Signal Processing Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
https://sps.ewi.tudelft.nl/

SPS-2025-00

M.Sc. Thesis

Motor Fault Detection Using Transformer-Based Models

Jiarui Zhou

Abstract

With the rapid development of industrial systems, the demand for stability, reliability, and robustness has become increasingly critical. Fault detection has emerged as a key research area, aiming to prevent unexpected failures and performance degradation. Recent advances in feature extraction techniques and machine learning have enabled the development of intelligent, autonomous fault detection systems.

This thesis proposes two Transformer-based models for motor fault detection. The first is a supervised classification model that incorporates discrete wavelet transform (DWT) to decompose time-series signals into multi-scale components, which are then processed by Transformer-based architectures to extract features for classification. Two structural variants are explored: one using masked attention over concatenated coefficients, and another employing upsampling and linear attention for efficient fusion. The second approach is an unsupervised forecasting-based model, where only normal samples are used for training. At inference time, samples are classified based on whether their forecasting error exceeds a threshold determined via ROC curve analysis on a validation set.

Experiments conducted on the JKU and CWRU datasets demonstrate the effectiveness of both approaches. The classification-based method achieves high accuracy in distinguishing between fault types, while the forecasting-based method shows strong robustness to previously unseen fault categories without retraining. The findings indicate that the models are capable of capturing informative temporal patterns for fault detection, showing promise for further exploration in real-world settings.

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Motor Fault Detection Using Transformer-Based Models

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Jiarui Zhou
born in Xiangyang, China

This work was performed in:

Signal Processing Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Delft University of Technology
Department of
Microelectronics

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Motor Fault Detection Using Transformer-Based Models"** by **Jiarui Zhou** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: August 21, 2025

Chairman:

prof.dr.ir. Justin Dauwels

Advisor:

Sinian Li

Committee Members:

prof.dr. Qing Wang

Abstract

With the rapid development of industrial systems, the demand for stability, reliability, and robustness has become increasingly critical. Fault detection has emerged as a key research area, aiming to prevent unexpected failures and performance degradation. Recent advances in feature extraction techniques and machine learning have enabled the development of intelligent, autonomous fault detection systems.

This thesis proposes two Transformer-based models for motor fault detection. The first is a supervised classification model that incorporates discrete wavelet transform (DWT) to decompose time-series signals into multi-scale components, which are then processed by Transformer-based architectures to extract features for classification. Two structural variants are explored: one using masked attention over concatenated coefficients, and another employing upsampling and linear attention for efficient fusion. The second approach is an unsupervised forecasting-based model, where only normal samples are used for training. At inference time, samples are classified based on whether their forecasting error exceeds a threshold determined via ROC curve analysis on a validation set.

Experiments conducted on the JKU and CWRU datasets demonstrate the effectiveness of both approaches. The classification-based method achieves high accuracy in distinguishing between fault types, while the forecasting-based method shows strong robustness to previously unseen fault categories without retraining. The findings indicate that the models are capable of capturing informative temporal patterns for fault detection, showing promise for further exploration in real-world settings.

Acknowledgments

After nine months of dedication, I have finally completed this research project. From initial exploration to eventual proficiency, this achievement is not only the result of my own effort and perseverance, but also owes much to the support and guidance of those around me.

First and foremost, I would like to thank myself, for the persistence and commitment that carried me through this journey and made this substantial work possible.

I am deeply grateful to my parents for their unwavering support and encouragement. Studying abroad, far from home, would not have been possible without their constant comfort and motivation.

I would also like to express my sincere appreciation to my supervisor, Justin, and my daily advisor, Sinian. Their insightful guidance and continuous support were instrumental in helping me navigate challenges and ultimately complete this thesis successfully.

My gratitude also goes to the member of my thesis defense committee, Prof. Qing, whose engagement helped bring this work to its final form.

I am thankful to all the professors and fellow students in the SPS group, from whom I have learned so much. Special thanks also go to my roommates and friends, your daily companionship and encouragement helped me stay positive and motivated throughout this intense process.

This thesis is part of the R-PODID project. R-PODID is supported by the Chips Joint Undertaking and its members, including the top-up funding by National Authorities of Italy, Turkey, Portugal, The Netherlands, Czech Republic, Latvia, Greece, and Romania under grant agreement n° 101112338.

As I look ahead, I see this thesis as a milestone marking the end of one chapter and the beginning of another. I hope to pursue new goals and challenges with the same determination and optimism that brought me here.

Jiarui Zhou
Delft, The Netherlands
August 21, 2025

Contents

List of Figures

List of Tables

Introduction 1

1.1 Background

Following the advent of the 21st century and the progress in the realms of science and technology, there has been a demonstrable increase in the efficiency of industrial manufacturing systems. However, the complexity of these systems can impede the process of troubleshooting and problem resolution when a malfunction occurs, whether in the field of semiconductor manufacturing [5] or aircraft control [6]. Inadequate diagnosis and maintenance may result in component failure, performance degradation and unexpected breakdown. Such occurrences often lead to a decline in productivity and consequent economic losses for the enterprises concerned. The high demand for reliability, stability and robustness makes more and more researchers devote to the field of fault detection.

Fault detection, a foundational element in system monitoring and maintenance, functions by identifying deviations from standard system behavior through the analysis of measurable signals and parameters. The core idea is to establish a baseline for normal operation status and monitor for anomaly behaviors that may indicate faults. Traditional fault detection approaches can be broadly categorized into two main types: model-based methods and signal-based methods. Model-based approaches rely on mathematical representations of the system to generate residuals that indicate differences between expected and actual behavior. Signal-based methods utilize statistical analysis and signal processing techniques to detect patterns of faults with sensor measurements.

However, conventional fault detection methods face some limitations in complex industrial environments. They generally require expert domain knowledge and understanding of system dynamics, which demands considerable time investment from experienced engineers. For example, the development of accurate mathematical or physical models for model-based methods requires comprehensive knowledge of system physics under various operating conditions. Similarly, the design of effective signal processing algorithms requires proficiency in both the fundamental signal processing techniques and the specific application domain. This dependence on human expertise makes traditional fault detection methods not only time-consuming but also labor-intensive. The development and validation in complex systems frequently last for a period of months or even years.

Furthermore, modern industrial systems are becoming increasingly complex. The establishment of accurate analytical models and reliable standard signal patterns is more challenging. At the same time, traditional methods also face difficulties in adapting to the dynamic characteristics of complex industrial production processes. These challenges create an urgent need for more automated, adaptive, and data-driven approaches to fault detection that can operate effectively while minimizing both time and human costs.

1.2 Motivation

In recent years, the rapid development of machine learning and artificial intelligence has demonstrated remarkable capabilities in feature extraction and pattern learning. These technologies are successfully applied across diverse domains, ranging from neural language processing to autonomous driving. Data-driven methods have significant advantages over traditional approaches, as they require minimal human intervention and primarily rely on large volumes of historical data. In industrial environments, such data can be obtained cost-effectively via existing sensor networks and monitoring systems. Machine learning is an emerging method in many cutting-edge applications. However, its usage in fault detection remains limited despite its promising potential.

The integration of machine learning techniques into fault detection systems provides an opportunity to address the fundamental limitations of conventional methods. The ability to adapt and improve their performance over time makes machine learning methods well-suited for dynamic industrial environments. The utilisation of data-driven methodologies holds considerable promise for the development of more robust, efficient, and cost-effective fault detection systems that can function with reduced human supervision while maintaining high levels of accuracy and reliability. This research aims to explore the application of machine learning techniques in fault detection, and to develop a more reliable and efficient fault detection system.

1.3 Problem Statement

The application of machine learning techniques makes automatic 24-hour monitoring and timely detection of faults more practical and easier to implement compared to traditional methods. This capability is particularly crucial in industrial settings where continuous operation is essential. In the context of industrial manufacturing systems, motors play a pivotal role. Their unexpected failure can result in substantial production losses and safety hazards.

This research focuses specifically on the motor fault detection domain. Motors are widely used across industries and generate measurable signals that can be effectively analyzed using data-driven approaches. Current signals and vibration signals represent two primary sources of information that contain valuable patterns useful in the assessment of motor health and operation status. These signals can be continuously monitored using standard sensors, making them ideal candidates for machine learning-based fault detection systems.

The primary objective of this study is to develop an end-to-end machine learning model capable of classifying motor states through the analysis of current signals or vibration signals. It is expected that this model will be able to distinguish between normal operation and various fault conditions. The development of such a system would demonstrate the practical application of machine learning in industrial fault detection while addressing the limitations of conventional approaches identified in the previous sections.

1.4 Outline

This thesis is structured as follows. Chapter 2 provides an overview of related work in the field of fault detection. Chapter 3 describes the two datasets utilized in this study. Chapter 4 presents the classification-based fault detection methods, while Chapter 5 introduces the forecasting-based approaches. Chapter 6 outlines the experimental setup and presents the results. These results are further analyzed and discussed in Chapter 7. Finally, Chapter 8 concludes the thesis and highlights potential directions for future research.

Literature Review 2

Data-driven fault detection methods have emerged as a powerful approach to industrial condition monitoring, making use of the large amounts of operational data generated by modern systems. Unlike traditional model-based approaches, which rely on a fundamental understanding of system dynamics, data-driven methods extract knowledge directly from historical and real-time data to identify patterns indicative of normal and faulty operation. These approaches are based on the idea that sufficient data containing examples of healthy and faulty system behaviour can be used to train algorithms to autonomously detect and classify faults.

The main idea behind data-driven motor fault detection is to use machine learning techniques to identify patterns in signals and detect faults. There are many machine learning techniques that can be used for this purpose. Depending on whether the data is labelled, machine learning can be divided into two categories: supervised and unsupervised. In this section, we also classify data-driven fault detection into these two categories and a hybrid approach.

2.1 Supervised Learning

Supervised learning approaches rely on the existence of labelled training data, containing examples of both normal and various fault conditions. These methods are designed to learn how to map input features to predefined fault classes, thereby enabling them to classify new observations into specific fault categories. Common supervised techniques include support vector machine (SVM), long short-term memory (LSTM), convolution neural network (CNN), and transformer, etc. A significant benefit of supervised learning is its capacity to yield reasonably precise classification results. However, the feasibility of this approach is constrained by the availability of labelled fault data, the acquisition of which can be expensive in practice.

2.1.1 Support Vector Machine

Support Vector Machine (SVM) [7] is a supervised learning algorithm that aims to find an optimal hyperplane that separates different classes in the feature space with the maximum margin. The principle of SVM is to find the decision boundary that maximizes the distance between the hyperplane and the nearest data points (support vectors) from each class. For a linearly separable binary classification problem, the optimal hyperplane can be expressed as:

$$\mathbf{w}^T\mathbf{x} + b = 0 \tag{2.1}$$

where $\mathbf{w}$ is the weight vector and b is the bias term. The optimization problem aims to minimize:

$$\frac{1}{2}|\mathbf{w}|^2 \tag{2.2}$$

subject to the constraints:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \tag{2.3}$$

for all training samples, where $y_i \in \{-1, +1\}$ represents the class labels. For non-linearly separable data, SVM employs kernel functions such as the radial basis function (RBF) kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma|\mathbf{x}_i - \mathbf{x}_j|^2) \tag{2.4}$$

to map the input space into a higher-dimensional feature space where linear separation becomes possible. In fault detection applications, SVM has been widely used in the field of thermal power plant turbines [8], wind turbines [9], wireless sensor networks [10], and induction motor bearings [11].

Multi-scale current or vibration signals are difficult to use directly as feature vectors due to their temporal complexity. Therefore, SVM-based methods typically compute statistical features from the signals to serve as classification inputs. These statistical features capture the essential characteristics of the signal while keeping the dimensionality in a manageable level. For instance, in [11], the authors employed three statistical features as classification criteria: root mean square (RMS), crest factor, and kurtosis. The RMS reflects the overall energy of the signal, while the crest factor captures impulsive behavior, and kurtosis quantifies deviations from a normal distribution. These features are highly effective in distinguishing between normal and faulty operating states.

2.1.2 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks [12] are a specialized variant of recurrent neural networks (RNNs), well-suited for modeling long-range dependencies in sequential data. The innovation of LSTM lies in its memory cell structure, which consists of three gating mechanisms: the forget gate, input gate, and output gate. The open and close of these gates control the flow of information through the network and allow it to selectively remember or forget information over time periods. The LSTM cell state C_t, hidden state h_t and output state o_t at time step t are updated according to the following equations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.5}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.6}$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2.7}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{2.8}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.9}$$
$$h_t = o_t \odot \tanh(C_t) \tag{2.10}$$

where σ represents the sigmoid function and W and b denote weight matrices and bias vectors respectively. The forget gate determines what information to discard from the cell state, the input gate and candidate values decide what new information to store in the updated cell state, and the output gate controls what parts of the cell state to output. In fault detection applications, LSTM networks play a pivotal role given the fact that most of the signals employed in fault detection are time series signals.

In [13], an end-to-end LSTM model is used to directly learn from raw process data without manual feature extraction. By using batch normalization, the model achieves faster convergence and improved stability. The result on the Tennessee Eastman Process shows LSTM outperforms other conventional model like dynamic principal component analysis (DPCA) + SVM and MLP in both accuracy and robustness. This demonstrates the effectiveness of LSTM in capturing dynamic information of input data.

2.1.3 Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning architecture widely used in image processing tasks. It uses the convolution operations to extract hierarchical features from input data. The convolutional layers apply learnable filters to detect local patterns such as edges, textures, and complex shapes. The filters are shifted across the input data to produce feature maps, followed by pooling layers for dimensionality reduction and fully connected layers for final classification.

However, in fault detection applications, the dataset typically consists of one-dimensional time-series signals. To address this challenge, the variant of CNNs, one-dimensional CNNs (1D-CNNs) are used. In 1D-CNNs, the convolution operation is performed along the time axis using one-dimensional kernels:

$$y[n] = \sum_{k=0}^{K-1} w[k] \cdot x[n-k] + b \tag{2.11}$$

where $y[n]$ is the output feature map, $w[k]$ represents the learnable kernel weights, $x[n]$ is the input signal, K is the kernel size, and b is the bias term.

In [14], the authors employed 1D-CNN for bearing fault diagnosis using raw vibration signals. The model uses multiple layers of one-dimensional convolution and pooling and includes dropout to enhance generalization. Experiments on the CWRU dataset show that the model achieves an average classification accuracy of 99.2% under single load condition and 98.8% under different loads.

Although classical two-dimensional CNNs cannot be directly applied to fault detection due to the one-dimensional nature of sensor signals, several innovative approaches have been developed to transform raw signals into image-like representations, thereby leveraging the well-developed CNN technologies.

As shown in Figure 2.1, a representative approach involves converting time-series signals into grayscale images through systematic data restructuring. In this method, N^2 consecutive signal points are selected from the raw signal and segmented into N equal parts. Each data point within these segments is normalized to a range of 0 to 255. Subsequently, each segment forms either a row or column in the resulting image matrix, creating an $N \times N$ grayscale image that serves as an ideal input for conventional CNN architectures. For instance, [1] employs this data preprocessing technique combined with transfer learning to accelerate online training process. Their experimental results demonstrate that the CNN-based approach significantly outperforms traditional methods including SVM, random forest, and artificial neural networks.

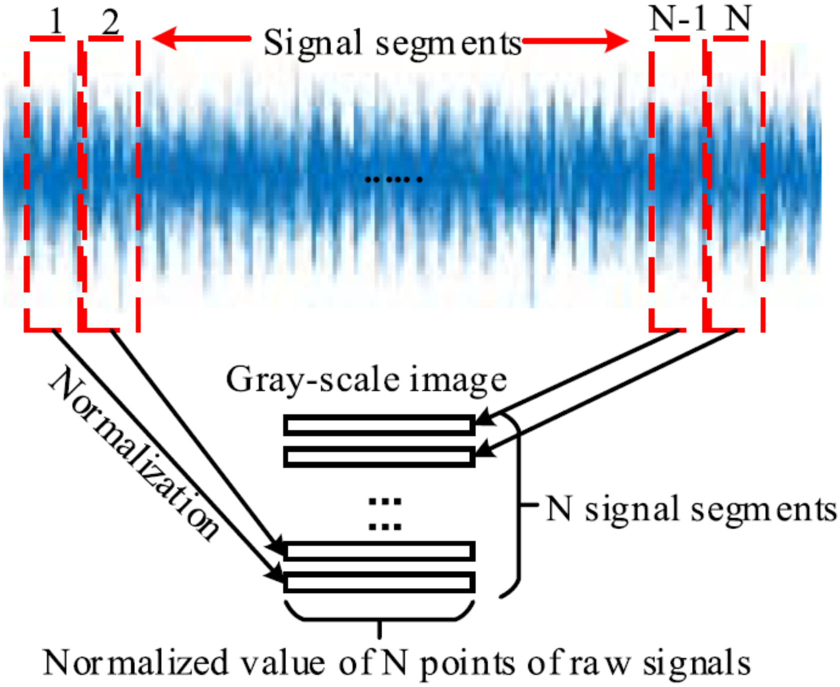


Figure 2.1: Converting 1-D signal to 2-D grayscale image [1]

Similarly, [15] adopts a similar signal-to-image conversion strategy with additional innovations. In their approach, multiple phases of current signals are each processed to train separate CNN models, creating a multi-model ensemble. The outputs from these individual CNNs are subsequently integrated to form an information fusion feature matrix. The final classification decision is then made using traditional machine learning algorithms such as SVM and multilayer perceptron operating on these newly constructed features. This methodology effectively combines information from different sources, resulting in enhanced reliability and robustness of the fault detection system.

Another idea applies advanced signal processing techniques to transform raw signals into time-frequency representations. Unlike the direct signal segmentation methods described above, these approaches utilize well-established signal analysis tools to extract both temporal and spectral information from the original signals. [16] and [2] follow this paradigm. Rather than utilizing spliced images derived directly from raw signals, they employ Short-Time Fourier Transform (STFT) and Wavelet Transform (WT), respectively, to obtain time-frequency images of the signals, which are subsequently provided as input to the CNN networks.

