicitly discourages overestimation, pushing the model to make earlier warnings without becoming overly pessimistic. This balance between caution and accuracy is essential for safety-critical systems. Overall, both the quantitative and qualitative results confirm that the custom loss function contributes meaningfully to safer and more operationally aligned predictions, completing the case for its integration into the proposed framework.

In summary, the experiments and ablation studies presented in this chapter validate the robustness, flexibility, and effectiveness of the proposed MCSFormer framework. Through systematic evaluation across intra- and cross-condition scenarios, along with targeted analysis of architectural and design choices, the framework demonstrated consistent performance advantages over state-of-the-art baselines. The empirical findings not only support the key design decisions made throughout this work, but also provide a strong foundation for the practical and theoretical reflections outlined in the final chapter.

Chapter 7

Conclusions and Future Work

7.1 Contributions and Practical Implications

This research addressed the problem of accurately predicting the remaining useful life (RUL) of rolling element bearings under variable operating conditions using vibration sensor data. The proposed framework, MCSFormer, combines a multi-channel design for handling dual-sensor signals with a Swin Transformer backbone, aiming to enhance both predictive accuracy and generalization. Throughout the development of the framework, several design choices were made to improve robustness, including a tailored denoising pipeline, an informative segmentation strategy, and a custom safety-aware loss function to penalize late predictions. These components were evaluated through extensive intra- and cross-condition experiments, demonstrating their individual and collective value.

At the core of this research is the design of the MCSFormer architecture, a multi-channel Swin Transformer framework tailored for dual-sensor vibration data. The architecture employs separate processing streams for horizontal and vertical signals, allowing the model to extract axis-specific temporal features before integrating them through a shared attention mechanism. This dual-branch design enhances the model sensitivity to nuanced degradation patterns that may manifest differently across sensor directions. Another contribution lies in the development of an effective preprocessing pipeline designed to enhance time-frequency representations. While denoising is a common step in signal processing, this work integrates a specific sequence, low-pass filtering, wavelet-based denoising, and Savitzky-Golay smoothing, followed by Wavelet Packet Decomposition (WPD). This combination was found to significantly improve

the clarity of extracted features, leading to more discriminative representations for downstream prediction. Additionally, a sliding window segmentation strategy was adopted to generate consistent-length input segments, which not only stabilized the training process but also avoided the complications introduced by variable-length contexts in expanding windows. Finally, a custom safety-aware loss function was introduced to penalize late predictions more heavily than early ones. This loss formulation explicitly aligns model training with the asymmetric cost structure of real-world maintenance scenarios, where delayed warnings carry higher operational risk.

These contributions collectively address critical challenges in real-world predictive maintenance. The dual-sensor design equips the framework to detect subtle failure patterns that might be overlooked in traditional setups, while the customized preprocessing pipeline ensures that the extracted features remain reliable even under noisy or degraded signal conditions. Consistent segmentation enables stable model operation regardless of how much historical data is available, making the approach well-suited for deployment in partially observed environments. Most crucially, the safety-aware loss drives the model toward conservative yet reliable RUL estimates, reducing the likelihood of late predictions, an outcome particularly vital in safety-critical applications where delayed intervention can have costly or catastrophic consequences. Together, these elements contribute to a predictive system that is not only accurate, but also robust, risk-aware, and deployment-ready.

7.2 Limitations

While the proposed framework demonstrates strong performance across a range of settings, several limitations remain that constrain its generalizability and practical deployment. First, the evaluation was conducted entirely on the PRONOSTIA and XJTU-SY datasets, both of which are collected in controlled laboratory environments. Although these datasets are widely used benchmarks in the prognostics community, they do not fully capture the variability, sensor noise, and operational complexity present in real-world industrial settings. The models trained on these datasets may therefore exhibit performance degradation when exposed to field data with different distributions, sensor placements, or failure modes.

