SIGNAL PROCESSING-EMPOWERED DEEP LEARNING FOR PROGNOSTICS AND HEALTH MANAGEMENT OF ROLLING BEARINGS

LIAO JINGXIAO

PhD

The Hong Kong Polytechnic University

This programme is jointly offered by The Hong Kong Polytechnic University and

Harbin Institute of Technology

2026

The Hong Kong Polytechnic University

Department of Industrial and Systems Engineering

Harbin Institute of Technology

School of Instrumentation Science and Engineering

Signal Processing-Empowered Deep Learning for Prognostics and Health Management of Rolling Bearings

Liao Jingxiao

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

August 2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

LIAO Jingxiao
_____ (Name of student)

Abstract

In recent years, deep learning has achieved significant success in various fields, including natural language processing, autonomous driving, and computer vision. In the realm of prognostics and health management (PHM) for rolling bearings in rotating machinery—such as aero engines, wind turbines, and high-speed trains—numerous intelligent PHM methodologies have emerged to provide accurate and adaptable machinery fault diagnostics and prognostics. However, methodologically speaking, there is no one-size-fits-all approach. It is widely acknowledged that these data-driven approaches still have considerable limitations, hindering their widespread adoption in industrial settings.

Three primary challenges persist: (1) the lack of interpretability in deep learning methods, particularly in machinery fault diagnosis, where diagnostic models must be transparent to foster trust in the results and inform maintenance decisions; (2) the limited generalizability and reliability of bearing remaining useful life (RUL) prediction models. When training data is scarce, even under identical operating conditions and with the same bearing types, current RUL models demonstrate suboptimal accuracy. In addition, ensuring the reliability of RUL predictions is an important consideration for making informed maintenance decisions in real-world scenarios; and (3) the difficulty in deploying intelligent diagnosis models to edge devices, which hinders their integration into real-world industrial settings.

Therefore, this dissertation aims to address these challenges by establishing a new paradigm of integrating traditional signal processing and modern deep learning methods. We formally define this approach as signal processing-empowered neural networks, which synthesize the complementary strengths of both domains. This framework provides three key advantages: (1) integrating rigorous signal processing theory to improve model interpretability; (2) leveraging the robust feature representation capabilities of signal processing techniques to enhance deep learning model generalizability and

auxiliary exponential model to quantify the reliability of RUL predictions; and (3) enabling faster computation and greater accuracy, thereby facilitating the edge device deployment of lightweight models. The research contents are summarized as follows:

(1) To address the interpretability issue in bearing fault diagnosis, we introduce the concept of neural blind deconvolution (Neural BD), which neuralizes the conventional blind deconvolution (BD) approach. We develop a classifier-guided blind deconvolution (ClassBD) framework that jointly learns BD extracted features and deep learning-based fault labels. The physics guaranteed features can be extracted by BD to provide interpretable information for intelligent fault diagnosis models.

Moreover, the superiority of quadratic neural networks in Neural BD inspired us to enhance the feature extraction ability for unsupervised anomaly detection. We mathematically prove that a heterogeneous network composed of quadratic and conventional neurons can approximate a function with a polynomial number of neurons, whereas a homogeneous network requires an exponential number of neurons to achieve the same level of approximation error. This theoretical result motivates the development of heterogeneous autoencoders (An autoencoder integrates conventional and quadratic neurons). Our proposed heterogeneous autoencoder exhibits both efficiency and effectiveness in unsupervised bearing fault anomaly detection.

(2) To address the generalizability issue of bearing remaining useful life (RUL) prediction, we introduce a logarithmic cumulative transformation (LCT) operator, consisting of cumulative, logarithmic, and another cumulative transformation, for feature extraction. This operator effectively aligns the diverse patterns across vibration degradation data of different bearings. The LCT can be seamlessly embedded into any deep learning or machine learning backbones to enhance the performance of RUL prediction.

Additionally, for reliable RUL estimation, we propose a fast method to quantify the reliability of each prediction by integrating a linear regression model and an auxiliary exponential model. This method provides an assurance on the reliability of model predictions.

(3) To improve the deployability of bearing fault diagnosis models, we propose a lightweight and deployable model for bearing fault diagnosis, referred to as BearingPGA-Net, to address the edge hardware deployment issue. Based on decoupled knowledge distillation, we construct a signal processing and deep learning integrated one-convolutional-layer bearing fault diagnosis model. This lightweight model is successfully deployed in an industrial-level FPGA, facilitating online bearing fault diagnosis

with potential for real-world applications.

Publications

(* Corresponding Author)

1. Journal Papers Completed at PolyU

[1] **J.-X. Liao**, J. Li, H.-C. Dong, M. Zhang*, S. Zhang*, X. Zhang*. Logarithmic cumulative transformation: A simple yet effective approach for bearing remaining useful life prediction. *IEEE Transactions on Industrial Informatics*, doi: 10.1109/TII.2024.3400299. (JCRQ1, IF:9.9).

[2] **J.-X. Liao**, C. He, J. Li, J. Sun, S. Zhang*, X. Zhang*, Classifier-guided neural blind deconvolution: a physics-informed denoising module for bearing fault diagnosis under heavy noise, *Mechanical Systems and Signal Processing*, doi: 10.1016/j.ymssp.2024.111750. (JCRQ1, IF:7.9).

[3] **J. -X. Liao**, S.-L. Wei, C.-L. Xie, T. Zeng, J. Sun, S. Zhang*, X. Zhang*, F.-L. Fan*, BearingPGA-Net: A lightweight and deployable bearing fault diagnosis network via decoupled knowledge distillation and FPGA acceleration, *IEEE Transactions on Instrumentation and Measurement*, doi: 10.1109/TIM.2023.3346517. (JCRQ1, IF:5.6).

[4] **J.-X. Liao**, B.-J. Hou, H.-C. Dong, H. Zhang, X. Zhang, J. Sun, S. Zhang*, F.-L. Fan*, Quadratic neuron-empowered heterogeneous autoencoder for unsupervised anomaly detection, *IEEE Transactions on Artificial Intelligence*, doi: 10.1109/TAI.2024.3394795.

[5] **J.-X. Liao**, H. Chao, J. Li, X.-C. Zhong, J. Sun, S. Zhang*, X. Zhang*, Heterogeneous neural blind deconvolution: A signal processing-empowered foundational feature extractor for bearing fault diagnosis. (Under Review in *Neural Networks*).

[6] W.-E. Yu, S. Zhang*, J. Sun, **J.-X. Liao***, X. Zhang*, A class-weighted supervised contrastive learning long-tailed bearing fault diagnosis approach using quadratic neural network, *arXiv preprint*, doi: arXiv:2309.11717. (Major Revision in *IEEE Transactions on Reliability*).

2. Journal Papers Completed at HIT

[1] F.-L. Fan, H.-C. Dong, Z. Wu, L. Ruan, T. Zeng*, Y. Cui*, **J.-X. Liao***, One neuron saved is

one neuron earned: On parametric efficiency of quadratic networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, doi: 10.1109/TPAMI.2025.3588894. (JCRQ1, IF:18.6).

[2] **J.-X. Liao**, H.-C. Dong, Z.-Q. Sun, J. Sun, S. Zhang*, F.-L. Fan*. Attention-embedded quadratic network (qttention) for effective and interpretable bearing fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, doi: 10.1109/TIM.2023.3259031 (JCRQ1, IF:5.6, ESI High Cited Paper).

[3] **J.-X. Liao**, H.-C. Dong, L. Luo, J. Sun, S. Zhang*. Multi-task neural network blind deconvolution and its application to bearing fault feature extraction, *Measurement Science and Technology*, doi: 10.1088/1361-6501/accbdb1.(JCRQ1, IF:2.7).

[4] M. Zhang, B. Zhao, J. Sun, Q. Wang*, D. Liu, J. Li, **J. Liao***, Research on driver fatigue detection based on information fusion, *Computers, Materials & Continua*, doi: 10.32604/cmc.2024.048643. (JCRQ2, IF:1.7).

3. Other Journal and Conference Papers

[1] H. Dong, **J. Liao**, Y. Wang, Y. Chen, B. Liu, D. Ye, G. Liu*. Training neural networks for solving 1-D optimal piecewise linear approximation. *Neurocomputing*, doi: 10.1016/j.neucom.2022.07.025. (JCRQ1, IF:5.5).

[2] X.-C. Zhong, Q Wang*, D Liu, Z. Chen, **J.-X. Liao**, J. Sun, Y. Zhang, F.-L. Fan*. EEG-DG: A multi-source domain generalization framework for motor imagery EEG classification. *IEEE Journal of Biomedical and Health Informatics*, doi: 10.1109/JBHI.2024.3431230. (JCRQ1, IF:6.7).

[3] X.-C. Zhong, Q. Wang*, D. Liu*, **J.-X. Liao**, R. Yang, S. Duan, G. Ding, J. Sun. A deep domain adaptation framework with correlation alignment for EEG-based motor imagery classification. *Computers in Biology and Medicine*, doi: 10.1016/j.compbiomed.2023.107235. (JCRQ1, IF:7.0).

[4] J. Tang, D. Liu, Q. Wang, J. Li, **J. Liao**, J. Sun. Marginal Contribution Spectral Fusion Network for Remote Hyperspectral Soil Organic Matter Estimation. *Remote Sensing*, doi:10.3390/rs17162806. (JCRQ1, IF:4.1).

[5] **J.-X. Liao**, J. Li, M. Zhang*, F.-L. Fan*, X. Zhang*. Uncertainty-Aware Heterogeneous Neural Blind Deconvolution Ensemble Network for Reliable System-Level Fault Diagnosis in Railway Transmission Systems. *The sixth International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence* (Accepted).

[6] J. Li, K. Yue, **J. Liao**, T. Wang, X. Zhang*. Global-local continual transfer network for intelli-

gent fault diagnosis of rotating machinery. *Annual Conference of the PHM Society*, doi: 10.36001/phm-conf.2024.v16i1.4064.

Acknowledgements

EVERY journey eventually reaches its conclusion. My doctoral journey has been an extraordinary and transformative odyssey. At this moment, I wish to express my deepest and heartfelt gratitude to all who have supported and guided me along this way.

My deepest appreciation goes to my chief supervisor, Prof. Xiaoge Zhang, whose guidance illuminated my scholarly exploration. He has bestowed upon my academic and personal growth. I am especially grateful for his rigorous standards and meticulous guidance throughout the writing of my dissertation. His unwavering support fostered an exceptional academic environment and provided invaluable opportunities for international collaborations.

I extend my sincere gratitude to my co-chief supervisor, Prof. Jinwei Sun, at Harbin Institute of Technology. Throughout my doctoral journey at HIT, Prof. Sun's steadfast support and encouragement enabled my research to thrive, offering enriching opportunities for international cooperation. His dedication to research and commitment to excellence will continue to inspire my future endeavors.

Heartfelt thanks to my co-supervisor, Prof. Shiping Zhang, who has guided me since my undergraduate period, leading me into the world of scientific research. His kindness and personal support are invaluable, helping me navigate both academic and personal milestones.

The successful completion of this dissertation is indebted to the cooperation and support of many mentors, friends, and colleagues. I am deeply grateful to my dear friends, Prof. Fenglei Fan at CityUHK and Mr. Hangcheng Dong at HIT, for their insightful academic assistance. My heartfelt thanks also go to Mr. Chao He at Beijing Jiaotong University, Dr. Jipu Li at PolyU and Prof. Jinwu Li at Dongguan University of Technology, Prof. Lei Luo at Chongqing University, Prof. Tieyong Zeng at CUHK, and Prof. Yiu-Ming Cheung at HKBU for their contributions and suggestions on my work. I am equally grateful to my lab mates—Ms. Zhiqi Sun, Mr. Shenglai Wei, Mr. Chenlong Xie, and Mr. Weien Yu—for their research support. Appreciation goes to Prof. Xiaoli Zhao and

Mr. Hongyuan Zhang at the MIIT Key Laboratory of Aerospace Bearing Technology and Equipment, HIT, for their assistance in bearing fault experiments. Lastly, I extend my sincere gratitude to the dissertation defense committee and reviewers for their constructive feedback.

My warmest thanks to all faculty members and lab mates at the Risk Reliability & Resilience Informatics Research Group, PolyU, including Dr. Jipu Li, Prof. Jinwu Li, Mr. Long Xue, Ms. Pingping Dong, Ms. Ruonan Zhu, Ms. Yanwen Jin, Mr. Chenyu Li, Mr. Tao Wang, Ms. Xinru Zhang, and Mr. Hang Ji. Their companionship and support have been deeply cherished.

Likewise, I extend my heartfelt thanks to all faculty members and lab mates at the Institute of Precision Electrical Measurement and Instrumentation, HIT, for their thoughtful guidance and encouragement. Particular appreciation to Prof. Qisong Wang, Prof. Dan Liu, Dr. Meiyan Zhang, Mr. Xiaocong Zhong, Mr. Boqi Zhao, Ms. Wenjia Gao, Mr. Jiaze Tang, and Ms. Meilin Huang.

Finally, to my beloved parents, my eternal gratitude for being my foundation. Their support, encouragement, and belief in my aspirations have endowed me with boundless strength, fueling my passion for perseverance throughout this journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Prognostics and health management in rotating machinery

These days, the rapid explosion of big data and artificial intelligence technologies has driven the fast development of industrial intelligence, such as intelligent manufacturing (B. Wang et al., 2021), industrial large vision-language models (Z. Gu et al., 2024), and autonomous industrial robotics (H. Fan et al., 2024). The Fourth Industrial Revolution, popularized by Klaus Schwab in 2016, describes a trend towards the increasing automation of conventional manufacturing and industrial practices (Philbeck & Davis, 2018). Specifically, intelligent manufacturing, benefiting from advances in computer science, has incorporated technologies such as industrial Internet of Things (IoT) (Da Costa et al., 2019), cloud computing (Armbrust et al., 2010), and prognostics and health management (PHM) (Zio, 2022). Among them, PHM is an engineering discipline that studies the range from failure generation mechanisms to systems life cycle management (Zio, 2016, 2022). It aims to provide just-in-time and just-right maintenance strategies, supporting condition-based and predictive maintenance decisions for reliable operations of industrial systems (Lei et al., 2018).

The research scope of PHM includes fault detection, fault diagnosis, and fault prognostics, as depicted in Figure 1.1. Fault detection, as an unsupervised or semi-supervised learning task, constructs a model using only healthy data (sometimes with limited abnormal data) to detect unknown faults. Fault diagnosis, as a supervised learning problem, classifies fault types and levels with pattern recognition models—also can integrate unsupervised (Xiao et al., 2022) or semi-supervised (L. Jiang, Xuan, & Shi, 2013) models. Fault diagnosis, as a supervised learning problem, classifies fault types and degradation levels. Lastly, fault prognostics utilizes limited historical data to predict the remaining useful

life (RUL) of the machine, facilitating to streamline maintenance operations and devise cost-effective production plans (Nemani et al., 2023). Nowadays, complex sub-tasks have emerged from these three foundational tasks, such as diagnosis under strong noise, cross-domain diagnosis, and lightweight diagnosis.

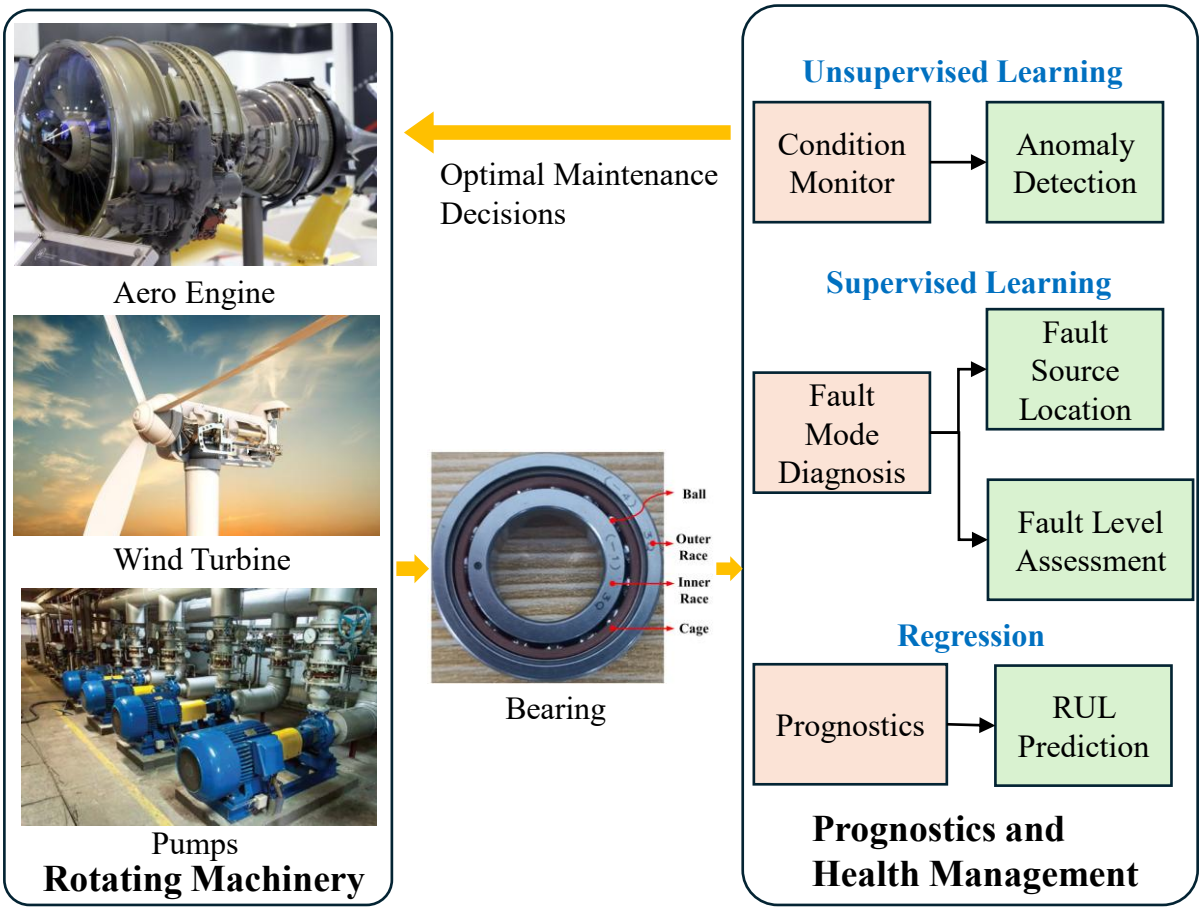


Figure 1.1: Prognostics and health management of rotating machinery.

Rotating machinery, including aero engines, wind turbines, and high-speed trains, plays a critical role in modern infrastructure and daily life. The PHM ensures sound and reliable operation of machinery, maximizing economic benefits and minimizing risks of unpredictable damage (Compare, Baraldi, & Zio, 2020). In particular, maintaining the healthy conditions of rolling bearings is essential for the performance and longevity of rotating equipment. Statistical analyses indicate that bearing faults account for approximately 40% to 70% of machinery failures (Bonnett & Yung, 2008). Consequently, the diagnosis of bearing failures has emerged as a prominent research focus in recent years. This dissertation specifically examines the application of PHM techniques for the predictive maintenance and fault detection of rolling bearings.

1.2 Research niche

The rapid advancement of big data and high-performance computing has led to the emergence of a plethora of artificial intelligence methods in the PHM area. In particular, deep learning approaches have brought new opportunities and challenges. By leveraging big data, deep learning methods have been employed to develop comprehensive and effective end-to-end fault diagnosis models, achieving the ability of automatic feature extraction and pattern recognition (X. Chen et al., 2023; R. Zhao et al., 2019). These approaches have liberated the traditional paradigm that relies on expert knowledge. However, it is recognized that these data-driven approaches still have significant limitations, hindering their widespread adoption in industrial settings:

- **Interpretability.** Deep learning models are proverbially considered as "black box" machines (Sarker, 2021). In other words, deep learning models only provide predictions without transparent insights into how they arrive at their conclusions. This is due to the inherent complex non-linear transformation embedded in deep learning. Until now, a fully transparent deep learning model has not been designed, and a complete mathematical theory behind deep learning has not been built to explain the inference process. However, in the context of fault diagnosis, it is of paramount importance to interpret the inference process, fostering trust in the results and assisting human maintenance decisions.

- **Generalizability and reliability.** In the field of remaining useful life (RUL) prediction, estimating the RUL of bearings remains a challenging task due to limited historical data and frequent variations in operational conditions. Even under nominally identical operating conditions, bearings of the same type may demonstrate divergent degradation patterns. This underscores the critical need in the generalizability of deep learning models for RUL prediction (X. Li et al., 2023; Qin et al., 2023). Moreover, since prediction accuracy is highly dependent on the progression of degradation over time, RUL models should incorporate reliability estimation to ensure that their predictions are trustworthy (T. Zhou, Han, & Droguett, 2022).

- **Deployability.** While deep learning models deliver superior performance, they often come with larger model sizes and computational burdens. These models require to deploy on high-performance computers (H. Fang, Deng, Bai, et al., 2021; S. Yan et al., 2024). This hinders their deployment on edge hardware in real-world industrial fields, which demand high speed,

lightweight, and low power consumption. Therefore, developing lightweight and deployable fault diagnosis models is an urgent issue that needs to be addressed.

1.3 Research objectives

To tackle the above challenges, the motif of this dissertation is to devise advanced neural networks that offer a more powerful feature extraction capability and post-hoc interpretability. The strong correlation between bearing fault signals and vibration physical mechanisms provides a promising avenue for exploration (Rai & Upadhyay, 2016). Specifically, bearing fault signals have distinct frequency components, and conventional signal processing methods are designed to extract fault-related features for diagnosis. The rigorous mathematical foundations of these signal processing methods can effectively complement the interpretability shortcomings of deep learning techniques. Therefore, this study aims to construct a novel paradigm that merges the advantages of traditional signal processing and modern neural networks. These networks are termed as signal processing-empowered neural networks (Ruqiang, Zheng, et al., 2024; Ruqiang, Zuogang, et al., 2024).

In this study, we propose several signal processing–empowered neural networks, developed from both theoretical and practical perspectives, to address the aforementioned challenges. It is important to note that signal processing is a broad concept. Techniques such as wavelet transform and mode decomposition are commonly employed to extract interpretable information from raw signals. This dissertation further develops their potential to construct interpretable fault diagnosis models. Moreover, in contrast to these methods designed to obtain interpretable features, those adopted in RUL prediction primarily aim to capture significant degradation trends from raw signals (Javed et al., 2015; M. Yan et al., 2020). Despite their different objectives, these approaches can also be regarded as signal processing techniques. Accordingly, this thesis employs both categories of methods to build deep learning models for PHM tasks, referred to as signal processing–empowered neural networks.

Our contributions are summarized as follows:

- To address the interpretability issue, we propose a novel approach called classifier-guided blind deconvolution (ClassBD). For the first time, blind deconvolution is successfully incorporated with classification networks. This approach provides a signal processing-based mathematical explanation for the neural network and achieves remarkable fault diagnostic performance under

heavy noise conditions (Chapter 3).

- To further enhance the capabilities of quadratic networks, we establish an efficient approximation theorem for heterogeneous neural networks (integrating quadratic and conventional neurons). Our analysis demonstrates that heterogeneous neural networks yield the most efficient representational framework. Building upon this theoretical foundation, we propose a heterogeneous autoencoder (HAE) architecture, which exhibits both computational efficiency and robust performance in unsupervised anomaly detection tasks, particularly for bearing fault signal analysis (Chapter 4).

- To tackle the generalizability and reliability issues in remaining useful life prediction, we propose a simple yet effective feature transformation method, logarithmic cumulative transformation (LCT), successfully addressing feature misalignment across bearings operated under varying conditions. Additionally, we propose a novel method to quantify the reliability of each RUL prediction, which provides an assurance in predicted results (Chapter 5).

- To resolve the deployability issue, we propose a lightweight and deployable neural network tailored for industrial settings. Based on the decoupled knowledge distillation, we construct a signal processing and deep learning integrated one-convolutional-layer bearing fault diagnosis model. This lightweight model is successfully deployed into an industrial-level FPGA, facilitating online bearing fault diagnosis (Chapter 6).

1.4 Thesis outline

In this dissertation, we mainly prototype several signal processing-empowered neural networks for the prognostics health management of rotating machinery. The technical route is summarized in Figure 1.2, and the thesis outline is illustrated as follows.

In Chapter 2, we present a systematic review of existing signal processing and deep learning approaches in machinery PHM. We conduct a critical analysis of current methodologies, examining their respective strengths and limitations. Furthermore, we provide a comprehensive survey of relevant works that form the foundational context for this dissertation's research.

In Chapter 3, we introduce a novel approach termed classifier-guided blind deconvolution (ClassBD), which integrates blind deconvolution (BD)-based feature extraction with deep learning-driven fault

diagnosis in a unified learning framework. To realize this, we design a neural blind deconvolution mechanism that replicates traditional BD using neural networks, enabling seamless joint optimization of BD components and the downstream classifier. The proposed architecture incorporates two specialized filters: (i) a time-domain quadratic filter leveraging quadratic convolutional networks to capture periodic impulse patterns, and (ii) a frequency-domain linear filter based on fully connected layers to enhance discrete spectral features. A deep classifier is then employed to guide the BD filters' learning process within a cohesive framework. Additionally, a physics-informed loss function combining kurtosis, l_2/l_4 norm, and cross-entropy is devised to jointly optimize both the BD filters and the classifier. This coupling allows the model to leverage label supervision to enhance noise-robust, class-discriminative feature extraction.

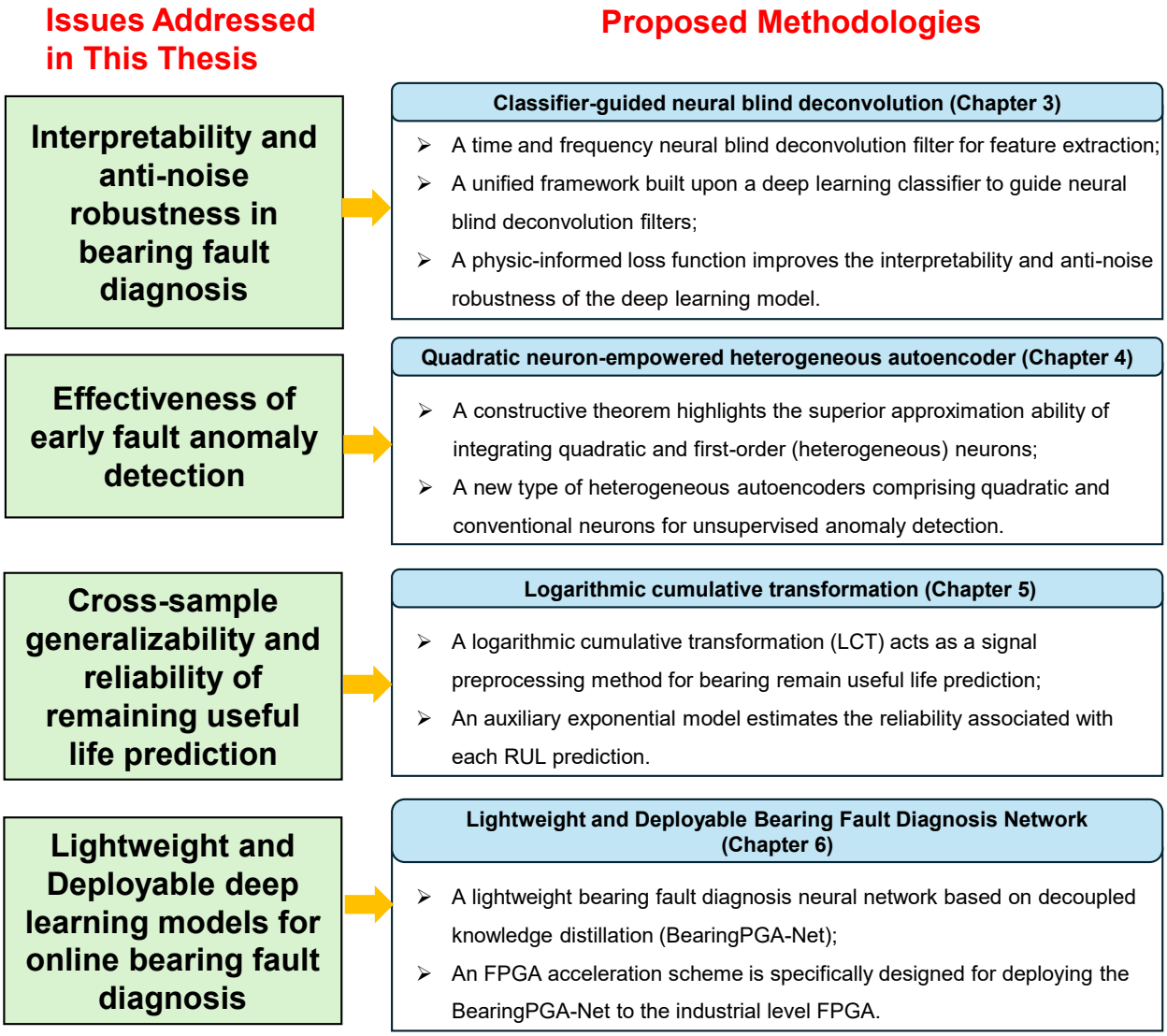


Figure 1.2: The outline of this dissertation.

In Chapter 4, we theoretically demonstrate that a heterogeneous neural network, composed of

mixed neuron types, can approximate specific functions with polynomial complexity—whereas conventional or purely quadratic networks require exponential scaling to achieve comparable accuracy. Motivated by this insight, we construct a new class of heterogeneous autoencoders by integrating conventional and quadratic neurons into a single architecture. To the best of our knowledge, this represents the first autoencoder constructed using heterogeneous neuron types. We then deploy this model for unsupervised anomaly detection in both tabular datasets and vibration signals related to bearing faults. This task poses challenges including unknown anomaly types, heterogeneous features, and low anomaly visibility—making it well-suited for the proposed model. The heterogeneous autoencoder demonstrates strong representational capacity: it can model diverse anomaly patterns (heterogeneity), distinguish subtle abnormal signals (unnoticeability), and learn the underlying distribution of normal data (unknownness) with high fidelity.

In Chapter 5, we present a novel logarithmic cumulative transformation (LCT) operator, designed to enhance feature extraction through a sequence of transformations—namely cumulative summation, logarithmic scaling, and a second cumulative operation. Additionally, we develop an innovative strategy to quantify the confidence level of each remaining useful life (RUL) prediction. This is achieved by combining a linear regression model with a complementary exponential model. The linear model serves to correct systematic deviations in point predictions from the neural network, while the exponential model captures variations in the regression slopes. These components jointly construct upper and lower predictive bounds, thereby facilitating the generation of a robust reliability metric.

In Chapter 6, we propose BearingPGA-Net, a compact and hardware-friendly neural network tailored for bearing fault diagnosis. The model is specifically designed to meet the constraints of edge-device deployment. The training of BearingPGA-Net is guided by a large-scale, pre-trained model through a decoupled knowledge distillation process. Despite its reduced size, the model achieves competitive performance relative to existing lightweight diagnostic networks. Furthermore, we implement an FPGA-accelerated version of BearingPGA-Net using Verilog. This implementation includes custom quantization and the design of dedicated logic circuits for each network layer. Key architectural features—such as parallel computation and component reuse—are incorporated to maximize throughput and computational efficiency.

In Chapter 7, we give a conclusion of this dissertation and provide our view for the future explorations.

Chapter 2

Literature Review

This chapter first establishes the theoretical framework for our study, beginning with fundamental concepts of bearing faults and their characteristic features. We then present a systematic review of current signal processing and deep learning approaches in machinery PHM, including a critical comparative analysis of existing models.

The review subsequently focuses on three key research domains relevant to this dissertation: i) Blind deconvolution for signal feature extraction; ii) Polynomial neural network architectures; iii) Lightweight fault diagnosis models for edge devices deployment.

2.1 Fault categories and characteristics of rolling bearings

Rotating machinery predominantly employs two types of rolling element bearings: deep groove ball bearings and angular contact ball bearings.

Deep groove ball bearings, characterized by their simple structure and extended service life, are widely implemented in common rotating machinery applications, such as gearboxes, electric motors, and household appliances. Notably, the majority of existing fault diagnosis datasets have utilized deep groove ball bearings for experimental validation (Lessmeier et al., 2016; B. Wang et al., 2018).

In contrast, angular contact ball bearings demonstrate superior load-bearing capacity, accommodating both higher radial and axial loads while achieving greater rotational speeds compared to their deep groove counterparts. These enhanced performance characteristics make them particularly suitable for high-speed, heavy-load applications such as aircraft engine spindles, industrial machine tools, and high-frequency motors.

Given their distinct operational advantages and widespread industrial applications, this study focuses on both deep groove ball bearings and angular contact ball bearings as primary research subjects.

As illustrated in Figure 2.1 (left), the basic architecture of rolling element bearings comprises four primary components: inner race, outer race, rolling elements (balls or rollers), and cage. During operation, the continuous frictional contact between the rolling elements and the raceways of both inner and outer rings creates three critical fatigue-prone zones. These are susceptible to various mechanical defects, including spalling (surface fatigue), plastic deformation, and pitting corrosion. A representative case of spalling damage is demonstrated in Figure 2.1 (right).

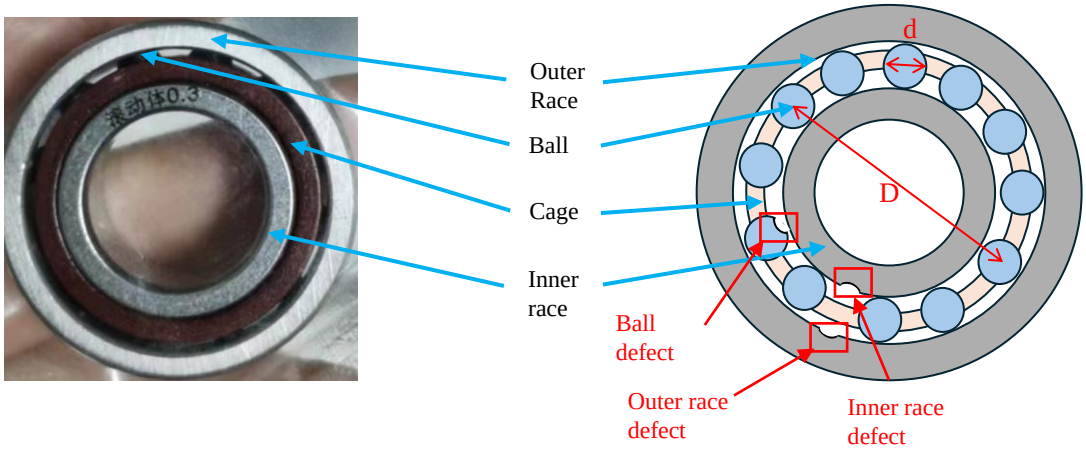


Figure 2.1: Structure of rolling bearings and their faults.

During operation, rolling element bearings generate characteristic vibration patterns that serve as effective indicators of their mechanical condition. In industrial practice, triaxial accelerometers are typically employed to capture these vibration signals, which contain crucial information about defect locations within the bearing assembly.