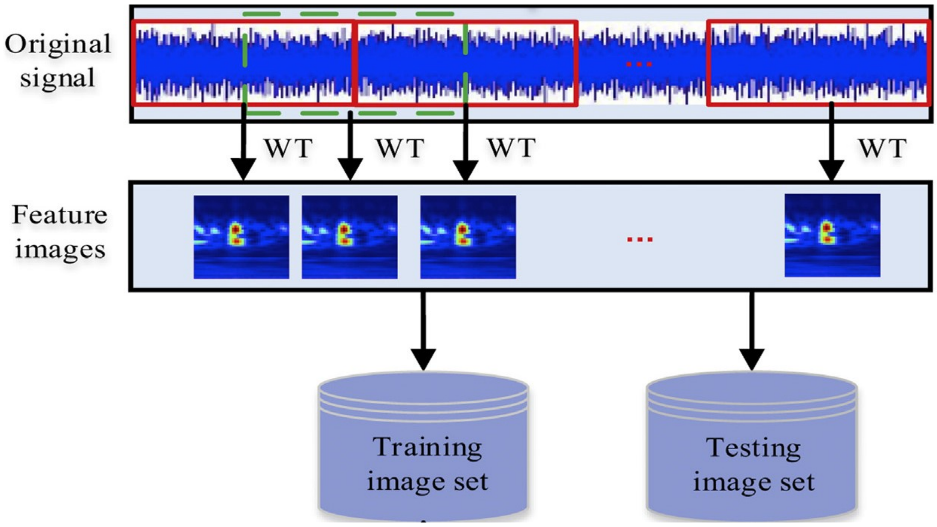


Figure 2.2: Converting 1-D signal to time-frequency image [2]

As illustrated in Figure 2.2, the raw signal is segmented using a sliding window. Each segment is then transformed into a time-frequency representation. This time-frequency representation has been shown to preserve crucial fault-related information that may be lost in purely temporal or frequency domain analysis. This makes it particularly suitable for detecting transient fault signatures that manifest as specific patterns in the time-frequency domain.

2.1.4 Transformer

The Transformer architecture [17] represents a significant advance in sequence modeling by relying entirely on attention mechanisms rather than recurrent or convolutional layers. The main innovation of the Transformer is its self-attention mechanism, which enables the model to learn and weigh the relative importance of different parts of the input sequence. The multi-head attention mechanism can be expressed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.12}$$

where Q, K, and V represent the query, key, and value matrices respectively, and d_k is the dimension of the key vectors. This architecture consists of an encoder-decoder structure with multiple layers, each containing multi-head self-attention and position-wise feed-forward networks, followed by residual connections and layer normalization. The Transformer model is able to capture long-range dependencies and process sequences in parallel, making it highly effective for handling long sequences.

Initially, the Transformer was employed for natural language processing (NLP) tasks, where it demonstrated remarkable accelerated training speed in machine translation when compared to traditional RNN-based models. More recently, with the emergence of Vision Transformer (ViT) [18], the application scope of Transformer architectures has expanded into the computer vision domain. ViT divides images into fixed-size patches and then flattens these patches into sequences as input to the Transformer. The success of ViT demonstrates that the self-attention mechanism can effectively capture spatial relationships and global dependencies in visual data.

In the fault detection domain, Transformers have gained significant attention in recent years, benefiting from their exceptional sequence relationship learning capabilities and fast computation speed. This corresponds well with the fact that industrial signals such as vibration and current are time-series data, as well as the industrial demand for low latency processing. Recent works have successfully applied Transformer-based models to fault detection applications. For example, [19] is one of the earliest studies to utilise transformer in the context of fault detection. The proposed model consists of a multi-head attention mechanism and positional encoding to capture temporal features, and it operates directly on raw one-dimensional sensor signals. Experiments on the CWRU dataset show that the Transformer model achieves higher classification accuracy than CNN, LSTM, and GRU models. In [3], three-directional vibration signals are utilized, and continuous wavelet transform (CWT) is applied to each signal to generate corresponding time-frequency spectrograms, as shown in Figure 2.3. These spectrograms are combined into a three-channel 2D image analogous to an RGB image, which can be directly fed into a ViT model. The experimental results show that the proposed method achieves higher classification accuracy compared to LSTM and CNN models. The author also compared the training time of different models, showing that the proposed method completes training faster than both LSTM and ResNet18 models.

2.2 Unsupervised Learning

In practical applications, it is often not feasible to obtain labeled data. In such cases, unsupervised learning methods are required to focus on learning the normal behavior patterns of the system. These approaches detect faults by identifying deviations between actual values and the values corresponding to the learned normal operating conditions. Techniques such as principal component analysis (PCA), clustering algorithms, and autoencoders fall into this category. Although unsupervised learning methods cannot provide specific fault classifications, they offer the advantage of detecting novel or previously unseen fault conditions without requiring pre-existing labeled datasets.

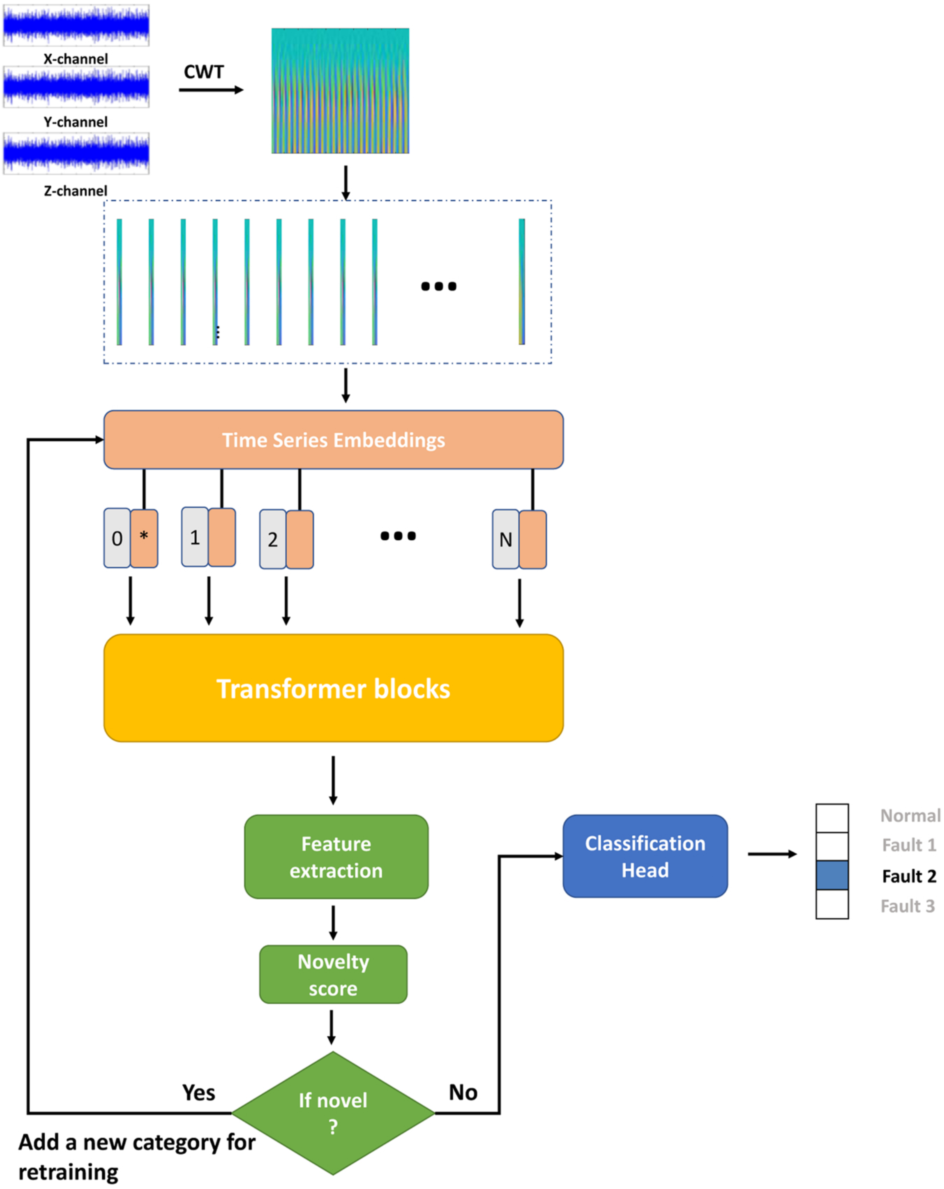


Figure 2.3: Transformer based fault detection architecture [3]

2.2.1 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique that transforms high-dimensional data into a lower-dimensional space while preserving the most important information. The main idea of PCA is to find the directions (principal components) that maximize the variance of the data. The first principal component is the direction that captures the most variance in the data, the second principal component is the direction that captures the most variance in the data that is orthogonal to the first principal component, and so on.

In fault detection applications, two widely used statistics are employed for monitoring purposes: the Q-statistic (also known as the SPE-statistic) and Hotelling's T^2-

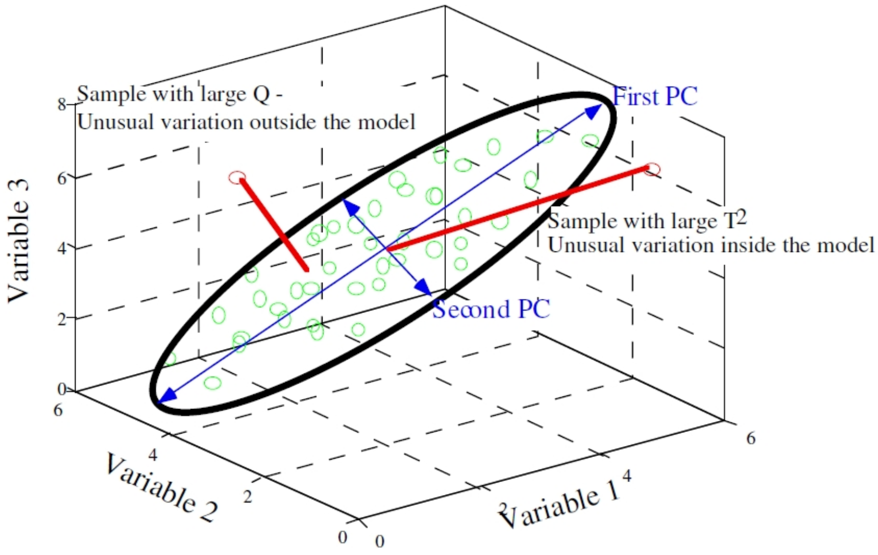


Figure 2.4: PCA model of a three-dimensional data set showing Q and T^2 outliers [4]

statistic (referred to as the D-statistic) [20]. As shown in Figure 2.4, the T^2-statistic captures the variability within the PCA model space, representing deviations in the mean and covariance structure of the retained principal components. In contrast, the Q-statistic measures the residual variation that cannot be explained by the selected principal components, indicating changes in the system behavior that fall outside the normal operating space. Control limits for both statistics can be established based on probability distribution assumptions derived from normal operating conditions, allowing for the detection of abnormal behavior while maintaining an acceptable false alarm rate.

An illustrative example is provided in [21], where Dynamic PCA (DPCA) combined with genetic algorithm (GA) feature selection is used for fault detection. By jointly monitoring the T^2 and Q statistics, the method captures both systematic variations and residual anomalies. Compared to traditional PCA, the DPCA approach notably improves the Q-based detection rate from 76.8% to 83.7%, demonstrating the effectiveness of using these statistics for fault detection.

2.2.2 K-means clustering

The K-means clustering algorithm is an unsupervised learning method that divides data into k clusters by finding the minimum within-cluster sum of squared distances. The algorithm assigns each data point to the nearest cluster center and updates the centers based on the mean of the assigned points in an iterative manner. The objective function to be minimized is:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2 \tag{2.13}$$

where C_i represents the i-th cluster, μ_i is the centroid of cluster C_i, and x denotes a data point. The algorithm converges when the centroids no longer change significantly between iterations or when a maximum number of iterations is reached.

In the field of fault detection, [22] employs a two-stage k-means clustering approach for bearing fault detection using frequency-domain features from vibration signals. The first stage identifies whether a fault exists, and the second classifies it as an inner- or outer-race fault. The proposed K-means clustering method achieves a 100% success rate across all measurements in the three industrial cases as well as the laboratory test case. Furthermore, it demonstrates strong robustness to the selection of initial cluster centers. In [23], vibration data from an exhaust fan was used to evaluate several unsupervised algorithms for early fault detection in predictive maintenance. The author compared different types of clustering methods, including K-Means, Hierarchical Clustering, Fuzzy C-Means, and model-based clustering. Although most algorithms produced similar results, each one provided deeper insight into the data.

2.2.3 Autoencoder

Autoencoder is a type of unsupervised neural network that is trained to reconstruct its input. It is composed of an encoder and a decoder. The encoder maps the input to a latent space, and the decoder maps this back to the input space. The autoencoder is trained to minimize the reconstruction error, typically expressed as:

$$L = \frac{1}{n} \sum_{i=1}^{n} ||x_i - \hat{x}_i||^2 \tag{2.14}$$

where x_i is the input and $\hat{x}_i$ is the reconstructed output.

In fault detection applications, autoencoders are commonly employed to learn the latent features of normal operating conditions. When abnormal data appears during testing, the reconstruction error becomes significantly large, thus serving as an indicator of faults. But the fundamental assumption is that the autoencoder, trained only on normal data, will struggle to accurately reconstruct abnormal patterns, resulting in higher reconstruction errors that can be used for fault detection.

There are several studies that have successfully applied autoencoders to fault detection. In [24], autoencoders are used for fault detection by learning nonlinear representations of normal data and identifying faults through high reconstruction errors. The study shows that autoencoders can detect subtle anomalies that linear PCA methods often miss. More recently, [25] proposed an optimized stacked variational denoising auto-encoder (OSVDAE) for bearing fault detection. This approach integrates the strengths of variational autoencoder (VAE) and denoising autoencoder (DAE) and adopts a seagull optimization algorithm (SOA) to adjust hyperparameters. By introducing noise to the input, the DAE component enhances the model's generalization and robustness, effectively mitigating overfitting. The stacked architecture, consisting of multiple variational denoising auto-encoder (VDAE) layers, enables the extraction of more representative and robust latent features. The SOA further improves the model's accuracy by automating hyperparameter selection, thus eliminating the need for manual tuning. The experimental comparisons showed that the OSVDAE achieves over 99%

fault classification accuracy, outperforming baseline models including VAE, SDAE, and CNN, particularly in noisy environments.

2.3 Hybrid Approaches

While supervised and unsupervised learning methods each offer distinct advantages, they also suffer from inherent limitations when applied independently to fault detection tasks. Supervised methods require extensive labeled data and may struggle with novel fault types, whereas unsupervised methods cannot provide specific fault classification and may have limited detection accuracy. To address these limitations, hybrid approaches that combine the strengths of both methods have become a promising solution for industrial fault detection.

Two typical hybrid approaches in fault detection are semi-supervised learning and pre-training and fine-tuning strategies.

2.3.1 Semi-supervised Learning

Semi-supervised learning offers a promising hybrid solution for fault detection by combining both limited labeled data and abundant unlabeled data. This is especially advantageous in industrial contexts, where acquiring labeled fault data is often costly and time-consuming. In [26], the author proposed a semi-supervised fault diagnosis framework using variational autoencoder (VAE)-based deep generative models for bearing fault detection. The method is particularly effective in scenarios where only a small portion of the data is labeled. In this framework, the VAE encoder and decoder are first trained in an unsupervised manner using all available data. Subsequently, a separate classifier is trained using the limited labeled data in a supervised fashion. Experimental results demonstrate that this approach outperforms conventional supervised and unsupervised methods. The study confirms that semi-supervised VAE-based generative models can significantly enhance classification performance by effectively utilizing the information in large volumes of unlabeled data.

2.3.2 Pre-training and Fine-tuning Strategies

Another effective hybrid approach is a two-stage training process involving unsupervised pre-training followed by supervised fine-tuning. This strategy enables the model to initially learn general feature representations from unlabelled data and then adapt these features for specific fault classification tasks. An example is provided by FaultFormer [27], where a Transformer model is first pretrained using self-supervised learning on unlabeled bearing vibration data to capture general signal characteristics. This is followed by supervised fine-tuning on a small amount of labeled data, which enables the model to classify faults accurately even in conditions where there is a lack of labels. The hybrid strategy significantly improves model adaptability and generalization across datasets.

2.4 Summary

This chapter presents a comprehensive overview of data-driven fault detection techniques, which have emerged as effective tools for industrial fault monitoring by using the large amounts of operational data produced by modern systems. The existing literature generally classifies these methods into three major categories, each with distinct strengths and limitations. Supervised learning methods, including SVM, LSTM, CNN, and Transformer, have shown impressive performance in fault classification when labeled training data is available. These methods can achieve high accuracy and provide detailed fault type identification, but their effectiveness is constrained by the availability and quality of labeled data, which can be expensive and time-consuming to acquire in industrial environments. Unsupervised learning methods, such as PCA, clustering algorithms, and autoencoders, address the challenge of labeled data scarcity by modeling normal behavior patterns and detecting deviations from these patterns. These techniques can identify novel or unseen fault conditions without the need for labeled training data. Nonetheless, they typically lack the ability to provide specific fault categorization and may underperform in terms of accuracy compared to supervised models. Hybrid approaches, including semi-supervised learning and pre-training with fine-tuning strategies, have been proposed as a solution to the current limitations of supervised and unsupervised methods. These hybrid approaches offer enhanced performance by leveraging both labeled and unlabeled data effectively.

The choice of fault detection approach is often guided by the specific requirements of the application, the availability of data, and computational limitations. In real-world scenarios, high-performing fault detection systems commonly integrate multiple methodologies to enhance both accuracy and reliability. Factors such as the availability of labeled data, the need for specific fault classification, the requirement for real-time processing, and the complexity of the industrial system all play crucial roles in determining the optimal approach.

Dataset 3

For data-driven fault detection methods, the quality and characteristics of the dataset play an important role in determining the effectiveness of the developed models. If the quality of the dataset is insufficient, the trained models often fail to achieve satisfactory performance. Additionally, when the dataset is too small, the models tend to lack generalization ability, limiting their practical applicability. Furthermore, if the dataset is collected under laboratory conditions or ideal environments, the resulting models may lack the robustness required for stable operation in industrial environments. Therefore, both the quantity and quality of the dataset significantly influence the final performance of the model.

In this project, two primary datasets are utilized: the Case Western Reserve University (CWRU) bearing vibration dataset and the Johannes Kepler University (JKU) current dataset. These datasets provide complementary perspectives on fault detection, with one focusing on vibration signals and the other on electrical current signatures. The following sections will introduce each of these datasets in detail, including their characteristics, data collection procedures, and relevance to the fault detection tasks addressed in this work.

3.1 CWRU Bearing Dataset

The Case Western Reserve University (CWRU) bearing dataset is one of the most widely used benchmark datasets in the field of bearing fault detection. This dataset provides ball bearing test data for both normal and faulty bearing conditions, making it an invaluable resource for developing and evaluating fault detection models.