Another limitation involves the computational complexity of the proposed architecture. While Swin Transformers offer excellent representational power, they also introduce significant memory and processing overhead compared to more lightweight

alternatives. This may limit the framework's suitability for deployment on resource-constrained devices, such as embedded systems or edge computing units commonly used in industrial IoT setups. Additionally, the current implementation relies solely on vibration data from two sensors, without incorporating complementary modalities such as temperature, acoustic emissions, or current signals. Although the framework is adaptable, its performance in multi-modal or cross-sensor fusion scenarios remains unexplored and may require architectural modifications.

Furthermore, the current framework operates in a fully offline manner, assuming access to complete input segments before making predictions. While this setup is sufficient for benchmarking and evaluation, real-world deployment often requires continuous monitoring and real-time inference capabilities. The framework has not been tested under streaming conditions or with online learning mechanisms that adapt to new data without retraining from scratch. Additionally, the model assumes that each bearing degrades over a single run-to-failure cycle, without accounting for complex usage patterns, intermittent faults, or varying load profiles that may arise in operational machinery. These assumptions, while simplifying model development, may limit the applicability of the system in settings where degradation is nonlinear or interrupted.

7.3 Future Work

Several directions remain open for extending this work, some of which stem directly from the limitations identified in the preceding section. First, applying the proposed framework to real-world industrial datasets would be a critical next step to evaluate its robustness under realistic conditions, including variable noise levels, sensor drift, and diverse failure types. Since the current experiments were conducted entirely on laboratory datasets, generalizing the model to field data would provide stronger evidence of its practical value. Second, future research could focus on making the model operate in an online or incremental learning setting, where it continuously updates its predictions as new data becomes available, rather than relying on pre-segmented input. This would align the framework more closely with real-time monitoring systems. Lastly, incorporating transfer learning techniques could help the model adapt more efficiently to new machines or operating conditions with limited labeled data, improving scalability across different deployment scenarios.

In addition to improving adaptability, future work could explore extending the

framework to support multi-modal sensor inputs. While this study focused on horizontal and vertical vibration signals, other sensor types, such as temperature, acoustic emission, or electrical current, could provide complementary information and further improve prediction accuracy. Another valuable extension would be to incorporate uncertainty estimation into the prediction process. By quantifying the confidence associated with each RUL estimate, the model could offer more actionable insights, particularly in scenarios where decisions must weigh prediction accuracy against risk. Exploring these directions could further enhance the robustness, reliability, and practical utility of the proposed approach in real-world maintenance settings.

Bibliography

[1] Muhammad Gibran Alfarizi, Bahareh Tajiani, Jørn Vatn, and Shen Yin. Optimized random forest model for remaining useful life prediction of experimental bearings. *IEEE Transactions on Industrial Informatics*, 2023.

[2] Mohammad Zawad Ali and Xiaodong Liang. Induction motor fault diagnosis using discrete wavelet transform. In *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, pages 1–4, 2019.

[3] Huaiqian Bao, Lijin Song, Zongzhen Zhang, Baokun Han, Jinrui Wang, Junqing Ma, and Xingwang Jiang. Prediction of the remaining useful life of rolling bearings by lstm based on multidomain characteristics and a dual-attention mechanism. *Journal of Mechanical Science and Technology*, 37(9):4583–4596, September 2023.

[4] RV Bhandare et al. Fault diagnosis and prediction of remaining useful life (rul) of rolling element bearing: A review state of art. *TRIBOLOGIA - Finnish Journal of Tribology*, 41(1-2):31–40, 2024.

[5] Manuel Arias Chao, Chetan Kulkarni, Kai Goebel, and Olga Fink. Fusing physics-based and deep learning models for prognostics. *Reliability Engineering & System Safety*, 217:107961, 2022.

[6] Cheng Cheng, Guijun Ma, Yong Zhang, Ye Yuan, Zhiwei Huang, Tao Peng, and Michael Pecht. Online bearing remaining useful life prediction based on a novel degradation indicator and convolutional neural networks. *arXiv preprint arXiv:1812.03315*, 2018.

[7] David L Donoho and Iain M Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.

[8] D.L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[10] Zhengyang Fan, Wanru Li, Kuo-chu Chang, and Ting Yuan. Performer: A permutation based vision transformer for remaining useful life prediction. *arXiv preprint arXiv:2506.00259*, 2025.