The underlying detection principle relies on the transient impulse response generated when rolling elements pass over defect sites. Each defect type produces distinctive frequency components determined by the defect location (inner race, outer race, or rolling element), the bearing's geometric parameters, and operational rotational speed. These characteristic defect frequencies can be theoretically calculated using the following equations.

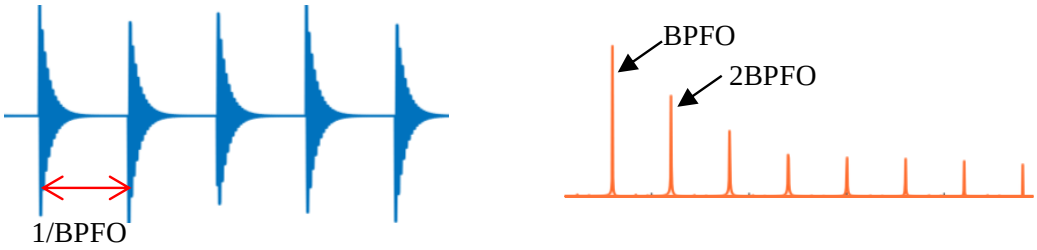
(1) Ball pass frequency outer race (BPFO) represents the characteristic frequency generated when rolling elements pass over a defect in the outer race. This frequency is determined by the bearing's

geometric parameters and rotational speed, as expressed by:

$$\text{BPFO} = \frac{mf_r}{2}\left(1 - \frac{d}{D}\cos\phi\right). \tag{2.1}$$

Where m denotes the number of rolling elements, f_r is the rotational frequency (Hz), d denotes the diameter of rolling elements (mm), D represents the pitch diameter (mm), and ϕ is the contact angle (degrees).

Figure 2.2 demonstrates the vibration signature of an outer race defect. The time-domain signal exhibits periodic transient impulses corresponding to each rolling element's passage over the defect location (McFadden & Smith, 1984; Randall & Antoni, 2011; Randall, 2021). In the frequency domain, the spectrum is characterized by a dominant BPFO component and its harmonic frequencies (n × BPFO).

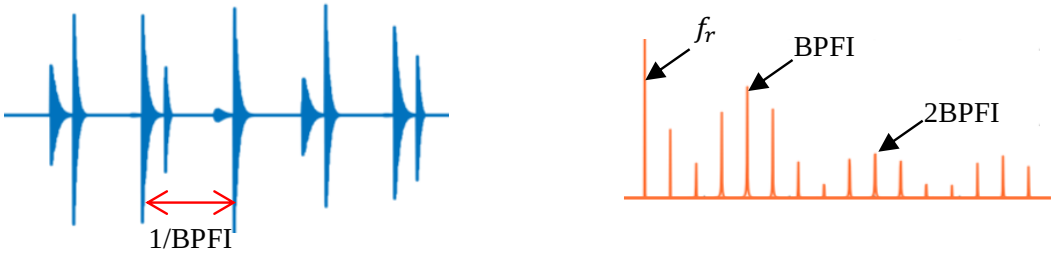


(a) Time domain signal (b) Envelope spectra of signal

Figure 2.2: Outer race fault signal and its envelope spectrum.

(2) Ball pass frequency inner race (BPFI). The generation mechanism of inner race fault signals is similar to that of outer race faults, and the expression of their characteristic frequency is as follows:

$$\text{BPFI} = \frac{mf_r}{2}\left(1 + \frac{d}{D}\cos\phi\right). \tag{2.2}$$



(a) Time domain signal (b) Envelope spectra of signal

Figure 2.3: Inner race fault signal and its envelope spectrum.

The theoretical inner race fault signal is illustrated in Figure 2.3. As the inner race rotates synchronously with the shaft, the passage of rolling elements over a raceway defect generates periodic impulse responses that are frequency-modulated by the shaft's rotational frequency f_r. Consequently, the spectrum exhibits fundamental BPFI frequencies, harmonics, and sideband components with a modulation frequency equal to the rotational frequency.

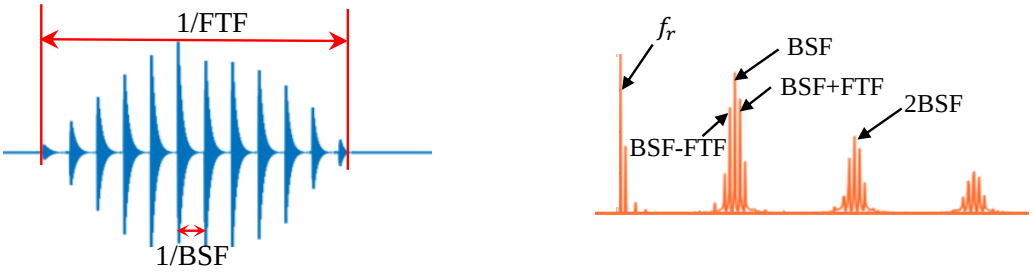
(3) Ball spin frequency (BSF). Because rolling elements not only rotate about their own axes but also revolve between the inner and outer races, their characteristic frequency depends on both the bearing's geometric parameters and rotational speed. The BSF is expressed as:

$$\mathrm{BSF} = \frac{D f_r}{2d}\left(1 - \left(\frac{d}{D}\cos\phi\right)^2\right). \tag{2.3}$$

Furthermore, unlike inner and outer race faults, rolling element defects exhibit distinct spectral characteristics due to additional kinematic constraints imposed by the cage. The resulting vibration spectrum incorporates both the BSF and the fundamental train frequency (FTF), which is given by:

$$\mathrm{FTF} = \frac{f_r}{2}\left(1 - \frac{d}{D}\cos\phi\right). \tag{2.4}$$

Figure 2.4 presents the time-domain signal and envelope spectrum of a rolling element fault, showing characteristic components including: i) Fundamental frequency and harmonics of BSF, and ii) sidebands modulated at FTF.



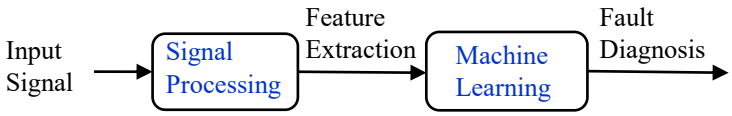
(a) Time domain signal (b) Envelope spectra of signal

Figure 2.4: Roll element fault signal and its envelope spectrum.

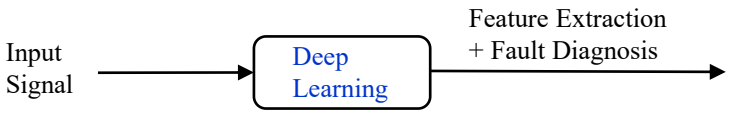
The distinctive features of bearing fault signals enable initial classification of defect types (inner race, outer race, or rolling elements). However, since characteristic frequencies depend on bearing dimensions and operating speeds, manual frequency calculation proves impractical for fault diagnosis. This necessitates intelligent diagnostic models to ensure accurate fault detection in rolling bearings.

2.2 Signal processing-empowered neural networks

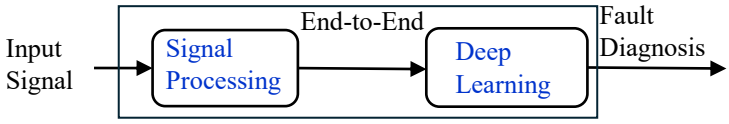
The development of intelligent PHM technology for rolling bearings is closely linked to advancements in artificial intelligence (AI). As AI has progressed from machine learning to deep learning, intelligent PHM methods have also undergone significant breakthroughs. These models integrate domain-specific features of fault detection into general AI frameworks. By incorporating fault mechanism models and signal processing techniques into intelligent models, they achieve high-accuracy fault diagnosis (Lei et al., 2020). Currently, intelligent PHM models can be categorized into three distinct paradigms, as illustrated in Figure 2.5. The introductions of these paradigms are as follows.



(a) **Paradigm 1:** Signal Processing + Machine Learning



(b) **Paradigm 2:** Deep Learning



(c) **Paradigm 3:** Signal Processing + Deep Learning

Figure 2.5: Three paradigms of machinery PHM models.

(1) Paradigm 1: Signal processing coupled with machine learning. This paradigm first employs signal processing methods—such as the wavelet transform and short-time Fourier transform—to extract relevant fault features from the measured signals. Those features are then input into some traditional machine learning models, including k-nearest neighbors (KNN) (R. Liu et al., 2018), artificial neural networks (ANN) (Hornik, Stinchcombe, & White, 1989), and support vector machines (SVM) (M. Wang et al., 2021) for fault diagnosis. For instance, Yaqub, Gondal, and Kamruzzaman (2011) proposed an adaptive feature extractor based on higher-order cumulants and wavelet transform and combined it with a KNN classifier to construct a fault detection framework. Their approach achieved a 90% diagnostic accuracy on the Case Western Reserve University (CWRU) bearing fault dataset under −10 dB noise conditions. Similarly, Gunerkar, Jalan, and Belgamwar (2019) employed a wavelet transform to extract time-domain fault features from raw vibration signals, which were subsequently fed into an ANN to verify the method's efficacy in diagnosing compound faults. In another study, M.

Wang et al. (2021) proposed using the Hilbert–Huang transform to extract bearing signal features and optimized the SVM model via particle swarm optimization, reaching a diagnostic accuracy of 98% on the CWRU dataset.

The primary advantage of this paradigm lies in the robust mathematical foundations in both signal processing modules and machine learning classifiers. Therefore, the model's inference process is inherently transparent, resulting in better interpretability compared to purely data-driven methods.

(2) Paradigm 2: End-to-end deep learning models. Around 2010, deep learning was widely introduced into the field of bearing PHM, resulting in notable advances. Deep learning relies on deep neural networks that enable a single model to perform both automatic feature extraction and fault diagnosis. The hierarchical stacking of multiple neural network layers enhances the model's non-linear representation capabilities, yielding performance in various regression and classification tasks that surpasses traditional machine learning approaches (LeCun, Bengio, & Hinton, 2015).

However, deep learning models are often criticized as "black boxes" as the extracted features do not correspond directly to the physical characteristics associated with bearing faults. This lack of interpretability can reduce the trustworthiness of the diagnostic outcomes (Ruqiang, Zheng, et al., 2024; Ruqiang, Zuogang, et al., 2024).

(3) Paradigm 3: Signal processing coupled with deep learning models. In this paradigm, a signal processing module is first used to extract fault features, and a deep learning module is subsequently cascaded to perform fault recognition. In this dissertation, we refer to this integrated technique as signal signal processing-empowered neural networks. Notably, a neural network is employed to construct the filter within the signal processing stage, and both modules are trained simultaneously. This concurrent training enables the signal processing module to automatically extract the most effective features for the classifier, thereby achieving higher diagnostic accuracy and interpretability.

Compared with methods that combine signal processing with traditional machine learning, replacing the machine learning module with a deep learning module significantly enhances the ability to learn from datasets with a larger number of fault categories, thus markedly improving diagnostic accuracy.

Moreover, compared to end-to-end deep learning approaches, incorporating an additional signal processing module enhances the model's capability to extract robust features in high-noise environments, thereby further improving diagnostic performance. In addition, because the signal processing

module extracts features based on the intrinsic characteristics of the fault signals, its outputs are more directly related to fault characteristics. This advantage significantly improves model interpretability.

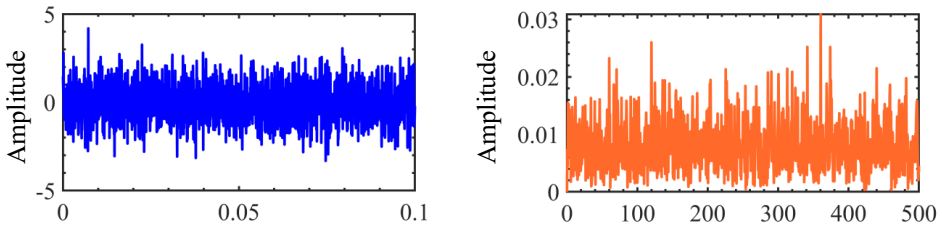
This thesis concentrates on Paradigm 3 to develop intelligent PHM models that offer both high diagnostic accuracy and robust interpretability. In the following sections, we introduce several representative methods within Paradigm 3. For fault diagnosis, we highlight envelope analysis-based (Lu et al., 2024; Yin et al., 2023), wavelet transform-based (Frusque & Fink, 2022; Xiong et al., 2020), and short-time Fourier transform-based neural networks (Q. Chen et al., 2024; C. He et al., 2024). For RUL predictions, we present feature transformation-based neural networks (Javed et al., 2015; L. Liu et al., 2021).

2.2.1 Envelope analysis-based neural networks

Envelope analysis is the benchmark method for analyzing bearing fault signals (Randall & Antoni, 2011; Randall, Antoni, & Chobsaard, 2001). The primary procedure involves applying the Hilbert transform (HT) to the signal to generate an envelope signal, and then executing the fast Fourier transform (FFT) (Bonnardot et al., 2004; N. E. Huang et al., 1998) on the envelope signal to obtain its envelope spectrum (ES). We have illustrated the ES of various bearing fault signals in Figures 2.2–2.4. The envelope analysis method accentuates the characteristic frequencies associated with bearing faults.

Due to the significant noise in measured signals, directly computing ES can not yield clear fault-characteristic frequencies, as illustrated in Figure 2.6. Therefore, it is necessary to pre-process the signal using an appropriate filtering method. Spectral kurtosis (SK) is one of the most widely utilized techniques for identifying optimal filter frequency bands (Antoni, 2006; Vrabie, Granjon, & Serviere, 2003; Y. Wang et al., 2016). Among the various approaches, the Fast Kurtogram proposed by Antoni and Randall (2006) is particularly representative. This method employs finite impulse response (FIR) filters or short-time Fourier transform (STFT) filters to process the signal across different frequency bands, synthesizing the outputs into a filter bank representation diagram. The optimal frequency band filter is then selected by calculating the kurtosis values of these filters.

Given that the ES is highly effective at extracting prominent bearing fault features, many approaches have integrated ES analysis with neural networks to enhance fault diagnosis performance. For instance, X. Wang et al. (2015) extracted fault features using the ES and then fed these into a



(a) Noisy signal (b) Envelope spectra of the signal

Figure 2.6: Noisy fault signal and its envelope spectrum.

deep belief network (DBN) for fault diagnosis, achieving a 99% accuracy on the CWRU bearing fault dataset. Yin et al. (2023) applied envelope analysis to extract both the frequency features and the corresponding frequency band ranges from fault signals, followed by a band-pass filter based on convolutional neural networks that was trained jointly with the model, thereby attaining a diagnostic accuracy of 91% on a cross-domain fault diagnosis task for industrial robot rotary vector reducers. Lu et al. (2024) introduced the envelope spectrum neural network (ESNN) , which employs convolutional neural networks as low-pass and band-pass filters in conjunction with the envelope spectrum computation process to achieve adaptive filtering and feature extraction, resulting in a 6.4% improvement in diagnostic accuracy.

2.2.2 Wavelet transform-based neural networks

Wavelet transform (WT) is one of the most commonly employed techniques in signal processing–empowered neural networks, due to its ability to analyze time-varying, nonstationary signals. This makes WT particularly effective for extracting bearing fault features in scenarios involving variable speed conditions. By applying scaling and translation factors to a finite-length or rapidly decaying "mother wavelet", WT enables multi-resolution signal analysis, facilitating precise feature extraction for improved fault diagnosis.

The basis function of the wavelet transform $\Psi(t)$ is given as follows:

$$\Psi_{u,s}(t) = \frac{1}{\sqrt{u}}\Psi\left(\frac{t-s}{u}\right). \tag{2.5}$$

Where u is the scaling factor and s is the translation factor.

The WT realizes multi-resolution by adjusting the scaling parameter. At high frequencies, a smaller scaling factor results in a narrower time window, improving time resolution while reducing

frequency resolution; At low frequencies, a larger scaling factor leads to a wider time window, enhancing frequency resolution while lowering time resolution. This property ensures that WT effectively balances time and frequency resolution, making it particularly well-suited for analyzing nonstationary bearing fault signals. Figure 2.7 illustrates the time-frequency resolution properties of WT.

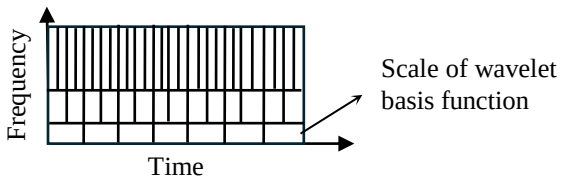


Figure 2.7: Time-frequency resolution properties of wavelet transform.

To integrate WT with deep learning models for joint training, some approaches construct wavelet basis function parameters or wavelet filters using convolutional neural networks. These methods are collectively referred to as wavelet neural networks (WNN) (Alexandridis & Zapranis, 2013).

In the field of fault diagnosis, a notable example is the wavelet kernel network (WaveKernelNet), proposed by T. Li et al. (2022). This method applies multiple wavelet basis functions with different parameters to convolve the input signal, as illustrated in Figure 2.8. A neural network is employed to learn the scale and translation parameters of the basis functions, after which feature maps with varying resolutions are forwarded to a subsequent deep learning classifier. This module enhances the diagnostic accuracy of deep learning methods by approximately 1%.

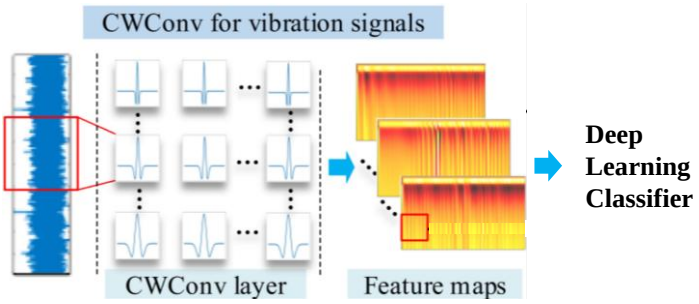


Figure 2.8: Multiple wavelet basis functions convlutional layer (T. Li et al., 2022).

Building upon this module, C. He et al. (2023) introduced an enhanced wavelet weight initialization network. Unlike directly modifying the first convolutional layer, this method initializes the first-layer weights of the neural network using wavelet basis functions, effectively incorporating wavelet parameters into the network and improving diagnostic accuracy by approximately 10%. H. Wang et al. (2023) combined discrete WT with an attention mechanism, proposing the Multilayer Wavelet

Attention Convolutional Neural Network (MWA-CNN) . This approach achieved 98% accuracy in diagnosing faults in high-speed aerospace bearings. Additionally, the multi-layer decomposition and reconstruction properties of wavelets enable the design of interpretable deep neural networks, such as wavelet packet transform-based neural networks (Frusque & Fink, 2022; Xiong et al., 2020).

2.2.3 Short-time Fourier transform-based neural networks

Short-time Fourier transform (STFT) has been widely applied in the field of bearing fault diagnosis. The process begins by segmenting the original signal using a window function, then obtaining localized signal segments:

$$x_\tau(t) = x(t) \cdot w(t-\tau), \tag{2.6}$$

where $w(t-\tau)$ represents the window function, and τ denotes the time frame.

Subsequently, the Fourier transform is performed on each windowed signal segment to obtain the frequency distribution in the vicinity of the given time frame. The discrete formulation is expressed as:

$$\hat{X}[m,k] = \sum_{n=0}^{N-1} x[n]w[n-m]e^{-j\frac{2\pi}{N}kn}, \tag{2.7}$$

where m is the time frame index and k is the frequency index.

A commonly used window function is the Hamming window, whose discrete form is given by:

$$w[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{L-1}\right),\ 0 \le n \le L-1. \tag{2.8}$$

In contrast to WT, Eq. (2.8) demonstrates that the time window length in STFT remains fixed. The time-frequency resolution property of STFT is illustrated in Figure 2.9.

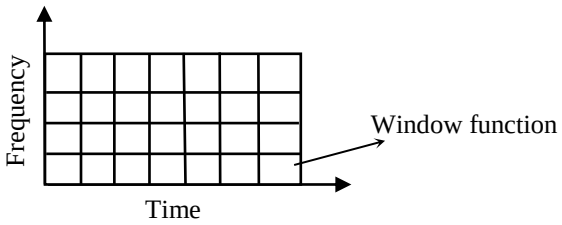


Figure 2.9: Time-frequency resolution property of short-time Fourier transform.

Therefore, STFT provides a more straightforward computational framework with fewer adjustable parameters, making it advantageous in applications requiring simplicity and efficiency.

Similar to WNN, the window function in STFT can be substituted with convolutional kernels, enabling the development of STFT-based neural networks. C. He, Shi, and Li (2023) introduced the interpretable and differentiable STFT cross-machine dual-driven adaptation network (IDSN), which leverages automatic backpropagation in neural networks to adjust the window length parameter of the STFT function. This approach achieved a remarkable 99% accuracy in bearing fault diagnosis under variable speeds and across multiple datasets. Building on this advancement, C. He et al. (2024) further refined the differentiable STFT, proposing the modulated differentiable STFT (MDSTFT). This method incorporates multiple learnable window functions of varying lengths and introduces a balanced spectrum quality (BSQ) regularization loss function, reaching approximately 97% accuracy in diagnosing faults in wheel bearings of variable-speed freight trains. Additionally, Q. Chen et al. (2024) integrated STFT, Chirplet transform, Morlet wavelet, and Laplace wavelet into a time-frequency convolutional layer, leading to the development of the time-frequency network (TFN). This approach demonstrated exceptional performance, achieving over 99% accuracy in small-sample fault diagnosis on the CWRU dataset.

In addition to the above methods, modern signal processing methods also have great potential in building accurate and interpretable bearing fault diagnosis neural networks:

(1) Sparsity measures serve as fundamental indicators for objectively quantifying the characteristic impulsiveness of data sequences, which is vital for fault feature characterization in the condition monitoring and diagnosis of rotating machinery. Traditional metrics, including kurtosis (Kurt) (Wiggins, 1978), negentropy (NE) (Pitkänen, 2006), and the D-norm (C.-S. Wang, Tang, & Zhao, 1991). These indicators have been applied to Blind Deconvolution (BD) (Miao, Zhang, Lin, et al., 2022) for capturing the transient impulse in the signal. However, their maximum values often manifest when the signal contains strong random impulse noise.

(2) Cyclostationary (CS) analysis is a method for extracting periodic or quasi-periodic components from non-stationary signals. Antoni et al. (2004) and Randall, Antoni, and Chobsaard (2001) demonstrated that fault signals from rotating machinery are approximately cyclostationary or an almost-cyclostationary (ACS) process (Napolitano, 2016). Bearing fault signals exhibit second-order cyclostationarity (CS2), which implies that their time-varying autocorrelation function is periodic.

Spectral correlation (SC) and spectral coherence (SCoh) are classical methods for CS analysis (Antoni, 2007b; Antoni, Xin, & Hamzaoui, 2017; Borghesani & Antoni, 2018; Randall, Antoni, & Chob-

saard, 2001). These approaches generate bi-frequency maps to illustrate the relationship between frequency, cyclic frequency, and amplitude within the vibration signal. However, the direct computation of SC is computationally prohibitive as it requires estimation over a wide range of cyclic frequencies (Antoni, Xin, & Hamzaoui, 2017). In recent years, scholars have proposed methods for the rapid computation of SC, including the periodogram (Antoni, 2007a), cyclic modulation spectrum (CMS) (Antoni & Hanson, 2012; Borghesani, 2015), fast spectral correlation (Fast SC) (Antoni, Xin, & Hamzaoui, 2017), and faster SC (Borghesani & Antoni, 2018).

(3) Interpretable weight optimization uses data-driven optimization not only to classify machine states but also to learn an "interpretable weight spectrum" that physically reflects a machine's fault characteristics. The foundational method, the optimized square envelope spectrum (OSES) proposed by Hou et al. (2022), reframes fault diagnosis as a convex optimization problem. Its key innovation is constraining the model to assign high positive weights to fault-specific frequencies, thereby making the weight parameters directly interpretable. Subsequently, T. Yan et al. (2025) provided a deeper theoretical justification for this "weight spectrum" concept and extended it to the time-frequency domain. They demonstrated that the resulting weight matrix w is not merely correlated with the fault but is a mathematical representation of the STFT of the fault component itself. To mitigate the frequency sensitivity of the original OSES method, Y. Wang et al. (2025) later proposed the robust optimized weights spectrum (ROWS). These weight optimization techniques effectively serve as enhanced, automatically filtered spectra, enabling the development of more interpretable and reliable tools for machinery fault diagnosis.

2.2.4 Feature extraction for remaining useful life prediction

Remaining useful life (RUL) prediction estimates the time remaining until a machine reaches the end of its operational lifespan (Nemani et al., 2023). Accurate and timely RUL prediction is crucial for optimizing maintenance strategies, minimizing machinery downtime, and facilitating cost-effective production planning. However, compared to fault diagnosis, forecasting future system states presents a significant challenge and is regarded as the most difficult task in PHM (X. Li, Zhang, & Ding, 2019; W. Peng, Ye, & Chen, 2019; Zhu et al., 2022).

RUL prediction models typically consist of two key steps: feature extraction and state prediction. Unlike fault diagnosis models, which employ signal processing techniques such as WT and STFT to

extract fault-related features, RUL models emphasize degradation trends within the signal. Consequently, various statistical functions, including root mean square (RMS) and kurtosis, are utilized to extract statistical features, referred to as health indicators (HI).

Mathematically, let the entire signal be represented as $\boldsymbol{X} \in \mathbf{R}^N$, which is partitioned into segments $[\boldsymbol{x}_1, \cdots, \boldsymbol{x}_t]$, $\boldsymbol{x}_i \in \mathbf{R}^n$. The HI is defined as a mapping function:

$$h_i = f(\boldsymbol{x}_i). \tag{2.9}$$

The feature extraction in RUL prediction is illustrated in Figure 2.10. It is obvious that HI provides a clearer representation of degradation trends compared to raw vibration signals.

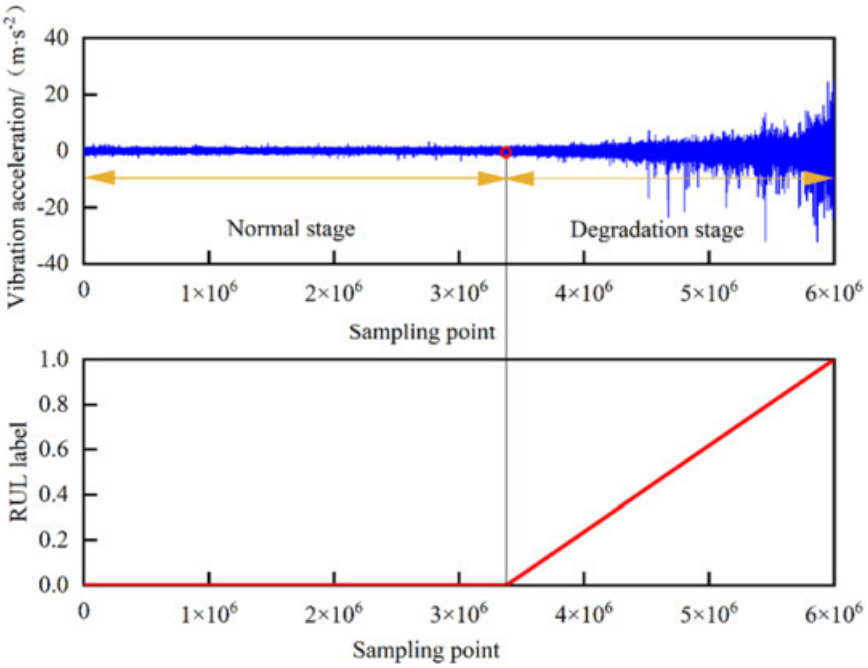


Figure 2.10: The process of feature extraction in RUL (Cai et al., 2024).

Existing feature extraction methods primarily focus on improving the trendability and monotonicity of derived features, thereby facilitating more accurate predictions (K. He et al., 2022; Javed et al., 2015; Lei et al., 2018). In general, vibration signals are transformed into statistical representations in the time, frequency, and time-frequency domains to construct predictive models (Javed et al., 2015). For instance, M. Yan et al. (2020) introduced two novel time-domain statistical features—relative root mean square (RRMS) and inertial relative root mean square (IRRMS)—which enhance degradation trends while mitigating random fluctuations. Similarly, T. Yan et al. (2022) proposed a weighted health index that fused spectrum amplitudes with weights to provide a monotonically degradation

trend. Javed et al. (2015) proposed a feature extraction approach that leverages discrete wavelet transform, supplemented by trigonometric functions and cumulative transformation. Building on this, L. Liu et al. (2021) validated the efficacy of Javed's method by integrating an attention-based encoder-decoder framework for RUL prediction.

Following the construction of HI, neural networks are employed to develop predictive models. Given that RUL prediction is highly dependent on temporal dynamics, specialized architectures have been designed to effectively capture time-dependent patterns in data. In particular, temporal convolutional networks (TCN) (Bai, Kolter, & Koltun, 2018), long short-term memory (LSTM) networks (C.-G. Huang, Huang, & Li, 2019), and gated recurrent units (GRU) (Bahdanau, Cho, & Bengio, 2014; Y. Chen et al., 2020) are widely utilized due to their capacity to model sequential dependencies and enhance prediction accuracy in time-series analyses.

2.3 Related works of proposed methodologies

In this section, we introduce several key methods that are related to this research. Different from signal processing-empowered neural networks introduced earlier, the proposed methodologies primarily rely on two fundamental techniques: blind deconvolution and polynomial neural networks.

2.3.1 Blind deconvolution

Blind deconvolution (BD) is an adaptive filtering technique that updates filter coefficients automatically through iterative optimization algorithms, guided by an objective function designed to capture the general characteristics of the signal. "Deconvolution" refers to the process in which the coefficient matrix of the filter serves as the inverse of the transfer function (including transmission paths and noise), thereby mitigating its impact on the original signal. The term "blind" indicates that this method operates without prior knowledge of the source signal, relying solely on the input signal to reconstruct the source signal (Haykin, 1986). Currently, BD has been widely applied across various domains, including image processing, seismic signal analysis, electroencephalography (EEG), and fault detection in bearings (Levin et al., 2011; Miao, Zhang, Lin, et al., 2022; Sanghvi, Chi, & Chan, 2024).

To further enhance the feature extraction capability and stability of BD, contemporary research

efforts focus on refining the methodology through the design of novel objective functions and the development of optimization algorithms (Miao, Zhang, Lin, et al., 2022).

2.3.1.1 Objective functions

(1) Kurtosis objective functions. These functions refer to the use of extended kurtosis-based objective functions. These functions characterize the transient impulsive nature of fault signals.

The earliest BD method was minimum entropy deconvolution (MED), proposed by Wiggins (1978). MED employs kurtosis (Pearson, 1905) as an optimization criterion, refining FIR filter coefficients based on kurtosis values. However, MED is susceptible to random impulse interference, potentially leading to the extraction of isolated impulses unrelated to faults (B. Zhang et al., 2021).

To address this limitation, McDonald, Zhao, and Zuo (2012) introduced maximum correlated kurtosis deconvolution (MCKD), which incorporates fault signal periodicity by using correlated kurtosis (CK) as the optimization objective. Subsequent advancements, such as Sparse Maximum Harmonics-to-Noise Ratio Deconvolution (SHMD) (Miao et al., 2016) and Multipoint Optimal MED Adjusted (MOMEDA) (McDonald & Zhao, 2017), were developed to improve anti-noise robustness.

Moreover, recent advancements include l_0 norm MED proposed by (X. Jiang et al., 2018), adaptive multipoint optimal MED adjusted (AMOMEDA) introduced by Y. Cheng, Chen, and Zhang (2019), and maximum average kurtosis deconvolution (MAKD) proposed by Liang et al. (2021). These approaches share a common fundamental objective: integrating fault periodicity into kurtosis functions to enhance signal extraction and improve diagnostic precision.

(2) Cyclostationarity objective functions. These functions are designed to capture the periodic characteristics of fault signals. A notable method in this category is second-order cyclostationarity blind deconvolution (CYCBD), introduced by Buzzoni, Antoni, and d'Elia (2018). This approach employs cyclostationary criteria as the objective function, significantly enhancing the model's ability to extract periodic features.

However, cyclostationary criteria require prior knowledge of fault cycle frequencies, prompting subsequent methods to focus on the automatic detection of signal periodicity. B. Zhang et al. (2021) proposed adaptive second-order cyclostationarity blind deconvolution (ACYCBD), integrating envelope harmonic product spectrum (EHPS) to detect cyclic frequencies and refine CYCBD. Z. Wang et al. (2022) introduced adaptive maximum second-order cyclostationarity blind deconvolution

(AMCBD), utilizing morphological envelope autocorrelation functions for cycle frequency detection. More recently, D. Peng et al. (2023) developed generalized Gaussian cyclostationarity blind deconvolution (CYCBD$_\beta$), addressing the limitations of cyclostationary objective functions in handling non-Gaussian noise.

These advancements collectively improve the robustness and adaptability of BD techniques in fault signal extraction.

(3) Sparsity Objective Functions. Bearing fault signals exhibit high sparsity due to their transient impulsive characteristics. Consequently, maximizing sparsity-based objective functions enables BD to enhance signal sparsity effectively. These objective functions are typically formulated as the ratio of two norms, with common statistical indicators such as kurtosis, skewness (Lei et al., 2008), and the l_1/l_2 norm (D. Wang et al., 2018) widely employed in BD applications.

A generalized sparsity function was introduced by L. Li (2014), leading to the incorporation of the l_p/l_q norm into BD. Build upon this, X. Jia, Zhao, Buzza, et al. (2017), X. Jia, Zhao, Di, et al. (2017), and X. Jia et al. (2018) explored the geometric properties of the generalized l_p/l_q norm for BD and proposed a multi-layer filtering structure. Subsequently, Peeters, Antoni, and Helsen (2020) introduced an l_2/l_1 norm based on envelope spectrum as a BD objective function for fault feature extraction in bearings and gearboxes. Further advancements include: L. He, Wang, et al. (2021) optimized signal envelope spectra using a generalized l_p/l_q norm, demonstrating its effectiveness in extracting cyclostationary components. Q. Zhou et al. (2021) proposed the improved correlated generalized l_p/l_q norm (ICG-l_p/l_q), extending the CK function to address compound fault diagnosis. X. Gu et al. (2022) introduced a multi-sparsity function objective, considering both impulse characteristics and cyclostationarity in the time domain. More recently, Z. Zhang et al. (2023) developed a multi-dimensional BD method based on cross-sparsity filtering, incorporating a generalized Sigmoid function.

In summary, sparsity objective functions enable the extraction of fault-related transient impulses without requiring additional prior information, making it a highly relevant and increasingly studied approach in recent years.

2.3.1.2 Optimization methods

(1) Eigenvector algorithm (EVA) iteratively solves the coefficients of BD filters using an eigenvalue equation, making it particularly suitable for single-objective function optimization (Lee & Nandi,

2000). Several methods, such as CYCBD (Buzzoni, Antoni, & d'Elia, 2018), the l_2/l_1 norm (Peeters, Antoni, & Helsen, 2020), and MAKD (Liang et al., 2021), have all adopted the EVA approach. However, EVA requires the objective function to have an analytical solution, which imposes constraints on the selection of objective functions in BD.