As shown in Figure 3.1, the experimental setup consists of a 2 horsepower Reliance Electric motor, a torque transducer/encoder, a dynamometer, and control electronics. Single point faults were introduced to the test bearings with fault diameters of 7 mils, 14 mils, 21 mils. These faults were introduced at three critical bearing locations: the inner raceway, the rolling element (ball), and the outer raceway. Acceleration data is measured at multiple locations both near to and remote from the motor bearings. This multi-point measurement approach allows for detailed analysis of vibration patterns under various fault conditions. The actual test conditions of the motor as well as the bearing fault status are also carefully documented for each experiment, ensuring reliability of the dataset.

The data acquisition was performed at two different sampling rates to accommodate various analysis requirements. The digital data was collected at 12,000 samples per second. For drive end bearing faults, the data was also collected at 48,000 samples per second. The data collection was performed under various operating conditions to ensure dataset diversity. Specifically, vibration data was recorded for motor loads ranging from

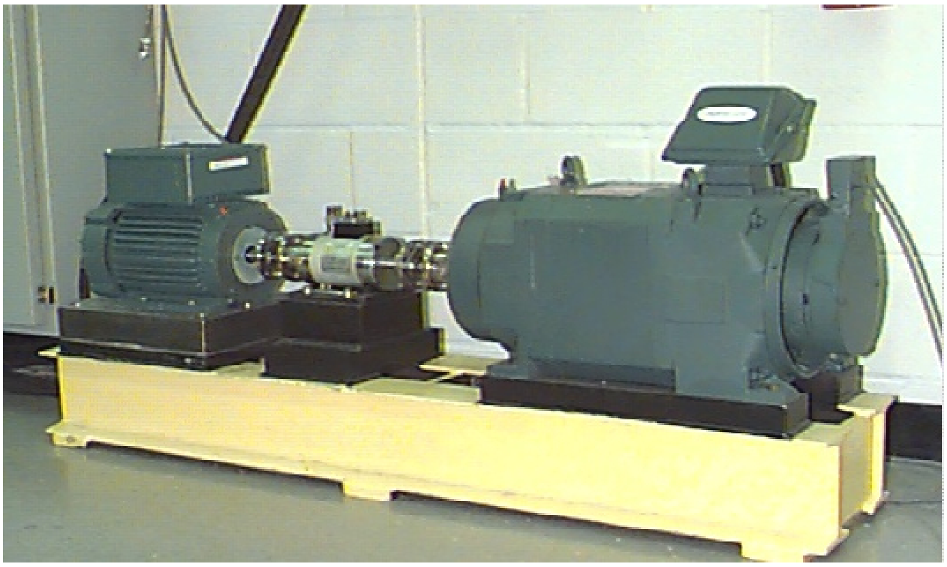


Figure 3.1: Test stand of CWRU dataset

0 to 3 horsepower, corresponding to motor speeds from 1797 to 1720 RPM. Speed and horsepower data were collected using the torque transducer/encoder system to provide complete operational context for each measurement. Some sample vibration signals from the CWRU bearing dataset are shown in Figure 3.2.

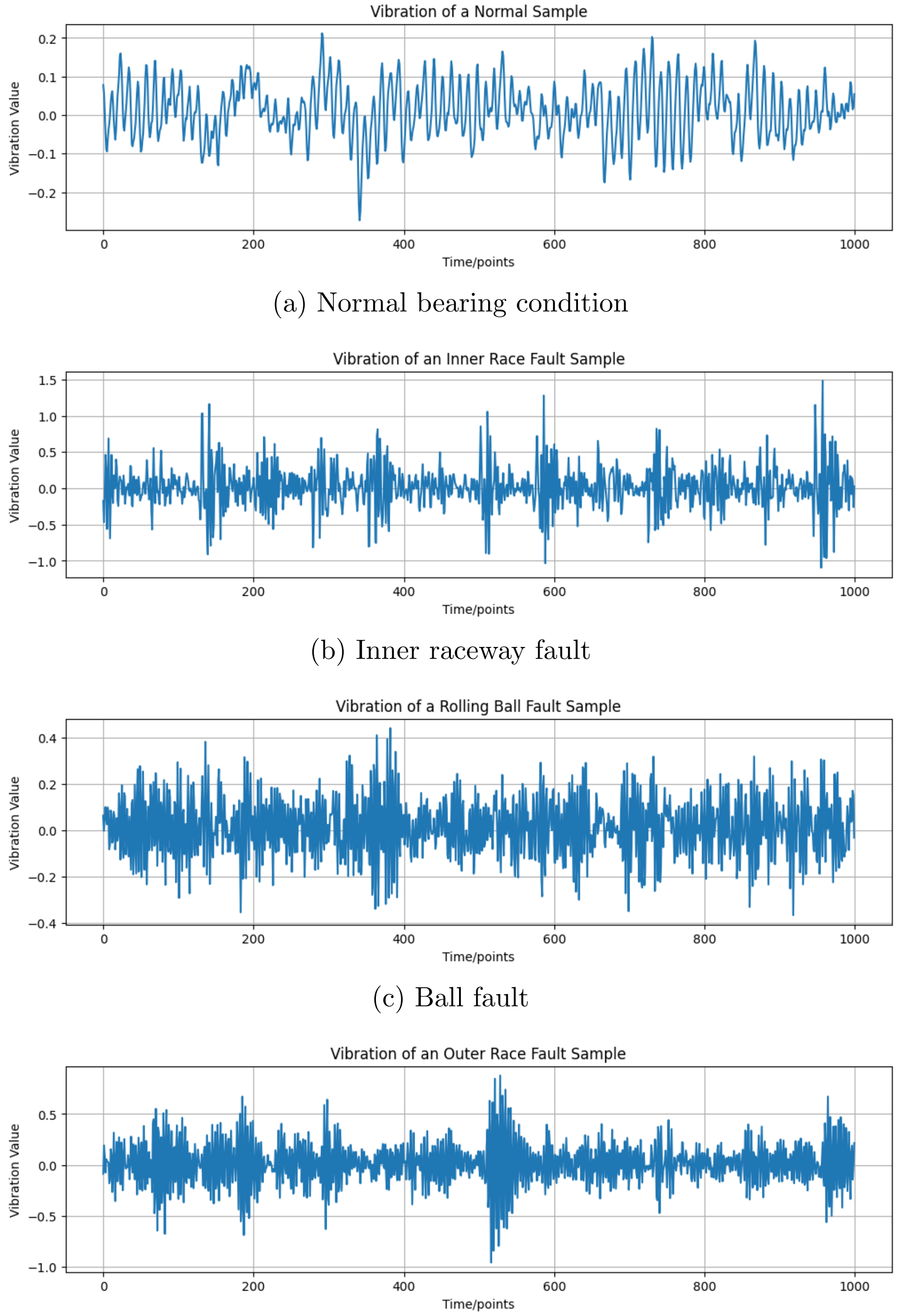
The CWRU bearing dataset has become a standard benchmark in the bearing fault detection community due to its well-documented experimental procedures, diverse fault types, and multiple operating conditions. The dataset's availability has facilitated numerous research studies and enabled fair comparison between different fault detection models.

3.2 JKU Current Dataset

The Johannes Kepler University (JKU) current dataset [28] contains three-phase electrical current measurements from a block-commutated 280W machine under both normal and faulty operating conditions. Unlike the CWRU dataset which focuses on vibration signals, the JKU dataset provides insights into fault detection through electrical current signature analysis.

The investigated fault condition involves misplaced hall sensors that result in commutation angle errors, denoted as φ_Δ. This parameter specifies by how many electrical degrees the rotary angle of the motor axle deviates from its nominal value, which directly affects the motor's operating efficiency. A commutation angle error of $\varphi_\Delta = 0^{\circ el}$ indicates a motor running in its nominal state (healthy condition), while $\varphi_\Delta \neq 0^{\circ el}$ indicates a faulty motor requiring maintenance attention. This type of fault typically occurs when all three hall sensors have the same offset, such as when sensors are mounted on a common carrier with rotational misalignment.

The dataset distinguishes three different types of signals based on their acquisition method and fault severity:



(a) Normal bearing condition

(b) Inner raceway fault

(c) Ball fault

(d) Outer raceway fault

Figure 3.2: Sample vibration signals from CWRU bearing dataset showing different bearing conditions

- **Measurement Nominal:** Real-world measurements of motors in nominal state ($\varphi_\Delta = 0^{\circ el}$), representing baseline healthy operation that is relatively easy to acquire in practice.

- **Measurement Faulty:** Real-world measurements of motors in faulty state ($\varphi_\Delta \in [-10^{\circ el}, 10^{\circ el}]$), which are more costly and challenging to obtain due to the need to intentionally introduce faults or wait for natural fault occurrence.
- **Simulation:** Simulated experiments of motors ($\varphi_\Delta \in [-20^{\circ el}, 20^{\circ el}]$) that are easy to produce and allow for controlled fault severity investigation across a wider range of conditions.

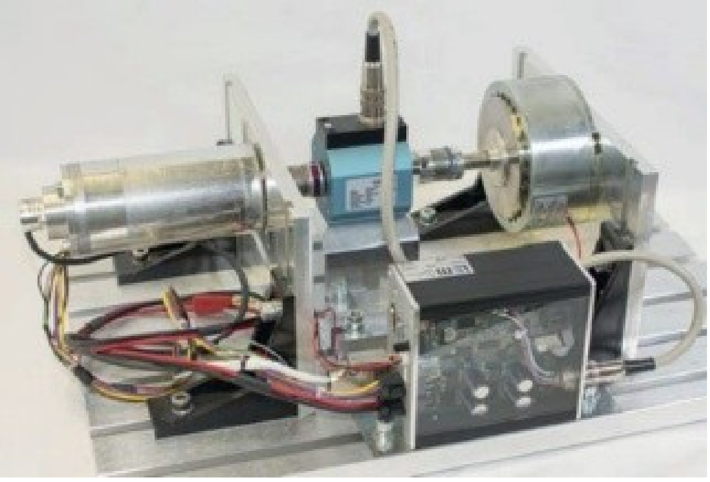
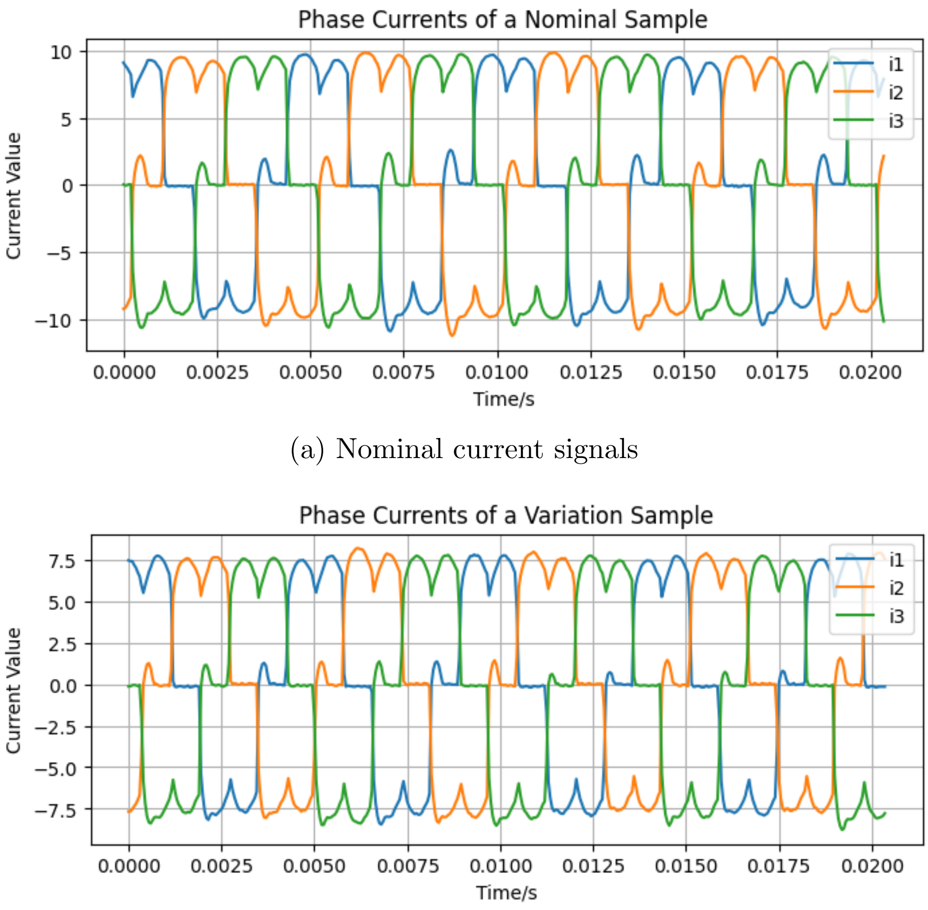


Figure 3.3: Test stand of JKU dataset

The simulation data is generated using the SyMSpace framework, which employs a flux-based surrogate motor model created from finite element simulations. This approach enables transient block-commutation simulation to determine the resulting phase currents under various fault conditions. The measurement setup consists of the block-commutated motor, a torque sensor, a hysteresis brake, and power electronics. In addition to the built-in hall sensors, an incremental encoder is mounted to provide absolute angle signals. Motor control, brake operation, and data acquisition are implemented using the X2C framework, with phase current measurement realized through shunt measurement integrated into the power electronics. The simulation framework provides the advantage of generating large amounts of training data with precise control over fault parameters, while the measurement data ensures real-world validation of the developed models.

The data acquisition follows a systematic approach where both nominal and faulty measurements are sampled with a fixed sample time and a fixed number of samples (408), regardless of motor speed. The simulation experiments vary in sample count according to motor speed and duration, ranging from 301 to 11923 samples. All experiments contain at least one full electrical period to ensure comprehensive signal characterization. Some sample current signals from the JKU dataset are shown in Figure 3.4.

In addition to the electrical current data, the JKU dataset also contains supplementary information including motor speed, torque, and commutation angle measurements. While these data can provide additional information for fault detection and potentially



(a) Nominal current signals

(b) Faulty current signals

Figure 3.4: Sample current signals from JKU dataset showing different motor conditions

improve classification accuracy, they suffer from the acquisition challenges for practical industrial implementation. Particularly, torque data acquisition requires expensive instrumentation and specialized sensors that increase the monitoring system cost. Furthermore, most industrial control systems do not require continuous torque monitoring for normal operation, making the integration of such sensors economically unfeasible for many applications. Therefore, while the availability of this supplementary data enhances the research potential of the dataset, the emphasis on current-based fault detection remains more practically viable for widespread industrial adoption.

3.3 Summary

This chapter introduced two datasets that serve different purposes in the development and validation of data-driven fault detection methods. The CWRU bearing dataset provides vibration-based fault detection data with multiple fault types (inner raceway, ball, and outer raceway faults) across various fault severities and operating conditions. The well-controlled experimental conditions as well as the detailed documentation make it an excellent benchmark for fault detection model development and comparison. In contrast, the JKU current dataset focuses on electrical current signature analysis for detecting commutation angle errors caused by misplaced hall sensors. This dataset incorporates both simulation and real-world measurement data, providing valuable insights into practical fault detection scenarios.

With regard to the volume of data, the CWRU dataset contains a substantially larger amount of data compared to the JKU dataset. This abundance of data makes it especially suitable for training complex machine learning models that require large datasets to achieve optimal performance and generalization capability.

However, the JKU dataset more closely aligned with real industrial applications. The dataset's focus on electrical current analysis also represents a more accessible monitoring approach in many industrial settings, where current sensors are often readily available and easier to implement than vibration monitoring systems.

Based on these considerations, the model development strategy in this work will utilize the CWRU dataset for development, training, and initial validation due to its data coverage and established benchmark status. Subsequently, the developed models will be trained and evaluated on the JKU dataset in order to assess their practical applicability and robustness in real-world industrial conditions. This approach guarantees both rigorous model development and practical validation of the proposed fault detection methods.

Classification-based Fault Detection 4

This chapter will introduce classification-based fault detection methods, which represent a fundamental but powerful approach in the field of operational condition monitoring. As the name suggests, classification-based methods utilize datasets to train models that can provide direct classification predictions for input samples.

The core principle of classification-based fault detection lies in its ability to learn patterns directly from labeled data and transfer these features into fault classification predictions. By training on datasets containing both normal and faulty conditions, these methods can develop models that map input signals to discrete fault categories. This approach enables the development of end-to-end models, where raw signal data can be directly fed into the system to obtain immediate classification results indicating whether the machinery is operating normally or experiencing specific types of faults.

4.1 Baseline Models

Before introducing the proposed model, it is essential to establish several baseline models that will serve as comparative benchmarks. These baseline models are trained under the same experimental conditions to ensure fair and consistent performance comparisons. Their primary role is to provide a solid reference point for assessing the effectiveness of the proposed methodology.

For classification-based fault detection, three representative baseline models are implemented: a one-dimensional Convolutional Neural Network (1D CNN), a Long Short-Term Memory (LSTM) network, and a Transformer model. Each of these architectures reflects a distinct deep learning paradigm and brings unique strengths to the task of sequential data analysis.

The selection of these baseline models is motivated by their success in various classification tasks. The 1D CNN is particularly effective in capturing local temporal patterns through convolutional operations. The LSTM excels at modeling long-term dependencies in sequential data, making it well-suited for tasks with temporal dynamics. The Transformer model, using self-attention mechanisms, enables the extraction of complex, non-local dependencies without the limitations of recurrent architectures.

The following subsections will provide a detailed overview of each baseline model, including their architectural design, implementation details, and the rationale behind the configuration choices.

4.1.1 1D CNN Model

The 1D CNN baseline model is designed as a hierarchical feature extraction network that processes sequential signals through multiple convolutional layers. The architecture consists of modular convolutional blocks followed by fully connected layers for final classification.

The basic building block of this model is the convolutional blocks, which combines a 1D convolutional layer, an activation function, and a pooling operation. Each block performs local feature extraction through convolution, applies non-linear transformation via activation (LeakyReLU), and reduces the temporal dimension through pooling operations. The pooling mechanism not only reduces computational complexity but also provides translation invariance, making the model more robust to small shifts in fault patterns.

The complete 1D CNN model employs a three-layer convolutional architecture with progressively increasing channel dimensions. The network starts with `d_model` channels in the first layer and expands to `2*d_model` channels in the subsequent layers, allowing the model to capture increasingly complex feature representations. Each convolutional block reduces the sequence length by half through pooling, effectively creating a multi-scale feature hierarchy.

Following the convolutional feature extraction, the model applies adaptive average pooling to obtain a fixed-size representation regardless of input sequence length. This global pooling operation aggregates temporal information across the entire sequence. The extracted features are then processed through two fully connected layers with ELU activation, where the first layer performs feature transformation and the second layer produces the final classification logits.