[11] Carlos Cesar Martins Ferreira and Gustavo Gonçalves. Remaining useful life prediction and challenges: A literature review on the use of machine learning methods. *Journal of Manufacturing Systems*, 63:550–562, 2022.

[12] Shaokun Fu, Yize Wu, Rundong Wang, and Mingzhi Mao. A bearing fault diagnosis method based on wavelet denoising and machine learning. *Applied Sciences*, 13(10):5936, 2023.

[13] Shuzhi Gao, Zeqin Li, Yimin Zhang, Sixuan Zhang, and Jin Zhou. Remaining useful life prediction method of rolling bearings based on improved 3σ and dbohkelm. *Measurement Science and Technology*, 35(10):106101, jul 2024.

[14] Dong Guo, Zhi Cao, Hongyong Fu, and Zhenxiang Li. Remaining useful life estimation for rolling bearings using msgcnn-tr. *IEEE Sensors Journal*, 22(24):24333–24343, 2022.

[15] Weixuan Hao, Zhen Li, Guangming Qin, Kai Ding, Xudong Lai, and Ke Zhang. A novel prediction method based on bi-channel hierarchical vision transformer for rolling bearings' remaining useful life. *Processes*, 11(4):1153, 2023.

[16] Yonghao He, Changjun Wen, and Wei Xu. Residual life prediction of sa-cnn-bilstm aero-engine based on a multichannel hybrid network. *Applied Sciences*, 15(2):966, 2025.

[17] Aik-Leong Heng, Shaoping Zhang, Andy CH Tan, and Joseph Mathew. Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing*, 23(3):724–739, 2009.

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[19] Aijun Hu, Yancheng Zhu, Suixian Liu, and Ling Xiang. A novel vision transformer network for rolling bearing remaining useful life prediction. *Measurement Science and Technology*, 35(2), 2023.

[20] HZ Huang, ZL Wang, and YF Li. Support vector machine based estimation of remaining useful life: Current research status and future trends. *Journal of Mechanical Science and Technology*, 29(1):151–163, 2015.

[21] Wanqing Huang, Yang Chen, Yongqi Chen, and Tao Zhang. Life prediction method of rolling bearing based on cnn-lstm-am. *Journal of Vibroengineering*, 26(1):1–12, 2024.

[22] Sadiqa Jafari and Yung-Cheol Byun. A cnn-gru approach to the accurate prediction of batteries' remaining useful life from charging profiles. *Computers*, 12(11):219, 2023.

[23] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020.

[24] Sunghyun Kim, Yun-Ho Seo, and Junhong Park. Transformer-based novel framework for remaining useful life prediction of lubricant in operational rolling bearings. *Reliability Engineering & System Safety*, 251:110377, 2024.

[25] H. Li, Z. Wang, and Z. Li. An enhanced cnn-lstm remaining useful life prediction model for aircraft engine with attention mechanism. *PeerJ Computer Science*, 8:e1084, 2022.

[26] Xiang Li, Wei Zhang, and Qian Ding. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. *Reliability Engineering & System Safety*, 182:208–218, 2019.

[27] Zhixuan Li, Kai Zhang, Xuwei Lai, Qing Zheng, and Guofu Ding. A remaining useful life prediction method for rolling bearing based on multi-channel fusion hierarchical vision transformer. In *2023 IEEE 12th Data Driven Control and Learning Systems Conference (DDCLS)*, pages 1025–1029, 2023.

[28] Bingguo Liu, Zhuo Gao, Binghui Lu, Hangcheng Dong, and Zeru An. Sal-cnn: Estimate the remaining useful life of bearings using time-frequency information. *arXiv preprint arXiv:2204.05045*, 2022.

[29] Lu Liu, Xiao Song, and Zhetao Zhou. Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture. *Reliability Engineering & System Safety*, 221:108330, 2022.

[30] Qiang Liu, Zhengwei Dai, Peirong Chen, Hongxi Lai, Youlin Liang, Minghao Chen, Xiaoming Xu, Mingxin Hou, and Guangbin Wang. Remaining useful life prediction of rolling bearings based on tcn-lstm. In *14th International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE 2024)*, pages 1016–1020. IET, 2024.