(2) Particle swarm optimization (PSO) has been applied to BD by Y. Cheng et al. (2018), who conceptualized it as an unconstrained multi-objective optimization problem spanning an N-dimensional Euclidean sphere. Their findings indicate that using PSO improves filter performance. This has led to the development of several PSO-based approaches, including PSO-MED, PSO-MCKD, PSO-MOMEDA (Y. Cheng, Wang, et al., 2019), and PSO-maximum impulse norm deconvolution (PSO-MIND) (Y. Cheng, Chen, Mei, et al., 2019). However, the effectiveness of PSO is heavily dependent on the search space, with larger search spaces negatively impacting computational efficiency.

(3) Backpropagation (BP). As BD can be treated as an unsupervised learning approach (Haykin, 1986), several studies have explored multilayer and multi-channel convolutional neural networks to optimize BD filters, utilizing BP to efficiently search for global optimal solutions. For instance, Du, Chen, and Zhang (2018) proposed the convolutional sparse learning model (ConvSLM), employing the ADMM optimizer to solve blind deconvolution. L. He, Yi, et al. (2021) introduced a blind deconvolution method based on fully connected neural networks with multi-node sparse structures. B. Fang et al. (2021, 2022) developed a blind deconvolution framework using convolutional neural networks, while Z. Zhang et al. (2023) proposed a multidimensional blind deconvolution approach based on cross-sparse filters. Studies have found that BP-based BD methods exhibit greater stability during optimization, leading to increasing interest in recent years.

2.3.2 Polynomial neural networks

The origins of high-order neural networks—commonly referred to as polynomial neural networks—can be traced back to the 1970s. One of the earliest developments in this area was the Group Method of Data Handling (GMDH), introduced by Ivakhnenko (1971), which employed polynomial-based structures for automatic feature extraction. Building upon this foundation, Shin and Ghosh (1991) later proposed the pi-sigma neural network architecture, which explicitly integrates higher-order poly-

nomial functions within its computational framework:

$$y_i = \sigma(\prod_j (\sum_k w_{kji} x_k + b_{ji})). \tag{2.10}$$

where w, b are learnable parameters, $\sigma(\cdot)$ represents the activation function, and x denotes the input. The high-order polynomials are implemented by multiplying several linear functions.

Recent progress in deep learning has reignited interest in incorporating polynomial operators within both fully connected and convolutional neural network architectures. These integration strategies generally fall into two main categories: polynomial structures and polynomial neurons. The former involves enhancing the network architecture through recursive polynomial expansion (Chrysos et al., 2023) or leveraging tensor decomposition techniques (Y. Cheng et al., 2024; Chrysos et al., 2022). In contrast, the latter replaces the conventional linear activation function of standard neurons with polynomial expressions of varying forms (F. Fan, Cong, & Wang, 2018; Z. Xu et al., 2022). The present work concentrates on the neuron-level approach.

Although polynomial neurons can theoretically support high-order terms, increasing the polynomial degree tends to introduce substantial computational overhead and instability during training. To balance expressiveness and efficiency, most implementations constrain the order to two, resulting in what is commonly known as a quadratic neural network.

A quadratic network is constructed by replacing the linear function with a quadratic function (F. Fan, Cong, & Wang, 2018):

$$y = \sigma((\boldsymbol{W}_1^\top \boldsymbol{x} + b_1)(\boldsymbol{W}_2^\top \boldsymbol{x} + b_2) + \boldsymbol{W}_3^\top (\boldsymbol{x} \odot \boldsymbol{x}) + b_3), \tag{2.11}$$

where $\odot$ is Hadamard product.

It is worth highlighting that many formulations of quadratic neurons have been introduced in prior research, as summarized in Table 2.1. In this thesis, we adopt the design proposed by F. Fan, Cong, and Wang (2018). Compared to alternative variants such as those by Z. Xu et al. (2022) and Goyal, Goyal, and Lall (2020), this formulation is more general in nature—it incorporates both the full inner-product term and an independent power term, providing greater expressive flexibility.

Table 2.1: A summary of the recently-proposed non-linear neurons. $\sigma(\cdot)$ denotes the nonlinear activation function. $\odot$ denotes Hadamard product. Here, $\boldsymbol{W} \in \mathbb{R}^{n\times n}$, $\boldsymbol{w}_i \in \mathbb{R}^{n\times 1}$, and we omit the bias terms in these neurons.

Authors	Formulations
Micikevicius et al. (2018) and Zoumpourlis et al. (2017)	$\mathbf{y} = \sigma(\boldsymbol{x}^\top \boldsymbol{W}\boldsymbol{x} + \boldsymbol{w}^\top \boldsymbol{x})$
Y. Jiang et al. (2020) Mantini and Shah (2021)	$\mathbf{y} = \sigma(\boldsymbol{x}^\top \boldsymbol{W}\boldsymbol{x})$
Goyal, Goyal, and Lall (2020)	$\mathbf{y} = \sigma(\boldsymbol{w}^\top (\boldsymbol{x} \odot \boldsymbol{x}))$
Bu and Karpatne (2021) X. Ma et al. (2024)	$\mathbf{y} = \sigma((\boldsymbol{w}_1^\top \boldsymbol{x}) \odot (\boldsymbol{w}_2^\top \boldsymbol{x}))$
Z. Xu et al. (2022)	$\mathbf{y} = \sigma((\boldsymbol{w}_1^\top \boldsymbol{x})(\boldsymbol{w}_2^\top \boldsymbol{x}) + \boldsymbol{w}_3^\top \boldsymbol{x})$
F. Fan, Cong, and Wang (2018)	$\mathbf{y} = \sigma((\boldsymbol{w}_1^\top \boldsymbol{x})(\boldsymbol{w}_2^\top \boldsymbol{x}) + \boldsymbol{w}_3^\top (\boldsymbol{x} \odot \boldsymbol{x}))$
Babiloni et al. (2023)	$\mathbf{y} = \boldsymbol{w}_3^\top (\sigma(\boldsymbol{w}_1^\top \boldsymbol{x} \odot \boldsymbol{w}_2 \top \boldsymbol{x}) \odot \boldsymbol{x})$

2.3.3 Lightweight fault diagnostic models

2.3.3.1 Lightweight CNNs for bearing fault diagnosis

Some lightweight networks were proposed for deployment in resource-constrained environments. D. Yao et al. (2020) introduced the stacked inverted residual convolution neural network (SIRCNN), comprising one convolutional layer and three inverse residual layers. Similarly, H. Fang, Deng, Bai, et al. (2021) developed the lightweight efficient feature extraction network (LFEE-NET), which is a feature extraction module conjugated with a lightweight classification module. However, despite the simple structure of the classification network, its feature extraction network is complex. Other lightweight models incorporated a multi-scale model (W. Li et al., 2023) or a self-attention mechanism (Zhong et al., 2022), demonstrating the superior performance in handling few-shot or cross-domain issues. But these networks have not yet been deployed on embedded devices. Therefore, it remains unclear how much performance will be sacrificed when deployed.

2.3.3.2 Bearing fault diagnosis models on edge device

To deploy large models onto edge devices with limited storage and computational resources, knowledge distillation (KD) is widely employed to construct lightweight models (G. Hinton, Vinyals, & Dean, 2015). For example, Q. Huang et al. (2023) proposed the cheap ghost network (CGhostNet), which integrates fine-grained feature KD (FFKD) and has been successfully deployed on the LubanCat2 device. Additionally, Y. Wang et al. (2024) introduced a lightweight intelligent fault diagnosis framework based on adaptive knowledge distillation, wherein fault knowledge is transferred from a cloud-based deep neural network model (T-model) to an edge-based lightweight model (S-model).

The S-model has been deployed on the NVIDIA Jetson Xavier NX suite, demonstrating a 10.7% performance improvement over existing methods.

Despite the various types of devices available for fault diagnosis, such as microcontroller units (MCUs), neural processing units (NPUs), and field-programmable gate arrays (FPGAs), FPGAs offer the fastest processing speed and lowest power consumption, making them ideal for real-time fault diagnosis. However, only a limited number of models have been successfully deployed on FPGAs. For instance, Kang, Kim, and Kim (2014) proposed an FPGA-based multicore system for real-time bearing fault diagnosis using acoustic emission signals. Their design incorporated a high-performance multicore architecture with 64 processing units running on an Xilinx Virtex-7 FPGA to enable online processing, utilizing time-frequency analysis and support vector machines (SVM) for fault diagnosis. Additionally, Toumi et al. (2022) implemented an envelope spectrum and a multi-layer perceptron (MLP) structure on a ZYNQ-7000 FPGA, achieving approximately 90% accuracy on the CWRU dataset. Furthermore, Ji et al. (2022) applied knowledge distillation to train a single-layer convolutional neural student network and deployed it onto a ZYNQ FPGA through parameter quantization, resulting in an approximate 8% performance improvement compared to direct network training.

Despite these achievements, the task of deploying fault diagnosis models in FPGAs still have a large room for improvement, as their performance has not yet reached the level of the large model.

Chapter 3

Classifier-Guided Neural Blind Deconvolution: A Physics-Informed Denoising Module for Bearing Fault Diagnosis Under Heavy Noise

3.1 Introduction

ROTATING machinery—including systems such as aero engines, pumps, and wind turbines—serves as a foundational component in a wide array of industrial operations. Yet, the core mechanical elements that sustain rotational motion, particularly rolling bearings, are highly vulnerable to degradation over time, especially under conditions of elevated temperature, high rotational speed, and mechanical stress (Miao, Zhang, Lin, et al., 2022; Randall & Monitoring, 2011). Failures in these components, such as cage fractures or cracks in the bearing races, can trigger sudden machinery breakdowns, leading to unplanned downtime, financial losses, and in severe cases, catastrophic consequences. As such, the ability to detect and identify bearing faults promptly and accurately is vital for maintaining the performance and safety of rotating systems (Z. Wang et al., 2022).

However, a primary challenge in bearing fault diagnostics lies in the fact that the collected vibration signals are frequently corrupted by substantial background noise. This interference stems from the mechanical transmission system and external environmental sources, often involving vibrations from adjacent machinery. These complex noise sources tend to mask or distort the fault-related signal

characteristics, thereby complicating the task of reliable diagnosis. Consequently, devising robust methods to extract discriminative, fault-specific features from noisy sensor data remains a prominent area of research.

Contemporary feature extraction techniques for vibration signal analysis can be broadly divided into two categories: data-driven approaches and signal processing-based methods. Data-driven techniques, which allow for automated pattern recognition and feature learning, have seen considerable advancement in recent years (Gao et al., 2024; Shao et al., 2024; Su et al., 2024; Xiao et al., 2024). For instance, K. Zhou, Diehl, and Tang (2023) developed a deep convolutional generative adversarial network (DCGAN) coupled with a semi-supervised learning framework to infer intrinsic physical relationships between known and previously unseen fault types, addressing the challenge of limited labeled data in gear fault diagnosis. Despite their promising performance, these models often suffer from a lack of interpretability—an inherent issue tied to their "black-box" nature—which restricts their adoption in critical, real-world decision-making scenarios (F.-L. Fan et al., 2021).

In contrast, a wide range of signal processing techniques—grounded in rigorous mathematical theory—have been developed to extract meaningful features from noisy data. These include wavelet transform (WT) (Kankar, Sharma, & Harsha, 2011; Y. Xu et al., 2019), variational mode decomposition (VMD) (S. Chen et al., 2017; Miao, Zhang, Li, et al., 2022; H. Wang, Chen, & Zhai, 2024), spectral kurtosis (SK) (Antoni, 2006; Y. Wang et al., 2016), cyclostationary analysis (McDonald, Zhao, & Zuo, 2012), energy spectrum analysis (Z. Liao et al., 2021, 2022), and blind deconvolution (BD) (L. He, Wang, et al., 2021; Miao, Zhang, Lin, et al., 2022; Miao et al., 2017). Among these, BD stands out due to its adaptive nature and independence from assumptions about center frequency or filter bandwidth. These advantages make BD particularly well-suited for isolating repeated transient impulses—a key indicator of bearing faults (L. He, Yi, et al., 2021). As a result, this chapter focuses on leveraging BD-based techniques for fault diagnosis in rotating machinery.

When analyzing vibration signals, blind deconvolution (BD) aims to recover repetitive transient impulses—key indicators of bearing faults—by optimizing an adaptive finite impulse response (FIR) filter (Miao, Zhang, Lin, et al., 2022). A central challenge in BD lies in formulating an effective objective function that captures the essential features of fault signals, such as sparsity and periodicity. The pioneering work in this field was Minimum Entropy Deconvolution (MED), introduced by Wiggins (1978), which utilized kurtosis (Pearson, 1905) as the objective to extract non-stationary components

from noisy signals. However, while kurtosis is sensitive to outliers, it lacks the ability to differentiate between randomly occurring and cyclic impulses (Y. Cheng, Chen, & Zhang, 2019), limiting its effectiveness in many fault diagnosis tasks.

To address this limitation, subsequent research has explored a variety of objective functions designed to better capture periodic fault-related features. Examples include kurtosis-based methods such as Maximum Correlated Kurtosis Deconvolution (MCKD) (McDonald, Zhao, & Zuo, 2012) and Multipoint Optimal Minimum Entropy Deconvolution Adjusted (MOMEDA) (McDonald & Zhao, 2017); cyclostationarity-based approaches like Adaptive Cyclostationarity Blind Deconvolution (ACYCBD) (Z. Wang et al., 2022) and its extension using the generalized Gaussian distribution (CYCBD$_\beta$) (D. Peng et al., 2023); and sparsity-driven techniques such as the generalized l_p/l_q norm (Gong et al., 2024; L. He, Wang, et al., 2021; L. He, Yi, et al., 2021) and the generalized sigmoid function (Z. Zhang et al., 2023). Despite their methodological diversity, these approaches share a common objective: to isolate meaningful, fault-related features from vibration data corrupted by complex noise sources.

However, one persistent challenge remains unresolved—*how to seamlessly integrate BD with deep classifiers to perform end-to-end fault diagnosis.* Most existing BD techniques evaluate performance based solely on signal reconstruction quality, without fully considering downstream classification objectives. Although some attempts have been made to combine BD with convolutional neural networks (CNNs) (S. Wang & Xiang, 2020), the integration has often resulted in suboptimal performance—or even degraded classification accuracy. This is largely due to a mismatch in the optimization landscape: BD and CNN components are typically trained with different loss functions (e.g., sparsity-driven vs. cross-entropy) and distinct optimization algorithms (e.g., filter-based optimizers vs. stochastic gradient descent). This disjointed optimization process creates inconsistencies during training. For instance, a BD module might enhance signal periodicity beneficial for one fault class but inadvertently suppress inter-class differences, undermining fault severity discrimination.

To overcome these limitations, this chapter proposes a unified framework that employs neural networks to jointly perform blind deconvolution and classification. At the core of this approach is a neural BD module, adapted from the Multi-task Neural Network Blind Deconvolution (MNNBD) architecture (J.-X. Liao, Dong, Luo, et al., 2023), in which traditional FIR filters are replaced by learnable convolutional layers. This design confers two main advantages: (i) it supports multi-channel, multi-layer filtering through convolutional kernels, enhancing its ability to extract complex signal structures—

unlike traditional BD, which is typically single-channel; and (ii) it leverages the powerful optimization capabilities of CNN training pipelines, avoiding the inefficiencies associated with matrix-based solvers (Buzzoni, Antoni, & d'Elia, 2018; Wiggins, 1978) or heuristic methods like particle swarm optimization (Y. Cheng, Chen, Mei, et al., 2019; Y. Cheng et al., 2018). Furthermore, by embedding the neural BD module directly within the classification network, we enable co-optimization of both feature extraction and classification objectives within a single, end-to-end learning framework.

Our proposed blind deconvolution (BD) framework comprises two neural network–based modules: a time-domain quadratic convolutional filter and a frequency-domain linear filter. We retain the term *filter* to maintain consistency with established terminology in signal processing and BD literature. The time-domain module consists of a two-layer symmetric Quadratic Convolutional Neural Network (QCNN) (F. Fan, Cong, & Wang, 2018; J.-X. Liao, Dong, Sun, et al., 2023), which is particularly adept at extracting periodic impulsive features characteristic of bearing faults. Complementing this, the frequency-domain module is implemented as a fully connected neural network operating on Fast Fourier Transform (FFT) outputs, allowing selective enhancement of critical frequency components relevant to fault detection.

In addition, drawing inspiration from recent developments in physics-informed neural networks (C. He, Shi, & Li, 2023; Ni et al., 2023; Yang et al., 2024), we propose a unified supervised learning framework, ClassBD, that effectively bridges BD and deep learning classifiers. Unlike traditional BD—which operates in an unsupervised manner—ClassBD leverages fault labels to steer the BD process toward extracting discriminative features under noisy conditions. The architecture follows a three-pronged design: (i) a neural BD module integrated as the initial stage of a deep classifier, functioning as a plug-and-play component; (ii) a physics-informed composite loss function that jointly optimizes both the BD filters and classifier weights; and (iii) an uncertainty-aware weighting strategy to dynamically balance the multiple loss terms during training.

The key contributions of this work are summarized as follows:

1. We introduce a modular, plug-and-play neural BD component composed of two cascaded filters: a time-domain quadratic convolutional neural filter and a frequency-domain linear neural filter. From a theoretical perspective, we show that the quadratic filter enhances the ability to extract periodic impulses in the time domain, while the linear filter—operating post-Fourier transform—enables adaptive filtering in the frequency domain. Together, they significantly

improve BD performance in noisy environments.

2. We propose ClassBD, a novel framework that seamlessly integrates blind deconvolution with deep learning–based classification. By leveraging fault labels, ClassBD reformulates BD as a supervised learning task—moving beyond traditional unsupervised filter optimization. The classifier not only drives the learning of BD filters but also ensures that extracted features are discriminative under noisy conditions. To the best of our knowledge, this is the first BD approach that combines interpretability, noise robustness, and end-to-end diagnosability for bearing faults.

3. We conduct comprehensive experiments to assess ClassBD's robustness under various noise levels and conditions. Results demonstrate that ClassBD consistently achieves state-of-the-art diagnostic performance across both synthetic and real-world datasets. Additionally, we show that ClassBD can be readily integrated into diverse neural network backbones, yielding consistent improvements and demonstrating strong generalizability.

3.2 Preliminaries

Table 3.1 presents some important symbols that will be used later in this chapter.

3.2.1 Blind deconvolution

In the realm of non-stationary signal processing, deconvolution reverses the effects of convolution operations performed by a linear time-invariant system on the input signal. A specialized form of this technique, known as blind deconvolution (BD) or more accurately, unsupervised deconvolution, aims to utilize the output signal to recover the input signal when both the signal transfer system and the input signal are unknown (Haykin, 1986). In the context of vibration signals associated with rotating machinery, the measured signals can be interpreted as the result of a convolution operation between the cyclic fault impulses and the transfer path functions originating from the fault source to the sensor (Miao, Zhang, Lin, et al., 2022). From a mathematical perspective, given the measured signal $\boldsymbol{x} \in \mathbb{R}^N$, the fault source signal $\boldsymbol{d} \in \mathbb{R}^N$, and the additive noise $\boldsymbol{n} \in \mathbb{R}^N$ (Gaussian noise, Laplace noise, etc.), the signal transfer process can be defined as follows:

Table 3.1: A list of mathematical notations and symbols.

Symbol	Description
N	Length of the input, constant
K	Length of the filter (convolution kernel), constant ($K < N$)
$\boldsymbol{x} \in \mathbb{R}^{1\times N}$ & $x(n),\ n = 1, \cdots, N$	Input signal measured by sensors
$\boldsymbol{d} \in \mathbb{R}^{1\times N}$	Fault source signal
$\boldsymbol{n} \in \mathbb{R}^{1\times N}$& $n(n),\ n = 1, \cdots, N$	Additive noise
$\boldsymbol{h}_d \in \mathbb{R}^{1\times K}$	Transfer function in accordance to fault source
$\boldsymbol{h}_n \in \mathbb{R}^{1\times K}$	Transfer function in accordance to noise
$\boldsymbol{f} \in \mathbb{R}^{1\times K}$	Blind deconvolution filter
$\boldsymbol{y} \in \mathbb{R}^{1\times N}$& $y(n),\ n = 1, \cdots, N$	Recovered signal after blind deconvolution
$\mathcal{K}(\cdot)$	Blind deconvolution objective function
$\boldsymbol{W} \in \mathbb{R}^{1\times K}$& $w(n),\ n = 1, \cdots, K$	Weight parameters of neural network
b	Bias parameters of neural network
$p(\cdot)$	Impulse response of bearing fault
$q(\cdot)$	Periodic modulation signal
T	Cyclic period, constant
R_{xx}	Instantaneous autocorrelation function
$\hat{x}(n),\ n = 1, \cdots, N$	Signal after time domain filter
$\mathcal{F}(\cdot), \mathcal{F}^{-1}(\cdot)$	Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT)
$\hat{X}(f),\ f = 1, \cdots, N$	Signal after FFT (frequency domain)
$\hat{Y}(f),\ f = 1, \cdots, N$	Signal after frequency domain filter (frequency domain)
$\hat{y}(n),\ n = 1, \cdots, N$	Signal after IFFT (time domain)
$h(n),\ n = 1, \cdots, N$	Hilbert Transform (HT) of the signal in time domain
$H(f),\ f = 1, \cdots, N$	Hilbert Transform of the signal in frequency domain
$z(n),\ n = 1, \cdots, N$	Analytic signal (time domain)
$ES(f),\ f = 1, \cdots, N$	Envelope spectrum (frequency domain)
$\mathcal{L}_c, \mathcal{L}_t, \mathcal{L}_f$	Cross-entropy, l_4/l_2 norm and l_2/l_4 norm loss functions
$\mathcal{G}_{l_p/l_q}(\cdot)$	Generalized sparsity blind deconvolution objective function

$$\boldsymbol{x} = \boldsymbol{d} * \boldsymbol{h}_d + \boldsymbol{n} * \boldsymbol{h}_n, \tag{3.1}$$

where $\boldsymbol{h}_d$ and $\boldsymbol{h}_n$ represent the transfer functions, and $*$ denotes the convolution operation. Hereafter, bold-faced symbols will be used to denote matrices.

The objective of BD is to extract fault-related features (cyclic impulses) from the measured signal. Towards this goal, it aims to recover the signal $\boldsymbol{y} \in \mathbb{R}^N$ that is closer to the fault source by constructing a filter $\boldsymbol{f} \in \mathbb{R}^L$:

$$\boldsymbol{y} = \boldsymbol{x} * \boldsymbol{f} = (\boldsymbol{d} * \boldsymbol{h}_d + \boldsymbol{n} * \boldsymbol{h}_n) * \boldsymbol{f} \to \boldsymbol{d}. \tag{3.2}$$

Please note that Eq. (3.1) represents a simplified description of the signal transfer process, which is not explicitly defined as a linear system. As per Miao, Zhang, Lin, et al. (2022), the noise $\boldsymbol{n}$ encompasses various noise and interference components, which may include periodic harmonics induced by machinery rotation and gear meshing, random impulse components resulting from external impacts

on the housing, and background noises. These noises pass the distinct transform paths to reach the sensor.

However, due to the complexity of machinery systems, it is often impractical to accurately estimate the transfer function and its frequency response. This challenge is further compounded by the presence of unpredictable noise. Consequently, in the absence of prior information, such as an accurate fault impulse period, BD is considered an ill-posed problem. Given the non-stationary and periodic nature of the fault characteristics, a variety of sparsity indexes have been proposed to function as optimization objective functions (L. He, Wang, et al., 2021; L. He, Yi, et al., 2021; McDonald & Zhao, 2017). A representative example is kurtosis (Pearson, 1905), which is utilized as the objective function in MED (Wiggins, 1978):

$$\mathcal{K} = \frac{\sum_{n=1}^{N} y(n)^4}{(\sum_{n=1}^{N} y(n)^2)^2}. \tag{3.3}$$

where $y(n)$ denotes the output of the BD filter, and its length is equal to the input N.

Essentially, Kurtosis is a statistical quantity that assesses the data distribution. An increase in the kurtosis value indicates a deviation from the standard normal distribution (Pearson, 1905). Intuitively, cyclic impulses emerge in the vibration signals when the fault occurs, and the kurtosis value of vibration signal increases due to the presence of more peaks (outliers). Consequently, maximizing kurtosis drives the adaptive filter to recover more impulses. Naturally, the optimization objective is defined as follows:

$$\begin{aligned} &\max_{\boldsymbol{f}} \mathcal{K}(\boldsymbol{y}) \\ &s.t. \quad \boldsymbol{y} = \boldsymbol{x} * \boldsymbol{f},\ \|\boldsymbol{f}\|_{l_2} = 1. \end{aligned} \tag{3.4}$$

Several effective optimization methods have been developed for BD, including matrix operations (Buzzoni, Antoni, & d'Elia, 2018; Wiggins, 1978), particle swarm optimization (PSO) (Y. Cheng, Chen, Mei, et al., 2019), and backpropagation (B. Fang et al., 2021, 2022). Obviously, the performance of BD is strongly influenced by the choice of optimization method.In recent years, considerable efforts have been made to identify more general objective functions, design new filters, and devise more powerful optimization tools (Miao, Zhang, Lin, et al., 2022).

3.3 Methodology

The proposed framework, as illustrated in Figure 3.1, primarily consists of two BD filters, namely a time domain quadratic convolutional filter and a frequency domain linear filter. These filters serve as a plug-and-play denoising module, and they are designed to perform the same function as conventional BD methods to ensure that the output is in the same dimension as the input.

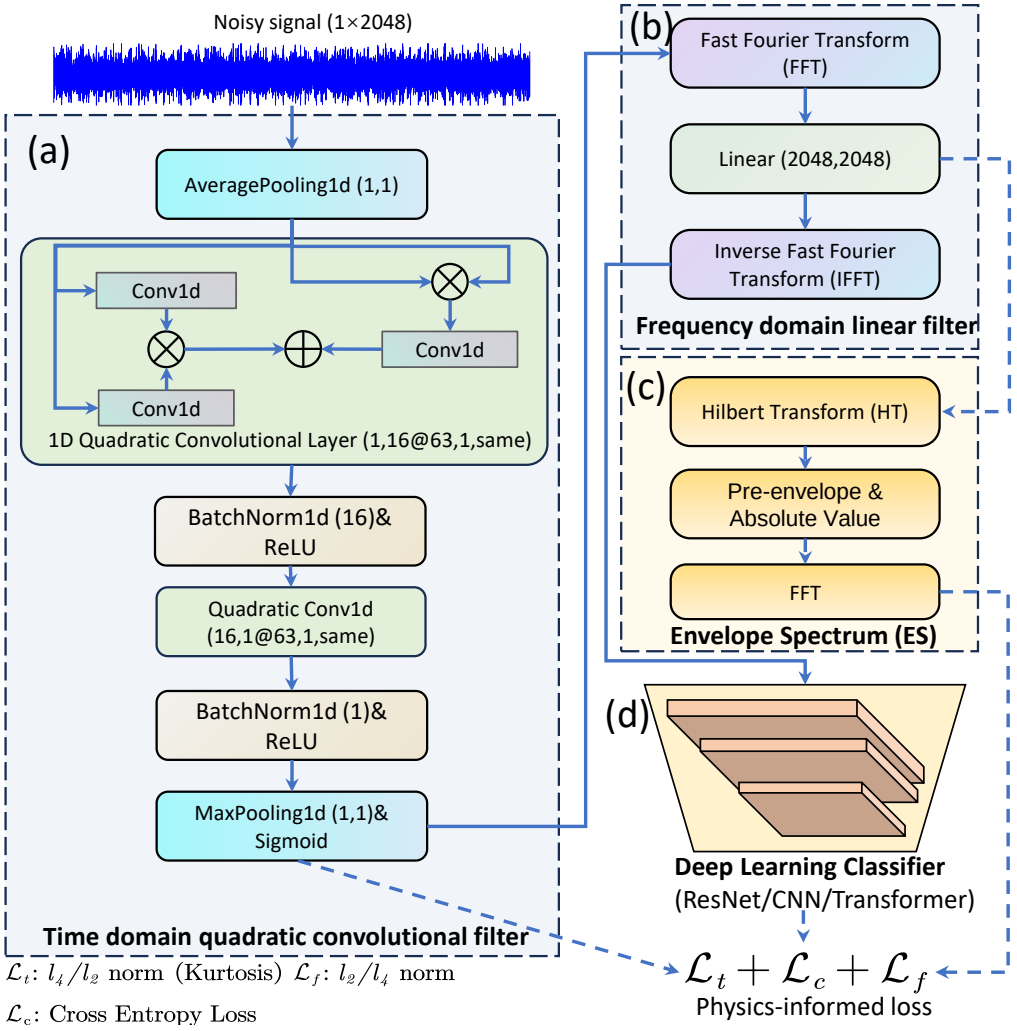


Figure 3.1: The proposed framework: (a) The time-domain filter, consisting of two symmetric quadratic convolutional neural network (QCNN) layers, is designed for time domain BD. (b) The frequency-domain filter, composed of a fully-connected layer, is utilized for frequency-domain BD. (c) The output from the fully-connected layer is extracted to compute the envelope spectrum (ES), which is crucial for constructing the objective function. (d) The output from the frequency domain linear filter is directed to the deep learning classifier to yield classification results.

1. The time domain filter is characterized by two symmetric quadratic convolutional neural network (QCNN) layers. A 16-channel QCNN is employed to filter the input signal (1×2048), and an inverse QCNN layer is used to fuse the 16 channels into one for recovering the input signal.

2. The frequency domain filter, on the other hand, starts with the fast Fourier transform (FFT) with an emphasis on highlighting the discrete frequency components. Subsequently, a linear neural layer filters the frequency domain of the signals, and the inverse FFT (IFFT) is conducted to recover the time domain signal. Moreover, an objective function in the envelope spectrum (ES) is designed for optimization.

After the neural BD filters, 1D deep learning classifiers, such as ResNet, CNN, or Transformer, can be directly used to recognize the fault type. In this chapter, we employ a popular and simple network – wide first kernel deep convolutional neural network (WDCNN) (W. Zhang et al., 2017) – as our classifier. Finally, a physics-informed loss function is devised as the optimization objective to guide the learning of the model. This function comprises a cross-entropy loss $\mathcal{L}_c$ and a kurtosis $\mathcal{L}_t$, and l_2/l_4 norm $\mathcal{L}_f$. It should be noted that $\mathcal{L}_t$ and $\mathcal{L}_f$ are used to calculate the statistical characteristics of the outputs of the time filter and frequency filter, respectively.

3.3.1 Time domain quadratic convolutional filter

As mentioned earlier, quadratic convolutional neural networks (QCNN) is a key constitutional component in the time domain filter. In this chapter, we utilize the quadratic neuron proposed by (F. Fan, Cong, & Wang, 2018), which is expressed as follows:

$$\boldsymbol{y} = \sigma((\boldsymbol{W}_1 * \boldsymbol{x} + \boldsymbol{b}_1) \odot (\boldsymbol{W}_2 * \boldsymbol{x} + \boldsymbol{b}_2) + \boldsymbol{W}_3 * (\boldsymbol{x} \odot \boldsymbol{x}) + \boldsymbol{b}_3). \tag{3.5}$$

where $*$ denotes the convolutional operation.

Currently, quadratic networks have been demonstrated to possess certain advantages in both theoretical and practical aspects. Firstly, in terms of efficiency, quadratic networks are capable of using neurons at a polynomial level to approximate radial functions, while conventional neural networks necessitate neurons at an exponential level (F.-L. Fan et al., 2025). Secondly, when it comes to feature representation, quadratic networks can achieve polynomial approximation. In contrast, conventional networks resort to piece-wise approximation via non-linear activation functions. The polynomial approximation is better than the piece-wise one in representing complex functions (F. Fan, Xiong, & Wang, 2020). Lastly, in practical applications, several studies have successfully incorporated quadratic neural networks into bearing fault diagnosis, and they have reported superior perfor-

mance under challenging conditions such as strong noise (J.-X. Liao, Dong, Sun, et al., 2023), data imbalance (W.-E. Yu et al., 2023), and variation loads (Tang et al., 2024). This further underscores the practical utility and robustness of quadratic networks in fault diagnosis.

Despite the impressive performance of quadratic networks, there is a substantial increase in the number of model parameters and non-linear multiplication operations in quadratic networks. Accordingly, the number of parameters to be optimized has increased largely. Previous studies have shown that conventional initialization techniques can significantly hinder the convergence of quadratic networks (Chrysos et al., 2023; F.-L. Fan et al., 2023). To overcome this problem, we design a dedicated strategy to initialize the quadratic network:

$$\begin{aligned} &\text{Normal initialization} \begin{cases} \boldsymbol{W}_1 \sim \mathcal{N}(0, \sigma^2), \text{ with } \sigma = \sqrt{1/32(k_n)}, \\ \boldsymbol{b}_1 \sim \ \mathcal{U}\left(-B, B\right), \text{ with } B = \sqrt{1/k_n}. \end{cases} \\ &\text{Zero initialization} \begin{cases} \boldsymbol{W}_2 = \mathbf{0}, \boldsymbol{W}_3 = \mathbf{0}, \\ \boldsymbol{b}_2 = \mathbf{1}, b_3 = \mathbf{0}. \end{cases} \end{aligned} \tag{3.6}$$

where $\mathcal{N}(0, \sigma^2)$ represents a Gaussian distribution with zero mean, $\mathcal{U}\left(-B, B\right)$ represents an uniform distribution within the range $(-B, B)$, and k_n denotes the kernel size of $\boldsymbol{W}_1$.

The group initialization strategy, also known as ReLinear (F.-L. Fan et al., 2023), aims to compel QCNN to commence from an approximately first-order linear neuron. The initial values of high-order weights are set to zero so that they grow slowly. This strategy greatly increases the stability of quadratic networks during training by avoiding gradient explosion. In terms of implementation, we employ two QCNN layers to form a symmetric structure, which mimics a multi-layer deconvolution filter. The first QCNN layer maps the input into 16 channels, while the second one consolidates these 16 channels into a single output. The dimension of the output is deliberately maintained the same as the input. This operation effectively implements a conventional BD filter using a convolutional neural network. Finally, as the QCNN functions as a time-domain BD, the widely used time-domain BD objective function kurtosis (Eq. (3.3)) is employed in this filter. Thus, we construct the time domain BD loss as follows:

$$\mathcal{L}_t = -\frac{\sum_{n=1}^{N} y(n)^4}{(\sum_{n=1}^{N} y(n)^2)^2}. \tag{3.7}$$

3.3.2 Superiority of cyclic features extraction by QCNN

In this section, we provide a theoretical derivation to address the following question: *Why are quadratic convolutional networks beneficial for the extraction of features from periodic and non-stationary signals?*

Denote the input signal as $\boldsymbol{x} = [x(1), x(2), \cdots, x(N)]$, three different weight parameters (Fan's quadratic neuron) as $\boldsymbol{W}_i = [w_i(1), w_i(2), \cdots, w_i(K)]$, where $i = 1, 2, 3$ and $K < N$. We can convert Eq. (2.11) into a sum-product form for simplification (note that the bias terms are omitted for the sake of simplicity):

$$y(n) = [\sum_{i=1}^{K} w_1(i)x(n-i)][\sum_{i=1}^{K} w_2(i)x(n-i)] + \sum_{i=1}^{K} w_3(i)x^2(n-i). \tag{3.8}$$

As shown in Eq. (3.8), a quadratic network employs a convolution kernel of size K to convolve over a segment of $\boldsymbol{x}$. The quadratic function has two parts: the product of two inner-product terms and one power term. The inner-product terms can be further factorized as follows:

$$\begin{aligned}
&[\sum_{i=1}^{K} w_1(i)x(n-i)][\sum_{i=1}^{K} w_2(i)x(n-i)] \\
=&[w_1(1)x(n-1) + w_1(2)x(n-2) + \cdots w_1(K)x(n-K)][w_2(1)x(n-1) + w_2(2)x(n-2) \\
&+ \cdots + w_2(K)x(n-K)] \\
=&w_1(1)(\sum_{j=1}^{K} w_2(j)x(n-1)x(n-j)) + w_1(2)(\sum_{j=1}^{K} w_2(j)x(n-2)x(n-j)) \\
&+ \cdots + w_K(2)(\sum_{j=1}^{K} w_2(j)x(n-K)x(n-j)) \\
=&\sum_{i=1}^{K}\sum_{j=1}^{K} w_1(i)w_2(j)x(n-i)x(n-j) \\
=&\sum_{i=1}^{K}\sum_{j=1}^{K} f(i,j)x(n-i)x(n-j),
\end{aligned} \tag{3.9}$$

with $f(i,j) = w_1(j)w_2(i)$.