This architecture leverages the strength of CNNs in capturing local temporal patterns while maintaining computational efficiency through pooling operations, making it well-suited for signal classification tasks.

4.1.2 LSTM Model

The LSTM baseline model is designed to capture long-term temporal dependencies in sequential signal data through recurrent neural network architecture. Unlike the CNN model that focuses on local patterns, the LSTM model excels at learning temporal relationships across the entire sequence, making it suitable for analyzing time-series data where historical information significantly influences current classifications.

The core component of this model is a standard LSTM layer implemented using PyTorch's built-in module. Each LSTM cell maintains both hidden state and cell state, enabling the network to selectively remember and forget information across time steps. The model extracts the final hidden state from the last LSTM layer as the sequence representation, which encodes the accumulated temporal information from the entire input sequence. This approach leverages the LSTM's ability to compress sequential information into a fixed-size vector that captures the most relevant temporal patterns for classification.

Following the LSTM feature extraction, the model includes a batch normalization layer that can stabilize training and improve convergence. The extracted temporal features are then processed through a two-layer fully connected network with ELU activation as before.

This architecture is effective for fault detection scenarios where the temporal evolution of signals contains critical diagnostic information. The LSTM's memory mechanism allows it to capture subtle changes in signal patterns over time, which may indicate the onset or progression of mechanical faults that are not immediately apparent in local signal segments.

4.1.3 Transformer Model

The Transformer baseline model applies the self-attention mechanism to capture complex relationships across the entire input sequence without the sequential processing limitations of recurrent networks. This architecture is good at modeling long-range dependencies in time-series data, as it can directly attend to any position in the sequence regardless of distance.

The model begins with a linear embedding layer that projects the input features from their original dimensionality to the transformer's model dimension. Following the successful approach used in vision transformers and BERT, the model employs a learnable classification token that is prepended to the sequence. This special token serves as a global representation that aggregates information from all positions in the sequence through the attention mechanism and is used for subsequent classification. Positional encoding is implemented through learnable parameters rather than fixed sinusoidal functions, allowing the model to adapt the positional representations to the specific characteristics of signals. The positional encoder adds location information to each token, enabling the model to understand the temporal ordering of the input sequence. A dropout layer is applied after positional encoding to prevent overfitting and improve generalization.

The feature extraction of the transformer architecture consists of multiple transformer encoder layers, each containing multi-head self-attention and feed-forward networks. The multi-head attention mechanism allows the model to simultaneously attend to information from different representation subspaces, capturing various types of temporal patterns and relationships. The feed-forward network in each layer provides additional non-linear transformation capability with a hidden dimension that can be configured independently.

After processing through the transformer layers, the model extracts the representation of the classification token. This global representation is then processed through a two-layer fully connected network with ELU activation to produce the final classification output.

This architecture excels at capturing both short-term and long-term temporal dependencies simultaneously, making it highly effective for complex fault detection tasks where different fault signatures may manifest at various time scales within the same signal.

4.2 Initial Proposed Model

The proposed model introduces a novel approach that combines discrete wavelet transform (DWT) preprocessing with transformer-based feature extraction for enhanced fault detection. Unlike traditional methods that process raw time-series data directly, this architecture takes the advantage of the multi-resolution characteristics of wavelet

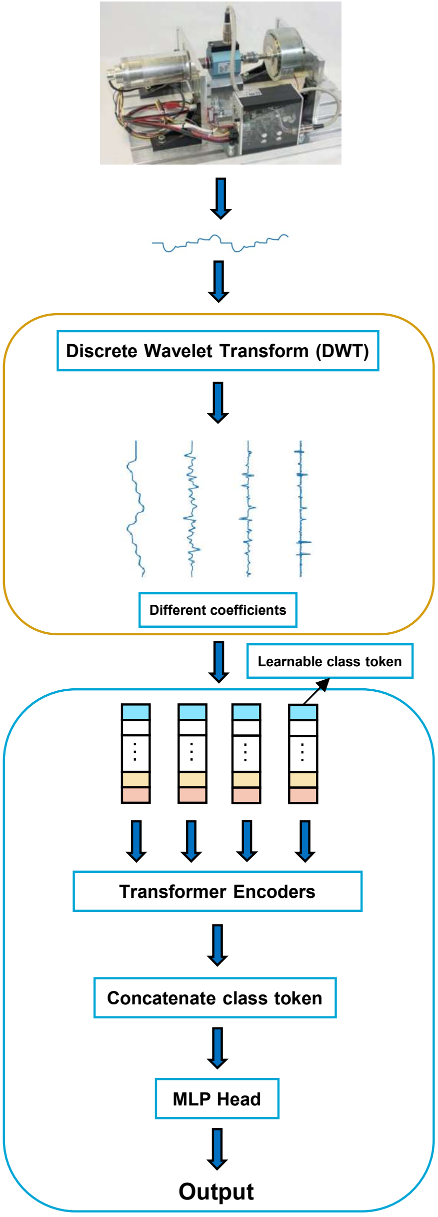


Figure 4.1: Pipeline of the initial proposed model

decomposition to capture both temporal and frequency domain information simultaneously.

As shown in Figure 4.1, the overall architecture consists of three main components: (1) DWT-based data preprocessing that decomposes input signals into multiple coefficient sets at different levels, (2) parallel transformer encoder modules that process each coefficient set independently, and (3) a fusion-based classification head that combines representations from all coefficients for final prediction. This design enables the model to analyze signal characteristics across multiple temporal resolutions, thereby capturing fault signatures that may appear in different frequency bands and time scales. A key innovation of the proposed model lies in its parallel processing of wavelet coefficients obtained through discrete wavelet transform (DWT) using Transformer encoders. This strategy allows the model to learn specialized representations for each decomposition level while preserving the capacity to integrate information across scales, enhancing its effectiveness in fault detection.

4.2.1 DWT-based Data Preprocessing

The discrete wavelet transform serves as the foundation of the proposed preprocessing pipeline, transforming one-dimensional time-series signals into multi-scale representations that capture both temporal and frequency characteristics. To understand the mathematical foundation of DWT, it is essential to begin with the continuous wavelet transform and derive the discrete form through systematic sampling.

The continuous wavelet transform (CWT) of a signal $f(t)$ is defined as:

$$W_f(a,b) = \int_{-\infty}^{+\infty} f(t) \cdot \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) dt \tag{4.1}$$

where a is the scale parameter, b is the translation parameter, and $\psi(t)$ is the mother wavelet. The scaled and translated wavelet function is given by:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) \tag{4.2}$$

To obtain the discrete wavelet transform, the scale and translation parameters are discretized using dyadic sampling:

$$a = 2^j, \quad b = k \cdot 2^j \tag{4.3}$$

where j and k are integers representing the scale and translation indices respectively.

This discretization leads to the discrete wavelet functions:

$$\psi_{j,k}(t) = 2^{j/2} \cdot \psi(2^j t - k) \tag{4.4}$$

However, it is important to note that the wavelet functions $\psi_{j,k}(t)$ alone are not sufficient for sparse decomposition. A set of orthogonal scaling functions $\phi_{j,k}(t)$ are also required in this context. To obtain a complete and exact reconstruction of the signal using an orthonormal wavelet basis, the function $f(t)$ must be decomposed into both approximation and detail components:

$$f(t) = \sum_k \langle f, \phi_{j_0,k} \rangle \cdot \phi_{j_0,k}(t) + \sum_{j=j_0}^{\infty} \sum_k \langle f, \psi_{j,k} \rangle \cdot \psi_{j,k}(t) \tag{4.5}$$

Here, $\phi_{j_0,k}(t)$ represents the scaling functions at a coarse resolution level j_0, which capture the low-frequency approximation of the signal, while $\psi_{j,k}(t)$ are the wavelet functions that capture increasingly finer detail information at higher scales $j > j_0$. The approximation and detail coefficients at scale j are given by the inner products:

$$a_j[k] = \langle f, \phi_{j,k} \rangle = \int_{-\infty}^{\infty} f(t) \cdot \phi_{j,k}(t)\, dt \tag{4.6}$$

$$d_j[k] = \langle f, \psi_{j,k} \rangle = \int_{-\infty}^{\infty} f(t) \cdot \psi_{j,k}(t)\, dt \tag{4.7}$$

In practice, the DWT is implemented using a filter bank approach with low-pass and high-pass filters. This implementation is grounded in Mallat's algorithm [29], which provides a fast and mathematically rigorous method for computing the discrete wavelet transform based on multiresolution analysis (MRA).

The connection between the inner product formulation of wavelet coefficients and their practical computation lies in the structure of the wavelet and scaling functions. Recall that the approximation and detail coefficients at a given scale j are obtained via inner products with the scaling and wavelet functions, as defined in (4.6) and (4.7). In the discrete domain, the functions $\phi_{j,k}(t)$ and $\psi_{j,k}(t)$ are localized, scaled, and shifted versions of the scaling and wavelet functions, and can be interpreted as finite-length convolution kernels.

Therefore, computing the inner products $\langle f, \phi_{j,k} \rangle$ and $\langle f, \psi_{j,k} \rangle$ becomes equivalent to filtering the signal with $h[k]$ and $g[k]$, respectively, followed by a downsampling operation. This equivalence allows the DWT to be implemented efficiently through a cascade of convolution and downsampling steps.

Formally, given a discrete signal $x[n]$, the decomposition at level $j+1$ is computed as:

$$A_{j+1}[n] = \sum_k h[k] \cdot A_j[2n-k] \tag{4.8}$$

$$D_{j+1}[n] = \sum_k g[k] \cdot A_j[2n-k] \tag{4.9}$$

where:

- $A_j[n]$ denotes the approximation coefficients at scale j,
- $D_{j+1}[n]$ denotes the detail coefficients at scale $j+1$,
- $h[k]$ and $g[k]$ are the low-pass and high-pass filter coefficients, respectively.

These filters are derived from the *scaling function* $\phi(t)$ and the *wavelet function* $\psi(t)$, and satisfy the two-scale relations:

$$\phi(t) = \sum_k h[k] \cdot \sqrt{2} \cdot \phi(2t-k) \tag{4.10}$$

$$\psi(t) = \sum_k g[k] \cdot \sqrt{2} \cdot \phi(2t-k) \tag{4.11}$$

This structure guarantees that the filter bank implementation of the DWT is mathematically equivalent to projecting the signal onto an orthonormal wavelet basis, as in

(4.5). Furthermore, the deepest approximation coefficients (i.e., the output after the final low-pass filtering and downsampling step) correspond to the inner products with the coarsest-scale scaling functions $\phi_{j_0,k}(t)$, while the detail coefficients at each level correspond to projections onto wavelet functions $\psi_{j,k}(t)$ at different resolutions. Mallat's algorithm thus enables both fast computation and exact reconstruction, making it a powerful tool for multiscale analysis of time-series signals.

The DWT offers several significant advantages over traditional Fourier transform methods for fault detection applications. First, unlike the Fourier transform which provides frequency information averaged over the entire signal duration, the DWT provides localized frequency analysis with variable time-frequency resolution. This characteristic is particularly valuable for detecting faults that may occur at specific time instances. Second, the multi-resolution nature of DWT enables analysis of both global signal trends and local anomalies, making it a powerful tool for capturing diverse fault signatures that may hide at different scales. The choice of wavelet basis function can be optimized for specific signal characteristics, providing additional flexibility in feature extraction.

4.2.2 Transformer-based Feature Extraction

Following the DWT preprocessing, the proposed model employs parallel transformer encoder modules to process different coefficients independently. This design recognizes that different wavelet coefficients contain distinct types of information and may benefit from specialized feature extraction mechanisms. In particular, the architecture utilizes separate transformer modules for the approximation coefficients (cA) and each level of detail coefficients ($cD_1, cD_2, ..., cD_L$), where L represents the maximum decomposition level. The transformer module is similar to the baseline model, which begins with a linear embedding layer that projects the input coefficients to the model's hidden dimension. A learnable classification token is added to aggregate information and positional embeddings are added to maintain temporal ordering information. A dropout regularization is applied to the input coefficients to prevent overfitting. Each encoder follows the standard encoder architecture with multi-head self-attention and feed-forward layers, but operates on coefficient sequences of different lengths corresponding to the natural dimensionality reduction at each DWT level.

The multi-head self-attention mechanism within each transformer module enables the model to capture complex relationships within each frequency band. This is important for fault detection, as mechanical faults often present as specific patterns or correlations within particular frequency ranges. The parallel processing of different coefficient sets allows the model to learn specialized attention patterns optimized for each scale of analysis.

4.2.3 Feature Fusion and Classification

The final stage of the proposed architecture involves fusing representations from all transformer modules and performing classification. After processing through their respective transformer encoders, the classification tokens from each module

$(cls_{cA}, cls_{cD_1}, ..., cls_{cD_L})$ are concatenated to form a feature vector that encodes information from all frequency bands and temporal scales. The concatenated representation is then processed through a two-layer fully connected network with ELU activation.

Although this fusion approach is just a simple concatenation, it can leverage diverse information from different frequency bands during training. The end-to-end nature of the architecture allows for joint optimization of both the feature extraction and classification components, potentially learning optimal representations for the specific fault detection task.

4.3 Model Improvements

While the initial proposed model demonstrates the potential of combining DWT preprocessing with transformer-based feature extraction, several practical limitations have emerged that prevent its deployment in real-world industrial applications. These challenges are particularly relevant in resource-constrained environments with limited computational capacity, where fault detection systems are frequently deployed.

Firstly, the DWT preprocessing step imposes a considerable computational burden, as it can only be executed on the CPU, thus preventing GPU acceleration and batch processing. This sequential operation introduces a processing bottleneck that significantly reduces model throughput and real-time responsiveness. Secondly, the use of multiple independent transformer encoders for different coefficient sets results in a dramatic increase in the number of model parameters, leading to higher memory requirements and storage costs. This parameter explosion poses a critical challenge for deployment on resource-constrained edge devices.

In addition, the traditional scaled dot-product attention mechanism exhibits quadratic time complexity with respect to sequence length, making it computationally expensive for longer input sequences. These challenges highlight the need for architectural improvement to reduce computational and memory overhead while preserving the model's strong performance in fault detection.

To address these limitations, improvements have been implemented at three key levels: (1) data preprocessing optimization to enable GPU acceleration and parallel computation, (2) encoder architecture refinement to reduce parameter count while preserving feature extraction capabilities, and (3) attention mechanism enhancement to achieve linear computational complexity for improved scalability.

4.3.1 Improvement On The Data Preprocessing

The computational bottleneck in the initial model's DWT preprocessing can be addressed by approximating the wavelet decomposition process using convolutional operations. The fundamental principle behind this approach lies in the mathematical similarity between DWT filtering operations and convolutional computations.

For discrete signals, the DWT coefficients are computed through recursive application of analysis filters. Starting from the original signal $f[n]$, the decomposition at level

j is obtained by:

$$cA_j[k] = \sum_n cA_{j-1}[n] \cdot h[n-2k] \tag{4.12}$$

$$cD_j[k] = \sum_n cA_{j-1}[n] \cdot g[n-2k] \tag{4.13}$$

where $cA_0[n] = f[n]$ represents the original signal.

By changing the summation index and applying the properties of discrete convolution, these equations can be reformulated as filtering operations followed by downsampling:

$$cA_j[k] = \sum_n h[n] \cdot cA_{j-1}[2k-n] = (h * cA_{j-1})[2k] \tag{4.14}$$

$$cD_j[k] = \sum_n g[n] \cdot cA_{j-1}[2k-n] = (g * cA_{j-1})[2k] \tag{4.15}$$

where $h[n]$ and $g[n]$ are the low-pass and high-pass analysis filters, and $*$ denotes convolution. This mathematical equivalence demonstrates that DWT decomposition can be implemented as convolution operations followed by downsampling, suggesting that the process can be approximated using 1D convolutional layers with fixed kernel parameters corresponding to the wavelet basis functions.

By replacing the traditional DWT preprocessing with fixed-parameter 1D convolutional layers, several significant advantages are achieved. First, the convolutional operations can be executed on GPU with full batch processing capabilities, dramatically improving computational throughput. Second, the parallel processing nature of convolutions eliminates the sequential dependency inherent in traditional DWT computation, enabling more efficient utilization of modern hardware architectures.

However, this approximation approach also introduces certain limitations. While the mathematical foundation is sound, achieving exact equivalence to traditional DWT requires precise configuration of convolutional parameters including padding, stride, and kernel initialization. In practice, the convolutional approximation serves as a close approximation rather than an exact replacement, potentially introducing minor numerical differences that may affect the decomposition quality.

4.3.2 Improvement On The Multiple Encoder Layers

The parameter explosion problem caused by multiple independent transformer encoders demands architectural modifications to reduce model complexity while maintaining feature extraction effectiveness. Two distinct approaches have been developed to address this challenge.

The first approach employs a unified transformer architecture with masked self-attention to process concatenated coefficient sequences. In this design, coefficients from different decomposition levels are concatenated in hierarchical order, and a carefully designed attention mask ensures that each coefficient can only attend to coefficients from the same or lower decomposition levels. Specifically, detail coefficients cD_j can compute attention with cD_i where $i \leq j$ and with themselves, while approximation coefficients cA can attend to all coefficient types. This hierarchical attention pattern respects the natural structure of wavelet decomposition while enabling cross-scale information exchange.

Mathematically, the masked attention can be expressed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T + M}{\sqrt{d_k}}\right) V \tag{4.16}$$

where M is the mask matrix with $M_{ij} = -\infty$ for prohibited attention connections and $M_{ij} = 0$ for allowed connections.

The second approach utilizes the structural properties of DWT decomposition by upsampling higher-level coefficients to match the length of lower-level coefficients, enabling the use of a single transformer encoder for all coefficient types. This method reduces the model parameter count significantly as only one transformer module is required regardless of the number of decomposition levels.

The first approach offers the advantage of enabling explicit cross-scale attention computation, helping the model to capture relationships between different frequency bands. However, the introduction of masked attention may increase computational cost due to the increase of the sequence length. The second approach achieves substantial parameter reduction and computational simplification, but requires careful design of the upsampling strategy to ensure proper alignment of information across different frequency scales.

4.3.3 Improvement On The Attention Mechanism

The quadratic computational complexity of traditional scaled dot-product attention poses significant challenges for processing long sequences. To mitigate this limitation, linear attention mechanisms [30] have been introduced, reducing the computational complexity of the attention operation from $O(L^2)$ to $O(L)$, where L denotes the sequence length.