[31] Ze Liu, Yutong Lin, Yu Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021.

[32] C.J. Lu and W.Q. Meeker. Using degradation measures to estimate a time-to-failure distribution. *Technometrics*, 35(2):161–174, 1993.

[33] Chun Fai Lui and Min Xie. Deep residual network for remaining useful life prediction. In *Proceedings of the 28th ISSAT International Conference on Reliability and Quality in Design*, page 247, 2023.

[34] Xu Lv, Fengxing Zhou, Bin Li, and Baokang Yan. Incipient fault feature extraction of rolling bearing based on signal reconstruction. *Electronics*, 12(18):3749, 2023.

[35] Ping Ma, Guangfu Li, Hongli Zhang, Cong Wang, and Xinkai Li. Prediction of remaining useful life of rolling bearings based on multiscale efficient channel attention cnn and bidirectional gru. *IEEE Transactions on Instrumentation and Measurement*, 73:1–13, 2024.

[36] Ali Mohajerzarrinkelk, Maryam Ahang, Mehran Zoravar, Mostafa Abbasi, and Homayoun Najjaran. Multi-channel swin transformer framework for bearing remaining useful life prediction. *arXiv preprint arXiv:2505.14897*, 2025.

[37] A. Vishwendra More, Pratiksha S. Salunkhe, Shivanjali V. Patil, Sumit A. Shinde, P. V. Shinde, R. G. Desavale, P. M. Jadhav, and Nagaraj V. Dharwadkar. A novel method to classify rolling element bearing faults using k-nearest neighbor machine learning algorithm. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 8(3), 2022.

[38] Patrick Nectoux, Rafael Gouriveau, Kamal Medjaher, Emmanuel Ramasso, Brigitte Chebel-Morello, Noureddine Zerhouni, and Christophe Varnier. PRONOSTIA : An experimental platform for bearings accelerated degradation tests. In *Conference on Prognostics and Health Management.*, volume sur CD ROM, pages 1–8, Denver, Colorado, United States, June 2012. IEEE Catalog Number : CPF12PHM-CDR.

[39] Oluwaseyi Ogunfowora and Homayoun Najjaran. A transformer-based framework for multi-variate time series: A remaining useful life prediction use case. *arXiv preprint arXiv:2308.09884*, 2023.

[40] Rajkumar Palaniappan. Comparative analysis of support vector machine, random forest and k-nearest neighbor classifiers for predicting remaining usage life of roller bearings. *Informatica*, 48(7):123–130, 2024.

[41] P.C. Paris and F. Erdogan. A critical analysis of crack propagation laws. *Journal of Basic Engineering*, 85(4):528–534, 1963.

[42] Kyungnam Park, Yohwan Choi, Won Jae Choi, Hee-Yeon Ryu, and Hongseok Kim. Lstm-based battery remaining useful life prediction with multi-channel charging profiles. *IEEE Access*, 8:20786–20798, 2020.

[43] C. Rajeswari, B. Sathiyabhama, S. Devendiran, and K. Manivannan. Bearing fault diagnosis using wavelet packet transform, hybrid pso and support vector machine. *Procedia Engineering*, 97:1772–1783, 2014.

[44] Robert Bond Randall. Vibration-based condition monitoring: industrial, aerospace and automotive applications. *John Wiley & Sons*, 2021.

[45] Xiaolei Si, Wenbin Wang, Chao Hu, and Donghua Zhou. Remaining useful life estimation–a review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1):1–14, 2011.

[46] Xiaolei Si, Wenbin Wang, Chao Hu, and Donghua Zhou. Remaining useful life estimation–a review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1):1–14, 2011.

[47] Xuanyuan Su, Hongmei Liu, Laifa Tao, Chen Lu, and Mingliang Suo. An end-to-end framework for remaining useful life prediction of rolling bearing based on feature pre-extraction mechanism and deep adaptive transformer model. *Computers & Industrial Engineering*, 161:107531, 2021.