Eq. (3.9) demonstrates that inner-product terms signify several convolution operations: cross-correlation between filters and inputs, and autocorrelation for input signals. These operations are crucial for cancelling noise in the bearing fault vibration signals.

Next, we establish the relations between QCNN and bearing fault signals. The ideal bearing fault mathematical model is given as follows (Antoni, 2006; Randall, Antoni, & Chobsaard, 2001):

$$x(t) = \sum_{i=-\infty}^{+\infty} p(t - iT)q(iT) + n(t), \tag{3.10}$$

where $p(t)$ is an impulse response by signal impact, $q(t) = q(t + T)$ is the periodic modulation with a period of T, T denotes the interval time between two consecutive impacts on the fault, and $n(t)$ denotes the additive noise.

The instantaneous autocorrelation function is defined as follows (Antoni, 2007b):

$$R_{xx}(t, \tau) = \sum(x(t - \tau/2)x(t + \tau/2)), \tag{3.11}$$

where τ is the time delay.

Generally, the bearing fault signal can be regarded as a second-order cyclostationary signal (Randall, Antoni, & Chobsaard, 2001). In other words, this signal presents periodicity in its second-order statistics (autocorrelation function). For Eq. (3.10), we have:

$$R_{pp}(t, \tau) = R_{pp}(t + iT, \tau). \tag{3.12}$$

Under this assumption, since the noise is randomized over the full time period and it has no periodicity, its autocorrelation function is expressed as:

$$R_{nn}(t, \tau) = 0, \tau \neq 0. \tag{3.13}$$

Lastly, let us show how the quadratic neuron enhances the fault-related signal from the noise. Combining Eqs. (3.9) and (3.10), we have:

$$\begin{aligned}
&\sum_{i=1}^{K}\sum_{j=1}^{K} f(i,j)x(n-i)x(n-j) \\
=&\sum_{i=1}^{K}\sum_{j=1}^{K} f(i,j)[\sum_{i=1}^{K}\sum_{j=1}^{K}(\sum_{k=-\infty}^{+\infty} p(t-i-kT)q(kT)+n(t-i))(\sum_{k=-\infty}^{+\infty} p(t-j-kT)q(kT)+n(t-j))] \\
\overset{(i)}{=}&\sum_{i=1}^{K}\sum_{j=1}^{K} f(i,j)(R_{pp}(t,1)+2R_{pn}(t,1)+R_{nn}(t,1))
\end{aligned}$$

$$\overset{(ii)}{\approx} \sum_{i=1}^{K}\sum_{j=1}^{K} f(i,j)R_{pp}(t,1). \tag{3.14}$$

where (i) follows from the convolutional step being 1, that is $\tau = 1$; (ii) follows from $p(t)$ and $n(t)$ being irrelevant, such that $R_{pn}(t,1) = 0$. In the ideal case, R_{pp} is not equal to 0 only on the cycle period. Clearly, the autocorrelation operation embedded in the QCNN enhances the second-order cyclostationary signal while suppressing the noise at the same time. Eq. (3.14) only keeps the fault-related signals.

In addition, the second term in Eq. (3.8), denoted as $\sum_{i=1}^{K} w_3(i)x^2(n-i)$, also plays a crucial role, as it calculates the power of the signal. This computation facilitates the amplification of the disparity between non-stationary impulses and stationary signals. Consequently, the QCNN is capable of enhancing the cyclostationary fault impulse while simultaneously suppressing the effect of random noise. Clearly, the conventional neural networks fall short in this regard. To showcase the superiority of QCNN, we conduct a comparative analysis of the feature extraction performance between quadratic and conventional networks in Section 3.5.6.

3.3.3 Frequency domain linear filter with envelope spectrum objective function

In our framework, the frequency domain filter serves as an auxiliary module, and it is endowed with the capability to directly manipulate the signal's frequency domain. The main idea involves the utilization of the neural network as a filter within the frequency domain, facilitated by the Fourier transform. This approach is commonly referred to as Fourier neural networks (Chi, Jiang, & Mu, 2020; H. Yu et al., 2023).

Denote the signal passing through the time domain filter as $\hat{x}(t)$, the FFT $\mathcal{F}(\cdot)$ is applied to convert the signal into the frequency domain:

$$\hat{X}(f) = \mathcal{F}(\hat{x}(t)). \tag{3.15}$$

According to the Convolution Theorem, the convolution of two time-domain data is equivalent to the inner product in their Fourier transform domain [1]. Such that, a frequency filter employs a

[1] https://en.wikipedia.org/wiki/Convolution_theorem

linear operation to filter the signal within the frequency domain, thereby substituting the convolution operation in the time domain. Consequently, it is reasonable to utilize a fully-connected neural network for the implementation of the frequency filter:

$$\hat{Y}(f) = \sum_{f=1}^{N} w_f(f)\hat{X}(f) + b_f, \tag{3.16}$$

where w_f, b_f are the weights and biases in the frequency domain, $\hat{Y}(f)$ denotes the filtered signal in frequency domain.

Subsequently, the IFFT $\mathcal{F}^{-1}$ is employed to recover the signal to the time domain:

$$\hat{y}(t) = \mathcal{F}^{-1}(\hat{Y}(f)). \tag{3.17}$$

Second, an objective function for the frequency domain filter is also required. Previous works have proposed some BD objective functions for the frequency domain, such as envelope spectra l_1/l_2 norm (Peter & Wang, 2013), envelope spectra kurtosis (ESK) (H. Zhang et al., 2016), and l_p/l_q norm (L. He, Wang, et al., 2021). The fundamental concept underlying these approaches is the enhancement of signal sparsity in the frequency domain, which effectively mitigates the impact of noisy frequency components. We adopt this idea to design the objective function based on the envelope spectrum (ES). Mathematically, the Hilbert transform (HT) is defined as:

$$h(t) = \frac{1}{\pi}\int_{-\infty}^{+\infty} x(\tau)\frac{1}{t-\tau}d\tau = x(t) * \frac{1}{\pi t}. \tag{3.18}$$

In our study, the HT is applied after the linear layer, which is computed in the frequency domain. According to the Convolution Theorem, we have:

$$\hat{H}(f) = (-\text{jsgn}(f))\hat{Y}(f), \tag{3.19}$$

where $\text{sgn}(f)$ denotes the sign function and j denotes the sign of the complex number.

The analytic signal (discrete form) is a complex-valued signal that is obtained by the HT:

$$z(n) = \hat{y}(n) + j\hat{h}(n), \tag{3.20}$$

where $\hat{h}(n) = \mathcal{F}^{-1}(\hat{H}(f))$.

Subsequently, the envelope is defined as the absolute value of $z(n)$, and the ES is the Fourier frequency spectrum of the envelope:

$$ES(f) = \mathcal{F}(\sqrt{\hat{y}^2(n) + \hat{h}^2(n)}). \tag{3.21}$$

At last, the objective function is designed to measure the sparsity of the ES as below:

$$\mathcal{L}_f = \frac{\sum_{f=1}^{N} ES(f)^2}{(\sum_{f=1}^{N} ES(f)^4)^{\frac{1}{2}}}. \tag{3.22}$$

Notably, both l_2/l_4 and l_4/l_2 (kurtosis) used in time and frequency domain filters are specific versions of the generalized sparsity criterion known as the $G - l_p/l_q$ norm (L. Li, 2014):

$$\begin{aligned} \mathcal{G}_{l_p/l_q}(x) &= \text{sgn}\left(\log\left(\frac{q}{p}\right)\right)\left(\frac{\|\boldsymbol{x}\|_{l_p}}{\|\boldsymbol{x}\|_{l_q}}\right)^p \\ &= \text{sgn}\left(\log\left(\frac{q}{p}\right)\right)\frac{\sum_{n=1}^{N}|x(n)|^p}{(\sum_{n=1}^{N}|x(n)|^q)^{\frac{p}{q}}},\ p,q>0, \end{aligned} \tag{3.23}$$

where $(sgn)(\cdot)$ denotes a symbolic function, which serves to guide the appropriate direction for optimization. Furthermore, the l_p norm of an input signal is defined as follows:

$$\|\boldsymbol{x}\|_{l_p} = (\sum_{n=1}^{N}|x(n)|^p)^{1/p}. \tag{3.24}$$

The $G - l_p/l_q$ norm exhibits a notable degree of flexibility, with specific choices of p and q encapsulating various high-order statistical properties. For instance, the selection of $p = 3$ and $q = 2$ corresponds to skewness, a measure of asymmetry in the distribution of the input signal. The configuration where $p > 2$ and $q = 2$ is referred to as Gary's variable norm, and the pq mean norm is characterized by the conditions $0 < p < 1$ and $q > 1$.

In addition, the values of p and q affect the monotonicity of the objective function. The derivative of $G - l_p/l_q$ norm is as follows (L. He, Wang, et al., 2021; J.-X. Liao, Dong, Luo, et al., 2023):

First, given the input vector $\boldsymbol{x} \in \mathbb{R}^K$, the $G - l_p/l_q$ norm can be unfolded as:

$$\begin{aligned}\mathcal{G}_{l_p/l_q}(\boldsymbol{x}) &= \left(\frac{\|\boldsymbol{x}\|_{l_p}}{\|\boldsymbol{x}\|_{l_q}}\right)^p \\ &= \frac{\sum_{n=1}^{K}|x(n)|^p}{\left(\sum_{n=1}^{K}|x(n)|^q\right)^{p/q}} \\ &= \frac{\sum_{n=1,n\neq n^{\max}}^{K}|x(n)|^p + |x(n^{\max})|^p}{\left(\sum_{n=1,n\neq n^{\max}}^{K}|x(n)|^q + |x(n^{\max})|^q\right)^{p/q}},\end{aligned} \tag{3.25}$$

where, $n^{\max}$ is the index of the maximum of $|\boldsymbol{x}|$.

Such that, we calculate the derivative of $\mathcal{G}_{l_p/l_q}$ with respect to the maximum of $\boldsymbol{x}$:

$$\begin{aligned}&\frac{\partial \mathcal{G}_{l_p/l_q}(\boldsymbol{x})}{\partial\left(|x(n^{\max})|\right)} = \\ &\frac{p|x(n^{\max})|^{p-1}\left(\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^q + |x(n^{\max})|^q\right)^{p/q}}{\left(\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^q + |x(n^{\max})|^q\right)^{2p/q}} \\ &- \frac{\left(\frac{p}{q}\right)\left(\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^q + |x(n^{\max})|^q\right)^{p/q-1} q|x(n^{\max})|^{q-1}\left(\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^p + |x(n^{\max})|^p\right)}{\left(\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^q + |x(n^{\max})|^q\right)^{2p/q}} \\ &= \frac{p|x(n^{\max})|^{p-1}\left(\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^q + |x(n^{\max})|^q\right)^{p/q}}{\left(\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^q + |x(n^{\max})|^q\right)^{2p/q}} \\ &- \frac{p|x(n^{\max})|^{q-1}\left(\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^q + |x(n^{\max})|^q\right)^{p/q-1}\left(\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^p + |x(n^{\max})|^p\right)}{\left(\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^q + |x(n^{\max})|^q\right)^{2p/q}}.\end{aligned} \tag{3.26}$$

Then, the numerator of (3.26) is simplified to:

$$\begin{aligned}&|x(n^{\max})|^{p-1}\left(\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^q + |x(n^{\max})|^q\right) - |x(n^{\max})|^{q-1}\left(\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^p + |x(n^{\max})|^p\right) \\ =&\frac{\left(\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^q + |x(n^{\max})|^q\right)}{|x(n^{\max})|^q} - \frac{\left(\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^p + |x(n^{\max})|^p\right)}{|x(n^{\max})|^p},\end{aligned} \tag{3.27}$$

when $p > q > 0$, the following inequality holds:

$$\frac{\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^q + |x(n^{\max})|^q}{|x(n^{\max})|^q} > \frac{\sum_{n=1,n\neq n^{\max}}^{N}|x(n)|^p + |x(n^{\max})|^p}{|x(n^{\max})|^p}. \tag{3.28}$$

Therefore, substituting (3.28) into (3.26), we have:

$$\frac{\partial \mathcal{G}_{l_p/l_q}(\boldsymbol{x})}{\partial |x(n^{\max})|} > 0. \tag{3.29}$$

Similarly, when $0 < p < q$, we have:

$$\frac{\partial \mathcal{G}_{l_p/l_q}(\boldsymbol{x})}{\partial |x(n^{\max})|} < 0. \tag{3.30}$$

Eq. (3.29) and Eq. (3.30) demonstrate that the $\mathcal{G}_{l_p/l_q}$ is a monotonic function with respect to the relationship between p and q.

This characteristic guides the design of the joint loss function. Comparing Eq. (3.7) and Eq. (3.22), they have opposite signs. As $\mathcal{L}_t$ and $\mathcal{L}_f$ are different in monotonicity, we rearrange them slightly so that they can be optimized towards the same direction.

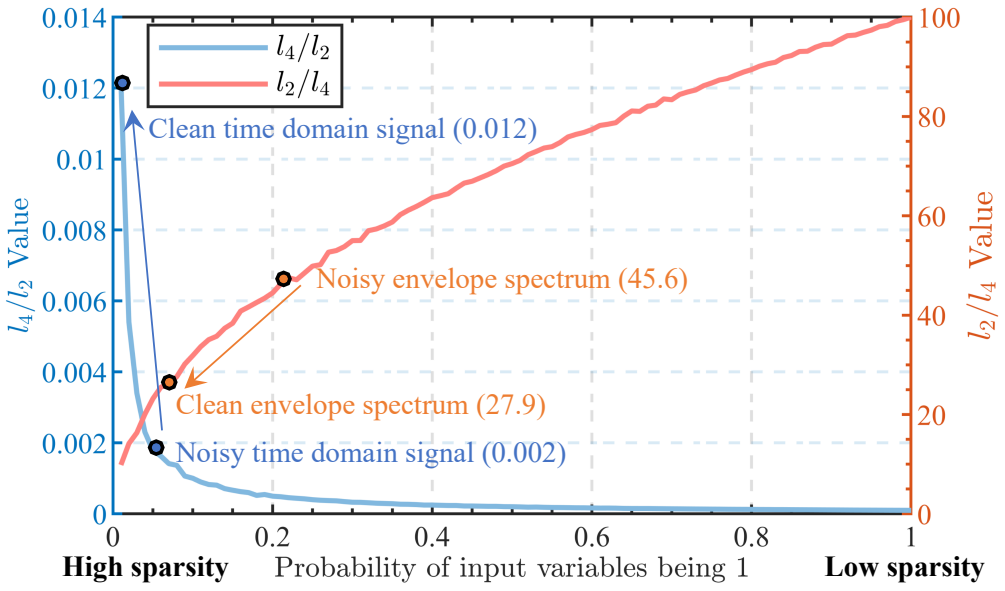


Figure 3.2: The optimization directions of l_4/l_2 and l_2/l_4 norm. Where the orange points display the l_2/l_4 values of the noisy and clean bearing fault signals in the envelope spectrum. Blue points are the l_4/l_2 values of the noisy and clean fault signals in the time domain. The optimization is to maximize the l_4/l_2 value of the time domain while minimizing the l_2/l_4 value of the frequency domain.

We further expound on the reason for employing the two distinct objective functions. The optimization trends of l_2/l_4 norm and l_4/l_2 norm are illustrated in Figure 3.2. Initially, we generate Bernoulli distributed random variables ($P(X = 1) = p_0,\ P(X = 0) = 1 - p_0,\ 0 < p_0 < 1$) and feed them into l_4/l_2 (kurtosis) and l_2/l_4 norms to demonstrate the trends. Although both functions are monotonic, they exhibit opposite directions. Clearly, the l_4/l_2 norm follows a more pronounced

exponential trend, while the l_2/l_4 norm exhibits an approximately linear trend. Secondly, we also present the l_p/l_q norm values of time domain and ES signals. The orange points represent the values of the l_2/l_4 norm of ES signals transitioning from noisy to clean, and the blue points represent the values of time domain signals. It is clear that the sparsity in the ES is much lower than in the time domain. Rapid optimization in the ES domain, resulting in the loss of too many frequency components, is undesirable. Consequently, the condition where $p < q$ offers a smoother gradient, rendering it more conducive for optimization within the ES domain. This insight has informed our decision to delineate two distinct functions tailored for BD optimization.

3.3.4 Uncertainty-aware multi-task optimization method

The fault-diagnosing task typically necessitates a deep learning classifier. Our module exhibits greater flexibility compared to other denoising models as it can be readily transferred to any 1D classifier, such as Transformer and CNN. Upon the addition of the classifier, the loss function evolves into a joint loss:

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_f + \mathcal{L}_t, \tag{3.31}$$

where $\mathcal{L}_t$ and $\mathcal{L}_f$ are derived from Eq. (3.7) and Eq. (3.22) respectively; $\mathcal{L}_c$ represents the cross-entropy loss:

$$\mathcal{L}_c = -\sum_i p_i \log q_i, \tag{3.32}$$

with p, q being the true label and predicted label, respectively.

Our framework seamlessly integrates the objective functions of BD and downstream classifier, thereby incorporating the classification labels as prior information into the BD optimization. Compared to other forms of prior knowledge, such as cyclic frequency (Buzzoni, Antoni, & d'Elia, 2018), classification labels are more readily obtainable without additional estimation, making them more suitable to guide BD in benefiting the downstream tasks.

In this context, optimizing ClassBD is framed as a multi-task learning problem (Ruder, 2017). In the context of multi-task learning, a key challenge lies in balancing different loss components. To address this problem, we employ the so-called uncertainty-aware weighing loss to automatically balance the importance of each loss component to the learning problem (Kendall, Gal, & Cipolla, 2018). Assume that all the tasks have task-dependent or homoscedastic uncertainty, the loss functions

for all tasks are subject to Gaussian noise, then the likelihood function can be defined as:

$$p(\boldsymbol{y}|f^{\boldsymbol{W}}(\boldsymbol{x})) = \mathcal{N}(f^{\boldsymbol{W}}(\boldsymbol{x}), \sigma^2), \tag{3.33}$$

where σ^2 denotes the variance of the noise.

Consequently, the joint loss is formulated as:

$$\mathcal{L} = \frac{1}{\sigma_c^2}\mathcal{L}_c + \frac{1}{\sigma_t^2}\mathcal{L}_t + \frac{1}{\sigma_f^2}\mathcal{L}_f + \log\sigma_c + \log\sigma_t + \log\sigma_f. \tag{3.34}$$

A large value of σ^2 will decrease the contribution of the corresponding loss component, and vice versa. Each σ is treated as a learnable parameter with value initialized at -0.5 as per (Kendall, Gal, & Cipolla, 2018; B. Lin & Zhang, 2023). During training, the scale is regulated by the last term $\log\sigma$, which will be penalized if the σ is too large. Although this strategy does not achieve perfect equilibrium, it allows each loss to decrease smoothly and prevents rapid convergence to zero.

Remark. This weighting strategy can mitigate training instability; however, it does not fully resolve the imbalance between time- and frequency-domain BD losses. As noted in (Antoni, 2016), achieving such a balance remains a challenging issue. A potential solution lies in employing a cyclic-sparsity function that jointly measures periodicity and sparsity, which may circumvent the balance issue (Hou et al., 2024).

3.4 Computational experiments

3.4.1 Experimental configurations

3.4.1.1 Signal preprocessing

In this experiment, we inject additive Gaussian white noise (AWGN) to simulate scenarios with significant noise and validate the classification performance of our method. The level of noise is determined based on the Signal-to-Noise Ratio (SNR), which is defined as follows:

$$\text{SNR} = 10\lg\left(\frac{P_s}{P_n}\right) = 20\lg\left(\frac{A_s}{A_n}\right), \tag{3.35}$$

where A_s and A_n denote the average amplitude of the signal and noise, respectively. An SNR of 0 indicates that the amplitude of the signals is equivalent to that of the noise. The lower the SNR, the higher the noise.

Under this configuration, we partition the datasets as per Z. Zhao et al. (2020). Firstly, the raw signals are segmented following the time sequence to separate training and test sets, thereby preventing information leakage. Secondly, the sub-sequence is split with or without overlapping sampling, contingent on the volume of the data. Thirdly, we add noise to the datasets at varying SNR levels, which are determined by the performance of models under severe degradation. The noisy signals are then normalized using Z-score standardization.

It is noteworthy that in our settings, we simulate a more challenging scenario where the noise is generated according to each sub-sequence. In other words, the noise power of each sub-sequence varies. This operation further increases the difficulty of discriminating between similar signals compared to calculating the noise using the entire signal. Finally, for the chosen datasets, the segmentation ratio differs slightly and will be elaborated upon subsequently.

3.4.1.2 Baselines and training settings

We have selected several state-of-the-art signal processing-integrated neural networks as the baselines for performance comparison: i) deep residual shrinkage neural network for bearing fault diagnosis (DRSN) (M. Zhao et al., 2020); ii) A wavelet convolutional neural network using Laplace wavelet kernel (WaveletKernelNet) (T. Li et al., 2022); iii) An enhanced semi-Shrinkage wavelet weight initialization network (EWSNet) (C. He et al., 2023); iv) A Gramian time frequency enhancement network (GTFENet) (L. Jia, Chow, & Yuan, 2023); v) A time-frequency transform-based neural network (TFN) (Q. Chen et al., 2024); vi) For ClassBD, we adopt WDCNN (W. Zhang et al., 2017) as our classifier. A detailed description of all the considered methods is provided in Table 3.2. Compared to other signal processing methods, ClassBD requires minimal settings and eliminates the need for center frequency bands, time window functions, and others.

Furthermore, to ensure fairness, we employ the model file of baseline methods provided by their officially released codes and use the same training configurations with Weights & Biases (wandb) management system. The hyperparameters of all the methods have an identical configuration. All the methods have a maximum training epoch of 200, a batch size of 128, learning rate within the range

[0.1, 0.3, 0.5, 0.8]. It is noteworthy that we employ SGD (Ruder, 2016) as the optimizer and utilize CosineAnnealingLR (Loshchilov & Hutter, 2016) to dynamically adjust the learning rate throughout the training process. The experiments are executed on an NVIDIA RTX 4090 24GB GPU and implemented using Python 3.8 with PyTorch 2.1. All reported results represent the average of ten independent runs.

Table 3.2: The descriptions of compared methods.

Methods	Signal Processing Techniques	Descriptions
DRSN (2020)	Soft thresholding	A residual shrinkage neural network module for threshold determination.
WaveletKernelNet (2022)	Continuous wavelet transform	A continuous wavelet convolutional (CWConv) layer replaces the first convolutional layer.
EWSNet (2023)	Continuous wavelet transform, Soft thresholding	A scale smoothing wavelet kernel weight replaces the first convolutional layer with a balanced dynamic adaptive threshold.
GTFENet (2023)	Gramian-based noise reduction	A branch generation module in the first layer converts signal into time domain, frequency domain, and denoised signal using Gramian-based noise reduction.
TFN (2024)	Time-frequency transform	A time-frequency convolutional layer (TFConv) in the first layer uses complex value convolutional kernel.
ClassBD	Blind deconvolution	A neural BD consists of a time domain quadratic filter and a frequency domain linear filter in the first layer with specific BD loss functions.

3.4.1.3 Evaluation metrics

We adopt the commonly used false positive rate (FPR) and F1 score to benchmark the performance of all the considered methods. Formally, these two metrics are defined as below:

$$\begin{aligned}
\text{FPR} &= \frac{FP}{FP+TN}, \\
\text{Recall} &= \frac{TP}{TP+FN}, \\
\text{Precision} &= \frac{TP}{TP+FP}, \\
\text{F1 score} &= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}},
\end{aligned}$$

where TP, TN, FP, and FN stand for the number of true positives, true negatives, false positives, and false negatives, respectively.

3.4.2 Case study 1: PU dataset

3.4.2.1 Dataset description

The PU dataset was collected by Paderborn University (PU) Bearing Data Center (Lessmeier et al., 2016). This dataset encompasses multiple faults for 32 bearings, including vibration and current signals. The bearings were categorized into three groups: i) Six healthy bearings; ii) Twelve bearings with manually-induced damage (seven with outer race faults, five with inner race faults); iii) Fourteen bearings with real damage, induced by accelerated lifetime tests (five with outer race faults, six with inner race faults, and three with multiple faults). Four operating conditions were implemented, varying the rotational speed (N = 1500 rpm or N = 900 rpm), load torque (M = 0.7 Nm or M = 0.1 Nm), and radial force (F = 1000 N or F = 400 N). The sampling frequency was set at 64 kHz. This dataset is one of the more difficult datasets in the field of bearing fault diagnosis (Z. Zhao et al., 2020).

In this study, we exclusively utilize the real damaged bearings for classification to validate the methods in real-world scenarios. We classify the fourteen faulty bearings and six healthy bearings into fourteen categories, as illustrated in Table 3.3. All four operating conditions are tested in the experiments, with the codes assigned as N09M07F10, N15M01F10, N15M07F4, and N15M07F10.

Table 3.3: Fourteen categories of PU datasets in our experiments. Where OR and IR denote outer race fault and inner race fault respectively; S, R, and M are single, repetitive, and multiple damages respectively; L1-L3 represent damage levels; F and P are fatigue and plastic deformation, respectively.

Category	Bearing Code	Damage	Category	Bearing Code	Damage
1	K01-K06	Healthy	8	KB24	(OR+IR)+M+L3+F
2	KA04	OR+S+L1+F	9	KB27	(OR+IR)+M+L1+P
3	KA15	OR+S+L1+P	10	KI04,KI14	IR+M+L1+F
4	KA16	OR+R+L2+F	11	KI16	IR+S+L3+F
5	KA22	OR+S+L1+F	12	KI17	IR+R+L1+F
6	KA30	OR+R+L1+P	13	KI18	IR+S+L2+F
7	KB23	(OR+IR)+M+L2+F	14	KI21	IR+S+L1+F

Given the large volume of data in the PU dataset, each bearing has 20 segments, so we do not use overlapping to construct the datasets. We slice 20×2048 sub-segments with a stride of 2048, resulting in 400 data points per bearing. The total number of datasets is 7600×2048. The data collected in the first 19 segments is randomly divided into training and validation sets at a ratio of 0.8:0.2, while the data from the 20th segment is allocated to the test set. Consequently, the dataset includes 5776 training data, 1444 validation data, and 380 test data. Lastly, we established four SNR levels (-4dB, -2dB, 0dB, 2dB) to evaluate the diagnosis performance.

3.4.2.2 Classification results

The classification results are presented in Table 3.4. We highlight key findings from the analysis. Notably, the ClassBD model demonstrates superior performance compared to its competitors across various noise levels and operating conditions. Overall, the average F1 scores of ClassBD on the four conditions are higher than 94%. Specifically, under high noise conditions (at -4 dB), the ClassBD exhibits a substantial performance gap relative to other methods. For instance, on the N15M01F10 dataset with -4dB noise, the ClassBD achieves an impressive 95% F1 score, while the second-best method, EWSNet, only attains 70% F1. These results underscore the efficacy of ClassBD as a robust anti-noise model, consistently delivering competitive performance across diverse high-noise scenarios.

Table 3.4: Classification results on the PU datasets. Where bold-faced numbers denote the better results.

Operating Condition	Method	SNR=-4dB		SNR=-2dB		SNR=0dB		SNR=2dB		Average	
		F1 ↑	FPR ↓	F1 ↑	FPR ↓	F1 ↑	FPR ↓	F1 ↑	FPR ↓	F1 ↑	FPR ↓
N09M07F10	WaveletKernelNet	67.45%	2.30%	74.09%	1.80%	83.14%	1.01%	89.04%	0.68%	78.43%	1.45%
	EWSNet	87.28%	0.91%	92.81%	0.49%	96.18%	0.28%	97.60%	0.18%	93.47%	0.47%
	GTFENet	70.68%	1.80%	82.19%	1.08%	92.39%	0.52%	90.34%	0.59%	83.90%	1.00%
	TFN	24.24%	5.87%	30.12%	5.45%	46.66%	4.37%	3.70%	7.69%	26.18%	5.85%
	DRSN	51.95%	2.83%	61.74%	2.28%	71.28%	1.84%	71.42%	1.49%	64.10%	2.11%
	ClassBD	**92.08%**	**0.60%**	**95.35%**	**0.34%**	**98.28%**	**0.13%**	**99.27%**	**0.05%**	**96.25%**	**0.28%**
N15M01F10	WaveletKernelNet	29.91%	3.94%	53.56%	2.67%	66.60%	1.95%	82.73%	1.08%	58.20%	2.41%
	EWSNet	69.66%	1.86%	82.65%	1.07%	94.89%	0.31%	99.06%	0.06%	86.56%	0.83%
	GTFENet	47.70%	3.22%	72.85%	1.51%	92.13%	0.45%	92.89%	0.44%	76.39%	1.41%
	TFN	32.46%	5.34%	37.94%	5.01%	51.75%	3.79%	60.85%	3.38%	45.75%	4.38%
	DRSN	52.29%	2.76%	67.15%	1.87%	81.88%	1.20%	88.81%	0.78%	72.53%	1.65%
	ClassBD	**95.19%**	**0.36%**	**98.87%**	**0.08%**	**99.63%**	**0.02%**	**99.71%**	**0.02%**	**98.35%**	**0.12%**
N15M07F04	WaveletKernelNet	17.16%	4.82%	42.61%	3.30%	75.14%	1.38%	53.78%	2.38%	47.17%	2.97%
	EWSNet	81.06%	1.12%	88.49%	0.65%	96.42%	0.23%	98.71%	0.08%	91.17%	0.52%
	GTFENet	65.56%	1.96%	78.51%	1.30%	95.60%	0.26%	98.40%	0.10%	84.51%	0.90%
	TFN	26.22%	5.75%	31.63%	5.41%	50.40%	3.89%	61.71%	3.24%	42.49%	4.57%
	DRSN	25.49%	4.14%	52.14%	2.81%	72.41%	1.56%	87.31%	0.75%	59.34%	2.32%
	ClassBD	**96.52%**	**0.24%**	**97.35%**	**0.18%**	**99.20%**	**0.05%**	**99.44%**	**0.03%**	**98.13%**	**0.13%**
N15M07F10	WaveletKernelNet	45.93%	3.13%	51.33%	2.81%	61.39%	2.14%	78.33%	1.18%	59.24%	2.31%
	EWSNet	57.38%	2.39%	76.01%	1.37%	95.84%	0.27%	98.60%	0.10%	81.96%	1.03%
	GTFENet	23.93%	4.26%	79.42%	1.14%	96.57%	0.21%	97.97%	0.12%	74.47%	1.43%
	TFN	30.47%	5.44%	33.87%	5.29%	68.23%	2.88%	8.50%	6.27%	35.27%	4.97%
	DRSN	33.80%	3.91%	62.51%	2.23%	86.19%	0.90%	92.14%	0.48%	68.66%	1.88%
	ClassBD	**80.13%**	**1.28%**	**97.35%**	**0.18%**	**99.48%**	**0.03%**	**99.44%**	**0.04%**	**94.10%**	**0.38%**

Moreover, the computational cost associated with the considered methods is summarized in Table 3.5. It can be observed that our proposed method is the most computationally efficient in terms of both inference time and FLOPs. Unlike the baseline methods, our approach utilizes a compact backbone (WDCNN) composed of only 67K parameters. Despite its compact size, the proposed ClassBD delivers a satisfactory performance, and it can thus be used as an efficient model for signal denoising.

Table 3.5: Summary on the computational cost of each method, where #FLOPs represents floating point operations and inference time denotes the elapsed time to infer one sample (1×2048) on an Intel i9-10900K CPU.

	#Parameters	#FLOPs	Inference time (ms)
WaveletNet	3.85M	0.70G	45.63
EWSNet	0.10M	93.81M	9.50
GFTENet	0.67M	0.27G	38.68
TFN	0.19M	45.67M	15.64
DRSN	5.25M	1.40G	107.93
ClassBD	4.27M	4.37M	5.39
ClassBD+WDCNN	4.27M (+67.30K)	5.98M	7.49

3.4.2.3 Classification under small sample conditions

In this study, we rigorously assess the performance of various methods under the challenging scenario of extremely limited sample availability. While our approach is not explicitly tailored for small sample issues, we conduct straightforward tests to gain insights into their behavior.

Specifically, we employ the N09M07F10 dataset, comprising 20 signal segments per category. In the most stringent case, we retain only one sample from each signal segment, reserving the last segment of the signal as the test set. Consequently, we collect one to three samples from each signal segment and finally compose 15, 30, and 45 samples per class for training sets. Notably, we exclude noise from this experiment.

The results, depicted in Figure 3.3, reveal that even under these constrained conditions, the ClassBD, EWSNet, and DRSN models exhibit commendable performance. Remarkably, all three methods achieve over 90% F1 scores with a training dataset of just 15 samples per class. Notably, the performance of ClassBD and EWSNet closely aligns. Their average F1 scores stand at 97.70% and 97.47%, respectively, positioning them as the top-performing approaches. In summary, our findings underscore the promising potential of ClassBD in addressing the challenges posed by small sample sizes

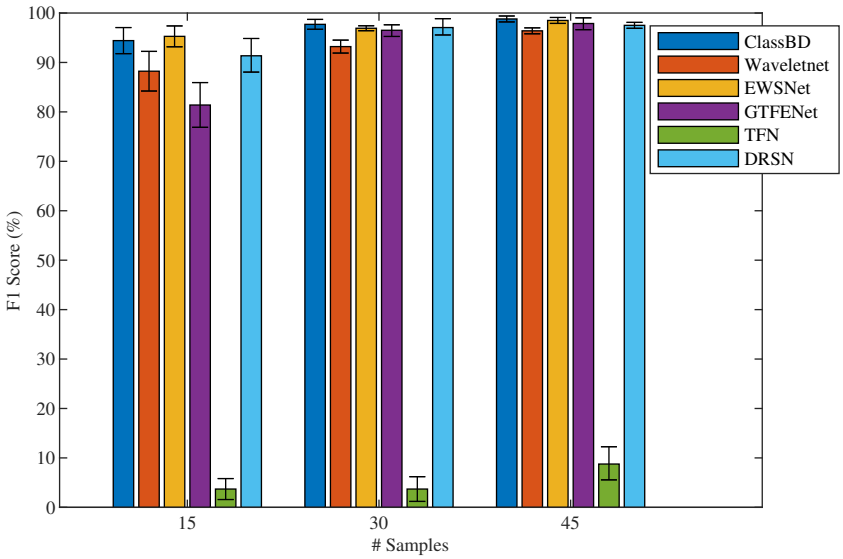


Figure 3.3: The F1 scores (%) of baseline methods on the PU N09M07F10 dataset under small sample conditions, where the error bar represents the standard deviation over ten runs.