Linear attention achieves this complexity reduction by reformulating the attention computation through kernel-based approximation. The derivation begins with the standard scaled dot-product attention mechanism:

For input $x \in \mathbb{R}^{L \times d}$, the query matrix Q, key matrix K, and value matrix V are computed as:

$$Q = xW_Q, \quad K = xW_K, \quad V = xW_V \tag{4.17}$$

where $Q, K, V \in \mathbb{R}^{L \times d}$. The standard attention output is:

$$A(x) = V' = \text{softmax}\left(\frac{QK^T}{\sqrt{D}}\right) V \tag{4.18}$$

This can be generalized by replacing the softmax operation with a similarity function $\text{sim}(Q_i, K_j)$:

$$V_i' = \frac{\sum_{j=1}^{N} \text{sim}(Q_i, K_j) V_j}{\sum_{j=1}^{N} \text{sim}(Q_i, K_j)} \tag{4.19}$$

The key insight is to replace the similarity function with a non-negative kernel function that can be decomposed using feature representations. By setting $\mathrm{sim}(Q_i, K_j) = \phi(Q_i)^T\phi(K_j)$, the attention becomes:

$$V_i' = \frac{\sum_{j=1}^{N} \phi(Q_i)^T\phi(K_j)V_j}{\sum_{j=1}^{N} \phi(Q_i)^T\phi(K_j)} \tag{4.20}$$

The computational complexity reduction is achieved by exploiting the associative property of matrix multiplication to reorder the operations:

$$V_i' = \frac{\phi(Q_i)^T \sum_{j=1}^{N} \phi(K_j)V_j^T}{\phi(Q_i)^T \sum_{j=1}^{N} \phi(K_j)} \tag{4.21}$$

This reformulation enables the computation to be reordered as $\phi(Q)\cdot(\phi(K)^T \cdot V)$ instead of $(\phi(Q)\cdot\phi(K)^T)\cdot V$, where $\phi(\cdot)$ is a feature mapping function that projects queries and keys to a higher-dimensional space. The terms $\sum_{j=1}^{N} \phi(K_j)V_j^T$ and $\sum_{j=1}^{N} \phi(K_j)$ can be precomputed and reused for all queries, reducing the complexity from $O(L^2)$ to $O(L)$.

The primary advantage of linear attention is the dramatic reduction in both time and space complexity, making it feasible to process much longer sequences with limited computational resources. This improvement is particularly valuable for industrial applications where real-time processing of extended signal sequences is required.

However, linear attention mechanisms have inherent limitations that restrict their applicability. Most notably, linear attention cannot effectively incorporate causal masks or complex attention patterns due to the mathematical constraints of the kernel approximation. This limitation prevents the use of linear attention in the masked attention approach described in the encoder improvements, restricting its application to the upsampling-based architecture variant.

Despite this limitation, the computational benefits of linear attention make it an attractive option for scenarios where the trade-off between computational efficiency and model expressiveness is acceptable, especially in resource-constrained deployment environments.

4.4 Final Model Architecture

The final model architecture integrates all the improvements discussed in the previous sections, resulting in a computationally efficient and deployable solution for fault detection while maintaining the multi-scale analysis capabilities of the original approach. Figure 4.2 illustrates the complete pipeline of the improved model architecture.

The pipeline of the final model follows a streamlined approach that addresses the computational bottlenecks identified in the initial design. Starting with raw signals, the model performs the following sequence of operations:

Step 1: Convolutional Wavelet Decomposition The input time-series signal $x[n]$ is processed through a series of fixed-parameter 1D convolutional layers that approximate the DWT decomposition. Instead of CPU-based DWT computation, the model employs convolutional layers with kernels initialized to correspond to wavelet

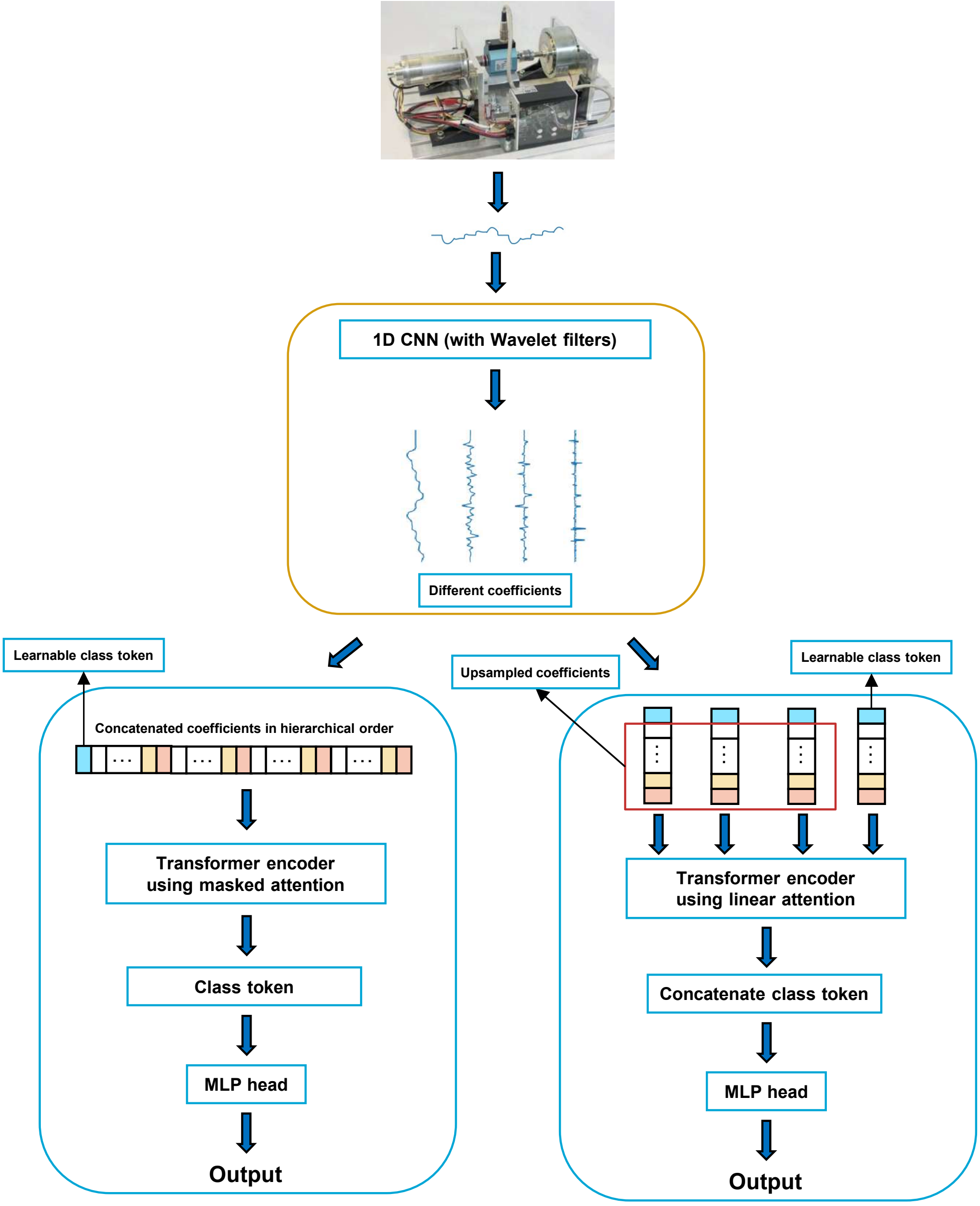

1D CNN (with Wavelet filters)
Different coefficients
Learnable class token
Concatenated coefficients in hierarchical order
Transformer encoder using masked attention
Class token
MLP head
Output
Upsampled coefficients
Learnable class token
Transformer encoder using linear attention
Concatenate class token
MLP head
Output

Figure 4.2: Pipeline of the final improved model with two variants

filter coefficients. This approach enables GPU acceleration and batch processing, significantly reducing computational cost. The decomposition produces approximation coefficients cA and multiple levels of detail coefficients $cD_1, cD_2, ..., cD_L$ at different scales.

Step 2: Coefficient Processing Depending on the specific architectural variant, the model processes the wavelet coefficients using one of the following two approaches:

- *Masked Attention Approach*: All coefficient sequences are concatenated in hierarchical order and processed by a single transformer encoder with carefully designed attention masks that respect the wavelet decomposition structure.

- *Upsampling Approach*: Higher-level coefficients are upsampled to match the length of the lowest-scale coefficients, enabling data processing through a single transformer encoder with linear attention mechanisms.

Step 3: Feature Extraction The transformer encoder processes the coefficient sequences using either standard scaled dot-product attention (for the masked approach) or linear attention mechanisms (for the upsampling approach). The encoder captures intra-scale relationships within the multi-resolution signal representation. The use of learnable positional embeddings and classification tokens enables the model to aggregate global information from the entire coefficient space.

Step 4: Classification The final classification stage uses the global representation from the transformer output and processes it through a two-layer fully connected network with ELU activation. For the upsampling approach, the classification token is concatenated for the final classification. For the masked approach, the class token is directly used.

The key advantages of the final architecture include: (1) GPU-accelerated preprocessing through convolutional approximation of DWT, (2) reduced parameter count through unified transformer architecture, (3) linear computational complexity for long sequences through efficient attention mechanisms, and (4) maintained multi-scale analysis capabilities for comprehensive fault detection. This design achieves a balance between computational efficiency and model performance, making it suitable for deployment in resource-constrained industrial environments while preserving the detection capabilities of the original multi-scale approach.

4.5 Summary

This chapter presented a detailed exploration of classification-based fault detection methods, progressing from baseline approaches to a novel multi-scale architecture and its improvements for practical deployment.

The chapter began with the establishment of three baseline models that represent different directions in deep learning for sequential data processing. The 1D CNN model demonstrated the effectiveness of hierarchical feature extraction through local convolutional operations. The LSTM model showcased the power of recurrent architectures in learning long-term temporal dependencies. The Transformer model illustrated the capabilities of attention mechanisms in capturing complex relationships across entire sequences without the sequential processing limitations of recurrent networks. These baseline models provided important reference points for evaluating the proposed methodology.

The initial proposed model introduced a novel approach that combines discrete wavelet transform preprocessing with parallel transformer-based feature extraction. The mathematical foundation was established from continuous wavelet transform to discrete implementation via filter banks, demonstrating how multi-resolution analysis enables simultaneous capture of temporal and frequency domain information. This architecture employed separate transformer encoders for different wavelet coefficient types, followed by feature concatenation for final classification. While this approach showed promise in using multi-scale signal characteristics, several practical limitations emerged regarding computational efficiency and deployment feasibility.

To address these limitations, several improvements were implemented at three critical levels. First, the data preprocessing bottleneck was resolved by approximating DWT operations through convolutional layers with fixed wavelet-based kernels, enabling parallel batch processing while maintaining mathematical equivalence. Second, the parameter explosion problem was tackled through two architectural variants: a transformer with masked attention that follows wavelet decomposition hierarchy, and an upsampling-based approach that processes all coefficients through a single encoder. Third, the attention mechanism was replaced with linear attention formulations that reduce computational complexity from quadratic to linear with respect to sequence length, making the approach scalable for longer signal sequences.

The final model architecture integrates all improvements into a system that maintains the multi-scale analysis capabilities of the original approach while achieving significant computational efficiency gains. The resulting architecture is ideal for implementation in resource-constrained industrial environments to offer real-time fault detection.

The classification-based approach presented in this chapter provides a solid foundation for end-to-end fault detection systems that can directly map raw sensor data to classification decisions, serving as a significant research contribution to the field of automated condition monitoring in mechanical and electrical systems.

5 Forecasting-based Fault Detection

While most existing models perform fault detection through direct classification, an alternative approach that leverages time series forecasting to enhance detection performance is proposed in this chapter. This methodology represents a significant difference from conventional classification-based methods, drawing inspiration from the remarkable success of state-of-the-art time series forecasting models in learning complex temporal dependencies and capturing complex patterns in sequential data.

The core idea of this approach is to transform the fault detection problem from a direct classification task into a forecasting-based inference mechanism. Instead of training a model to directly discriminate between normal and faulty states, it employs an unsupervised (self-supervised) learning strategy, where a forecasting model is trained solely on data from normal operating conditions. The forecasting error is then used as an indicator for state classification. This method offers distinct advantages in scenarios where the temporal patterns of signals differ between normal and faulty operating conditions.

5.1 Related Forecasting Models

Recent advances in time series forecasting have produced several state-of-the-art models that achieve outstanding performance in modeling long-term dependencies and predicting subsequent values in sequential data. These models have enriched the field of time series analysis and provide the basis for the forecasting-based fault detection approach proposed in this chapter. Here are some of the most notable models:

Informer [31] introduces a novel ProbSparse self-attention mechanism to address the quadratic time and memory complexity of standard Transformers in long sequence time-series forecasting. By selecting only the top-u dominant queries based on sparsity measurement, it reduces the attention complexity to $O(L \log L)$ while preserving long-range dependency modeling. In addition, Informer introduces a self-attention distilling operation that hierarchically compresses the input sequence length across layers, further enhancing efficiency and scalability for long-term forecasting tasks.

Autoformer [32] proposes a novel decomposition architecture that explicitly separates time series into trend and seasonal components within the Transformer framework. The model introduces an Auto-Correlation mechanism that identifies and aggregates similar sub-series based on periodic dependencies, enabling more accurate modeling of long-term and repeating patterns. Its progressive decomposition across layers allows the model to capture complex temporal dynamics with multiple seasonalities.

FEDformer [33] introduces a dual-domain forecasting framework that combines frequency-domain attention with time-domain decomposition. It employs Fourier transforms to extract dominant periodic patterns in the frequency domain while modeling long-term trends in the time domain. This design enables effective learning of both global periodic structures and local variations, making the model especially relevant for complex long-term forecasting tasks.

PatchTST [34] proposes a patch-based representation of time series inspired by Vision Transformers, where input sequences are segmented into fixed-length patches. This strategy preserves local temporal structures while reducing input length, leading to better efficiency. Channel-independent Transformers are used to model each variable separately, enabling the model to generalize well across diverse multivariate time series forecasting tasks.

These models have been shown to succeed in handling complex temporal dependencies, capturing long-range relationships, modeling complex patterns, and giving accurate forecasts in time series data. This proves their potential for applications beyond traditional forecasting problems, making them ideal candidates for fault detection applications.

5.2 Proposed Model

To integrate forecasting capabilities into fault detection, a new framework that uses the strength of advanced forecasting architectures is proposed. This approach transforms fault detection from a direct classification task into a forecasting error comparison problem.

5.2.1 Framework Overview

Similar to the idea of autoencoders, the proposed framework is trained using only normal samples from the dataset, with the objective of learning the characteristics of normal operating conditions and predicting future measurements. During the inference phase, test samples are fed into the model to generate future predictions. The mean squared error (MSE) between the predicted and actual future values is then calculated and compared against a predefined threshold. If the error is below the threshold, the sample is classified as normal; otherwise, it is classified as faulty. This approach assumes that the model is capable of achieving higher prediction accuracy on samples that exhibit temporal patterns similar to those observed in the normal data.

Mathematically, for a test sample x_t with corresponding future sequence y_t, the classification decision is formulated as:

$$\hat{y} = f(x_t) \tag{5.1}$$

$$\text{MSE} = \frac{1}{L}\sum_{i=1}^{L}(y_{t,i} - \hat{y}_i)^2 \tag{5.2}$$

$$\text{Class} = \begin{cases} \text{Normal} & \text{if MSE} < \text{threshold} \\ \text{Fault} & \text{otherwise} \end{cases} \tag{5.3}$$

where f represents the forecasting model, L is the prediction horizon length, and $\hat{y}$ denotes the predicted future sequence.

5.2.2 PatchTST Implementation

In the implementation of this framework, PatchTST is selected as the backbone forecasting architecture due to its outstanding performance in time series forecasting. The PatchTST architecture processes raw time series signals by dividing them into non-overlapping patches of fixed length. Each patch is treated as a token and linearly embedded into a high-dimensional representation space. The model employs a standard transformer encoder to capture dependencies between patches, followed by a linear projection layer that generates predictions for future time steps.

For fault detection applications, the input is historical signal segments of length T, which are divided into patches of size P. The model learns to predict the subsequent H time steps, where H represents the forecasting horizon. The forecasting value is predicted in a non-autoregressive manner, meaning that the output is generated at once.

The training process follows the standard time series forecasting practice. Historical signal segments serve as input features, while future segments constitute the prediction targets. The models are optimized using mean squared error loss, encouraging accurate prediction of subsequent signal values.

5.3 Advantages and Limitations

As a forecasting-based approach falls under the category of unsupervised learning, it shares both the advantages and challenges common to other unsupervised methods. The success of this approach also relies on several key assumptions, which must be carefully considered when applying it to real-world fault detection scenarios.

5.3.1 Advantages

Like most unsupervised learning methods, one of the key advantages of forecasting-based fault detection is that it does not require a large amount of accurately labeled data. Unlike supervised approaches, where model training depends on the availability of labeled samples, this method enables the model to learn features of the data without the need for labels. Specifically, in the self-supervised learning framework employed here, the model learns to predict future values based on historical patterns in the data, and the prediction error is used as an indicator for classification. By eliminating the reliance on class labels and utilizing the data itself along with the prediction error for decision-making, this approach becomes particularly well-suited for industrial applications, where labeled data are often costly or difficult to obtain. The ability to perform fault detection using only the collected data makes this method a highly attractive solution in such practical settings.

Another notable advantage of this approach is its inherent scalability to multi-class problems, particularly in scenarios involving previously unseen fault types. Traditional classification models are typically limited to distinguishing among the categories encountered during training. When new fault types emerge, the model lacks prior knowledge of their characteristics and cannot classify them accurately. From an implementation perspective, the classification head is usually fixed to a predefined number of classes, meaning the appearance of new classes often necessitates retraining the model. In contrast, the proposed forecasting-based model is largely unaffected by the emergence of new fault types, as long as their features differ sufficiently from those of the normal data. For a well-trained model, if a newly encountered sample yields a prediction error exceeding the predefined threshold, it can still be classified as a fault without requiring additional retraining. This property makes the method particularly suitable for real-world scenarios where fault categories are uncertain or subject to change over time.

5.3.2 Limitations and Challenges

The fundamental assumption of this approach is that normal and faulty samples can be distinguished based on their patterns and features. This assumption is critical, as the classification mechanism relies on comparing the prediction error to a predefined threshold. If the temporal dynamics of normal and faulty states are similar, the prediction error may not provide sufficient discriminative power for reliable classification. Therefore, the method assumes that normal operating conditions exhibit stable and predictable temporal patterns that can be effectively modeled using prediction techniques, while faulty states introduce significant deviations from these patterns, which occur as increased prediction errors.