[48] Kwok L Tsui, Nan Chen, Qiang Zhou, Yizhen Hai, and Wenbin Wang. Prognostics and health management: A review on data driven approaches. *Mathematical Problems in Engineering*, 2015:793161, 2015.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[50] Biao Wang, Yaguo Lei, Naipeng Li, and Ningbo Li. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability*, 69(1):401–412, 2020.

[51] Biao Wang, Yaguo Lei, Tao Yan, Naipeng Li, and Liang Guo. Recurrent convolutional neural network: A new framework for remaining useful life prediction of machinery. *Neurocomputing*, 379:117–129, 2020.

[52] Zhaozong Wang, Jiangfeng Cheng, Hui Zheng, Xiaofu Zou, and Fei Tao. Multistage convolutional autoencoder and bcm-lstm networks for rul prediction of rolling bearings. *IEEE Transactions on Instrumentation and Measurement*, 72:1–13, 2023.

[53] Fuhui Wu, Qingbo Wu, Yusong Tan, and Xinghua Xu. Remaining useful life prediction based on deep learning: A survey. *Sensors*, 24(11):3454, 2024.

[54] Haixu Wu, Jinpeng Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

[55] Jian-Da Wu and Jian-Ji Chan. Faulted gear identification of a rotating machinery based on wavelet transform and artificial neural network. *Expert Systems with Applications*, 36(5):8862–8875, 2009.

[56] Yancai Xiao, Jinyu Xue, Mengdi Li, and Wei Yang. Low-pass filtering empirical wavelet transform machine learning based fault diagnosis for combined fault of wind turbines. *Entropy*, 23(8):975, 2021.

[57] Zaimi Xie, Chunmei Mo, and Baozhu Jia. A novel swin-transformer with multi-source information fusion for online cross-domain bearing rul prediction. *Journal of Marine Science and Engineering*, 13(5):842, 2025.

[58] Mingming Yan, Xiaofei Wang, Bing Wang, et al. Bearing remaining useful life prediction using support vector machine and hybrid degradation tracking model. *ISA Transactions*, 91:331–342, 2019.

[59] Dechen Yao, Boyang Li, Hengchang Liu, Jianwei Yang, and Limin Jia. Remaining useful life prediction of roller bearings based on improved 1d-cnn and simple recurrent unit. *Measurement*, 175:109166, 2021.

[60] Xianbiao Zhan, Zixuan Liu, Hao Yan, Zhenghao Wu, Chiming Guo, and Xisheng Jia. A two-stage framework for predicting the remaining useful life of bearings. *Open Physics*, 22(1):20230187, 2024.

[61] Jingliang Zhang and Jay Lee. A review on prognostics and health monitoring of li-ion battery. *Journal of Power Sources*, 196(15):6007–6014, 2011.

[62] Jiusi Zhang, Yuchen Jiang, Shimeng Wu, Xiang Li, Hao Luo, and Shen Yin. Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism. *Reliability Engineering & System Safety*, 221:108297, 2022.

[63] Jianfei Zheng, Qing Dong, Xuanjun Wang, Qingchao Zhang, and Dangbo Du. Adaptive wiener process–based remaining useful life prediction method considering multi-source variability. *Heliyon*, 10(16):e35925, 2024.

[64] Shuai Zheng, Konstantin Ristovski, Ahmed Farahat, and Chetan Gupta. Long short-term memory network for remaining useful life prediction. *Proceedings of the 2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–7, 2017.

[65] Shuai Zheng, Kristijan Ristovski, Ayman Farahat, and Chetan Gupta. Long short-term memory network for remaining useful life estimation. In *2017 IEEE*

International Conference on Prognostics and Health Management (ICPHM), pages 88–95. IEEE, 2017.

[66] Haoyi Zhou, Shanghang Zhang, Jie Peng, Shuai Zhang, Jianyi Li, Hui Xiong, and Weiping Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115, 2021.

[67] Yisheng Zou, Zhixuan Li, Yongzhi Liu, Shijiao Zhao, Yantao Liu, and Guofu Ding. A method for predicting the remaining useful life of rolling bearings under different working conditions based on multi-domain adversarial networks. *Measurement*, 188:110393, 2022.