3.4.3 Case study 2: JNU dataset

3.4.3.1 Dataset description

This dataset was provided by Jiangnan University (JNU) (K. Li et al., 2013). Two types of roller bearings were artificially injected with inner race defects (bearing N205), outer race defects, and ball defects (bearing NU205) by a wire-cutting machine. Three different rotation speeds (600 rpm, 800 rpm, 1000 rpm) were implemented to test the bearings. The sampling rate was set to 50 kHz, resulting in signal segments of 20 seconds each. This dataset is one of the more difficult datasets in the field of bearing fault diagnosis (Z. Zhao et al., 2020).

To facilitate varying rotation speed classification, we organized the data into ten classes. These classes encompass three fault types across the three different speeds, along with a healthy class. Furthermore, compared to the PU dataset, the JNU dataset is characterized by a limited data volume. Consequently, we adopted an overlapping strategy to extract signal segments, with a stride of 100. The final dataset configuration involved allocating the last 25% of the data as the test set. Prior to this, the remaining data were randomly partitioned into training and validation sets, maintaining an 80:20 split ratio.

In summary, the dataset statistics are as follows: 54,064 samples in the training set, 13,517 samples in the validation set, and 22,269 samples in the test set. Also, we introduce four SNR levels: -10 dB, -8 dB, -6 dB, and -4 dB for evaluation.

3.4.3.2 Classification results

The classification results, as summarized in Table 3.6, yield valuable insights. Notably, the ClassBD model consistently outperforms its competitors across various noisy conditions. Overall, the ClassBD achieves an impressive F1 score exceeding 96%, surpassing the second-best method, EWSNet, which attains an average F1 score of 92.75%. To be specific, even in the presence of severe noise (at -10 dB), our method maintains a commendable 93% F1 score. This remarkable performance underscores the ClassBD's excellent anti-noise capabilities. In comparison, under the same noisy conditions, other methods, excluding EWSNet, experience significant degradation. At last, the JNU dataset experiment substantiates that our approach effectively realizes bearing fault diagnosis across varying rotational speeds, even in challenging high-noise environments.

Table 3.6: Classification results on the JNU dataset. Where bold-faced numbers denote the better results.

Method	SNR=-10dB		SNR=-8dB		SNR=-6dB		SNR=-4dB		Average	
	F1 ↑	FPR ↓	F1 ↑	FPR ↓	F1 ↑	FPR ↓	F1 ↑	FPR ↓	F1 ↑	FPR ↓
WaveletKernelNet	42.19%	4.43%	74.72%	1.66%	70.75%	1.66%	74.36%	1.52%	65.50%	2.32%
EWSNet	89.39%	0.63%	90.06%	0.62%	95.03%	0.29%	96.51%	0.21%	92.75%	0.44%
GTFENet	63.35%	2.13%	80.19%	1.28%	91.45%	0.52%	93.98%	0.34%	82.24%	1.07%
TFN	58.90%	3.01%	39.86%	5.30%	6.69%	10.00%	18.74%	6.58%	31.05%	6.22%
DRSN	33.60%	4.80%	50.55%	3.36%	56.89%	2.54%	57.96%	2.42%	49.75%	3.28%
ClassBD	**93.06%**	**0.42%**	**96.20%**	**0.23%**	**97.33%**	**0.17%**	**98.54%**	**0.09%**	**96.28%**	**0.23%**

3.4.4 Case study 3: HIT dataset

3.4.4.1 Dataset description

The Harbin Institute of Technology (HIT) dataset was collected by our team. The data acquisition for faulty bearings was conducted at the MIIT Key Laboratory of Aerospace Bearing Technology and Equipment at HIT. The bearing test rig and faulty bearings are illustrated in Figure 3.4. This test rig follows the Chinese standard rolling bearing measuring method (GB/T 32333-2015). The test bearings used were HC7003 angular contact ball bearings, typically employed in high-speed rotating machines. For signal collection, an acceleration sensor was directly attached to the bearing, and the NI USB-6002 device was used at a sampling rate of 12 KHz. Approximately 47 seconds of data were recorded for each bearing, resulting in 561,152 data points per class. The rotation speed was set to 1800 rpm, significantly faster than previous datasets.

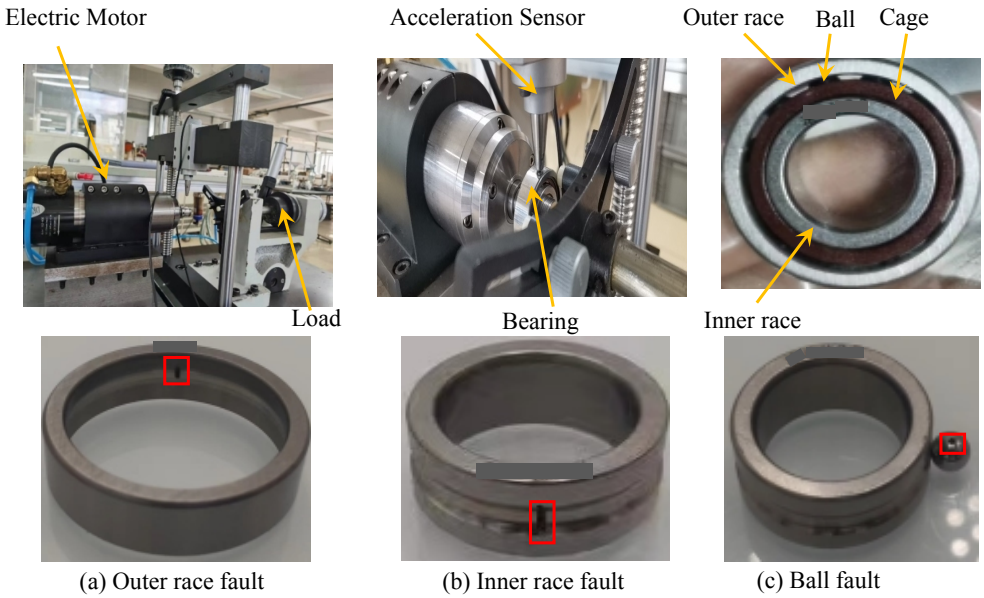


Figure 3.4: The bearing fault test rig and three types of faults.

Moreover, defects were manually injected at the outer race (OR), inner race (IR), and ball, similar to the JNU dataset. However, we established three severity levels (minor, moderate, severe), resulting in ten classes, as detailed in Table 3.7. This dataset presents a greater challenge than the JNU dataset due to the faults being cracks of the same size but varying depths, leading to more similarity in the features of different categories.

Table 3.7: Ten healthy statuses in our HIT dataset. OR and IR denote that the faults appear in the outer race and inner race, respectively.

Category	Faulty Mode	Category	Faulty Mode
1	Health	6	OR (Moderate)
2	Ball cracking (Minor)	7	OR (Severe)
3	Ball cracking (Moderate)	8	IR (Minor)
4	Ball cracking (Severe)	9	IR (Moderate)
5	OR cracking (Minor)	10	IR (Severe)

The segmentation of the dataset is similar to that of the JNU dataset. A stride of 28 was set to acquire a larger volume of data. The last quarter of the data was designated as the test set, while the remaining data were randomly divided into training and validation sets at a ratio of 0.8:0.2. Consequently, the training set, validation set, and test set comprise 113920, 28480, and 46732 samples, respectively. Lastly, the SNR levels were set at -10dB, -8dB, -6dB, -4dB.

3.4.4.2 Classification performance

The classification results are presented in Table 3.8. Several observations are as follows.

Table 3.8: Classification results on the HIT dataset. Where bold-faced numbers denote the better results.

Method	SNR=-10dB		SNR=-8dB		SNR=-6dB		SNR=-4dB		Average	
	F1 ↑	FPR ↓	F1 ↑	FPR ↓	F1 ↑	FPR ↓	F1 ↑	FPR ↓	F1 ↑	FPR ↓
WaveletKernelNet	20.93%	7.68%	51.64%	4.64%	66.58%	3.17%	83.89%	1.49%	55.76%	4.25%
EWSNet	48.23%	5.10%	63.91%	3.58%	78.39%	2.02%	83.99%	1.49%	68.63%	3.05%
GTFENet	42.01%	5.63%	52.53%	4.93%	76.79%	2.58%	82.90%	1.68%	63.56%	3.70%
TFN	7.52%	9.53%	19.90%	8.54%	36.14%	6.67%	61.65%	4.02%	31.30%	7.19%
DRSN	32.83%	6.80%	61.67%	3.77%	59.31%	4.12%	84.85%	1.38%	59.67%	4.01%
ClassBD	**86.15%**	**1.29%**	**86.84%**	**1.42%**	**93.74%**	**0.69%**	**97.23%**	**0.32%**	**90.99%**	**0.93%**

Firstly, ClassBD outperforms its competitors in terms of both F1 score and FPR across all noise conditions. The average F1 score of ClassBD exceeds 90%, compared to the second-best method,

EWSNet, which achieves an average F1 score of 68%. Secondly, it is noteworthy that even under severe noise conditions (-10dB), ClassBD maintains an F1 score of 86%, indicating its robust anti-noise performance. Under these conditions, the performance of all other methods degrades significantly. This phenomenon highlights the superiority of ClassBD.

Furthermore, we reduce the dimensions of the last layer features of all methods to 2D space using t-SNE (G. E. Hinton, 2008) under -10dB noise. The results are depicted in Figure 3.5. Overall, the features of the five methods can generate distinct clusters, with the exception of TFN, which aligns with their categorical performance. However, all methods exhibit some degree of misclassification. For instance, the red points (OR3) of DRSN and EWSNet appear within the blue clusters (B3), indicating that some instances of OR3 are misclassified as B3. Similarly, in GTFENet and WaveleKernelNet, some green points (IR3) overlap with the IR2 and B3 clusters. In contrast, ClassBD only has a few outliers that are misclassified into other clusters, demonstrating its superior feature extraction ability under high noise conditions.

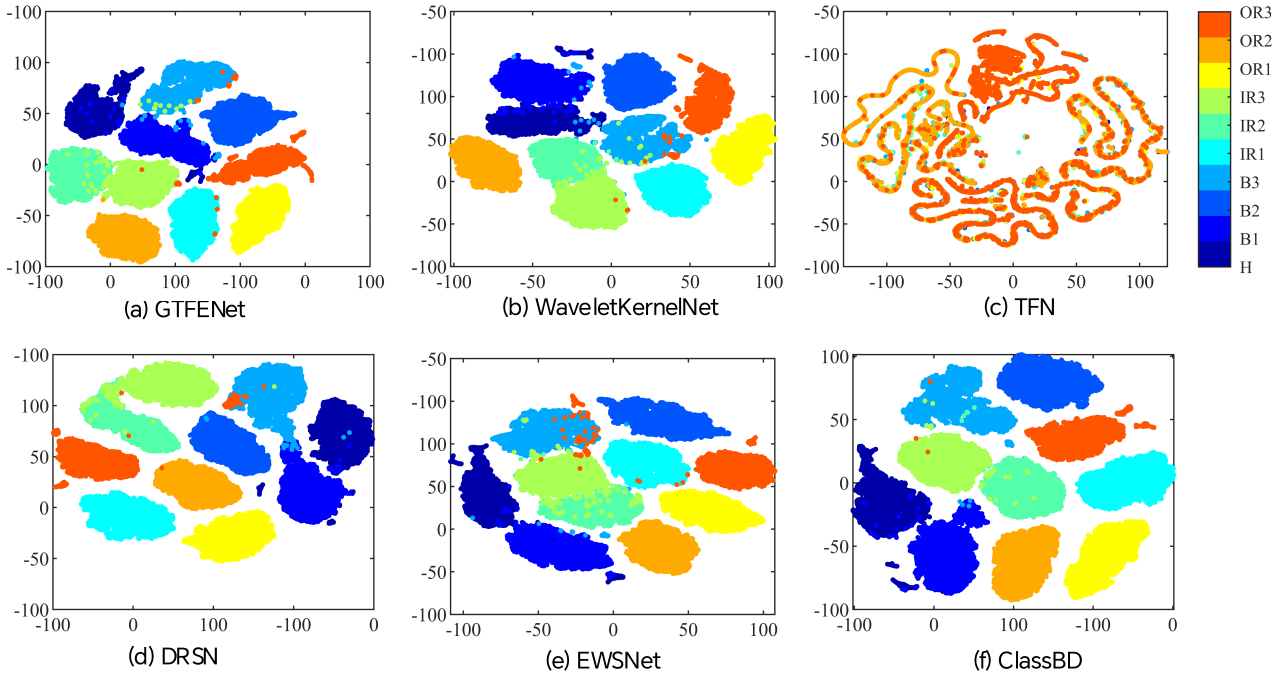


Figure 3.5: The t-SNE visualization of the last convolutional features of all methods on the -10dB HIT dataset.

In summary, based on the performance across three datasets, we illustrate the average F1 scores of all methods in Figure 3.6. The results indicate that our method, ClassBD, outperforms other baseline methods on all datasets, with average F1 scores exceeding 90%. This underscores the robust anti-noise performance of ClassBD.

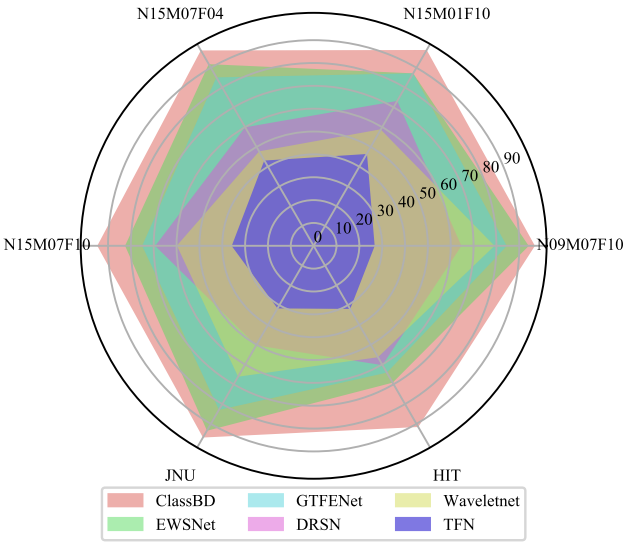


Figure 3.6: The average F1 scores of all methods across the compared datasets.

3.5 Analysis experiments

3.5.1 Comparison of feature extraction methods

Despite the recent proposal of many feature extraction methods, few works have validated their fault diagnosis performance via the integration of classifiers. Initially, we assess the performance of recently proposed BD and other denoising methods by employing WDCNN as the post-BD classifier. For comparative analysis, we utilize four prior-free BD methods: i) minimum entropy deconvolution (MED) (Wiggins, 1978), ii) sparse maximum harmonics-to-noise-ratio Deconvolution (SMHD) (Miao et al., 2016), iii) improved maximum correlated kurtosis deconvolution (IMCKD) (Miao et al., 2017), and iv) multi-task neural network blind deconvolution (MNNBD) (J.-X. Liao, Dong, Luo, et al., 2023). Additionally, we consider three VMD-based methods: i) variational nonlinear chirp mode decomposition (VNCMD) (S. Chen et al., 2017), ii) feature mode decomposition (FMD) (Miao, Zhang, Li, et al., 2022), iii) variational generalized nonlinear mode decomposition (VGNMD) (H. Wang, Chen, & Zhai, 2024). F1 score is employed as the performance metric across all the comparisons.

The classification results over the three datasets are reported in Table 3.9. Evidently, our approach consistently outperforms other methods across the three considered datasets, thus indicating that the classifier-guided optimization is apt for fault diagnosis. Unlike the traditional BD methods adopting an unsupervised optimization paradigm, our method takes a supervised learning approach by incorporating class information.

Table 3.9: The F1 scores (%) of different feature extraction methods on the three considered datasets.

	VNCMD	FMD	VGNMD	MED	SHMD	IMCKD	MNNBD	ClassBD
N09M07F10 0dB	60.15%	55.36%	64.52%	63.21%	75.12%	73.35%	72.52%	**98.28%**
JNU -6dB	80.33%	82.56%	79.36%	88.64%	89.88%	90.02%	90.33%	**97.51%**
HIT -6dB	69.35%	75.65%	71.26%	66.12%	70.61%	69.63%	73.23%	**98.47%**

Moreover, BD-based methods exhibit superior performance compared to VMD-based methods. This is attributed to the fact that VMD-based methods decompose signals into multiple modes, but they fail to select fault-related frequency components. As illustrated in Figure 3.7, VMD-based methods necessitate manual or additional methods to select fault-related modes. Finally, our proposed approach is end-to-end trainable and forms a streamlined pipeline. When compared to the combination of conventional BD and classifiers, our method shows superior performance under noise-affected conditions.

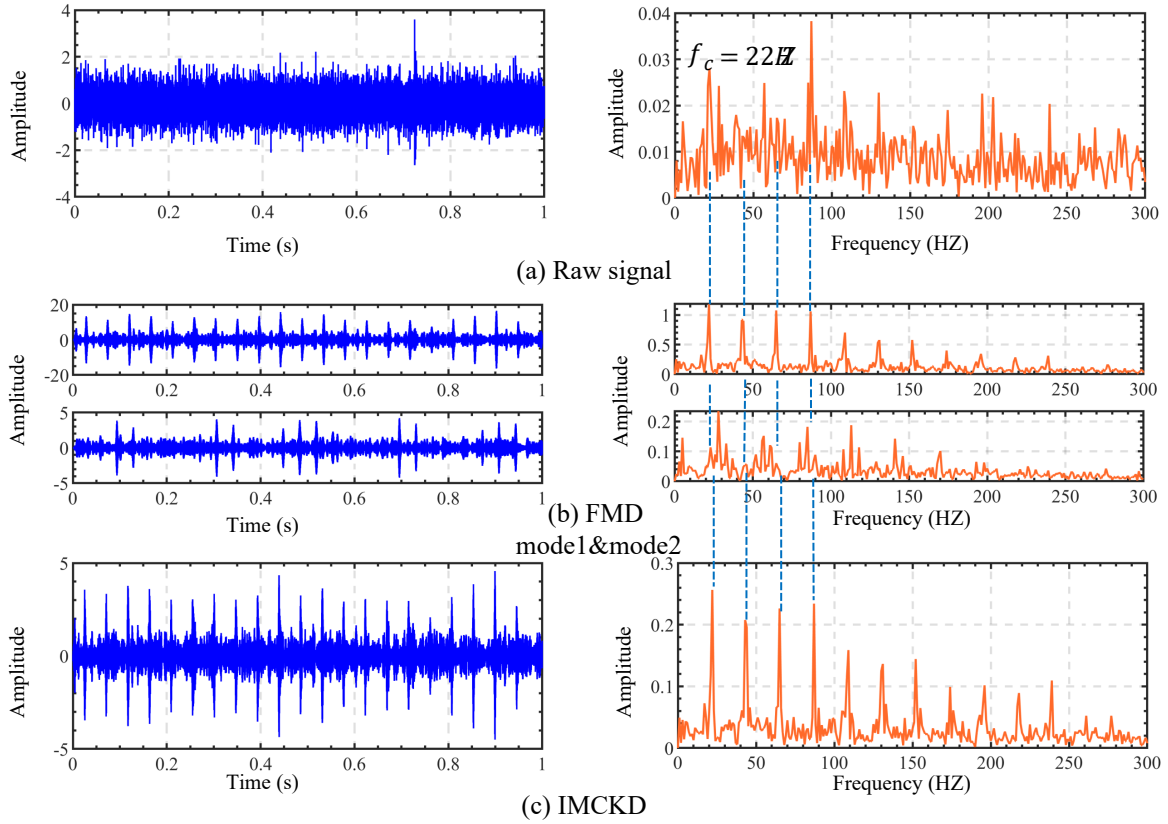


Figure 3.7: The time and envelope spectrum domain signal using VMD-based and BD-based methods.

On the other hand, it is imperative to validate the denoising performance of BD methods in the frequency domain, thereby demonstrating that these methods indeed extract fault-related features and provide interpretability. Therefore, we design a frequency domain metric called fault frequency index (FFI), which is defined as follows:

$$\text{FFI} = \frac{1}{I}\sum_{i=1}^{I} \max[EES(if_c - 0.1f_c), EES(if_c + 0.1f_c)]. \tag{3.36}$$

Where, f_c represents the fault characteristic frequency, I is the number of harmonic frequencies (set to $I = 5$), ± 0.1 constitutes an interval of the random drift of f_c in the actual measured signals (Antoni, 2006), The term $EES(\cdot)$ signifies the enhanced envelope spectrum, obtained by performing Fast-SC (Antoni, Xin, & Hamzaoui, 2017):

$$EES(\alpha) = \frac{1}{N}\sum_{f=1}^{N} |\gamma_x(\alpha, f)|, \tag{3.37}$$

where $\gamma_x(\alpha, f)$ is the spectral coherence, N denotes the number of discrete frequency, and α is the cyclic frequency, which encompasses fault characteristic frequencies f_c and other cyclic frequency components. Compared to the squared envelope spectrum (ES), the EES can amplify the non-zero cyclic frequency. Hence, the EES is apt for identifying the amplitude of fault characteristic frequencies and computing the FFI.

Subsequently, we utilize the JNU dataset to demonstrate the feature extraction performance. The SNR is set to -10dB, and we apply BD methods to all faulty signals, and then calculate the FFI of the signals post-BD. The results are illustrated in Table 3.10. From these results, we can make several observations. Firstly, ClassBD is highly effective in extracting fault-related features, as evidenced by the highest FFIs across all signals, surpassing even the raw clean signals. Secondly, on average, all BD methods are capable of enhancing the fault characteristic frequencies. The FFIs are consistently higher than the noisy signals, with MNNBD ranking second-best, achieving an average improvement of 41% compared to the noisy signals.

Table 3.10: The FFI of different BD methods on the JNU dataset. Higher is better. OR, IR, B represent the outer, inner and ball faults respectively and the numbers (600,800, 1000) denote the rotating speeds.

	OR600	OR800	OR1000	IR600	IR800	IR1000	B600	B800	B1000	Average
Raw signal	0.51	0.38	0.38	0.38	0.45	0.39	0.31	0.60	0.49	0.43
Noisy signal (-10dB)	0.25	0.17	0.16	0.17	0.24	0.20	0.11	0.10	0.09	0.17
IMCKD	0.25	0.18	0.15	0.18	0.22	0.19	0.12	0.25	0.19	0.19
SHMD	0.18	0.14	0.11	0.17	0.16	0.21	0.19	0.25	0.23	0.18
MED	0.25	0.18	0.15	0.18	0.22	0.19	0.12	0.26	0.19	0.19
MNNBD	0.13	0.23	0.29	0.24	0.31	0.28	0.20	0.17	0.29	0.24
ClassBD	0.62	0.42	0.44	0.68	0.86	0.73	0.37	0.87	0.48	**0.61**

Finally, we present a comparison of the feature extraction performance on the B1000 signal using

Fast-SC. As depicted in Figure 3.8, the bright lines in the spectral coherence suggest that the magnitudes of the cyclic frequencies α and corresponding frequency bands are elevated. And the EES displays the intensity of the extracted cyclic frequencies of the signal post-BD. Several observations can be made from this. Firstly, the noise significantly attenuates the characteristics of the signal. A comparison between Figure 3.8 (a) and (b) reveals that the intensity of all frequency bands is suppressed, with the lower frequency bands being more severely drowned out by the noise. Secondly, all BD methods are capable of enhancing the cyclic frequency characteristics of the signal, indicating the effectiveness of BD in extracting fault-related features from noisy signals. Specifically, MNNBD focuses on the fundamental cyclic frequency, resulting in the extraction of full frequency bands at the first cyclic frequency. While SHMD and MED enhance all cyclic frequency bands, the amplitude of the individual cyclic frequencies is lower. Lastly, ClassBD exhibits the best performance. It significantly enhances the cyclic frequency characteristics of the signal. More importantly, ClassBD can provide valuable interpretability in decision-making.

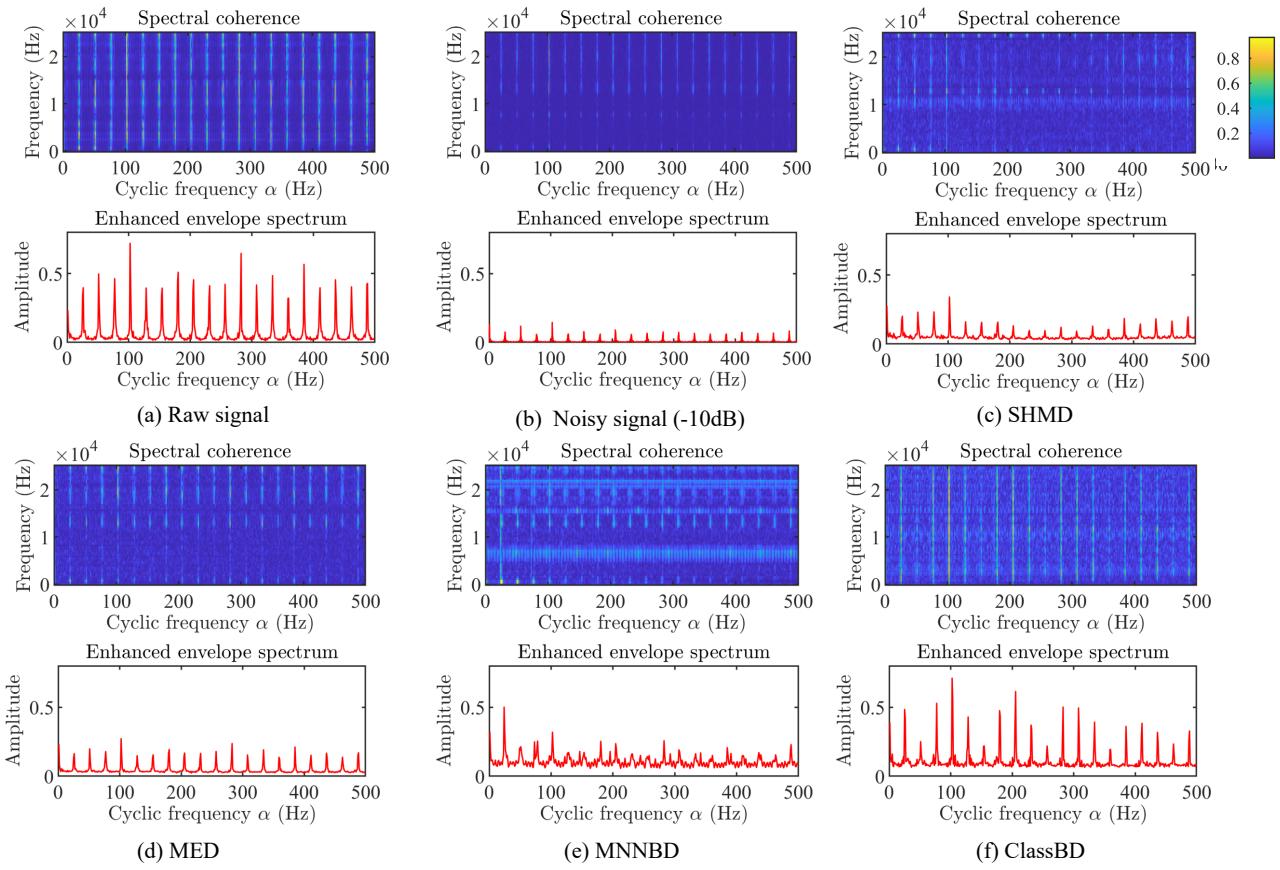


Figure 3.8: Comparison of BD methods on the JNU B1000 signal.

3.5.2 Classification performance under noisy conditions

To underscore the superior performance of ClassBD in handling various types of noise, we conduct a comparative analysis using several synthetic and real-world noises.

First, we generate Gaussian and Laplace noises from their respective probability density functions:

$$\begin{aligned} \text{Gaussian Noise} : g(x|\mu,\sigma) &= \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}, \\ \text{Laplace Noise} : l(x|\gamma,b) &= \frac{1}{2b} e^{\frac{-|x-\gamma|}{b}}, \end{aligned} \tag{3.38}$$

where μ and σ denote mean and standard deviation respectively, and are set $\mu = 0, \sigma = 1$ for Gaussian noise. For Laplace noise, γ represents the location parameter and b is the scale parameter, with $\gamma = 0, b = 1$.

Furthermore, pink noise is generated in the frequency domain, with its power spectral density inversely proportional to the signal frequency:

$$\text{Pink Noise} : S(f) \propto \frac{1}{f^{\alpha}}, \tag{3.39}$$

where f denotes the randomly determined frequency of the noise, and α is set to 1. In this experiment, we validate the classification performance on the PU "N09M07F10" dataset and set the SNR to -4dB for all synthetic noises.

Lastly, we employ two types of real-world noises, collected from an airplane and a truck using acoustic sensors, to simulate bearings operating under real-world conditions [2]. Due to the discrepancy between the signal power captured by the acoustic and vibration sensors, we adjust the noise power to match the power of the vibration signal, thereby simulating a real-world bearing failure scenario.

The experimental results, presented in Table 3.11, reveal that our method delivers the highest average F1 score, outperforming the second-best method by 20%. This underscores the superior anti-noise performance of ClassBD across different types of noise. Specifically, Gaussian noise is the easiest to handle, with several methods (WaveletKernelNet, EWSNet, GTFENet, ClassBD) achieving optimal results at the same SNR. Conversely, pink noise appears to be the most challenging, as evidenced by the highest-performing method only achieving an 83% F1 score. This is likely due to pink noise being generated in the frequency domain, which can interfere with the signal frequency. Finally, when con-

[2] https://github.com/markostam/active-noise-cancellation

sidering the two real-world noises, ClassBD significantly outperforms other methods, demonstrating approximately 40% improvement over the second-best method. A comparison of the performance of several methods reveals that the difficulty of handling real-world noise lies somewhere between Gaussian and Laplace noise, indicating the efficacy of using synthetic noise to simulate real-world scenarios.

Table 3.11: The F1 scores (%) of all methods on the PU ”N09M07F10” dataset with different noise.

	Airplane	Truck	Pink=-4dB	Laplace=-4dB	Gaussian=-4dB	Average
WaveletKernelNet	49.88%	47.94%	48.36%	59.08%	67.45%	54.54%
EWSNet	61.19%	58.30%	80.58%	60.51%	87.28%	69.57%
GTFENet	61.19%	67.36%	68.58%	41.27%	70.68%	61.81%
TFN	26.64%	26.07%	24.05%	14.39%	24.24%	23.07%
DRSN	58.96%	55.36%	48.13%	51.49%	51.95%	53.17%
ClassBD	**90.01%**	**91.01%**	**83.68%**	**90.52%**	**92.08%**	**89.46%**

3.5.3 Influence of BD parameters

We select the most challenging scenario, -4dB Pink noise, to investigate potential strategies for further improving the performance of ClassBD. One straightforward strategy is to increase the number of parameters in the BD module. Consequently, we vary the number of channels and the size of the kernels in the quadratic layers to understand their impact on performance. The overall results, as depicted in Table 3.12, indicate that introducing moderate channels and kernel sizes can lead to a performance improvement. When comparing inference speed, it is found that increasing the kernel size yields more benefits.

Table 3.12: Comparison of varying number of QCNN channels and kernel sizes with -4dB Pink noise on the PU ”N09M07F10” dataset, where inference time denotes the elapsed time to infer one sample (1×2048) on an Intel i9-10900K CPU.

#Channels (Kernel=63)	8	16	32	64	128
F1 Scores (%) ↑	80.59	83.68	88.97	**91.31**	83.37
Inference Time (ms) ↓	4.72	7.49	8.81	9.68	61.02
#Kernel size (Channel=16)	31	63	127	255	511
F1 Scores (%) ↑	81.64	83.68	**90.18**	87.99	88.88
Inference Time (ms) ↓	7.48	7.49	8.00	8.74	9.98

We present the features extracted by the time BD, as illustrated in Figure 3.9. In corroboration with the classification results, it is observed that the feature extraction ability of ClassBD can be enhanced

by increasing the number of channels and the kernel size. Specifically, configurations with C=64, K=63 (Figure 3.9 (e)) and C=16, K=127 (Figure 3.9 (h)) are able to extract accurate fault-related features. However, configurations with excessively large parameters extracted incorrect frequency components (Figure 3.9 (f) and (i)), leading to a decline in classification performance.

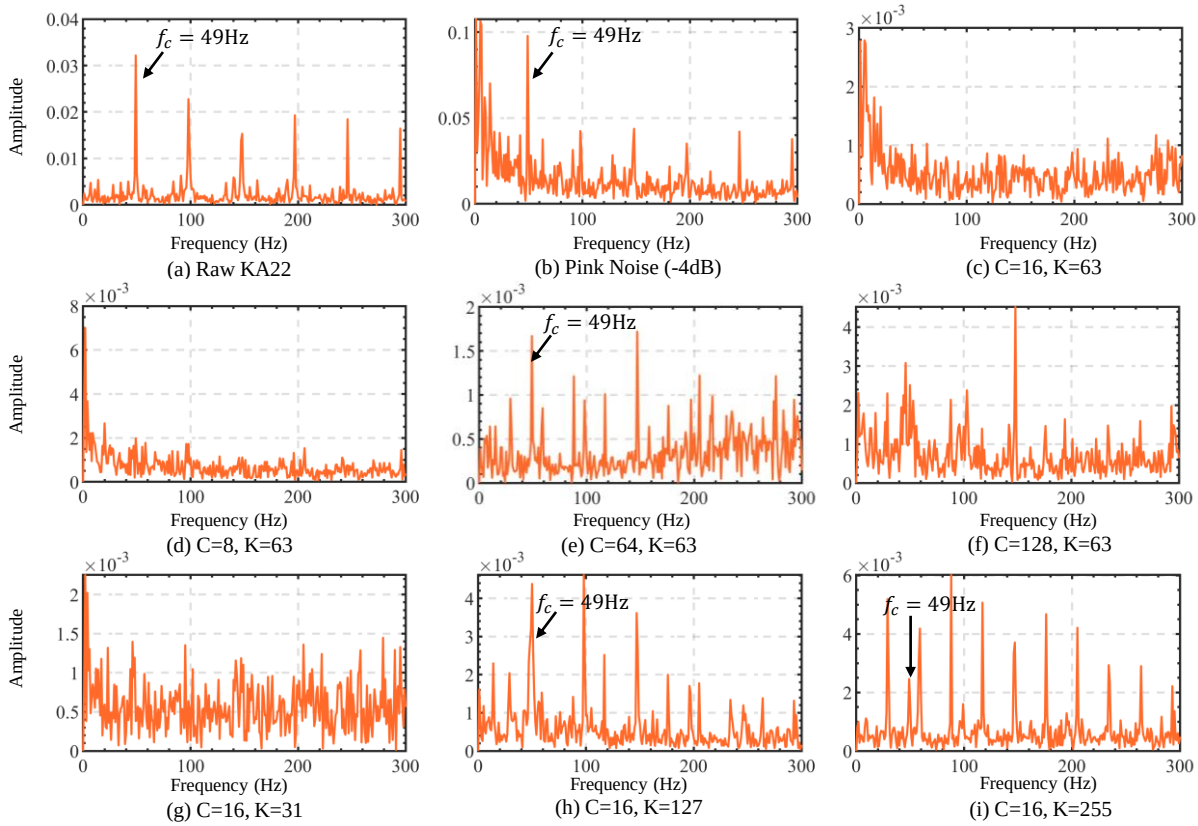


Figure 3.9: The envelope spectrum of the signals and the output of time domain BD filters on the PU ”N09M07F10” dataset KA22 signal. Where ”C” denotes the number of convolutional channels and ”K” denotes the size of a convolutional kernel.