At the same time, forecasting-based methods are also sensitive to distribution shifts between datasets collected under different operating conditions. This implies that the method may produce false alarms when encountering normal data that deviates from the training distribution. Environmental changes, load variations, or external disturbances that significantly alter the signal characteristics may lead to misclassifications, even when the system remains in a healthy state. The choice of prediction horizon H also significantly affects performance. Short horizons may not capture sufficient information for discrimination, while long horizons may introduce prediction uncertainty that degrades classification accuracy.

In addition to the aforementioned limitations, the proposed method also suffers from several challenges associated with unsupervised learning. For instance, although it does not require accurately labeled data, training the model still demands a large volume of historical data to effectively learn the patterns. Furthermore, despite its improved robustness to previously unseen fault types, the model is limited to binary classification and is therefore not suitable for applications where precise identification of fault categories is required. Finally, in terms of classification accuracy, unsupervised approaches typically underperform compared to supervised classification models.

Despite these limitations, the forecasting-based approach offers unique perspectives on fault detection that can complement traditional classification approaches or serve as a standalone solution in appropriate applications.

5.4 Summary

This chapter introduced a novel forecasting-based approach for fault detection that is different from conventional direct classification methodologies. The approach applies the advanced temporal modeling capabilities of state-of-the-art time series forecasting models to enable fault detection through forecasting accuracy comparison.

Firstly, some recent advances in time series forecasting are reviewed. Highlight models such as Informer, Autoformer, FEDformer, and PatchTST have demonstrated remarkable performance in capturing complex temporal dependencies. These models provide the technical implementation for the proposed forecasting-based fault detection framework.

The core method involves training a forecasting model on normal data, then using forecasting accuracy as a discriminative feature for classification. PatchTST was selected as the backbone architecture due to its excellent performance in patch-based temporal modeling and computational efficiency. The framework transforms fault detection from a direct classification problem into a forecasting error comparison task.

Like most unsupervised learning models, the forecasting-based approach is more robust to unseen classes and does not rely on the availability of labeled data. However, it also faces limitations such as its dependence on the learnability of feature patterns, lower classification accuracy compared to direct classification models, and its inherent restriction to binary classification tasks.

Despite its limitations, the forecasting-based approach offers a valuable alternative perspective for fault detection. Its effectiveness largely depends on the assumption that there exist sufficient differences in the temporal patterns between normal and faulty states. When this assumption holds, the forecasting-based method is expected to provide fault detection capabilities comparable to those of traditional classification-based approaches.

6 Experiments and Results

To evaluate the effectiveness and robustness of the proposed classification-based and forecasting-based fault detection models, a series of experiments using both a project-specific dataset (JKU) and a publicly available benchmark dataset (CWRU) were conducted. This chapter presents the experimental details and key results obtained from both modeling approaches. The experiments are designed to first validate model performance in the target application domain, followed by broader comparisons on a benchmark dataset to assess generalization ability and comparability with prior work.

6.1 Experiments

Prior to the presentation and discussion of the results, it is important to clarify the experimental details. This section presents the experimental setup, dataset preprocessing, evaluation strategy, and metrics used in the experiments.

6.1.1 Experimental Setup

All experiments were carried out on a computer equipped with an NVIDIA RTX 3060 GPU (6GB RAM). To ensure a fair comparison between models, consistent training hyperparameters were used across all experiments, except where model-specific or dataset-specific configurations were required. The number of training epochs was kept constant throughout all experiments to maintain comparability.

For the classification-based approach, the JKU dataset was first investigated, as it is closely related to the target application scenario of this project. This dataset enables direct evaluation of the model's effectiveness in the specific domain of interest. Subsequently, we extended the training and evaluation to the CWRU dataset, which is a widely used benchmark in fault detection research. Using this benchmark allows for comprehensive comparisons with existing works and also serves as a means to assess the generalization capability of the proposed model. During training, the cross-entropy loss function was employed to optimize classification performance, and the Adam optimizer was used to ensure stable and efficient convergence.

In contrast, the forecasting-based approach was evaluated only on the CWRU dataset. This choice was made because the CWRU dataset contains a variety of fault types, including categories that can be treated as previously unseen, making it well-suited for assessing the generalization capability of the model. Such a data structure provides an ideal environment for evaluating the performance and robustness of forecasting-based fault detection methods. For this approach, the mean squared error (MSE) was used as the loss function, and the Adam optimizer was employed for model training.

6.1.2 Dataset Preprocessing

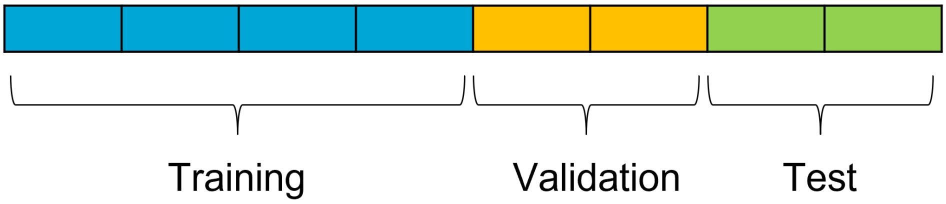


Figure 6.1: Segmentation of the CWRU dataset

For the CWRU dataset, each data file contains signals recorded under a specific fault type and load condition. Since each file includes continuous data, segmentation was necessary. To avoid data leakage and ensure rigorous evaluation, each file was sequentially divided into training, validation, and testing subsets according to a fixed ratio, as shown in Figure 6.1. Each segment was created using a window length of 2048 and a stride of 2048, ensuring no overlap between adjacent samples.

In contrast, each sample in the JKU dataset is of fixed length (408) and represents an independent observation. Thus, no segmentation was required during preprocessing.

All datasets were standardized during preprocessing. This normalization step ensures that the input features fall within a similar range, which facilitates faster convergence during training and prevents the model from being biased toward input dimensions with larger absolute values. Additionally, for the forecasting-based approach, the datasets were normalized to the range of $[-1, 1]$ to ensure that the model focuses on the pattern of the input signals rather than the magnitude.

6.1.3 Evaluation Strategy

To account for the randomness in machine learning training processes, such as weight initialization, batch shuffling, dropout layers, and optimization dynamics, specific evaluation strategies aimed at reducing variance and improving the reliability of experimental results were adopted.

For experiments based on the JKU dataset, k-fold cross-validation was used. In this strategy, the dataset is divided into k subsets (folds) with equal size. In each experiment, one fold is chosen as the test set. The remaining k-1 folds are used for training. This process is repeated k times to ensure that every sample in the dataset is used exactly once as a test sample. The final performance metrics are obtained by averaging the results across all folds, which helps to mitigate overfitting and yields a more stable evaluation of the model's generalization ability.

However, for experiments involving the CWRU dataset, cross-validation is not applicable due to the time-dependent structure of the data segmentation. Each file in the CWRU dataset represents a continuous time-series signal under specific working conditions, and splitting this data across folds could violate temporal consistency, potentially resulting in data leakage between training and testing sets. To address this, the instead solution was to perform multiple independent runs of the training and evaluation process and reported the average results. This approach helps smooth out fluctuations caused by random initializations and provides a fair estimate of model performance on temporally structured data.

For the prediction-based approach, the evaluation procedure differs from that of conventional classification methods. The dataset is first divided into three parts: the training set contains only normal samples and is used to train the prediction model, while the validation and test sets include both normal and abnormal samples. The trained model is first applied to the validation set to conduct receiver operating characteristic (ROC) curve analysis, which evaluates the trade-off between true positive rate and false positive rate across a range of decision thresholds. To determine the optimal threshold, Youden's J statistic [35] is used, defined as

$$J = \text{sensitivity} + \text{specificity} - 1 \tag{6.1}$$

where sensitivity is the true positive rate and specificity is the true negative rate. This aims to maximize the difference between the true positive rate and the false positive rate. The threshold that yields the highest J value is selected as the optimal decision point. This threshold is then applied to the test set to distinguish between normal and faulty samples, thereby assessing the final performance of the model.

6.1.4 Metrics

To evaluate the performance of the proposed approaches, the following classification metrics were used. These metrics provide a comprehensive understanding of model behavior from different perspectives, especially in the presence of class imbalance or when different types of misclassifications have different implications.

Accuracy: Accuracy measures the proportion of correctly classified samples among the total number of samples. It is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.2}$$

where TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives, respectively. Accuracy is applicable to both binary and multi-class classification tasks and provides an overall measure of correctness.

Precision: In binary classification, precision evaluates the proportion of true positives among all predicted positives:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6.3}$$

For multi-class classification, precision is calculated in a *macro-averaged* manner, where the precision is first calculated independently for each class and then averaged:

$$\text{Macro-Precision} = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i + FP_i} \tag{6.4}$$

where C is the number of classes, and TP_i, FP_i denote the true positives and false positives for class i.

Recall: In binary classification, recall measures the proportion of actual positives that are correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6.5}$$

Similarly, for multi-class classification, macro-averaged recall is defined as:

$$\text{Macro-Recall} = \frac{1}{C} \sum_{i=1}^{C} \frac{TP_i}{TP_i + FN_i} \tag{6.6}$$

F1-score: The F1-score is the harmonic mean of precision and recall. For binary classification:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6.7}$$

In the multi-class case, the macro-averaged F1-score is defined as:

$$\text{Macro-F1} = \frac{1}{C} \sum_{i=1}^{C} 2 \cdot \frac{\text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{6.8}$$

These metrics were computed for each model and dataset configuration to ensure a consistent and comprehensive evaluation. Macro-averaging ensures that all classes contribute equally to the final score, which is particularly important when class distributions are imbalanced.

6.2 Results

This section presents the results of the classification-based and forecasting-based approaches. Results are presented separately for each dataset, providing insights into how each method performs under different data characteristics and application scenarios. Since the ultimate objective of these models is to be deployed on edge devices with limited resources, the parameter size of each model will also be provided.

6.2.1 Classification-based Approach On The JKU Dataset

First, the training and evaluation of the models on the JKU dataset are described. Since each sample in the JKU dataset has a moderate length, there is no need to divide the time series into shorter segments. Thus, 5-fold cross-validation is employed to evaluate model performance, providing stable and reliable assessment across different data partitions.

Furthermore, for wavelet-based models, a series of ablation studies are conducted to investigate the effects of different wavelet settings. The wavelet transform allows for various choices of mother wavelets (such as Daubechies, Symlets, Coiflets, Biorthogonal, etc.) and decomposition levels, both of which can significantly influence the ability to capture time-frequency features. To identify the optimal configuration, experiments are carried out by varying the mother wavelet type and decomposition depth. The best-performing setup is selected and used in the subsequent experiments. The ablation study is implemented on the previous proposed initial model. The results are shown in Figure 6.2 and Figure 6.3. From the results, it can be observed that the model achieves the best performance when using the Daubechies 4 wavelet and a decomposition level of 3. So these settings are used in the subsequent experiments.

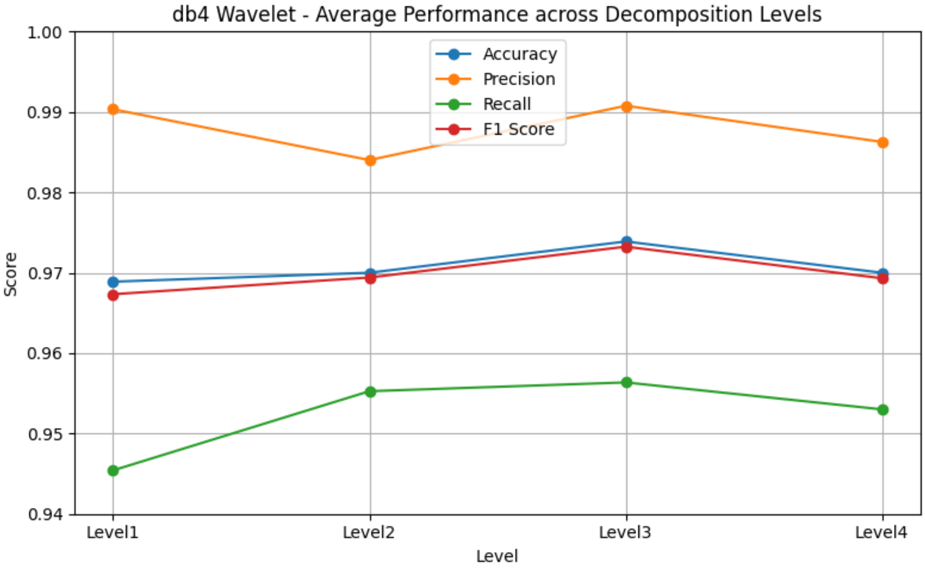


Figure 6.2: Performance on JKU dataset for different decomposition levels

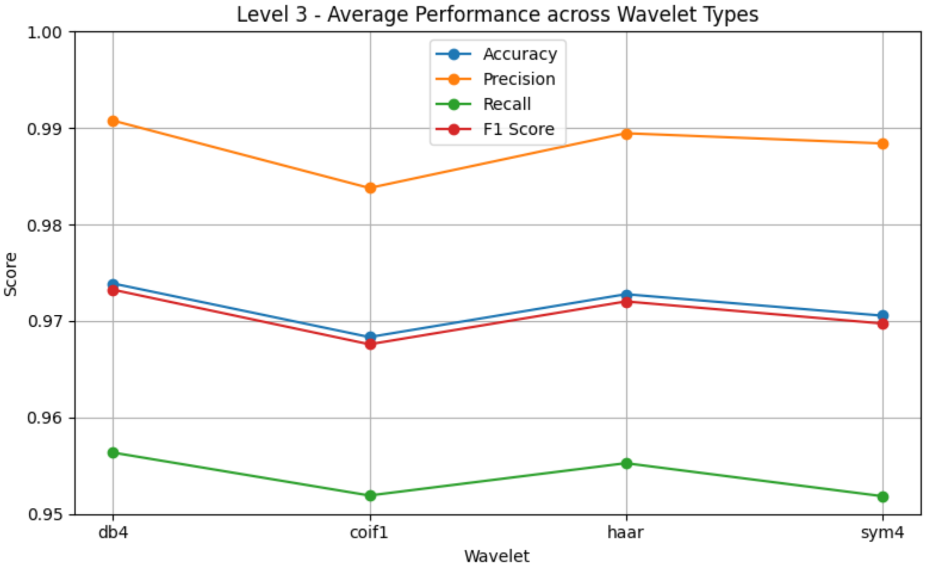


Figure 6.3: Performance on JKU dataset for different mother wavelets

The performance of the proposed model is shown in Figure 6.4. For the exact results, please refer to Table A.1 in the Appendix. Besides, the model size of different architectures on the JKU dataset is shown in Table 6.1.

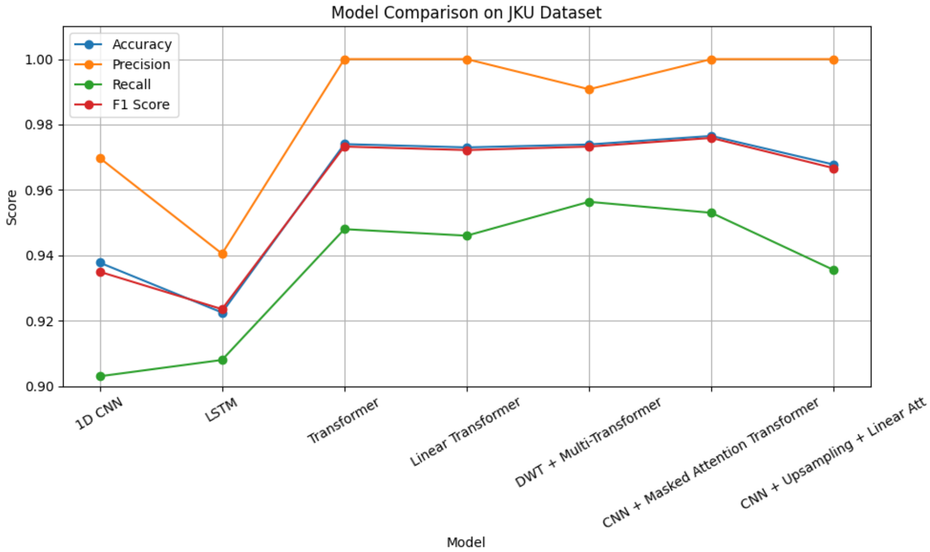


Figure 6.4: Performance on JKU dataset for the proposed model

Table 6.1: Model Size of Different Architectures on JKU Dataset

Model	Model size
1D CNN	324.26 KB
LSTM	86.26 KB
Transformer	315.51 KB
Linear Transformer	315.51 KB
DWT + Multi-Transformer	1739.76 KB
CNN + Masked attention Transformer	319.79 KB
CNN + Upsampling + Linear attention	313.04 KB

6.2.2 Classification-based Approach On The CWRU Dataset

Before presenting the experimental results on the CWRU dataset, it is important to consider the nature of the data. Since the samples from the CWRU dataset are obtained through segmentation of continuous time-series signals, the choice of input window length can have a significant impact on both model performance and complexity. Shorter windows may limit the amount of fault-related information available to the model, while longer windows may increase computational cost and potentially introduce redundant information.

To investigate this effect, a series of comparative experiments were conducted for each model using four different input lengths: 256, 512, 1024, and 2048. These experiments aim to analyze the relationship between the input sequence length, the resulting model size, and the classification accuracy. By doing so, it is possible to identify the optimal input configuration that balances performance and efficiency for practical deployment.

The results of the models on the CWRU dataset for different input lengths are

Table 6.2: Model Size of Different Architectures on CWRU Dataset

Model	Input Length	Model size
1D CNN	256	324.79 KB
1D CNN	512	324.79 KB
1D CNN	1024	324.79 KB
1D CNN	2048	324.79 KB
LSTM	256	86.29 KB
LSTM	512	86.29 KB
LSTM	1024	86.29 KB
LSTM	2048	86.29 KB
Transformer	256	202.16 KB
Transformer	512	234.16 KB
Transformer	1024	298.16 KB
Transformer	2048	426.16 KB
Linear Transformer	256	202.16 KB
Linear Transformer	512	234.16 KB
Linear Transformer	1024	298.16 KB
Linear Transformer	2048	426.16 KB
DWT + Multi-Transformer	256	1542.66 KB
DWT + Multi-Transformer	512	1574.66 KB
DWT + Multi-Transformer	1024	1638.66 KB
DWT + Multi-Transformer	2048	1766.66 KB
CNN + Masked attention Transformer	256	204.48 KB
CNN + Masked attention Transformer	512	236.48 KB
CNN + Masked attention Transformer	1024	300.48 KB
CNN + Masked attention Transformer	2048	428.48 KB
CNN + Upsampling + Linear attention	256	206.60 KB
CNN + Upsampling + Linear attention	512	222.60 KB
CNN + Upsampling + Linear attention	1024	254.60 KB
CNN + Upsampling + Linear attention	2048	318.60 KB

shown in Figure 6.5, Figure 6.6, Figure 6.7, and Figure 6.8. The original results are shown in Table A.2 in the Appendix. The model size of different architectures on the CWRU dataset is shown in Table 6.2.