In summary, a useful strategy when dealing with challenging scenarios is to moderately increase the kernel size. For general cases, our designed parameters strike a good balance between performance and computational speed. Therefore, we posit that the structural parameters of ClassBD should be carefully set against task requirements.

3.5.4 Employing ClassBD to deep learning classifiers

Given that ClassBD serves as a signal preprocessing module, it possesses the flexibility to be integrated into various backbone networks. In the context of this experiment, we assess the classification performance by incorporating ClassBD into four widely recognized deep learning classifiers: ResNet (K. He et al., 2016), MobileNetV3 (Howard et al., 2019), WDCNN (W. Zhang et al., 2017), and Trans-

former (Vaswani et al., 2017). It is important to note that some networks were initially designed for image classification, so some parameters are revised to accommodate the 1D signal input. The properties of these backbone networks are illustrated in Table 3.13.

Table 3.13: The properties of compared backbone networks. Where #FLOPs represents floating point operations, inference time denotes the elapsed time to infer one sample (1×2048) on an Intel i9-10900K CPU.

Backbone network	#Params	#FLOPs	Inference time (ms)
WDCNN	67.30k	1.61M	3.99
Transformer	0.41M	7.45M	5.98
ResNet18	3.83M	0.35G	13.96
MobilieNetV3	5.32M	77.4M	22.98

Firstly, the classification results on the PU dataset are illustrated in Table 3.14. Predominantly, ClassBD has demonstrated its efficacy in enhancing performance. For instance, in the case of N09M07F10 with -4dB noise, the application of ClassBD results in an improvement exceeding 10% across all backbones. On average, when the SNR is -4dB, ClassBD achieves F1 scores of 88.40%, whereas the raw networks only yield 64.07%.

Table 3.14: The F1 scores (%) of the compared backbone networks on the PU datasets.

Working Condition	Backbone Network	SNR=-4dB			SNR=-2dB		
		Raw	ClassBD	Improvement	Raw	ClassBD	Improvement
N09M07F10	MobilieNetV3	71.86%	84.49%	12.64%	77.84%	97.20%	19.36%
	ResNET18	63.39%	84.99%	21.60%	73.11%	74.29%	1.17%
	WDCNN	50.75%	92.08%	41.33%	65.74%	95.35%	29.61%
	Transformer	61.15%	82.34%	21.19%	64.87%	87.69%	22.82%
N15M01F10	MobilieNetV3	84.30%	82.65%	-1.66%	91.32%	96.15%	4.83%
	ResNET18	3.70%	88.57%	84.88%	22.78%	94.22%	71.44%
	WDCNN	78.67%	95.19%	16.51%	90.71%	98.87%	8.16%
	Transformer	75.94%	81.45%	5.51%	82.76%	89.29%	6.54%
N15M07F04	MobilieNetV3	86.95%	94.55%	7.60%	91.37%	97.82%	6.45%
	ResNET18	40.57%	87.28%	46.71%	56.53%	96.46%	39.93%
	WDCNN	81.59%	96.52%	14.94%	92.60%	97.35%	4.75%
	Transformer	83.68%	88.37%	4.69%	79.41%	80.34%	0.93%
N15M07F10	MobilieNetV3	89.19%	91.99%	2.80%	91.80%	97.66%	5.86%
	ResNET18	3.68%	94.05%	90.37%	88.89%	95.41%	6.52%
	WDCNN	74.35%	80.13%	5.78%	59.33%	97.35%	38.02%
	Transformer	75.35%	89.97%	14.62%	76.97%	94.88%	17.91%
Average		64.07%	88.40%	24.30%	75.38%	93.15%	17.77%

Subsequently, we also test these models on the JNU and HIT datasets, setting the SNRs to -6dB and -4dB respectively. As depicted in Table 3.15, ClassBD exhibits commendable performance on

both datasets, and all backbones experience a performance boost. Specifically, on the JNU dataset, the F1 scores exceed 90% post the employment of ClassBD, irrespective of the backbone performance. Furthermore, a substantial improvement is observed in the Transformer. The raw Transformer initially yields F1 scores around 50% on the JNU and HIT datasets, which, after the application of ClassBD, escalates to an average F1 score of 90%.

Table 3.15: The F1 scores (%) of the compared backbone networks on the JNU and HIT datasets.

Dataset	Backbone Network	SNR=-6dB			SNR=-4dB		
		Raw	ClassBD	Improvement	Raw	ClassBD	Improvement
JNU	MobilieNetV3	84.16%	93.59%	9.42%	90.61%	96.73%	6.12%
	ResNET18	69.73%	95.31%	25.59%	6.69%	97.73%	91.04%
	WDCNN	94.35%	97.33%	2.98%	95.98%	98.54%	2.57%
	Transformer	54.25%	94.58%	40.32%	53.30%	96.81%	43.51%
Average		75.62%	95.20%	19.58%	61.65%	97.45%	35.81%
HIT	MobilieNetV3	68.62%	94.31%	25.69%	86.21%	95.38%	9.17%
	ResNET18	49.24%	86.18%	36.94%	85.03%	86.57%	1.53%
	WDCNN	85.11%	93.74%	8.63%	94.15%	97.23%	3.07%
	Transformer	42.95%	80.14%	18.26%	61.88%	95.58%	52.63%
Average		61.48%	88.59%	22.38%	81.82%	93.69%	16.60%

Finally, the results on the three datasets suggest that ClassBD can function as a plug-and-play denoising module, thereby enhancing the performance of deep learning classifiers under substantial noise. Considering both performance and model size, the simplest backbone, WDCNN, achieves consistent performance under all conditions. Consequently, we recommend it as the backbone for bearing fault diagnosis.

3.5.5 Employing ClassBD to machine learning classifiers

In comparison of deep learning models, classical machine learning (ML) classifiers offer some distinct advantages, including robust interpretability and lightweight models. However, these “shallow” ML methods invariably rely on human-engineered features for bearing fault diagnosis, thereby exhibiting limited generalization ability in the design of end-to-end diagnosing models (S. Zhang et al., 2020). Given that ClassBD can enhance the performance of deep learning classifiers, we posit that it can also serve as a feature extractor to augment the performance of classical ML classifiers.

Consequently, in this experiment, we utilize the pre-trained ClassBD as a feature extractor and

feed the output of ClassBD into several ML classifiers for comparison: support vector machine (SVM) (Cortes & Vapnik, 1995), k-nearest neighbor (KNN) (Mucherino et al., 2009), random forest (RF) (Ho, 1995), logistic regression (LR) (Cox, 1958), and a highly efficient gradient boosting decision tree (LightGBM) (Ke et al., 2017).

The results are presented in Table 3.16. Evidently, ClassBD significantly facilitates the performance of ML methods. Directly inputting raw signals into these ML classifiers results in markedly poor performance. On the JNU and HIT datasets, SVM and RF even fail to converge. However, with the incorporation of ClassBD, the classification performance experiences a substantial improvement. For instance, the KNN achieves a 90.71% F1 score on the JNU dataset, compared to a mere 2.89% F1 score without ClassBD. This performance even surpasses some deep learning methods. Nonetheless, ML methods exhibit instability across different datasets. The best-performing ML methods can only achieve a 49.77% score on the PU dataset. Despite this, we believe that the combination of ClassBD and ML methods presents a promising solution, promoting the study of high interpretability and efficiency in diagnostic approaches. We will explore this topic further in our future work.

Table 3.16: The F1 scores (%) of different machine learning methods on three datasets. Where the bold-faced numbers are the better results, '-' denotes the model failed to converge.

Dataset	SVM		KNN		RF		LR		LightGBM	
	Raw	ClassBD	Raw	ClassBD	Raw	ClassBD	Raw	ClassBD	Raw	ClassBD
PU-N09M07F10 (-4dB)	16.59%	40.28%	6.80%	**49.77%**	7.70%	31.99%	15.48%	52.73%	12.30%	38.68%
JNU (-10dB)	-	89.74%	2.89%	**90.71%**	16.28%	90.30%	13.58%	88.06%	18.36%	89.38%
HIT (-10dB)	-	72.58%	1.90%	73.69%	-	68.91%	23.85%	49.28%	35.82%	**75.78%**

3.5.6 Feature extraction ability of quadratic and conventional networks

Previously, we have theoretically demonstrated that quadratic networks possess superior cyclostationary feature extraction ability to conventional networks. It is also necessary to validate the performance in practice. Therefore, we construct two time-domain filters using quadratic convolutional layers and conventional convolutional layers with an identical structure and then evaluate their feature-extraction performance on the JNU dataset subjected to -10 dB noise.

The signals are analyzed using the Fast-SC method (Antoni, Xin, & Hamzaoui, 2017). The results, as depicted in Figure 3.10, clearly demonstrate that the quadratic network outperforms in terms of feature extraction capability. The bright lines in the spectral coherence, highlight the quadratic network can extract cyclic frequency across high and low frequency bands. Despite the severe attenuation of

the signal amplitude due to the noise, the quadratic network effectively recovers the cyclic frequency of the signal. Remarkably, the amplitude of the initial few cyclic frequencies is even higher than that of the raw signal.

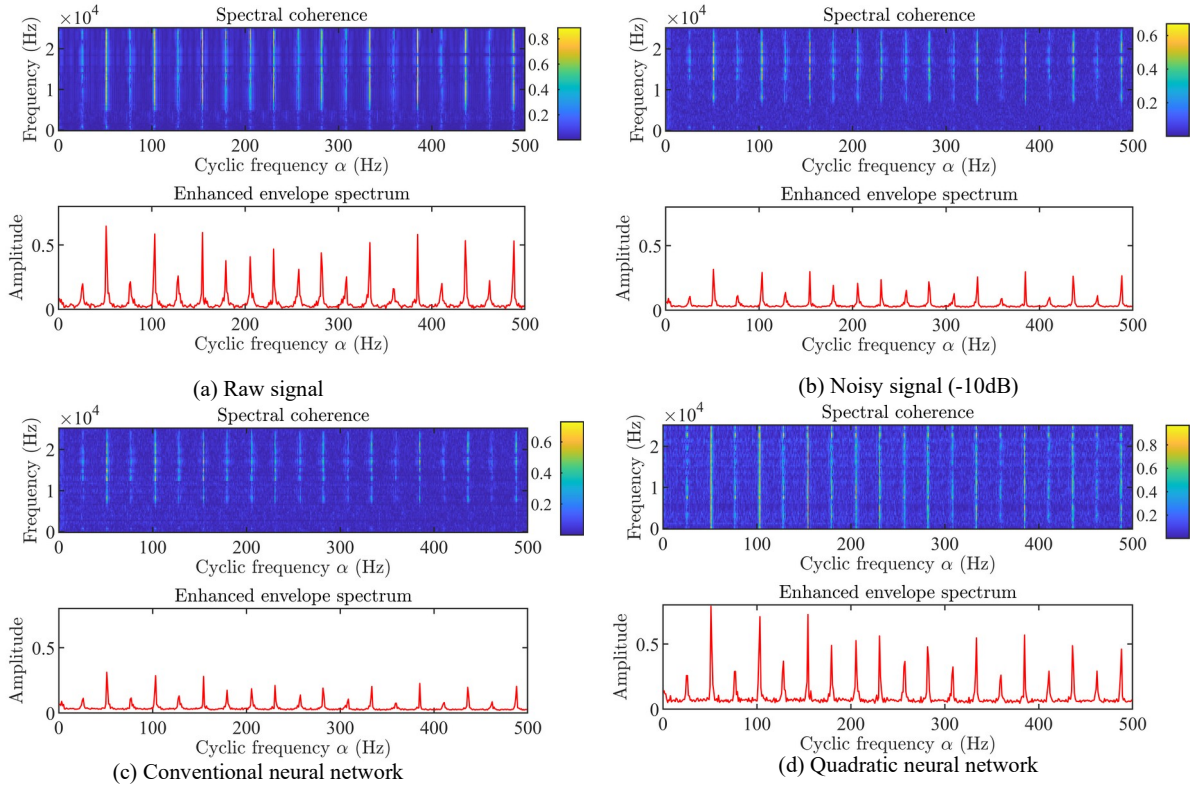


Figure 3.10: Denosing performance of the quadratic neural network and conventional network on the JNU dataset.

3.5.7 Comparison of ClassBD filters

Given that ClassBD comprises two filters, we explore their respective contributions in this experiment. Here we test four combinations: a standalone time domain filter (T-filter), a standalone frequency domain filter (F-filter), an F-filter followed by a T-filter, and our proposed scheme (a T-filter followed by an F-filter).

The results are presented in Table 3.17. Primarily, our scheme exhibits superior performance, indicating that both filters contribute significantly to the classification. Secondly, when comparing the single T-filter and F-filter, their performances are found to be dataset-dependent. On the PU and HIT datasets, T-filters outperform F-filters, whereas their performance is inferior to F-filters on the JNU dataset.

Table 3.17: The F1 scores (%) of different neural filters on three datasets. Where T-filter represents time domain quadratic convolutional filter, F-filter represents frequency domain filter

	T-filter	F-filter	F-T filter	T-F filter (Ours)
PU-N09M07F10 (-4dB)	63.01%	56.48%	58.23%	92.08%
JNU (-10dB)	82.28%	89.26%	88.44%	93.06%
HIT (-10dB)	61.99%	60.83%	64.11%	86.15%
Average	69.09%	68.86%	70.26%	**90.43%**

Lastly, the F-T filter demonstrates instability across the three datasets, with the average F1 score showing only about a 1% improvement compared to the single filter module. A plausible explanation for this is that the F-filter operates across the entire frequency domain. If it is employed as the first filter, it is susceptible to significant information loss. This is illustrated in Figure 3.11, where we plot the envelope spectrum of the outputs of the F-filter and T-filter. A comparison of raw and noisy signals reveals that the fault characteristic frequencies f_c are severely suppressed and distorted. While T-filters can still recover the f_c in different faults, the F-filter amplifies the energy across all frequency bands, causing the f_c to be overwhelmed. Therefore, our scheme initially employs a T-filter to extract f_c from the noisy signals, followed by an F-filter to enhance the energy in the frequency domain.

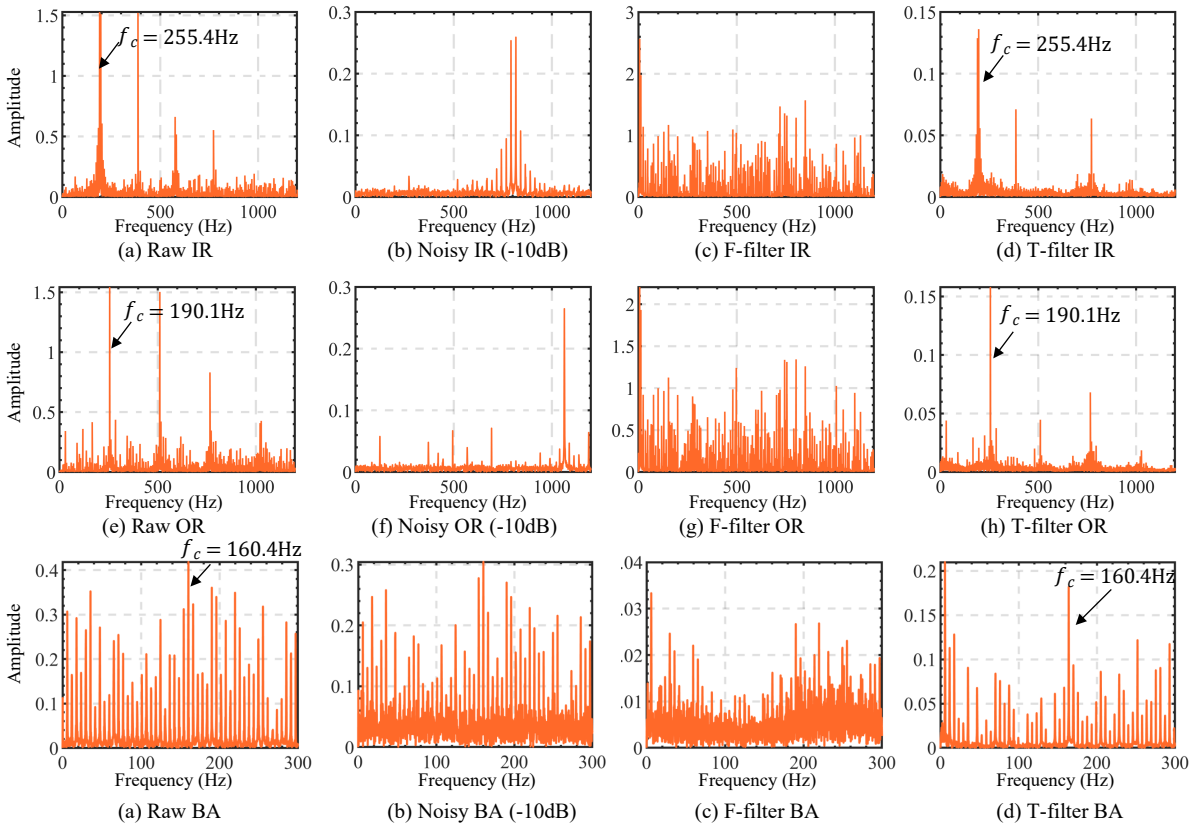


Figure 3.11: The envelope spectrum of the signals and the output of F/T filters on the HIT dataset. Where IR denotes inner race fault, OR denotes outer race fault, and f_c is the fault characteristic frequency.

3.6 Summary

(1) Discussions. ClassBD has demonstrated its effectiveness in enhancing the performance when integrated into various backbone models. This suggests a promising direction for applying ClassBD across diverse learning paradigms to address other challenging tasks. i) ClassBD can be utilized to construct a lightweight model suitable for online deployment. By leveraging feature-based knowledge distillation (KD) (Gou et al., 2021; G. Hinton, Vinyals, & Dean, 2015), our proposed framework could serve as a teacher model with ClassBD functioning as a feature extractor. The feature learned by ClassBD can be transferred to a lightweight student model. ii) It is also feasible to incorporate ClassBD into the transfer learning paradigm (Su et al., 2024; K. Zhou, Diehl, & Tang, 2023). ClassBD can function as a signal preprocessing module that can be embedded into any 1D neural network, such as domain adaptation networks (An et al., 2023; Long et al., 2015), deep adversarial networks (Ganin et al., 2016; S. Liu et al., 2022), and universal domain adaptation networks (You et al., 2019; Y. Zhang et al., 2023), thereby addressing cross-domain open-set, partial-set, and source-free issues. These ideas will be pursued in our future research study.

(2) Limitations. While ClassBD demonstrates promising denoising performance for bearing fault signals, it is not without its limitations. Firstly, the incorporation of additional BD loss functions may exacerbate the challenge of locating global optimal values. The difficulty of optimization escalates when adapting ClassBD to alternative learning paradigms. Secondly, the interpretability of our method remains partially unresolved. Our approach continues to rely on the stacking of multilayer neural networks, and the features learned in the hidden layers remain unexplained. This issue awaits future advancements in the field of deep learning interpretability. Thirdly, the BD method aims to reconstruct the transient impulse, which inherently broadens the frequency band. The primary challenges lie in phase shifts and the difficulty in restoring the amplitude of the raw signal. Future work could mitigate these limitations by redesigning BD filters. For instance, integrating filters across the time, frequency, and time–frequency domains to preserve more information from the raw signal, thereby reducing amplitude ambiguity.

(3) Conclusions. In this study, we have introduced a novel approach termed as ClassBD for bearing fault diagnosis under heavy noisy conditions. ClassBD is composed of cascaded time and frequency neural BD filters, succeeded by a deep learning classifier. Specifically, the time BD filter incorporates quadratic convolutional neural networks (QCNN), and we have mathematically proved

its superior capability in extracting periodic impulse features in the time domain. The frequency BD filter includes a fully-connected linear filter, supplementing the frequency domain filter subsequent to the time filter. Furthermore, a deep learning classifier is directly integrated to empower classification capabilities. We have devised a physics-informed loss function composed of kurtosis, l_2/l_4 norm, and cross-entropy loss to facilitate the joint learning. This unified framework transforms traditional unsupervised BD into supervised learning, providing interpretability due to its retention of conventional BD operations. Finally, comprehensive experiments conducted on three public and private datasets reveal that ClassBD outperforms other state-of-the-art methods, achieving an average of 95% F1 scores on several bearing datasets under heavy noise. ClassBD represents the first BD method that can be directly applied to classification and it exhibits good noise resistance and portability. Therefore, ClassBD holds great potential for further generalization to other difficult tasks such as cross-domain and small sample issues in future research.

Chapter 4

Quadratic Neuron-Empowered Heterogeneous Autoencoder for Unsupervised Anomaly Detection

4.1 Introduction

Deep learning has achieved great success in many important fields (LeCun, Bengio, & Hinton, 2015). However, so far, deep learning innovations mainly revolve around structure design, such as increasing depth and proposing novel shortcut architectures (F.-L. Fan et al., 2022; G. Huang et al., 2017; Panda et al., 2023). Almost exclusively, the existing advanced networks only involve the same type of neurons characterized by i) the inner product of the input and the weight vector; ii) a nonlinear activation function. For convenience, we call this type of neuron *conventional neurons*. Although these neurons can be interconnected to approximate a general function, we argue that the type of neurons useful for deep learning should not be unique. As we know, neurons in a biological neural system are diverse, collaboratively generating all kinds of intellectual behaviors. Since a neural network was initially devised to imitate a biological neural system, neuronal diversity should be carefully examined in neural network research.

In this context, a new type of neurons called *quadratic neurons* was recently proposed in (F. Fan et al., 2019), which replaces the inner product in a conventional neuron with a simplified quadratic function. Hereafter, we call a network made of quadratic neurons a *quadratic network* and a network made of conventional neurons a *conventional network*. The superiority of quadratic networks over

conventional networks in representation efficiency and capacity was demonstrated in (F.-L. Fan et al., 2023). Intuitively, nonlinear features ubiquitously exist in the real world, and quadratic neurons are empowered with a quadratic function to represent nonlinear features efficiently and effectively. Mathematically, when the rectified linear unit (ReLU) is employed, a conventional model is a piecewise linear function, whereas a quadratic one is a more powerful piecewise nonlinear mapping. Since a quadratic neuron shows great potential, we are encouraged to keep pushing its theory and application boundaries.

Previously, it was shown that there exists a function such that a conventional network needs an exponential number of neurons to approximate, but a quadratic network only needs a polynomial number of neurons to achieve the same level of error (F. Fan, Xiong, & Wang, 2020). Despite the attraction of this construction, with the help of techniques in (Eldan & Shamir, 2016), we find that one can also construct a function that can realize the opposite situation, *i.e.,* a function that is hard to approximate by a quadratic network but easy to approximate by a conventional network (Theorem 2). Combining these two constructed functions, one can naturally get a function easy to approximate by a heterogeneous network of both quadratic and conventional neurons but hard by a purely conventional or quadratic network, where the difference remains polynomial vs exponential (Theorem 1). Moreover, the opposite case will not happen because the purely conventional or quadratic is a degenerated case of the heterogeneous network. Such a theoretical derivation makes the employment of heterogeneous models well grounded, implying that there are at least some tasks pretty suitable for the combination of conventional and quadratic neurons so that an exponential difference is reached. While for the general task, heterogeneous networks' performance will not be worse, since a purely conventional or quadratic network is a special case of a heterogeneous network.

This theoretical derivation agrees with our intuition. There is no one-size-fits-all artificial neuron in deep learning. Neither conventional nor quadratic neurons can perform well on all kinds of tasks. Encouraged by the inspiring theoretical result on heterogeneous networks and the idea of neuronal synergy, here we directly integrate conventional and quadratic neurons in an autoencoder to make a new type of *heterogeneous autoencoders*. Autoencoders (Dewangan & Maurya, 2022; F. Fan et al., 2019; Y. Li et al., 2022; Ng, 2011) are widely used in unsupervised learning tasks such as feature extraction (X. Zhang, Zhang, & Song, 2023), denoising (Majumdar, 2019), anomaly detection (Tao et al., 2022), and so on. To the best of our knowledge, all the previous mainstream autoencoders use

the same type of neurons regardless of their innovations. The proposed heterogeneous autoencoder is the first heterogeneous autoencoder made up of diverse neurons in the family of autoencoders. Furthermore, we apply the heterogeneous autoencoder to solve the anomaly detection problem.

Anomaly detection is widely applied in different contexts such as intelligent manufacturing, data management, and intelligent transportation (H. Liu et al., 2021; Takiddin et al., 2023). However, anomaly detection is challenging due to the following issues (Pang et al., 2021):

- **Unknownness.** The anomaly features may include the unknown data, which requires an anomaly detection model to accurately construct the distribution of normal samples and be sensitive to anomalous samples.

- **Heterogeneity.** Anomalies are irregular, and different anomalies may display completely different abnormal features. For instance, when detecting network traffic attacks, four types of attacks are all anomalies (DOS, R2L, U2R, Probing).

- **Unnoticeability.** Anomalies are typically rare and are easily overwhelmed by normal instances. Therefore, it is difficult to detect the anomaly.

Based on the above analyses, unsupervised anomaly detection highly relies on the models' ability to feature representation. A model with a high feature representation ability can characterize a variety of anomaly data (heterogeneity), discriminate the anomaly from the normal (unnoticeability), and accurately learn the distribution of normal samples (unknownness). Our theory indicates that a heterogeneous network has a more powerful representation ability than a homogeneous network; therefore, a heterogeneous network is suitable to address the anomaly detection problem. Experiments demonstrate that a heterogeneous autoencoder achieves competitive performance relative to other state-of-the-art methods on anomaly detection datasets and real-world bearing fault detection. In summary, our contributions are

1. We prove that to approximate a constructed function, a heterogeneous network is more efficient and has a more powerful capability than a homogeneous one. Different from ensemble learning, our result characterizes the ability of heterogeneous networks from the perspective of approximation theory, which is a novel theoretical angle and paves a way for the employment of heterogeneous networks.

2. To further verify our theory, we develop a first-of-its-kind autoencoder that integrates essentially different types of neurons in one model, which is a methodological contribution to autoencoder design.

3. Experiments suggest that the proposed heterogeneous autoencoder delivers competitive performance compared to homogeneous autoencoders and other baselines on unsupervised tabular data anomaly detection.

4.2 Related works

4.2.1 Anomaly detection

Anomaly detection (AD) algorithms strive to identify outliers that deviate significantly from the majority of data (Han et al., 2022). This study concentrates on unsupervised AD, which exclusively utilizes samples from the normal class to construct a model for outlier detection (Hojjati & Armanfard, 2023). Broadly, AD algorithms can be categorized into traditional machine learning (shallow) and deep learning-based (deep) methods (Ruff et al., 2021). First, traditional machine learning methods are roughly summarised as follows: i) Classification-based method, one-class support vector machine (OCSVM) (Schölkopf et al., 2001) and support vector data description (SVDD) (Tax & Duin, 2004); ii) Probabilistic-based method, Gaussian mixture model (GMM) (Reynolds et al., 2009) and kernel density estimation (KDE) (Parzen, 1962); iii) Reconstruction-based method, probabilistic principal component analysis (p-PCA) (Tipping & Bishop, 1999); iv) Distance-based method, local outlier factor (LOF) (Breunig et al., 2000) and empirical cumulative outlier detection (ECOD) (Z. Li et al., 2022).

On the other hand, deep learning-based methods that are capable of handling larger and higher-dimensional data have recently garnered increased attention. Compared to machine learning methods, deep neural network methods demonstrate superior generalizability to various types of data. Several approaches integrated deep neural networks into machine learning methods, such as DeepSVDD (Ruff et al., 2018), deep autoencoding GMM (DAGMM) (Zong et al., 2018), and deep autoencoding support vector data descriptor (DASVDD) (Hojjati & Armanfard, 2023). These approaches are predicated on certain prior assumptions. Conversely, other approaches adopted pure deep neural networks to construct the end-to-end AD models without requiring any prior knowledge. The primary objective

of these approaches is to enhance the performance and generalizability of AD methods across a broad spectrum of data. The most representative methods are autoencoders (AEs) (Y. Yao et al., 2024; J. Yu et al., 2024) and generative adversarial networks (GANs) (Randhawa et al., 2023; Schlegl et al., 2017), which automatically learn the latent features of normal samples to enlarge their differences from the outliers in the latent space.

This study specifically focuses on autoencoders for anomaly detection. Compared to GANs, AEs can primarily be adopted for anomaly detection and exhibit lower computational complexity. This also facilitates the easier integration of different neurons.

4.2.2 Autoencoder

Autoencoders (AEs) constitute a vibrant research field in machine learning and have attained significant success in a variety of tasks, including data generation (Ojo & Bouguila, 2024; Troullinou et al., 2024), fault diagnosis (Dewangan & Maurya, 2022; X. Zhang, Zhang, & Song, 2023), and anomaly detection (Y. Yao et al., 2024; J. Yu et al., 2024). Presently, AEs, as a class of reconstruction-based nonlinear dimension reduction models, have emerged as the most extensively adopted methods for unsupervised anomaly detection, owing to their lightweight and user-friendly variants (Ruff et al., 2021). The advantages of AEs, such as automatic latent feature extraction, reduction of dimensional curses, and potent non-linear representation ability, have been proven beneficial in addressing the challenges associated with anomaly detection. Recent developments in AEs can be categorized into two aspects. i) Ensemble learning has been adapted to the autoencoder: Pan *et al.* (Pan et al., 2021) proposed a synchronized heterogeneous autoencoder that uses an item-oriented and a user-oriented autoencoder to serve as a teacher and a student, respectively, towards the distillation of both label-level and feature-level knowledge. Some studies combine multiple AEs for heterogeneous data, *e.g.*, combining a denoising autoencoder and a recurrent autoencoder for non-sequential and sequential data, respectively (T. Li et al., 2018). ii) Some novel metrics to measure the reconstruction error: Hojjati *et al.* (Hojjati & Armanfard, 2023) proposed an anomaly score that combines the reconstruction error and the distance of the enclosing hypersphere in the latent representation, and constructed an autoencoder called DASVDD. The low-rank and sparse priors were integrated into the loss function of deep AEs and employed in the hyperspectral anomaly detection (S. Lin et al., 2024).

However, regardless of representation regularization, architecture modification, and direct com-

bination, these AEs are built upon the same type of neurons, which are conventional neurons. In contrast, the proposed heterogeneous autoencoders design an AE by replacing neuron types, which provides a fresh perspective on AE design.

4.2.3 Heterogeneous neural network

Heterogeneous neural networks, characterized by the integration of various sub-networks or models, have shown significant potential in tackling complex scenarios such as big data and multi-modality. Several notable works include the heterogeneous Rybak neural network (HRYNN) introduced by Qi et al. (2021), which is composed of multiple Rybak neural networks (RYNNs) and has demonstrated excellent image enhancement effects, particularly on the image edge details. Sadr, Pedram, and Teshnehlab (2020) constructed heterogeneous neural networks by integrating intermediate features from convolutional and recursive neural networks, exhibiting superior efficiency and generalization performance in sentiment analysis. Another example is a heterogeneous neural network that integrates a hierarchical fused neural fuzzy and neural network, providing effective representation learning of users and items for multiple recommendation problems (Pham et al., 2023).

On the other hand, some multimodal large language models (MLLMs), despite not being explicitly named "heterogeneous networks", can be broadly considered as a variant of heterogeneous neural network (Fu et al., 2023). These models utilize different sub-networks to handle data of different modalities. For instance, contrastive language-image pre-training (CLIP) (Radford et al., 2021) and OpenCLIP (Cherti et al., 2023) construct both image and text encoders to train image-text pairs. These large multimodal models have shown powerful performance in various downstream tasks, such as zero-shot classification, retrieval, linear probing, and end-to-end fine-tuning. The concept of heterogeneous networks is also implied in some text-to-image diffusion models. For example, ControlNet injects additional conditions into a neural network block and incorporates it as a branch of the large pre-trained stable diffusion model (Rombach et al., 2022) for fine-tuning (L. Zhang, Rao, & Agrawala, 2023). Another text-conditional image diffusion model, RAPHAEL, is capable of generating highly artistic images through text descriptions. It stacks space and time mixture-of-experts (MoEs) layers, which consist of distinguished neural structures and construct billions of diffusion paths from the input to the output (Xue et al., 2024).

In summary, the principle of heterogeneity is prevalent in the design of deep learning networks. Integrating diverse architecture is beneficial in handling complex tasks. This characteristic is also advantageous for unsupervised AD, which requires a strong representation ability to represent distinguished latent features. Our heterogeneous network is different from the above in terms of that the heterogeneity lies in the incorporation of two kinds of neurons, providing superior representation of both linear and quadratic features.

4.3 Power of heterogeneous networks

Here, we prove a non-trivial theorem that there exists a function that is easy to approximate by a heterogeneous network but hard to approximate by a purely conventional or quadratic network, *i.e.*, the difference in the number of neurons needed for the same level of error is polynomial vs exponential. This result suggests that at least for some tasks, combining conventional and quadratic neurons is instrumental to model's power and efficiency, which makes the employment of heterogeneous networks well-supported.

4.3.1 Preliminaries and assumptions

A quadratic neuron computes two inner products and one power term of the input vector and aggregates them for a nonlinear activation function (F. Fan et al., 2019). The formulation (Eq. 2.11) and related works of quadratic neural networks are put in Section 2.3.2. Moreover, symbols used in our theorem are shown in Table 4.1.

Given arbitrary functions p and q, we have the following assumptions.

(1) $\|\cdot\|$ is the L_2 norm, *i.e.*, $\|p\|^2 = \int_{\boldsymbol{x}} p^2(\boldsymbol{x})d\boldsymbol{x}$.

(2) The inner product of two functions is $\langle p, q\rangle_{L_2} = \int_{\boldsymbol{x}} p(\boldsymbol{x})q(\boldsymbol{x})d\boldsymbol{x}$.

(3) $\varphi(\boldsymbol{x}) = (\frac{R_d}{\|\boldsymbol{x}\|})^{d/2} J_{d/2}(2\pi R_d\|\boldsymbol{x}\|)$, where $J_{d/2} : \mathbb{R} \to \mathbb{R}$ is the Bessel function of the first kind. $\varphi(\boldsymbol{x})$ is the Fourier transform of the unit-volume ball in the frequency domain $\mathbf{1}_{\boldsymbol{\omega}\in R_d\mathbb{B}_d}$.