6.2.3 Forecasting-based Approach On The JKU Dataset

The ROC curve with input length of 720 and prediction horizon of 720 is shown in Figure 6.9. This figure reflects the trade-off between the true positive rate and the false positive rate. The area under the ROC curve (AUC) serves as an aggregate measure of the model's classification performance. A larger AUC indicates better discriminative ability between normal and faulty samples. And the optimal threshold is selected as the point that maximizes the Youden's J statistic.

Figure 6.10 shows the performance of the proposed forecasting-based model on the

Table 6.3: Performance of the proposed forecasting-based model on the CWRU dataset for different input classes

Input class	**Accuracy**	**Precision**	**Recall**	**F1-score**
2-class	0.8923	0.9309	0.8570	0.8924
3-class	0.9048	0.9457	0.8856	0.9147

CWRU dataset for different input lengths. The results of different prediction horizons are compared in Figure 6.11. All of the original results are reported in Table A.3 and Table A.4.

Additionally, Table 6.3 compares the performance of the forecasting-based model when exposed to different input classes after training. It can be observed that the model's performance remains relatively stable even when encountering previously unseen data, indicating its robustness to novel fault types.

6.3 Summary

This chapter presented the experimental settings and results of the proposed classification-based and forecasting-based approaches on the JKU and CWRU datasets. For classification-based approach, the results include the performance of the proposed models, the model size of different architectures, and the comparison of the proposed models with the baseline and unimproved models on the JKU and CWRU datasets. For forecasting-based approach, the influence of different input lengths and prediction horizons on the performance of the proposed model is investigated. More discussions regarding the results are provided in the next chapter.

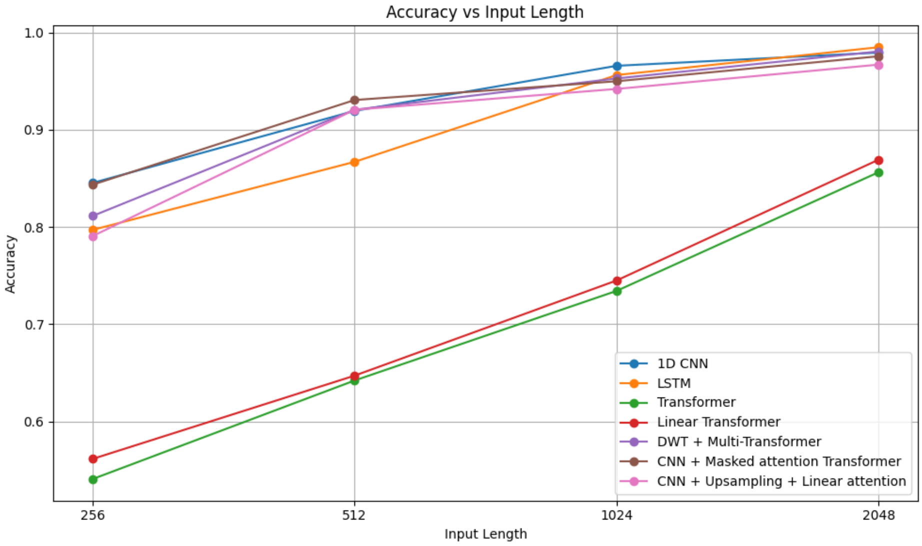


Figure 6.5: Accuracy on CWRU dataset for different input lengths

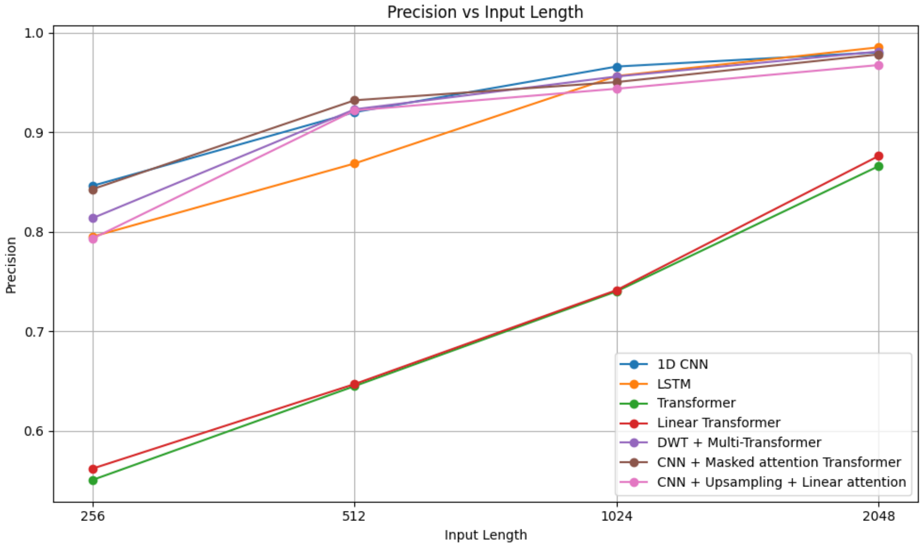


Figure 6.6: Precision on CWRU dataset for different input lengths

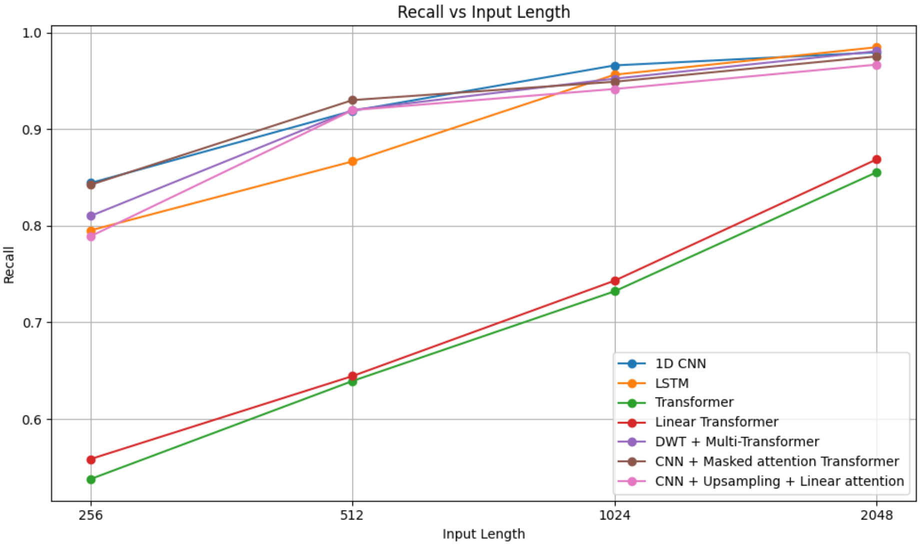


Figure 6.7: Recall on CWRU dataset for different input lengths

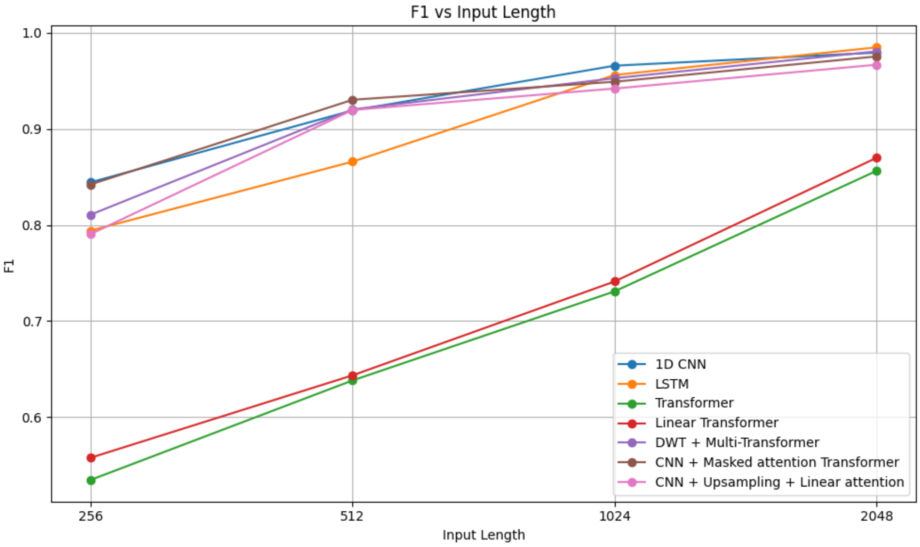


Figure 6.8: F1-score on CWRU dataset for different input lengths

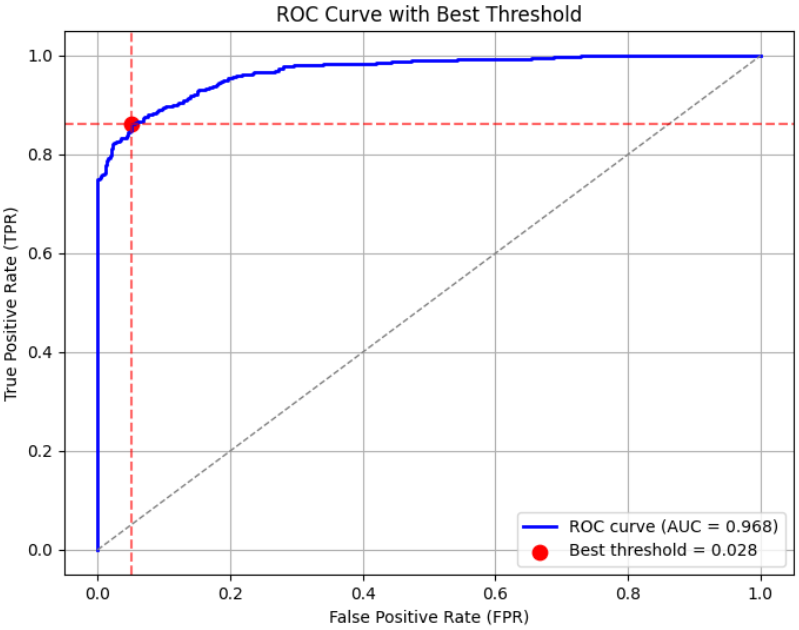


Figure 6.9: ROC curve of the proposed forecasting-based model on the CWRU dataset

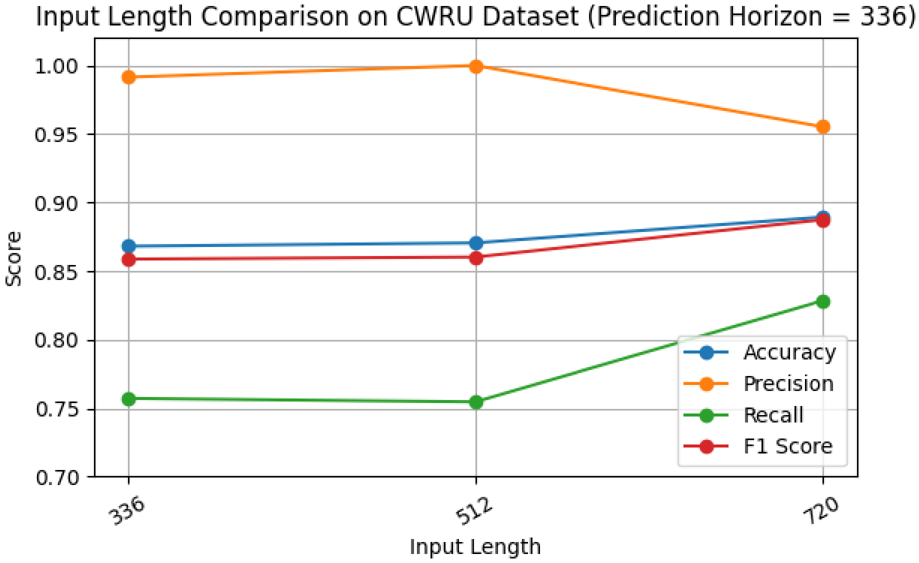


Figure 6.10: Forecasting-based approach on CWRU dataset for different input lengths

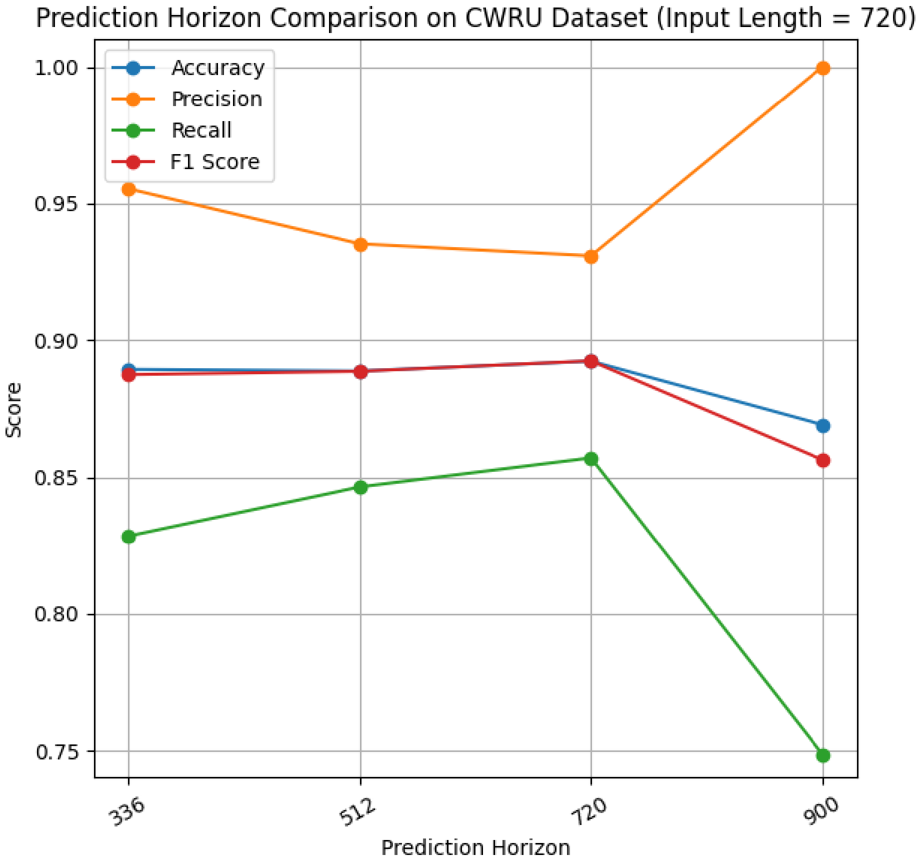


Figure 6.11: Forecasting-based approach on CWRU dataset for different prediction horizons

Discussion 7

This chapter discusses the results presented in the previous experiments, with a particular focus on comparing the proposed models with baseline and unimproved models on two datasets in terms of both performance and model size. In addition, several observed trends and phenomena are analyzed, and possible reasons behind them are explored by considering both the model architectures and the characteristics of the datasets.

7.1 Classification-based Approach

7.1.1 Ablation Study: Impact of Decomposition Parameters

One key factor investigated in the ablation study is the impact of wavelet decomposition level on model performance. As shown in Figure 6.2, the overall performance improves when increasing the decomposition level from 1 to 3, with Level 3 achieving the best results across all evaluation metrics. Specifically, Level 3 yields the highest accuracy (0.9739), precision (0.9908), recall (0.9564), and F1 score (0.9732), suggesting that this level offers an optimal trade-off between temporal resolution and feature abstraction. When the decomposition level is increased further to Level 4, a slight decline is observed. This indicates that overly deep decomposition may lead to information loss in high-frequency components, thereby affecting the model's ability to detect certain fault features. Therefore, Level 3 is considered the most balanced and effective configuration for this task.

In addition to decomposition level, the choice of mother wavelet also plays a crucial role in the model's performance. Figure 6.3 summarizes the average results for four commonly used wavelets: `db4`, `coif1`, `haar`, and `sym4`. Among them, the `db4` wavelet consistently achieves the best overall performance, with the highest accuracy (0.9739), precision (0.9908), recall (0.9564), and F1 score (0.9732). This suggests that `db4` is particularly well-suited for capturing the underlying features of the fault signals in the given datasets. The `haar` and `sym4` wavelets also produce competitive results, indicating their effectiveness in certain cases. However, `coif1` lags slightly behind in all four metrics, which may be attributed to its different time-frequency localization properties and longer filter length. Overall, the results indicate that selecting an appropriate wavelet basis is essential, and `db4` provides a favorable balance between localization and representation capacity for this task.

The varying performance across different wavelets and decomposition levels reflects the fact that these wavelet bases were originally designed with different signal characteristics and applications in mind. Each wavelet possesses distinct time-frequency localization properties, which can significantly affect how signal features are captured and represented. As a result, their classification performance differs when applied to fault detection tasks.

Moreover, the choice of decomposition level introduces a trade-off between signal resolution and sparsity. Deeper decomposition levels may lead to sparser coefficient representations, which in some cases improve generalization, but may also increase sensitivity to noise or result in the loss of discriminative details. These findings highlight the importance of carefully selecting both the wavelet basis and decomposition depth according to the specific nature of the dataset and task.

7.1.2 Model Performance on JKU Dataset

The comparison of different models on the JKU dataset reveals several important insights. As shown in Figure 6.4, traditional sequence models such as 1D CNN and LSTM achieve reasonable performance, with the CNN slightly outperforming the LSTM in terms of both accuracy and F1 score. However, both models fall short in recall, especially the CNN (0.9030), indicating a tendency to miss certain fault patterns.

In contrast, Transformer-based models demonstrate consistently strong results. The standard Transformer already achieves an accuracy of 0.9740 and perfect precision, while the Linear Transformer performs comparably with slightly lower recall. These results confirm the Transformer's strong capacity for capturing long-range dependencies in time series data.

Notably, models that incorporate domain-aware components, such as wavelet transform or attention masking, further improve performance. The DWT + Multi-Transformer and CNN + Masked Attention Transformer achieve the highest F1 scores (0.9732 and 0.9759, respectively), indicating superior overall classification ability. These improvements can be attributed to better feature localization in time-frequency space (from wavelets) and refined attention guidance (from masking). Interestingly, the CNN + Upsampling + Linear Attention model, though conceptually simpler, also performs competitively (F1: 0.9666), suggesting that upsampling and weight sharing may help reduce the model size and retain the feature extraction ability.