(4) For simplicity, we use $\hat{p}(\boldsymbol{\omega})$ to denote the Fourier transform of $p(\boldsymbol{x})$.

(5) The L_2-norm of $p(\boldsymbol{x})$ under a general measure $\mu = \varphi^2(\boldsymbol{x})$ is $\left[\int_{\boldsymbol{x}} p^2(\boldsymbol{x})\varphi^2(\boldsymbol{x})d\boldsymbol{x}\right]^{1/2}$. The

distance between p and q under a general measure $\mu = \varphi^2(\boldsymbol{x})$ equals to

$$\int (p-q)^2 \varphi^2 d\boldsymbol{x} = \|p\varphi - q\varphi\|^2 = \|\hat{p} * \mathbf{1}_{\boldsymbol{\omega} \in R_d \mathbb{B}_d} - \hat{q} * \mathbf{1}_{\boldsymbol{\omega} \in R_d \mathbb{B}_d}\|^2, \tag{4.1}$$

where $*$ is a convolution. For simplicity, let $\mathbb{E}_{\boldsymbol{x} \sim \mu}(p-q)^2 = \int (p-q)^2 \varphi^2 d\boldsymbol{x}$.

Table 4.1: A table of important symbols

Symbol	Description
$\boldsymbol{x}$	Input
d	The dimension of the input
$\boldsymbol{w}^r$, $\boldsymbol{w}^g$, $\boldsymbol{w}^b$	Weight vectors in a quadratic neuron
b^r, b^g, c	Biases
σ	The activation function
$\boldsymbol{\omega}$	Frequency elements
$\boldsymbol{v}$	Unit vector
$\mathbb{B}_d$	Unit Euclidean ball
$R_d \mathbb{B}_d$	Unit-volume Euclidean ball, R_d is a selected constant.
$\mathbf{1}_{\boldsymbol{\omega} \in R_d \mathbb{B}_d}$	Indicator function that elements within $R_d \mathbb{B}_d$ are one
$\varphi(\boldsymbol{x})$	Fourier transform of $\mathbf{1}_{\boldsymbol{\omega} \in R_d \mathbb{B}_d}$
$\mu = \varphi^2(\boldsymbol{x})$	A general measure
$\mathbb{E}_{\boldsymbol{x} \sim \mu}(p-q)^2$	The distance between p and q under the measure μ
f_1	A one-hidden-layer conventional network
f_2	A one-hidden-layer quadratic network
f_3	A one-hidden-layer heterogeneous network
$\tilde{g}_{ip}(\boldsymbol{x})$	The constructed inner-product function
$\tilde{g}_r(\boldsymbol{x})$	The constructed radial function
c_σ	The constant dependent on σ
c_1, c_2	Constants associated with the quadratic network
ϵ_1	Approximation error associated with quadratic networks
c_3, c_4	Constants associated with the conventional network
ϵ_2	Approximation error associated with conventional networks
C_1, C_2	Constants associated with heterogeneous networks
Span$\{\boldsymbol{v}\}$	The set of all linear combinations of $\boldsymbol{v}$
Span$\{\boldsymbol{v}\} + R_d \mathbb{B}_d$	The region resultant by convoluting Span$\{\boldsymbol{v}\}$ and $R_d \mathbb{B}_d$
$m_i(\boldsymbol{x})$	A continuous radial function
$\mathbb{S}^{d-1}$	The unit Euclidean sphere in $\mathbb{R}^d$
T	$T =$ Span $\{\boldsymbol{v}\} + R_d \mathbb{B}_d$
r	The radius
$h(r)$	$\frac{Area\left(r\mathbb{S}^{d-1} \cap T\right)}{Area\left(r\mathbb{S}^{d-1}\right)}$
c	A constant that restricts the upper bound of $h(r)$
l	The function $l = \frac{\widehat{\tilde{g}_{ip}\varphi}}{\|\tilde{g}_{ip}\varphi\|_{L_2}}$
q	The function $q = \frac{\widehat{f_2\varphi}}{\|f_2\varphi\|_{L_2}}$
$\tilde{g}_{ip}^{(t)}(\boldsymbol{x})$	Truncating $\tilde{g}_{ip}(\boldsymbol{x})$

(6) A one-hidden-layer conventional network of k neurons is $f_1(\boldsymbol{x}) = \sum_{i=1}^{k} a_i \sigma(b_i \boldsymbol{x}^\top \boldsymbol{v}_i + c_i)$, while a one-hidden-layer quadratic network is simplified as $f_2(\boldsymbol{x}) = \sum_{i=1}^{k} a_i \sigma(b_i \boldsymbol{x}^\top \boldsymbol{x} + c_i)$, in which keeping only the power term is sufficient to express the construction.

(7) All theorems assume that the activation function $\sigma(\cdot)$ satisfies $|\sigma(x)| \leq C(1 + |x|^\alpha), x \in \mathbb{R}$ for some constants C and α, where C and α are dependent on $\sigma(\cdot)$.

(8) Span$\{\boldsymbol{v}\}$ is the set of all linear combinations of $\boldsymbol{v}$.

(9) Span$\{\boldsymbol{v}\} + R_d\mathbb{B}_d$ is the region resultant from convolution between two regions: Span$\{\boldsymbol{v}\}$ and $R_d\mathbb{B}_d$. Geometrically, Span$\{\boldsymbol{v}\} + R_d\mathbb{B}_d$ is a hyper-cylinder. As Figure 4.1 shows, Fourier transform of a radial function is also radial in the frequency domain, while Fourier transform of an inner-product based function $g(\boldsymbol{x}^\top \boldsymbol{v})$ is supported over a line, denoted as Span$\{\boldsymbol{v}\}$. Then, $\hat{g} * \mathbf{1}_{\boldsymbol{\omega} \in R_d\mathbb{B}_d}$ will be supported over a region resultant from convolution between two regions: Span$\{\boldsymbol{v}\}$ and $R_d\mathbb{B}_d$. Geometrically, Span$\{\boldsymbol{v}\} + R_d\mathbb{B}_d$ is a hyper-cylinder.

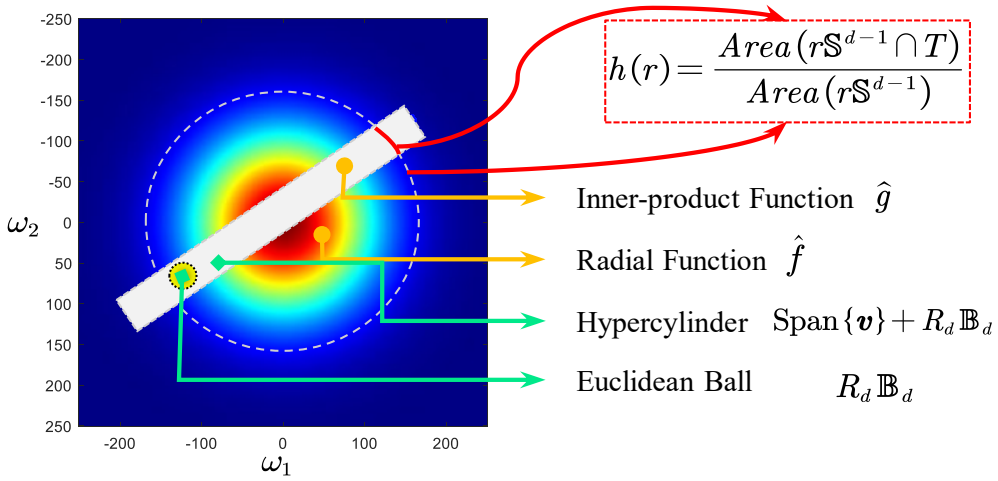


Figure 4.1: Fourier transforms of a radial function f and an inner-product based function g in the frequency domain.

4.3.2 Hetergeneous networks approximation theorem

Theorem 1 (Main). *There exist universal constants c_σ, $c_1, c_2, c_3, c_4, C_1, C_2, \epsilon_1, \epsilon_2 > 0$ such that the following holds: For every dimension d, there exists a measure μ and a function $\tilde{g}_{ip}(\boldsymbol{x}) + \tilde{g}_r(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$ with the following properties:*

1. *Every f_1 expressed by a one-hidden-layer conventional network of width at most $\frac{1}{2}c_1 e^{c_2 d}$ has*

$$\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_1 - (\tilde{g}_{ip} + \tilde{g}_r)]^2 \geq \epsilon_2. \tag{4.2}$$

2. *Every f_2 expressed by a one-hidden-layer quadratic network of width at most $\frac{1}{2}c_3 e^{c_4 d}$ has*

$$\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_2 - (\tilde{g}_{ip} + \tilde{g}_r)]^2 \geq \epsilon_1. \tag{4.3}$$

3. *There exists a function f_3 expressed by a one-hidden-layer heterogeneous network that can approximate $\tilde{g}_{ip} + \tilde{g}_r$ with $C_1 c_\sigma d^{3.75} + C_2 c_\sigma$ neurons.*

The sketch of proof. A comparison of three theorems are depicted in Table 4.2. It has been shown that there exists a function $\tilde{g}_r$ hard to approximate by a one-hidden-layer conventional network and easy to approximate by a one-hidden-layer quadratic network (Theorem 3). We only need to construct a function $\tilde{g}_{ip}$ hard to approximate by a one-hidden-layer quadratic network and easy to approximate by a one-hidden-layer conventional network (Theorem 2). Then, $\tilde{g}_{ip} + \tilde{g}_r$ will be hard for both conventional and quadratic networks but easy for a heterogeneous network (Theorem 1).

Table 4.2: The approximability of $\tilde{g}_{ip}$, $\tilde{g}_r$, and $\tilde{g}_{ip} + \tilde{g}_r$.

	construction	conventional f_1	quadratic f_2	heterogeneous $f_1 + f_2$
Theorem 3	$\tilde{g}_r$ (radial)	✗	✓	-
Theorem 2	$\tilde{g}_{ip}$ (inner-product)	✓	✗	-
Theorem 1	$\tilde{g}_{ip} + \tilde{g}_r$	✗	✗	✓

Remark 1. The standard quadratic neuron is $\sigma((\boldsymbol{x}^\top \boldsymbol{w}^r + b^r)(\boldsymbol{x}^\top \boldsymbol{w}^g + b^g) + (\boldsymbol{x} \odot \boldsymbol{x})^\top \boldsymbol{w}^b + c)$. By excluding interaction terms, we do not intend to weaken the expressive ability of quadratic neurons to establish the superiority of heterogeneous networks. Instead, this is because the target function $\tilde{g}_{ip}$ is a function made of an inner product. Our theorem is existential. Even if we include interaction terms, to represent $\tilde{g}_{ip}$, we can find a network whose every quadratic neuron actually takes $\boldsymbol{w}^g = 0, b^g = 1, \boldsymbol{w}^b = 0, c = 0$. Such a network is essentially a conventional network, with quadratic neurons degenerating into conventional ones. In this regard, it is true that a standard quadratic network can express $\tilde{g}_{ip} + \tilde{g}_r$. But because some quadratic neurons degenerate into conventional neurons, this quadratic network is essentially a heterogeneous network. If the constructed function $\tilde{g}_{ip}$ is not based on inner-product, the original form of quadratic neurons must be adopted.

Theorem 2. *There exist universal constants c_σ, $c_1, c_2, C_1, \epsilon_1 > 0$ such that the following holds: For every dimension d, there exists a measure μ and an inner-product based function $\tilde{g}_{ip}(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$ with the following properties:*

1. *Every f_2 that is expressed by a one-hidden-layer quadratic network of width at most $\frac{1}{2}c_1 e^{c_2 d}$ has*

$$\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_2 - \tilde{g}_{ip}]^2 \geq \epsilon_1. \tag{4.4}$$

2. *There exists a function f_1 that is expressed by a one-hidden-layer conventional network that can approximate $\tilde{g}_{ip}$ with $C_1 c_\sigma$ neurons.*

Based on our proof, the relation between ϵ_1 and c_1, c_2 is characterized by

$$\epsilon_1 \leq \|\tilde{g}_{ip}\|_{L_2(\mu)} \sqrt{2(c_1/2 - k\exp(-c_2 d)}, \tag{4.5}$$

while

$$0 < \epsilon_1 \leq \|\tilde{g}_r\|_{L_2(\mu)} \sqrt{2(c_3/2 - k\exp(-c_4 d)} \tag{4.6}$$

according to Theorem 1 of (F. Fan, Xiong, & Wang, 2020).

Theorem 3 (Theorem 1 of (F. Fan, Xiong, & Wang, 2020))**.** *There exist universal constants $c_\sigma, c_3, c_4, C_2, \epsilon_2 > 0$ such that the following holds: For every dimension d, there exists a measure μ and a radial function $\tilde{g}_r(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$ with the following properties:*

1. *Every f_1 that is expressed by a one-hidden-layer conventional network of width at most $\frac{1}{2}c_3 e^{c_4 d}$ has*

$$\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_1 - \tilde{g}_r]^2 \geq \epsilon_2. \tag{4.7}$$

2. *There exists a function f_2 expressed by a one-hidden-layer quadratic network that can approximate $\tilde{g}_r$ with $C_2 c_\sigma d^{3.75}$ neurons.*

Proof. The detailed proof can be found in Theorem 1 of (F. Fan, Xiong, & Wang, 2020) and Proposition 13 of (Eldan & Shamir, 2016). □

The key idea of constructing $\tilde{g}_{ip}$. The purpose of construction is to enlarge the difference between $\tilde{g}_{ip}$ and f_2. First, since $\hat{f}_2(\boldsymbol{\omega})$ is radial, $\hat{\tilde{g}}_{ip}(\boldsymbol{\omega})$ is preferred to be supported over as small area as possible such that $\hat{f}_2$ and $\hat{\tilde{g}}_{ip}$ have a small overlap. Therefore, $\tilde{g}_{ip}$ is cast as an inner-product-based function whose Fourier transform is only supported over a line. Second, still because $\hat{f}_2(\boldsymbol{\omega})$ is radial, we wish

$\tilde{g}_{ip}$ to have unneglectable high-frequency elements because the portion of overlap between f_2 and $\tilde{g}_{ip}$ is lower in high-frequency domain than in low-frequency domain. Specifically, given a unit vector $\boldsymbol{v}$, let $\tilde{g}_{ip}(\boldsymbol{x}) = \frac{1}{2\pi}\sqrt{\frac{4R_d}{1-\delta}}\text{sinc}\Big(\frac{2R_d}{1-\delta}\boldsymbol{x}^\top\boldsymbol{v}\Big), \delta \in [0,1]$. In the following, Lemma 1 implies that $\tilde{g}_{ip}$ has certain high-frequency elements, and Lemma 2 suggests that an inner-product based function and a radial function is incompatible, since their inner-product is upper bounded exponentially.

Lemma 1. *Given a unit vector* $\boldsymbol{v}$, $\tilde{g}_{ip}(\boldsymbol{x}) = \frac{1}{2\pi}\sqrt{\frac{4R_d}{1-\delta}}\text{sinc}\Big(\frac{2R_d}{1-\delta}\boldsymbol{x}^\top\boldsymbol{v}\Big)$ *satisfies* $\int_{2R_d\mathbb{B}_d}\hat{\tilde{g}}^2_{ip}(\boldsymbol{\omega})d\boldsymbol{\omega} = 1-\delta, \delta \in [0,1]$.

Proof. Denote the support of $\hat{\tilde{g}}_{ip}(\boldsymbol{\omega})$ as $\text{Span}\{\boldsymbol{v}\}$, which is $\{\boldsymbol{\omega} = t\boldsymbol{v} \mid t \in \mathbb{R}\}$. $\tilde{g}_{ip}(\boldsymbol{x})$ is mathematically formulated as

$$\hat{\tilde{g}}_{ip}(\boldsymbol{\omega}) = \begin{cases} \sqrt{\frac{1-\delta}{4R_d}}, & \boldsymbol{\omega} = t\boldsymbol{v}, t \in [-\frac{2R_d}{1-\delta}, \frac{2R_d}{1-\delta}] \\ 0, & \boldsymbol{\omega} = t\boldsymbol{v}, t \notin [-\frac{2R_d}{1-\delta}, \frac{2R_d}{1-\delta}]. \end{cases} \tag{4.8}$$

Because $\hat{\tilde{g}}_{ip}$ is a constant over $[-\frac{2R_d}{1-\delta}, \frac{2R_d}{1-\delta}]$, we have $\int_{2R_d\mathbb{B}_d}\hat{\tilde{g}}^2_{ip}(\boldsymbol{\omega})d\boldsymbol{\omega} = \int_{2R_d\mathbb{B}_d}\frac{1-\delta}{4R_d}d\boldsymbol{\omega} = \frac{1-\delta}{4R_d}\int_{[-2R_d,2R_d]}dt = 1-\delta$, which concludes the proof. □

Lemma 2. *Let* $g, f : \mathbb{R}^d \to \mathbb{R}$ *be two functions of unit norm. Moreover,* $\int_{2R_d\mathbb{B}_d} g^2(\boldsymbol{x})d\boldsymbol{x} \le 1-\delta$, $\delta \in [0,1]$, g *is supported over* $\text{Span}\{\boldsymbol{v}\} + R_d\mathbb{B}_d$, *and* f *is a combination of* k *radial functions, denoted as* $f = \sum_{i=1}^k m_i(\|\boldsymbol{x}\|)$, *where* $m_i(\|\boldsymbol{x}\|)$ *is a continuous radial function. Then,*

$$\langle f, g\rangle_{L_2} \le 1 - \frac{\delta}{2} + k\exp(-cd/2),$$

where $c > 0$ *is a constant dependent on* f.

Proof. Let $T = \text{Span}\{\boldsymbol{v}\} + R_d\mathbb{B}_d$. For any radius $r > 0$, define $h(r) = \frac{Area\big(r\mathbb{S}^{d-1}\cap T\big)}{Area\big(r\mathbb{S}^{d-1}\big)}$, where $\mathbb{S}^{d-1}$ is the unit Euclidean sphere in $\mathbb{R}^d$, and $Area(\cdot)$ is to compute the area. The geometric meaning of $h(r)$ is the ratio of the area of intersection of T and $r\mathbb{S}^{d-1}$ vs the area of $r\mathbb{S}^{d-1}$. Because T is a hyper-cylinder, following the illustration in page 7 of (Eldan & Shamir, 2016), $h(r)$ is exponentially small, *i.e.,* $\exists\, c > 0$, such that

$$h(2R_d) \le \exp(-cd). \tag{4.9}$$

As r increases, the area of the intersection decreases but the area of the sphere increases. Therefore, $h(r)$ is monotonically decreasing.

Our goal is to show $\langle f, g\rangle_{L_2}$ can be upper bounded. We decompose the inner product $\langle f, g\rangle_{L_2}$ into the integrals in a hyperball $2R_d\mathbb{B}_d$ and the region out of the hyperball $(2R_d\mathbb{B}_d)^C$. Mathematically,

$$\langle f, g\rangle_{L_2} = \int_{2R_d\mathbb{B}_d} f(\boldsymbol{x})g(\boldsymbol{x})d\boldsymbol{x} + \int_{(2R_d\mathbb{B}_d)^C} f(\boldsymbol{x})g(\boldsymbol{x})d\boldsymbol{x}. \tag{4.10}$$

Now, we calculate the above two integrals, respectively. For the first integral, we bound it as follows:

$$\begin{aligned}\int_{2R_d\mathbb{B}_d} f\left(\boldsymbol{x}\right) g\left(\boldsymbol{x}\right) d\boldsymbol{x} &\overset{(1)}{\leq} \left\|g \cdot \mathbf{1}_{\{2R_d\mathbb{B}_d\}}\right\|_{L_2} \|f\|_{L_2} \\ &\overset{(2)}{\leq} \sqrt{1-\delta},\end{aligned} \tag{4.11}$$

(1) follows from the Cauchy-Schwartz inequality; (2) follows from the fact that $f(\cdot)$ has unit L_2 norm and $\int_{2R_d\mathbb{B}_d} g^2(\boldsymbol{x})d\boldsymbol{x} \leq 1-\delta$.

For the second integral, we have $\int_{(2R_d\mathbb{B}_d)^C} f\left(\boldsymbol{x}\right) g\left(\boldsymbol{x}\right) d\boldsymbol{x} = \int_{(2R_d\mathbb{B}_d)^C \bigcap T} f\left(\boldsymbol{x}\right) g\left(\boldsymbol{x}\right) d\boldsymbol{x}$ because $g(\boldsymbol{x}) = 0$ when $\boldsymbol{x} \notin T$. Then, we have

$$\begin{aligned}&\int_{(2R_d\mathbb{B}_d)^C \bigcap T} f\left(\boldsymbol{x}\right) g\left(\boldsymbol{x}\right) d\boldsymbol{x} \\ =&\sum_{i=1}^{k} \int_{(2R_d\mathbb{B}_d)^C \bigcap T} m_i\left(\|\boldsymbol{x}\|\right) g\left(\boldsymbol{x}\right) d\boldsymbol{x} \\ \leq&\sum_{i=1}^{k} \left(\sqrt{\int_{(2R_d\mathbb{B}_d)^C \bigcap T} m_i^2\left(\|\boldsymbol{x}\|\right) d\boldsymbol{x}} \sqrt{\int_{(2R_d\mathbb{B}_d)^C \bigcap T} g^2\left(\boldsymbol{x}\right) d\boldsymbol{x}}\right) \\ \overset{(1)}{\leq}&\sum_{i=1}^{k} \sqrt{\int_{r\geq 2R_d} \int_{\left(r\mathbb{S}^{d-1}\right) \bigcap T} m_i^2\left(r\right) d\mathbb{S} dr} \\ =&\sum_{i=1}^{k} \sqrt{\int_{r\geq 2R_d} \left(\int_{r\mathbb{S}^{d-1}} m_i^2\left(r\right) d\mathbb{S}\right) \cdot \left[\frac{\int_{\left(r\mathbb{S}^{d-1}\right)\cap T} m_i^2\left(r\right) d\mathbb{S}}{\int_{r\mathbb{S}^{d-1}} m_i^2\left(r\right) d\mathbb{S}}\right] dr} \\ \overset{(2)}{=}&\sum_{i=1}^{k} \sqrt{\int_{r\geq 2R_d} \int_{r\mathbb{S}^{d-1}} m_i^2\left(r\right) d\mathbb{S} \cdot h\left(r\right) dr} \\ \overset{(3)}{\leq}&\sum_{i=1}^{k} \sqrt{h\left(2R_d\right) \int_{r\geq 2R_d} \int_{r\mathbb{S}^{d-1}} m_i^2\left(r\right) d\mathbb{S} dr} \\ \overset{(4)}{=}&\sum_{i=1}^{k} \sqrt{h\left(2R_d\right)} \overset{(5)}{\leq} k\exp\left(-cd/2\right).\end{aligned} \tag{4.12}$$

In the above, (1) follows from the fact that $g(\cdot)$ has a unit L_2 norm; (2) holds because $m_i(r)$ is radial; (3) follows from $h(r)$ is monotonically decreasing; (4) follows from $f(\cdot)$ has a unit norm; (5) follows from Eq. (4.9).

Combining Eqs. (4.10), (4.11) and (4.12), we have $\langle f, g\rangle_{L_2} \leq \sqrt{1-\delta} + k\exp(-cd/2) \leq 1 - \frac{\delta}{2} + k\exp(-cd/2)$.

□

*Proof of **Theorem** 2.* The proof of Theorem 2 consists of two parts: (1) the inapproximability of a quadratic network to $\tilde{g}_{ip}$; (2) the approximability of a conventional network to $\tilde{g}_{ip}$.

(1) Define $l = \frac{\widehat{\tilde{g}_{ip}\varphi}}{\|\tilde{g}_{ip}\varphi\|_{L_2}}$. Because for $\tilde{g}_{ip}$, we have $\int_{2R_d\mathbb{B}_d} \tilde{g}_{ip}^2(\boldsymbol{x})d\boldsymbol{x} = 1-\delta$, there must exist a universal constant $c_1 \in [0,1]$ such that $\int_{2R_d\mathbb{B}_d} l^2(\boldsymbol{x})d\boldsymbol{x} \leq 1-c_1$. Define the function $q = \frac{\widehat{f_2\varphi}}{\|f_2\varphi\|_{L_2}}$, where $f_2 = \sum_i^k a_i\sigma(\omega_i\boldsymbol{x}^\top\boldsymbol{x}+b_i)$ is a radial function expressed by a quadratic network. Thus, according to Lemma 2, the functions $l(\cdot)$, $q(\cdot)$ satisfy

$$\langle l(\cdot), q(\cdot)\rangle_{L_2} \leq 1 - \frac{c_1}{2} + k\exp(-c_2 d/2), \tag{4.13}$$

with $c_2 > 0$ being a universal constant.

For every scalars $\beta_1, \beta_2 > 0$, we have

$$\|\beta_1 l(\cdot) - \beta_2 q(\cdot)\|_{L_2} \geq \frac{\beta_2}{2}\|l(\cdot) - q(\cdot)\|_{L_2}. \tag{4.14}$$

The reason why it holds true is as follows:

Without loss of generality, we assume that $\beta_2 = 1$. For two unit vectors u, v in one Hilbert space, it has

$$\begin{aligned}&\min_\beta \|\beta v - u\|^2 = \min_\beta \left(\beta^2\|v\|^2 - 2\beta\langle v,u\rangle + \|u\|^2\right)\\ =&\min_\beta \left(\beta^2 - 2\beta\langle v,u\rangle + 1\right) = 1 - \langle v,u\rangle^2 = \frac{1}{2}\|v-u\|^2.\end{aligned}$$

Next, combining Eqs. (4.13) and (4.14), and utilizing that q, l have unit L_2 norm, we have

$$\begin{aligned}&\|f_2 - \tilde{g}_{ip}\|_{L_2(\mu)} = \|f_2\varphi - \tilde{g}_{ip}\varphi\| = \|\widehat{f_2\varphi} - \widehat{\tilde{g}_{ip}\varphi}\|\\ =&\|(\|f_2\varphi\|)\, q(\cdot) - (\|\tilde{g}_{ip}\varphi\|_{L_2})\, l(\cdot)\|\\ \geq&\frac{1}{2}\|\tilde{g}_{ip}\varphi\|\|q(\cdot) - l(\cdot)\| = \frac{1}{2}\|\tilde{g}_{ip}\|_{L_2(\mu)}\|q(\cdot) - l(\cdot)\|_{L_2}\\ \geq&\frac{1}{2}\|\tilde{g}_{ip}\|_{L_2(\mu)}\sqrt{2\left(1 - \langle q,l\rangle_{L_2}\right)}\\ \geq&\|\tilde{g}_{ip}\|_{L_2(\mu)}\sqrt{2\max\left(c_1/2 - k\exp\left(-c_2 d\right), 0\right)},\end{aligned} \tag{4.15}$$

where the first inequality is due to Eq. (4.14). Therefore, as long as $k \leq \frac{c_1}{2}e^{c_2 d}$, $\|f_2 - \tilde{g}_{ip}\|_{L_2(\mu)}$ is larger than a positive constant ϵ_1.

(2) In contrast, f_1 can approximate $\tilde{g}_{ip}(\boldsymbol{x}) = \frac{1}{2\pi}\sqrt{\frac{4R_d}{1-\delta}}\mathrm{sinc}\left(\frac{2R_d}{1-\delta}\boldsymbol{x}^\top\boldsymbol{v}\right)$ with a polynomial number

of neurons. From the proof of Theorem 1 in (Debao, 1993), $\forall H : \mathbb{R} \to \mathbb{R}$ which is constant outside a bounded interval $[r_1, r_2]$, there exist scalars $a, \{\alpha_i, \beta_i, \gamma_i\}_{i=1}^k$, where $k \leq c_\sigma \frac{(r_2-r_1)L}{\epsilon}$ such that the function $h(t) = a + \sum_{i=1}^k \alpha_i \cdot \sigma(\beta_i t - \gamma_i)$ satisfies that $\sup_{t\in\mathbb{R}} |H(t) - h(t)| \leq \epsilon$.

According to this theorem, we define an auxiliary function $\tilde{g}_{ip}^{(t)}(\boldsymbol{x})$ by truncating $\tilde{g}_{ip}(\boldsymbol{x})$ when $\boldsymbol{x}^\top \boldsymbol{v} \notin [-\frac{1-\delta}{R_d\epsilon}, \frac{1-\delta}{R_d\epsilon}]$ to be zero, then $|\tilde{g}_{ip}^{(t)}(\boldsymbol{x}) - \tilde{g}_{ip}(\boldsymbol{x})| \leq |\tilde{g}_{ip}(\boldsymbol{x})| \leq \frac{1}{2\pi}\sqrt{\frac{4R_d}{1-\delta}}/(\frac{2R_d}{1-\delta} \cdot \boldsymbol{x}^\top \boldsymbol{v}) \leq \epsilon/2$ when $\boldsymbol{x}^\top \boldsymbol{v} \notin [-\frac{\sqrt{1-\delta}}{\pi\sqrt{R_d\epsilon}}, \frac{\sqrt{1-\delta}}{\pi\sqrt{R_d\epsilon}}]$. Applying the aforementioned theorem to $\tilde{g}_{ip}^{(t)}(\boldsymbol{x})$, we have $f_1(\boldsymbol{x}) = a + \sum_{i=1}^k \alpha_i \cdot \sigma(\beta_i \boldsymbol{x}^\top \boldsymbol{v} - \gamma_i)$ that can approximate $\tilde{g}_{ip}^{(t)}$ in terms of $\sup_{\boldsymbol{x}\in\mathbb{R}^d} |\tilde{g}_{ip}^{(t)}(\boldsymbol{x}) - f_1(\boldsymbol{x})| \leq \epsilon/2$. Here, $r_1 = -\frac{\sqrt{1-\delta}}{\pi\sqrt{R_d\epsilon}}$ and $r_2 = \frac{\sqrt{1-\delta}}{\pi\sqrt{R_d\epsilon}}$, the number of neurons needed is no more than $c_\sigma \frac{2\sqrt{1-\delta}L}{\pi\sqrt{R_d\epsilon}} \leq c_\sigma \frac{2\times\sqrt{5.264}\sqrt{1-\delta}L}{\pi\epsilon} = C_1 c_\sigma$. Because the geometric meaning of $1/R_d$ is the volume of d-hyperball, which is upper bounded by $(1/R_d)_{d=5} \approx 5.264$[1], the number of needed neurons is irrelevant to d. Furthermore, f_1 fulfills

$$\begin{aligned} &\sup_{\boldsymbol{x}\in\mathbb{R}^d} |\tilde{g}_{ip}(\boldsymbol{x}) - f_1(\boldsymbol{x})| \\ \leq &\sup_{\boldsymbol{x}\in\mathbb{R}^d} |\tilde{g}_{ip}(\boldsymbol{x}) - \tilde{g}_{ip}^{(t)}(\boldsymbol{x})| + \sup_{\boldsymbol{x}\in\mathbb{R}^d} |\tilde{g}_{ip}^{(t)}(\boldsymbol{x}) - f_1(\boldsymbol{x})| \leq \epsilon. \end{aligned} \tag{4.16}$$

□

*Proof of **Theorem** 1.* Theorems 3 and 2 suggest that a conventional network f_1 can express $\tilde{g}_{ip}$ with a polynomial number of neurons, and a quadratic network f_2 can express $\tilde{g}_r$ with a polynomial number of neurons. Let a heterogeneous network be $f_3 = f_1 + f_2$, f_3 can straightforwardly express $\tilde{g}_{ip} + \tilde{g}_r$ with a polynomial number of neurons.

Next, we need to show how to deduce from $\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_2 - \tilde{g}_{ip}]^2 \geq \epsilon_1$ to $\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_2 - (\tilde{g}_{ip} + \tilde{g}_r)]^2 \geq \epsilon_1$. Here, we slightly abuse ϵ_1 for succinctness. Both formulas mean that there exists a gap between functions. We can rewrite $\tilde{g}_r$ as

$$\tilde{g}_r(\|\boldsymbol{x}\|) = \tilde{g}_r \circ \sqrt{\|\boldsymbol{x}\|^2}. \tag{4.17}$$

Thus, we can express $\tilde{g}_r$ by a quadratic neuron using a special activation function $\sigma' = \tilde{g}_r \circ \sqrt{\cdot}$. It holds that $\sigma'(x) \leq C'(1 + |x|^{\alpha'})$ for certain constants C' and α', since both $\tilde{g}_r$ and s are bounded. As a result, regarding $f_2 - \tilde{g}_r$ as a quadratic network, we can get $\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_2 - (\tilde{g}_{ip} + \tilde{g}_r)]^2 = \mathbb{E}_{\boldsymbol{x}\sim\mu}[(f_2 - \tilde{g}_r) - \tilde{g}_{ip}]^2 \geq \epsilon_1$.

Similarly, $\tilde{g}_{ip}$ can be re-expressed by a conventional neuron with a bounded activation. We can

[1] https://en.wikipedia.org/wiki/Volume_of_an_n-ball

also get $\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_1 - (\tilde{g}_{ip} + \tilde{g}_r)]^2 \geq \epsilon_2$ from $\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_1 - \tilde{g}_r]^2 \geq \epsilon_2$.

□.

Remark 2. Will **Theorem 1** change if we use more layers? Indeed, our theorem and the associated proof cannot be extended into deeper networks. As Table 4.2 shows, the key ingredient of our main theorem is constructing $\tilde{g}_{ip} + \tilde{g}_r$, wherein $\tilde{g}_{ip}$ can be approximated by a one-hidden-layer conventional network with a polynomial number of neurons but a one-hidden-layer quadratic network needs an exponential number of neurons, and $\tilde{g}_r$ can be approximated by a one-hidden-layer quadratic network with a polynomial number of neurons but a one-hidden-layer conventional network needs an exponential number of neurons. The main reason accounting for such exponential differences is that it is hard for a one-hidden-layer conventional network to represent a radial function and a one-hidden-layer quadratic network to represent an inner-product-based function. However, if one allows a network to have more hidden layers, a conventional network can use a polynomial number of neurons to represent the function $g(x) = x^2$ first, and then use x^2 to construct a radial function with a polynomial number of neurons. Thus, the total number of neurons is still polynomial. In this light, although heterogeneous networks still enjoy the advantage of efficiency in representing $\tilde{g}_{ip} + \tilde{g}_r$, the difference is not exponential. A similar analysis also holds for $\tilde{g}_{ip}$ and the quadratic. We leave the extension to deeper networks as our future work.

Despite assuming a simple structure, our main theorem holds significant implications for unsupervised anomaly detection. As per the manifold hypothesis, anomalies often exhibit evident abnormal features in a low-dimensional space but become hidden or unnoticeable in a high-dimensional space (Fefferman, Mitter, & Narayanan, 2016). AEs are employed to extract the latent features of the data, thereby amplifying the differences between normal instances and anomalies (Ruff et al., 2021). When dealing with high-dimensional and complicated data, particularly in the context of unsupervised learning, our theorem assures that heterogeneous networks can approximate more flexible, more expressive, and nonlinear decision functions, thereby demonstrating superior representational ability compared to conventional networks. Furthermore, in our primary experiments, encoders and decoders for anomaly detection are designed to consist of a single hidden layer. Such heterogeneous networks indeed achieve competitive performance, as suggested by our theory, thereby showcasing high efficiency in the design of anomaly detection AEs.