The model size is also a critical factor when considering deployment in resource-constrained environments. As summarized in Table 6.1, most of the Transformer-based models maintain relatively compact sizes, all around 315KB. The 1D CNN is slightly larger (324KB), while the LSTM remains the most lightweight at only 86KB, which makes it potentially attractive for edge applications despite its lower performance. On the other hand, the DWT + Multi-Transformer model exhibits a significantly larger footprint of 1739KB, more than five times the size of other models. This is due to the added complexity from the multiple transformer encoders. While this model achieves excellent performance, the increased cost in memory may limit its applicability in real-time or embedded settings.

The CNN + Masked Attention Transformer and CNN + Upsampling + Linear Attention models strike a good balance between performance and parameter size, both achieving high F1 scores with model sizes below 320KB. This suggests that carefully designed architectural improvements can enhance performance without incurring a large increase in model size.

7.1.3 Model Performance on CWRU Dataset

Experiments on the CWRU dataset reveal several important trends. First, increasing the input length generally improves model performance across all architectures. This aligns with the expectation that longer sequences provide richer fault-relevant context. Second, models equipped with DWT or CNN-based DWT mechanisms show noticeable advantages at all input lengths, confirming that wavelet decomposition enhances signal representation.

However, a striking observation is that classical models like 1D CNN outperform Transformers at some input lengths. For example, at 256-sample input, the CNN achieves an F1 score of 0.8445, while the best Transformer model reaches 0.8423. At 1024 samples, the CNN achieves 0.9659 in F1, whereas the Transformer remains lower at 0.9529. A possible explanation is that CNNs are inherently good at capturing localized patterns, which is an inductive bias well aligned with the structure of vibration signals in CWRU. In contrast, Transformers are designed to model long-range dependencies, which may not be as critical in this dataset where the most discriminative features are local.

Moreover, the performance gap may also stem from the nature of the dataset: CWRU's highly structured, repetitive fault signals favor models that can exploit local temporal features. Transformers, being more data-driven and position-invariant, may require larger datasets or more complex regularization to match the performance of simpler, bias-driven models like CNNs.

Overall, the results support the conclusion that combining signal processing priors (e.g., wavelet transforms) with attention-based architectures leads to more effective fault classification, particularly in complex datasets like JKU.

7.2 Forecasting-based Approach

The results of the forecasting-based approach on the CWRU dataset demonstrate the effectiveness of this method, particularly in scenarios involving previously unseen fault types. As shown in Figure 6.9, when the input length and prediction horizon are both set to 720, the area under the curve (AUC) reaches 0.968, indicating strong discriminative capability between normal and faulty samples. The ROC curve also provides a visual means to select the optimal threshold, facilitating a more reliable evaluation of the model's final performance.

Moreover, comparative analysis across different input lengths and prediction horizons reveals several trends. For instance, when the prediction horizon is fixed, increasing the input length generally leads to improved classification performance. This aligns with expectations, as longer input sequences allow the model to capture more informative features, thereby enhancing its predictive capability.

Conversely, when the input length is held constant and the prediction horizon is varied, the model performance first improves and then declines. The initial improvement may be attributed to the fact that abnormal patterns tend to occur more frequently in the near-future trajectories, resulting in a greater divergence in the distribution of prediction errors between normal and faulty samples. However, when the prediction horizon exceeds the model's effective prediction capacity, the proportion of anomalous segments in the predicted window diminishes, and their contribution to the mean squared error (MSE) becomes diluted. As a result, normal segments dominate the error calculation, reducing the separability between classes and ultimately degrading classification performance.

When the trained model was evaluated using test sets containing two and three fault types respectively, its performance showed no significant variation. The accuracy was 0.8923 in the two-class case and 0.9048 in the three-class case, indicating that the presence of previously unseen fault types did not substantially impact the model's predictive ability. This observation further supports the robustness of the proposed unsupervised learning approach, demonstrating its capability to maintain stable classification results even in the presence of unknown fault categories.

7.3 Summary

This chapter first discusses the performance of classification-based models. On the JKU dataset, the effects of discrete wavelet transform (DWT) decomposition levels and wavelet basis functions were evaluated. Through comparative experiments, the optimal configuration was identified as a decomposition level of 3 with the 'db4' wavelet. This configuration was subsequently adopted for DWT-based preprocessing in further validation on the JKU dataset. The results showed that the CNN + Masked Attention Transformer and CNN + Upsampling + Linear Attention models achieved a favorable balance between performance and model size, indicating that architectural optimizations can reduce the number of parameters without sacrificing model effectiveness.

The performance of classification models was then examined on the CWRU dataset. Since the CWRU dataset requires segmentation before use, the length of the input sequence naturally became a focus of investigation. Across input lengths of 256, 512, 1024, and 2048, all models exhibited better performance with longer input sequences. However, it is noteworthy that although the improved CNN + Masked Attention Transformer model outperformed their unmodified counterparts and the vanilla Transformer, their accuracy remained inferior to that of the conventional CNN at some input lengths. A possible explanation is that fault features in the CWRU dataset are highly localized, such as sharp spikes in vibration signals. This local feature is what CNNs excel, whereas Transformer-based models tend to emphasize global contextual features.

For the forecasting-based approach, the model also demonstrated strong performance. The investigation into input length and prediction horizon revealed that the optimal configuration should be determined based on the specific characteristics of the signal. Furthermore, comparisons involving different fault categories highlighted the robustness of the method to previously unseen fault types.

Conclusion 8

The proposed classification-based and forecasting-based fault detection models both demonstrate strong performance in the target application domain. However, there are still some limitations and potential improvements that can be explored in future work.

8.1 Conclusion

In this work, two Transformer-based models for motor fault detection were proposed. The first is a supervised learning model based on classification. In this model, the input signal is first processed by a convolutional layer designed to simulate discrete wavelet transform (DWT) decomposition. The resulting multi-level coefficients are then concatenated and passed through a masked attention mechanism to control the flow of information. A variant of this approach involves upsampling the coefficients to a uniform length, after which a shared linear attention module is applied. The final output is represented by a class token, which is used for classification. These architectural designs aim to fully leverage the multi-scale decomposition capabilities of DWT, while optimizing memory efficiency through structural adjustments.

The second model is an unsupervised learning approach based on forecasting. In this framework, only normal operating data are used during training. During validation, both normal and faulty samples are introduced to identify the optimal decision threshold using the receiver operating characteristic (ROC) curve. In the testing phase, samples are classified based on whether the prediction error exceeds the threshold.

Experimental results demonstrate that both models perform effectively on the target datasets, validating their potential for practical deployment in real-world fault detection scenarios. Each approach has distinct advantages and limitations: the supervised model offers high accuracy given sufficient labeled data, while the unsupervised model provides robustness and label-independence, making them suitable for different industrial contexts with varying constraints on computational resources, accuracy requirements, and real-time processing demands.

8.2 Future Work

Despite the promising performance demonstrated on the given datasets, there remain several aspects that can be further optimized. The following section outlines potential directions for future research, focusing on both model-level improvements and hardware-level implementation considerations.

8.2.1 Model-level Improvements

1. **Multi-scale Decomposition Alignment**: In the aforementioned classification models, the multi-level DWT coefficients are either concatenated and processed with a masked attention mechanism or upsampled to a uniform length for subsequent attention-based processing. However, this approach does not fully exploit the hierarchical frequency characteristics inherent in different coefficients. There remains significant research potential in how these features are represented and utilized. Similar to challenges encountered in multimodal learning where features from different modalities must be aligned within a shared semantic space, the fusion of multi-scale coefficients may benefit from more advanced integration strategies. For instance, employing cross-attention mechanisms or introducing additional encoders to align features across different frequency levels may further enhance the model's performance.

2. **Backbone Model for Forecasting**: The current forecasting-based model adopts PatchTST as the backbone architecture. In future work, a wider range of forecasting models could be explored. These models vary in architectural design and may offer better suitability for the specific characteristics of the current task, enabling more effective feature extraction and more accurate predictions. By reducing prediction errors, such models could improve the model's sensitivity to abnormal patterns and ultimately enhance classification accuracy.

3. **Hyperparameter Tuning**: In DWT-based methods, the choice of decomposition level and wavelet basis function are two critical parameters. There is a wide range of wavelet function families, each with different time-frequency localization properties. The selected decomposition level determines the frequency bands to which different coefficients correspond. For different types of signals and tasks, alternative configurations of these parameters may offer improved performance by enabling more accurate signal decomposition and providing richer multi-scale representations.

8.2.2 Hardware-level Implementation

1. **Hardware Deployment**: This work was designed with a focus on improving model performance while minimizing the number of parameters, in consideration of future deployment in industrial scenarios where computational resources are limited. As such, a key direction for future research will involve adapting the proposed models for deployment on resource-constrained devices. This includes challenges such as translating the models into suitable programming environments and applying model compression techniques to reduce computational and memory overhead, ensuring practical applicability in real-world industrial settings.

2. **Latency Considerations**: Another important factor to consider in hardware deployment and practical applications is latency. The model's ability to generate predictions in a timely manner is a critical evaluation metric, particularly in scenarios with strict real-time requirements. Future deployment efforts should

therefore include a thorough assessment of model latency, alongside accuracy and memory consumption, to comprehensively evaluate overall performance under real-world constraints.

Bibliography

[1] G. Xu, M. Liu, Z. Jiang, W. Shen, and C. Huang, "Online fault diagnosis method based on transfer convolutional neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 2, pp. 509–520, 2019.

[2] P. Liang, C. Deng, J. Wu, Z. Yang, J. Zhu, and Z. Zhang, "Compound fault diagnosis of gearboxes via multi-label convolutional neural network and wavelet transform," *Computers in Industry*, vol. 113, p. 103132, 2019.

[3] H. Wu, M. J. Triebe, and J. W. Sutherland, "A transformer-based approach for novel fault detection and fault classification/diagnosis in manufacturing: A rotary system application," *Journal of Manufacturing Systems*, vol. 67, pp. 439–452, 2023.

[4] B. M. Wise, N. B. Gallagher, S. W. Butler, D. D. White, and G. G. Barna, "Development and benchmarking of multivariate statistical process control tools for a semiconductor etch process: impact of measurement selection and data treatment on sensitivity," *IFAC Proceedings Volumes*, vol. 30, no. 18, pp. 35–42, 1997.

[5] S.-K. S. Fan, C.-Y. Hsu, D.-M. Tsai, F. He, and C.-C. Cheng, "Data-driven approach for fault detection and diagnostic in semiconductor manufacturing," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1925–1936, 2020.

[6] J. Chen, S. Chen, C. Ma, Z. Jing, and Q. Xu, "Fault detection of aircraft control system based on negative selection algorithm," *International Journal of Aerospace Engineering*, vol. 2020, no. 1, p. 8833825, 2020.

[7] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[8] K.-Y. Chen, L.-S. Chen, M.-C. Chen, and C.-L. Lee, "Using svm based method for equipment fault detection in a thermal power plant," *Computers in industry*, vol. 62, no. 1, pp. 42–50, 2011.

[9] P. Santos, L. F. Villa, A. Reñones, A. Bustillo, and J. Maudes, "An svm-based solution for fault detection in wind turbines," *Sensors*, vol. 15, no. 3, pp. 5627–5648, 2015.

[10] S. Zidi, T. Moulahi, and B. Alaya, "Fault detection in wireless sensor networks through svm classifier," *IEEE Sensors Journal*, vol. 18, no. 1, pp. 340–347, 2017.

[11] P. Konar and P. Chattopadhyay, "Bearing fault detection of induction motor using wavelet and support vector machines (svms)," *Applied Soft Computing*, vol. 11, no. 6, pp. 4203–4211, 2011.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] H. Zhao, S. Sun, and B. Jin, "Sequential fault diagnosis based on lstm neural network," *Ieee Access*, vol. 6, pp. 12 929–12 939, 2018.

[14] C.-C. Chen, Z. Liu, G. Yang, C.-C. Wu, and Q. Ye, "An improved fault diagnosis using 1d-convolutional neural network model," *Electronics*, vol. 10, no. 1, p. 59, 2020.

[15] D. T. Hoang and H. J. Kang, "A motor current signal-based bearing fault diagnosis using deep learning and information fusion," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 6, pp. 3325–3333, 2019.

[16] L.-H. Wang, X.-P. Zhao, J.-X. Wu, Y.-Y. Xie, and Y.-H. Zhang, "Motor fault diagnosis based on short-time fourier transform and convolutional neural network," *Chinese Journal of Mechanical Engineering*, vol. 30, pp. 1357–1368, 2017.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[19] Y. Jin, L. Hou, and Y. Chen, "A new rotating machinery fault diagnosis method based on the time series transformer," *arXiv preprint arXiv:2108.12562*, 2021.

[20] L. Mujica, J. Rodellar, A. Fernandez, and A. Güemes, "Q-statistic and t2-statistic pca-based measures for damage assessment in structures," *Structural Health Monitoring*, vol. 10, no. 5, pp. 539–553, 2011.

[21] C. Liu, J. Bai, and F. Wu, "Fault diagnosis using dynamic principal component analysis and ga feature selection modeling for industrial processes," *Processes*, vol. 10, no. 12, p. 2570, 2022.

[22] C. T. Yiakopoulos, K. C. Gryllias, and I. A. Antoniadis, "Rolling element bearing fault detection in industrial environments based on a k-means clustering approach," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2888–2911, 2011.

[23] N. Amruthnath and T. Gupta, "A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance," in *2018 5th international conference on industrial engineering and applications (ICIEA)*. IEEE, 2018, pp. 355–361.

[24] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, 2014, pp. 4–11.

[25] X. Yan, Y. Xu, D. She, and W. Zhang, "Reliable fault diagnosis of bearings using an optimized stacked variational denoising auto-encoder," *Entropy*, vol. 24, no. 1, p. 36, 2021.

[26] S. Zhang, F. Ye, B. Wang, and T. G. Habetler, "Semi-supervised bearing fault diagnosis and classification using variational autoencoder-based deep generative models," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6476–6486, 2020.

[27] A. Y. Zhou and A. B. Farimani, "Faultformer: pretraining transformers for adaptable bearing fault classification," *IEEE Access*, vol. 12, pp. 70 719–70 728, 2024.

[28] E. Marth, P. Zorn, F. Schmid, S. Masoudian, K. Koutini, and W. Amrhein, "Simulation-assisted training of neural networks for condition monitoring of electrical drives: Approach and proof of concept," in *IKMT 2022; 13. GMM/ETG-Symposium.* VDE, 2022, pp. 1–7.

[29] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 11, no. 7, pp. 674–693, 2002.

[30] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International conference on machine learning.* PMLR, 2020, pp. 5156–5165.

[31] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[32] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in neural information processing systems*, vol. 34, pp. 22 419–22 430, 2021.

[33] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International conference on machine learning.* PMLR, 2022, pp. 27 268–27 286.

[34] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *The Eleventh International Conference on Learning Representations*, 2023.

[35] W. J. Youden, "Index for rating diagnostic tests," *Cancer*, vol. 3, no. 1, pp. 32–35, 1950.

Extra Tables and Figures A

Table A.1: Performance of Different Classification Models on the JKU Dataset

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
1D CNN	0.9378	0.9698	0.9030	0.9350
LSTM	0.9225	0.9405	0.9080	0.9235
Transformer	0.9740	1.0000	0.9480	0.9733
Linear Transformer	0.9730	1.0000	0.9460	0.9722
DWT+Multi-Transformer	0.9739	0.9908	0.9564	0.9732
CNN+Masked Attention	0.9765	1.0000	0.9530	0.9759
CNN+Upsampling+Linear Att	0.9678	1.0000	0.9355	0.9666

Table A.2: Performance of Different Classification Models on the CWRU Dataset

Model	**Input Length**	**Accuracy**	**Precision**	**Recall**	**F1-score**
1D CNN	256	0.8454	0.8461	0.8443	0.8445
1D CNN	512	0.9195	0.9201	0.9190	0.9193
1D CNN	1024	0.9659	0.9659	0.9660	0.9659
1D CNN	2048	0.9795	0.9802	0.9795	0.9795
LSTM	256	0.7971	0.7950	0.7953	0.7942
LSTM	512	0.8671	0.8685	0.8667	0.8660
LSTM	1024	0.9564	0.9564	0.9564	0.9562
LSTM	2048	0.9850	0.9852	0.9848	0.9850
Transformer	256	0.5407	0.5505	0.5379	0.5346
Transformer	512	0.6421	0.6448	0.6394	0.6382
Transformer	1024	0.7343	0.7400	0.7322	0.7309
Transformer	2048	0.8564	0.8658	0.8554	0.8564
Linear Transformer	256	0.5614	0.5619	0.5586	0.5577
Linear Transformer	512	0.6471	0.6468	0.6445	0.6435
Linear Transformer	1024	0.7450	0.7412	0.7432	0.7412
Linear Transformer	2048	0.8693	0.8760	0.8687	0.8700

Model	Input Length	Accuracy	Precision	Recall	F1-score
DWT+Multi-Transformer	256	0.8114	0.8136	0.8102	0.8108
DWT+Multi-Transformer	512	0.9207	0.9229	0.9199	0.9204
DWT+Multi-Transformer	1024	0.9529	0.9559	0.9523	0.9529
DWT+Multi-Transformer	2048	0.9807	0.9810	0.9807	0.9808
CNN+Masked Attention	256	0.8436	0.8427	0.8425	0.8423
CNN+Masked Attention	512	0.9307	0.9320	0.9301	0.9304
CNN+Masked Attention	1024	0.9500	0.9503	0.9491	0.9492
CNN+Masked Attention	2048	0.9757	0.9781	0.9753	0.9755
CNN+Upsampling+Linear Att	256	0.7907	0.7931	0.7892	0.7906
CNN+Upsampling+Linear Att	512	0.9207	0.9218	0.9197	0.9198
CNN+Upsampling+Linear Att	1024	0.9421	0.9436	0.9416	0.9421
CNN+Upsampling+Linear Att	2048	0.9671	0.9674	0.9668	0.9669

Table A.3: Performance of Forecasting Models of Different Input Lengths (Prediction Horizon: 336)

Input Length	Accuracy	Precision	Recall	F1 Score
336	0.8681	0.9915	0.7573	0.8587
512	0.8705	1.0000	0.7548	0.8602
720	0.8893	0.9554	0.8284	0.8874

Table A.4: Performance of Forecasting Models of Different Prediction Horizons (Input Length: 720)

Prediction Horizon	Accuracy	Precision	Recall	F1 Score
336	0.8893	0.9554	0.8284	0.8874
512	0.8887	0.9353	0.8464	0.8886
720	0.8923	0.9309	0.8570	0.8924
900	0.8691	1.0000	0.7487	0.8563