4.4 Heterogeneous autoencoder

Given the input $\boldsymbol{x} \in \mathbb{R}^n$, let $E(\cdot)$ be the encoder and $D(\cdot)$ be the decoder, an autoencoder attempts to learn a mapping from the input to itself by optimizing the following loss function: $\min_{\boldsymbol{W}} \mathcal{J}(\boldsymbol{W}; \boldsymbol{x}) = \|D(E(\boldsymbol{x})) - \boldsymbol{x}\|^2$, where W is a collection of all network parameters for simplicity. $\boldsymbol{z} = E(\boldsymbol{x})$ is the learned embedding in the low-dimensional space. The objective of the autoencoder is to learn $\boldsymbol{z}$ such that the output layer can accurately reproduce the original input. Within this structure, we refer to AEs that only use conventional neurons or quadratic neurons in the encoder and decoder as homogeneous AEs, which are referred to as AE and QAE, respectively.

Our theoretical derivation suggests combining the conventional and quadratic neurons in a network is a promising direction to explore. But it does not provide specific guidelines on which structure is the best for HAEs. Thus, we can only heuristically design the structures of the autoencoder based on a symmetry consideration: 1) utilizing equal depth and width in the encoder and decoder, which fulfills the standard practice of AEs; 2) arranging conventional and quadratic layers in a symmetric fashion to put them on an equal booting, since we don't have any prior knowledge which neuron type is more important. Symmetrical consideration can naturally result in three heterogeneous autoencoder designs, referred to as HAE-X, HAE-Y, and HAE-I, respectively, as shown in Figure 4.2.

• The HAE-X conjugates a conventional and a quadratic branch in the encoder, which are fused in the intermediate layer by summation. Then, this sum is transmitted to a conventional and a quadratic branch in the decoder. Next, the yields of two branches in the decoder are summed again to produce the final model output. This approach not only facilitates the heterogeneity in the model but also preserves its existing backbone.

• The HAE-Y is a simplification of the HAE-X, which respects the HAE-X in the encoder but only has a quadratic branch in the decoder. The reason why we put quadratic neurons into the decoder is that quadratic neurons are more capable of information extraction, which can help reconstruct the input more accurately. HAE-Y explores how the model's outputs are affected when an encoder and a decoder possess a similar number of neurons but different types of neurons. This serves to reduce the number of parameters and complexity of heterogeneous autoencoders.

• Other than using conventional and quadratic layers in parallel as the HAE-X and HAE-Y models do, the HAE-I interlaces conventional and quadratic layers in both an encoder and a decoder while keeping the symmetry. In the HAE-I model, we make the first layer a quadratic layer, as the quadratic

layer is capable of extracting more information and ensures sufficient information to pass to subsequent layers. Since we hardly know which heterogeneous design is preferable, we examine them all in the experiments.

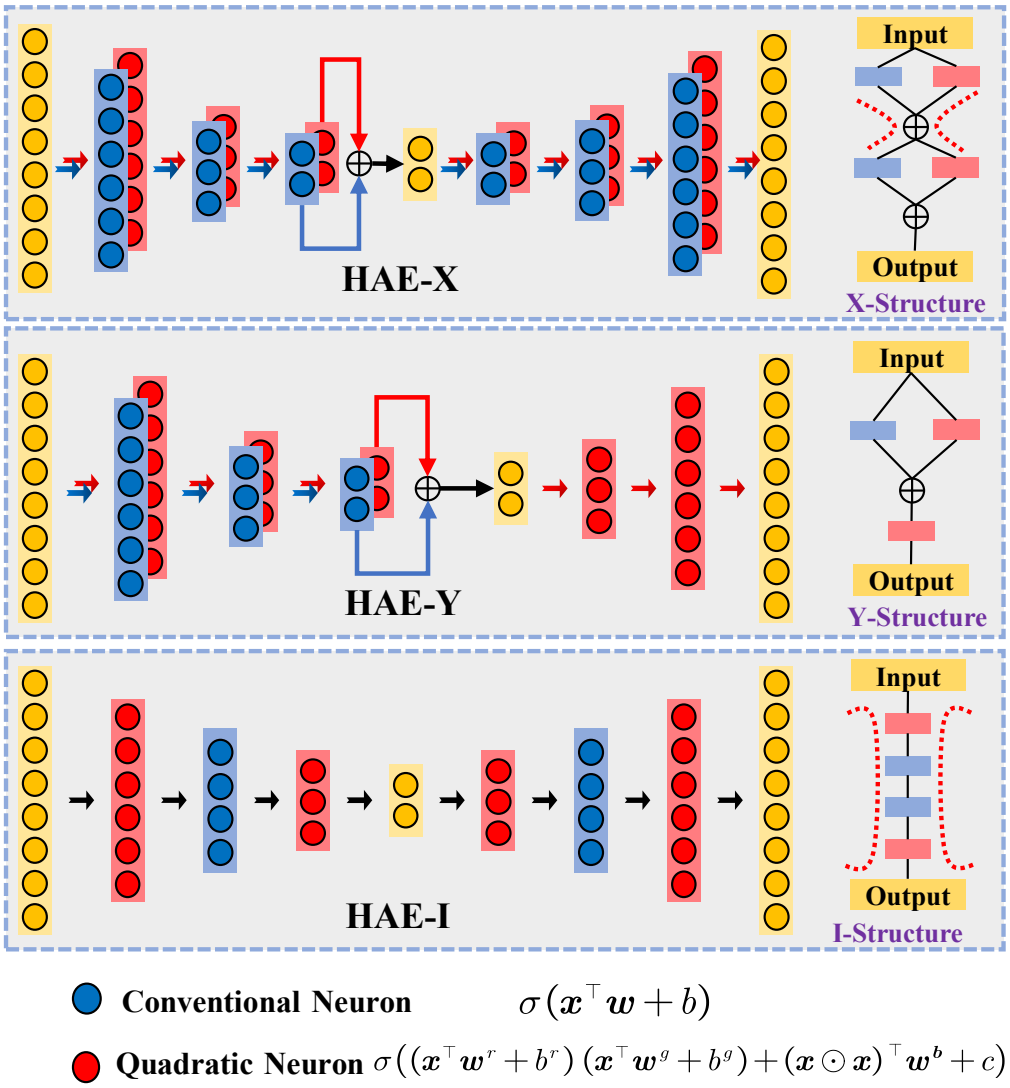


Figure 4.2: Three heterogeneous autoencoder designs.

4.5 Experiments

In this section, we apply the HAE to solve the anomaly detection task, by which we can also investigate if the HAE is a powerful model, as suggested by theory. The anomaly detection is to distinguish between the outliers and the normal via a threshold of contamination percentage after calculating the outlier score of each sample with a model. The experiments are twofold. First, we compare HAE with a purely conventional or quadratic autoencoder. Second, we compare the HAE with either classical or SOTA methods. The results show that the HAE outperforms its competitors on 8 benchmarks. All

quadratic neurons, except those used in an ablation study, are based on the standard quadratic neurons (Eq. 2.11).

4.5.1 Experimental settings

4.5.1.1 Tabular datasets descriptions

We choose eight datasets from ODDS (Outlier Detection DataSets)[2]: multi-dimensional point datasets and seven datasets from DAMI[3] (Campos et al., 2016), which consist of a variety of tabular datasets from different fields. We eliminate duplicate datasets from DAMI repositories. Data in these datasets are labeled into two classes: normal and abnormal. For instance, the Wisconsin-Breast Cancer (WBC) dataset includes measurements for breast cancer cases, classified as either benign or malignant. The characteristics of all datasets are summarized in Table 4.3. In all experiments, we use 80% normal samples for training and contaminated data for testing. The test set includes normal samples that are not used in the training set and all abnormal samples.

Table 4.3: Statistics of anomaly detection benchmark datasets.

Repositories	Datasets	#Samples	#Features	Outlier Ratio
OODs	Arrhythmia	452	274	14.60%
	Glass	214	9	4.21%
	Musk	3062	166	3.16%
	Optdigits	5216	64	2.88%
	Pendigits	6870	16	2.27%
	Pima	768	8	34.90%
	Verterbral	240	6	12.50%
	Wbc	378	30	5.56%
DAMI	ALOI	50000	27	3.01%
	Ionosphere	351	7	35.89%
	KDDCUP99	60632	41	0.41%
	Shuttle	1013	9	1.28%
	Waveform	3443	21	2.90%
	WDBC	367	30	2.72%
	WPBC	198	33	23.7%

4.5.1.2 Bearing datasets descriptions

We conduct experiments on bearing fault detection, which is a high-dimensional problem. Bearings are widely used in rotating machinery, but they are prone to damage. Anomaly detection determines

[2]http://odds.cs.stonybrook.edu
[3]http://www.dbs.ifi.lmu.de/research/outlier-evaluation/DAMI

whether the bearing is working under normal conditions, which is the key to bearing health management. Different from tabular data, bearing fault detection is based on long vibration signals. We use two datasets of run-to-failure bearing data to validate our method: one from the public(B. Wang et al., 2018) and the other provided by a nuclear power plant in Hainan Province, China.

(1) XJTU-SY dataset is collected by the Institute of Design Science and Basic Component at Xi'an Jiaotong University (XJTU) and the Changxing Sumyoung Technology Co., Ltd. (SY) (B. Wang et al., 2018). Two accelerometers of type PCB 352C33 (sampling frequency set to 25.6 kHz) are mounted at 90° on the housing of the tested bearings, which measure the vibration of the bearing. The accelerometers record 1.28s (32768 points) per minute until the bearing fails completely. We select the Bearing2_5 data which runs at 2250 rpm (37.5 Hz) and 11 kN load. The bearing's lifetime is 5 h 39 min and ends with an outer race fault.

(2) The seawater booster pump (SBP) dataset is gathered from Hainan Nuclear Power Co., Ltd. The SBP transfers cooling water to steam turbines and serves as a crucial component of the auxiliary cooling water system in nuclear power plants. Given the intricate nature of nuclear plants, mechanical systems require periodic monitoring to ensure their reliability. Figure 4.3 illustrates that acceleration sensors are placed at four points on both the pump and the motor when performing measurement.

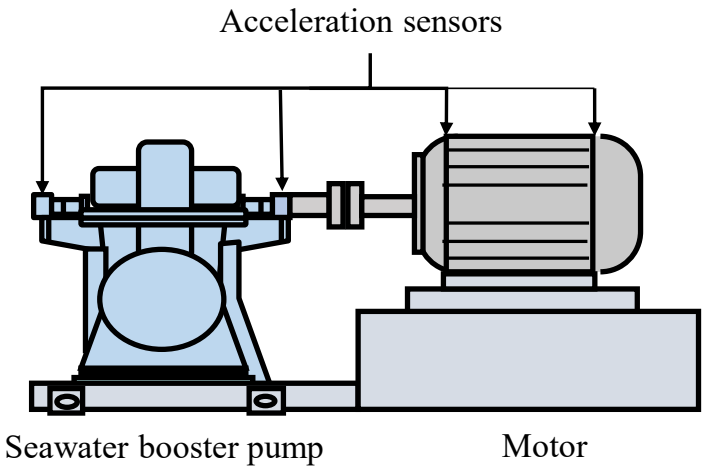


Figure 4.3: The measurement structure of the seawater booster pump.

The vibration signal for the seawater booster pumps is measured every two months, with a sampling rate of 16.8 kHz and a recording period of 0.4 seconds, which results in 6,721 data points. The faulty and healthy signals are shown in Figure 4.4. Data in this study were collected from 2015 to 2023. During this period, a bearing fault at the outer race is observed.

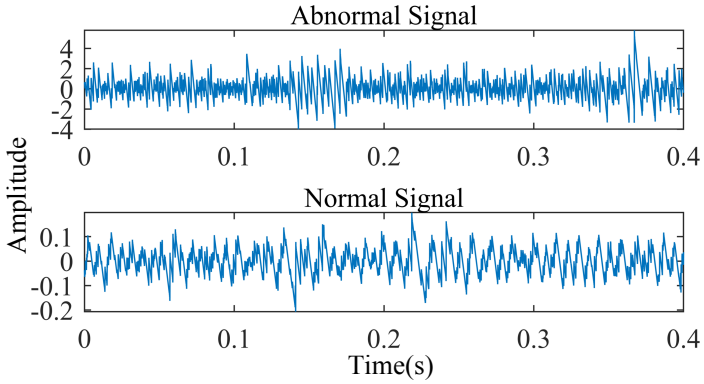


Figure 4.4: Abnormal and normal signals of the seawater booster pump.

We resample all signals to 1024 points per sample and use the FFT to split a single side of it into 512 points. The summary of the two datasets is in Table 4.4. Outliers in the XJTU dataset are identified based on the method described in (X. Huang et al., 2021). However, it should be noted that, unlike simulation tests, the availability of faulty data from real industrial scenes is very limited. As a result, only 6 samples were available for outlier detection in the SBP dataset, which makes this task pretty challenging.

Table 4.4: The summary of two bearing datasets. a×b represents the number of samples and sample dimensions.

Datasets	XJTU	SBP
#Raw Sample	334×32768	57×6721
#Abnormal Raw Sample	222×32768	1×6721
#Resample	8544×512	342×512
#Abnormal Resample	4671×512	6×512
Outlier Ratio	54.66%	1.75%

4.5.1.3 Baseline methods and training settings

We compare the proposed heterogeneous autoencoders with 7 baselines, SUDO (Y. Zhao et al., 2021), SO-GAAL (Y. Liu et al., 2019), DeepSVDD (Ruff et al., 2018), DAGMM (Zong et al., 2018), RCA (B. Liu et al., 2021), AE, and QAE (an autoencoder based on quadratic neurons). All these baselines are either classical methods with many citations or published in prestigious venues. To implement them, except for autoencoders, we utilize packages in the PyOD (Python Outlier Detection) package[4] (Y. Zhao, Nasrullah, & Li, 2019) or follow official scripts[5]. We program all autoencoder models by ourselves. We use the area under the receiver operating characteristic curve (AUC) and precision

[4]https://github.com/yzhao062/pyod
[5]https://github.com/illidanlab/RCA

(PRE) as evaluation metrics, which are widely used in unsupervised anomaly detection (Aggarwal & Sathe, 2017; Z. Li et al., 2022; X. Wang et al., 2020).

Suppose that the dimension (the number of features of the input) of the input is d, the structures of all autoencoders are $(d - \frac{d}{2} - \frac{d}{4} - \frac{d}{2} - d)$. The activation function is ReLU. The QAE and HAE are initialized with ReLinear (F.-L. Fan et al., 2023), whereas the AE is initialized with a random initialization. The dropout probability is 0.5. The number of epochs is 100. We use grid search for the learning rate and batch size.

4.5.2 Performance comparison

4.5.2.1 Classification results in tabular datasets

Table 4.5 and Table 4.6 display the AUCs and PREs of all models, where we have the following observations. First, by comparing the AUCs, in the majority of datasets, HAEs are the best or second best among all benchmark methods, where HAE-X has three best performances and six second best, HAE-Y has two best and two second best. Although DeepSVDD has three best results, its AUCs are lower on other datasets. Second, HAEs demonstrated the most consistent performance on average AUC and average rank, with all three HAE models leading.At last, the same trend is observed when considering the PREs, with HAEs consistently outperforming other baselines in terms of average rank. While DeepSVDD showed competitive results on the DAMI repository, its average PRE is 1.3% lower than the best-performing HAE-X model. Note that the HAEs present varying performances on different datasets, indicating that the design of a heterogeneous autoencoder is still an open question. As a summary, our anomaly detection experiments show that the employment of diverse neurons is beneficial, supporting the earlier-developed theory for heterogeneous networks.

Table 4.5: Comparison of AUCs for different anomaly detection baseline methods. Bold-faced numbers are the best and underlined numbers are the second best.

Repositories	Datasets	SUOD	SO-GAAL	DeepSVDD	DAGMM	RCA	AE	QAE	HAE-X	HAE-Y	HAE-I
OODs	Arrhythmia	0.770	0.568	0.704	0.603	0.806	0.807	0.807	**0.832**	0.816	0.817
	Glass	**0.775**	0.386	0.517	0.474	0.602	0.571	0.587	0.605	0.608	0.591
	Musk	0.568	0.438	0.502	0.903	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	Optdigits	0.469	0.367	0.494	0.290	0.622	0.652	0.599	0.772	**0.787**	0.668
	Pendigits	0.562	0.657	0.507	0.872	0.856	0.943	0.949	0.962	0.943	**0.967**
	Pima	0.624	0.292	0.485	0.531	0.711	0.657	0.652	**0.739**	0.695	0.701
	Verterbral	0.273	0.634	0.279	0.487	0.456	0.568	0.567	**0.574**	0.573	0.573
	Wbc	**0.950**	0.078	0.834	0.834	0.906	0.924	0.932	0.926	0.876	0.915
DAMI	ALOI	0.784	0.542	0.531	**1.000**	0.554	0.555	0.555	0.556	0.547	0.553
	Ionosphere	0.850	0.688	**0.979**	0.904	0.917	0.907	0.906	0.929	0.910	0.910
	KDDCUP99	0.953	0.889	0.914	**1.000**	0.989	0.989	0.986	0.989	0.989	0.989
	Shuttle	0.493	0.534	**0.892**	0.652	0.458	0.365	0.371	0.473	0.500	0.495
	Waveform	0.655	0.382	0.635	0.638	**0.704**	0.658	0.670	0.692	0.651	0.680
	WDBC	0.972	0.019	0.973	0.963	0.990	**0.997**	0.986	0.985	0.978	0.980
	WPBC	0.587	0.457	**0.703**	0.495	0.460	0.352	0.449	0.438	0.443	0.439
Avg. ↑		0.686	0.462	0.663	0.710	0.735	0.730	0.734	**0.765**	0.754	0.752
Avg. Rank ↓		6.250	8.438	7.250	6.438	4.875	5.500	5.125	**3.063**	4.313	3.750

Table 4.6: Comparison of PREs for different anomaly detection baseline methods. Bold-faced numbers are the best and underlined numbers are the second best.

Repositories	Datasets	SUOD	SO-GAAL	DeepSVDD	DAGMM	RCA	AE	QAE	HAE-X	HAE-Y	HAE-I
OODs	Arrhythmia	0.726	0.585	0.685	0.271	0.699	0.739	0.725	**0.758**	0.740	0.745
	Glass	0.593	0.516	0.573	0.497	0.596	0.572	0.576	0.582	0.589	**0.682**
	Musk	**0.794**	0.648	0.748	0.430	0.572	0.768	0.766	0.775	0.773	0.773
	Optdigits	0.448	0.447	0.473	0.513	**0.566**	0.466	0.447	0.514	0.515	0.481
	Pendigits	0.737	0.615	0.541	0.585	0.553	0.721	0.723	**0.757**	0.726	0.753
	Pima	0.638	0.382	0.511	0.567	0.541	0.591	0.636	**0.650**	0.629	0.639
	Verterbral	0.422	0.559	0.427	0.461	**0.656**	0.567	0.495	0.495	0.574	0.593
	Wbc	0.745	0.360	0.657	0.730	0.621	0.739	0.750	**0.829**	0.704	0.712
DAMI	ALOI	**0.576**	0.432	0.519	0.432	0.527	0.516	0.527	0.518	0.529	0.528
	Ionosphere	0.773	0.631	**0.915**	0.787	0.769	0.808	0.812	0.832	0.816	0.816
	KDDCUP99	0.589	0.490	0.793	0.489	0.515	0.726	0.490	0.870	**0.872**	0.871
	Shuttle	0.469	0.482	**0.835**	0.495	0.530	0.500	0.469	0.513	0.512	0.512
	Waveform	0.435	0.517	**0.655**	0.600	0.567	0.527	0.548	0.557	0.548	0.547
	WDBC	0.874	0.437	0.809	0.714	0.563	**0.993**	0.864	0.859	0.748	0.747
	WPBC	0.477	0.474	**0.674**	0.495	0.442	0.315	0.466	0.493	0.495	0.486
Avg. ↑		0.620	0.505	0.654	0.538	0.581	0.637	0.620	**0.667**	0.651	0.659
Avg. Rank↓		5.313	8.813	5.375	7.313	6.125	5.813	5.938	**3.000**	3.813	3.500

4.5.2.2 Classification results in bearing datasets

Table 4.7 presents the classification results, indicating that HAE-based methods outperform other competitors in both datasets. It should be noted that DAGMM is not included as a comparison method because its training fails. The SUOD model performs well in the SBP but achieves the limited performance in XJTU dataset, while the GAAL is the opposite. The RCA model demonstrates the best precision in the XJTU dataset and the best recall in the SBP dataset. However, it fails in some other metrics.

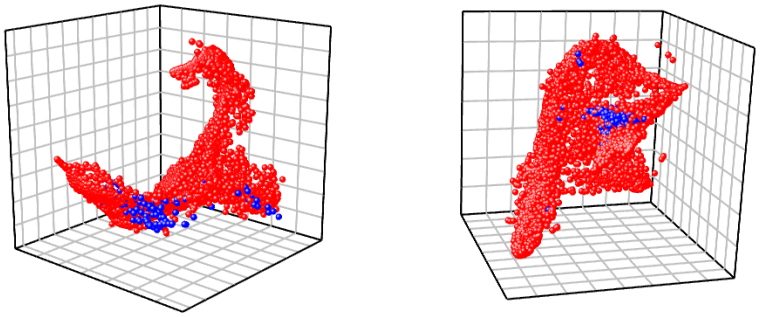
Autoencoder-based models demonstrated better results in both datasets, implying that autoencoder-based models are relatively suitable for handling higher dimensional data. In particular, HAE models are consistently better than AE and QAE, which shows that the combination of different neuron types is a wise strategy.

Table 4.7: Classification results of all baseline methods over two bearing fault datasets. Bold-faced numbers are better results.

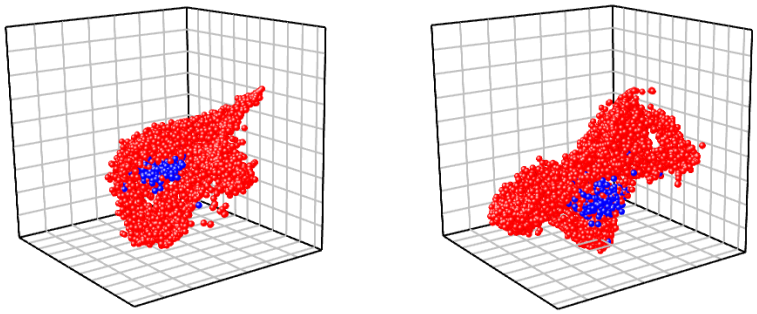
Datasets	Methods	AUC	Pre	Recall	F1
XJTU	SUOD	0.874	0.714	0.668	0.686
	SO-GAAL	0.958	0.896	0.724	0.777
	DeepSVDD	0.828	0.633	0.652	0.641
	RCA	0.946	**1.000**	0.528	0.691
	AE	0.942	0.755	0.712	0.731
	QAE	0.944	0.790	0.715	0.744
	HAE-X	0.957	0.956	**0.740**	**0.803**
	HAE-Y	0.958	0.957	0.737	0.800
	HAE-I	**0.961**	0.959	0.730	0.795
SBP	SUOD	**1.000**	0.875	0.985	0.921
	SO-GAAL	0.831	0.723	0.894	0.781
	DeepSVDD	**1.000**	0.700	0.967	0.769
	RCA	**1.000**	0.018	**1.000**	0.035
	AE	**1.000**	0.850	0.956	0.911
	QAE	**1.000**	0.873	0.963	0.904
	HAE-X	**1.000**	0.876	0.997	0.933
	HAE-Y	**1.000**	**0.883**	0.995	**0.935**
	HAE-I	**1.000**	0.881	0.990	0.932

4.5.3 Visualization

Because HAE variants show a better performance than a conventional one, we perform a learned embedding visualization to understand what these methods have learned. Here, with the optidigits dataset, we conduct t-SNE (G. E. Hinton, 2008) to visualize the latent space learned by AE and HAEs into 3D space. As shown in Figure 4.5, compared to the AE result, abnormal data show distinguishable and concentration clusters by HAEs, suggesting that a heterogeneous autoencoder better HAE better expresses the divergences between the different categories of samples.



(a) AE (b) HAE-X



(c) HAE-Y (d) HAE-I

Figure 4.5: The t-SNE results of four autoencoders on optidigits. The red and blue points are learned latent variables from abnormal and normal data, respectively. It is observed that the outliers are more concentrated in HAEs than AE.

4.6 Analysis experiments of HAEs

4.6.1 Neuron types

Earlier, we compared the performance between HAEs and purely conventional or quadratic autoencoders. However, the proposed HAEs use different architectures from AE and QAE to better synergize the power of conventional and quadratic neurons. Consequently, it is not sure if the superior performance of HAEs is due to the synergy of different neurons or due to the architecture. To resolve this ambiguity, we replace neurons in HAEs to prototype homogeneous autoencoder, as Table 4.8 shows. Then, we conduct experiments on 15 anomaly detection datasets to compare the difference in performance between HAEs and homogeneous autoencoders using the same architecture.

The results are presented in Tables 4.9 and 4.10. First, both AUC and precision metrics show that the performance of HAEs is superior to their homogeneous counterparts. Specifically, HAE-X achieves the highest AUC scores on 11 datasets, while HAE-Y performs best on 8 datasets and HAE-I outperforms its counterparts on 10 datasets. All quadratic neuron-based autoencoders are better than conventional ones, which underscores the powerful feature representation capabilities of quadratic

neurons. At last, we observe that HAE-I with quadratic neurons in the first layer outperforms those with conventional neurons (HAE-Ic), suggesting that quadratic neurons are more adept at extracting hidden features.

Table 4.8: The structure of autoencoders with different neurons. ”Q” represents quadratic neurons, ”C” represents conventional neurons, ”Yc” denotes the HAE-Y structure using a conventional decoder, and ”Ic” denotes the HAE-I structure using the conventional layers at first.

Structures	Encoder	Decoder
HAE-X	Q+C	Q+C
QAE-X	Q+Q	Q+Q
AE-X	C+C	C+C
HAE-Y	Q+C	Q
QAE-Y	Q+Q	Q
AE-Y	C+C	C
HAE-Yc	Q+C	C
HAE-I	Q→C	C→Q
HAE-Ic	C→Q	Q→C

Table 4.9: Comparison of AUCs for all structures. Bold-faced numbers are the best in an identical autoencoder structure with different neurons.

Datasets	HAE-X	QAE-X	AE-X	HAE-Y	QAE-Y	AE-Y	HAE-Yc	HAE-Iq	HAE-Ic
arrhythmia	**0.832**	0.819	0.815	0.816	**0.826**	0.813	0.816	**0.817**	0.814
glass	**0.605**	0.600	0.551	**0.608**	0.585	0.554	0.559	**0.591**	0.581
musk	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
optdigits	**0.772**	0.604	0.476	**0.787**	0.656	0.475	0.487	**0.668**	0.510
pendigits	**0.962**	0.945	0.931	0.943	**0.947**	0.934	0.935	**0.967**	0.937
pima	**0.739**	0.690	0.577	**0.695**	0.606	0.582	0.581	**0.701**	0.581
verterbral	**0.574**	0.481	0.571	**0.573**	0.561	0.570	0.571	**0.573**	0.572
wbc	**0.926**	0.909	0.857	**0.876**	0.868	0.857	0.856	**0.915**	0.855
ALOI	**0.556**	0.553	0.546	**0.547**	0.554	0.546	**0.547**	**0.553**	0.547
Ionosphere	**0.929**	0.920	0.906	0.910	**0.919**	0.905	0.907	**0.910**	0.908
KDDCUP99	0.989	**0.994**	0.989	0.989	0.989	0.989	0.989	0.989	0.989
Shuttle	0.473	0.385	**0.488**	**0.500**	0.453	0.491	0.491	0.495	0.495
Waveform	**0.692**	0.681	0.643	0.651	**0.688**	0.655	0.647	**0.680**	0.657
WDBC	**0.985**	0.981	0.978	0.978	**0.980**	0.979	**0.980**	0.980	**0.981**
WPBC	0.438	0.440	**0.445**	**0.443**	0.439	0.442	0.442	0.439	**0.442**
Avg.	**0.765**	0.733	0.718	**0.754**	0.738	0.719	0.721	**0.752**	0.725

Table 4.10: Comparison of PREs for all structures. Bold-faced numbers are the best in an identical autoencoder structure with different neurons.

Datasets	HAE-X	QAE-X	AE-X	HAE-Y	QAE-Y	AE-Y	HAE_Yc	HAE_Iq	HAE_Ic
arrhythmia	**0.758**	0.742	0.729	**0.740**	0.735	0.724	0.736	**0.745**	0.734
glass	**0.582**	0.576	0.556	0.589	**0.636**	0.574	0.574	**0.682**	0.572
musk	0.775	**0.776**	0.770	0.773	**0.917**	0.771	0.773	**0.773**	0.770
optdigits	**0.514**	0.443	0.431	**0.515**	0.433	0.432	0.431	**0.481**	0.431
pendigits	**0.757**	0.717	0.712	0.726	**0.818**	0.711	0.715	**0.753**	0.715
pima	**0.650**	0.626	0.557	**0.629**	0.573	0.555	0.555	**0.639**	0.553
verterbral	0.495	0.504	**0.582**	0.574	0.572	0.586	**0.589**	**0.593**	0.589
wbc	**0.829**	0.723	0.699	**0.704**	0.695	0.700	0.703	**0.712**	0.700
ALOI	**0.518**	**0.518**	0.515	**0.529**	0.517	0.516	0.516	**0.528**	0.516
Ionosphere	**0.832**	0.827	0.812	**0.816**	0.815	0.806	0.806	**0.816**	0.807
KDDCUP99	**0.870**	0.713	0.724	**0.872**	0.727	0.729	**0.872**	**0.871**	0.726
Shuttle	0.513	0.509	**0.518**	0.512	0.511	**0.519**	0.469	0.512	**0.522**
Waveform	**0.557**	0.546	0.537	0.548	**0.558**	0.541	0.544	**0.547**	0.537
WDBC	**0.859**	0.747	0.747	0.748	0.749	0.746	**0.855**	0.747	0.747
WPBC	0.493	0.487	**0.494**	**0.495**	0.490	0.484	0.487	0.486	**0.488**
Avg.	**0.667**	0.630	0.626	**0.651**	0.650	0.626	0.642	**0.659**	0.627

4.6.2 Formula of quadratic neurons

Quadratic neurons have different variants, such as only keeping the power terms and replacing the interaction term with a linear term. What we use in a quadratic neuron is the standard form of the quadratic function. But is it suitable for anomaly detection? Here, we investigate if the standard form of quadratic neurons fits. We have chosen several datasets for comparison, with the results for AUC and precision presented in Tables 4.11 and 4.12, respectively. We highlight that models using standard quadratic neurons display superior performance in the majority of cases. This outcome strongly suggests that the standard quadratic neuron, with its inclusion of both power and inner-product terms, possesses the best representation capabilities. Additionally, it is worth noting that models utilizing the power term generally perform better than those omitting it. This observation highlights the important role of the power term within the quadratic neuron.

Table 4.11: Comparison of AUC for different quadratic functions on some datasets. Bold-faced numbers are the best compared in every structure.

Methods	Quadratic Function	Arrhythmia	Glass	Optdigits	ALOI	Shuffle	Waveform	XJTU	Avg.
HAE-X	$(\boldsymbol{x}^\top \boldsymbol{w}^r + b^r)(\boldsymbol{x}^\top \boldsymbol{w}^g + b^g)$	0.817	0.591	0.627	0.552	0.471	0.682	0.942	0.669
	$\boldsymbol{x}^\top \boldsymbol{w}^r + (\boldsymbol{x} \odot \boldsymbol{x})^\top \boldsymbol{w}^b + c$	0.830	0.580	0.676	0.554	0.471	0.687	0.945	0.678
	Standard	**0.832**	**0.605**	**0.772**	**0.556**	**0.473**	**0.692**	**0.957**	**0.698**
HAE-Y	$(\boldsymbol{x}^\top \boldsymbol{w}^r + b^r)(\boldsymbol{x}^\top \boldsymbol{w}^g + b^g)$	0.820	0.578	0.659	0.553	0.453	0.689	0.942	0.671
	$\boldsymbol{x}^\top \boldsymbol{w}^r + (\boldsymbol{x} \odot \boldsymbol{x})^\top \boldsymbol{w}^b + c$	**0.829**	0.571	0.661	**0.554**	0.452	**0.695**	0.943	0.672
	Standard	0.816	**0.608**	**0.787**	0.547	**0.500**	0.651	**0.958**	**0.695**
HAE-I	$(\boldsymbol{x}^\top \boldsymbol{w}^r + b^r)(\boldsymbol{x}^\top \boldsymbol{w}^g + b^g)$	**0.818**	0.567	0.627	0.551	0.450	0.671	0.944	0.661
	$\boldsymbol{x}^\top \boldsymbol{w}^r + (\boldsymbol{x} \odot \boldsymbol{x})^\top \boldsymbol{w}^b + c$	0.815	0.568	0.628	**0.553**	0.450	0.670	**0.966**	0.664
	Standard	0.817	**0.591**	**0.668**	**0.553**	**0.495**	**0.680**	0.961	**0.681**

Table 4.12: Comparison of precision for different quadratic functions on some datasets. Bold-faced numbers are the best compared in every structure.

Methods	Quadratic Function	Arrhythmia	Glass	Optdigits	ALOI	Shuffle	Waveform	XJTU	Avg.
HAE-X	$(\boldsymbol{x}^\top \boldsymbol{w}^r + b^r)(\boldsymbol{x}^\top \boldsymbol{w}^g + b^g)$	0.716	**0.634**	0.457	0.516	0.517	0.548	0.747	0.591
	$\boldsymbol{x}^\top \boldsymbol{w}^r + (\boldsymbol{x} \odot \boldsymbol{x})^\top \boldsymbol{w}^b + c$	0.742	0.568	0.481	0.515	**0.523**	0.554	0.818	0.600
	Origin	**0.758**	0.582	**0.514**	**0.581**	0.513	**0.557**	**0.956**	**0.637**
HAE-Y	$(\boldsymbol{x}^\top \boldsymbol{w}^r + b^r)(\boldsymbol{x}^\top \boldsymbol{w}^g + b^g)$	**0.753**	0.572	0.460	0.516	**0.522**	0.550	0.746	0.588
	$\boldsymbol{x}^\top \boldsymbol{w}^r + (\boldsymbol{x} \odot \boldsymbol{x})^\top \boldsymbol{w}^b + c$	0.744	0.572	0.475	0.515	0.520	**0.557**	0.852	0.605
	Origin	0.740	**0.589**	**0.515**	**0.529**	0.512	0.548	**0.957**	**0.627**
HAE-I	$(\boldsymbol{x}^\top \boldsymbol{w}^r + b^r)(\boldsymbol{x}^\top \boldsymbol{w}^g + b^g)$	0.741	0.573	0.455	0.514	**0.551**	0.539	0.728	0.586
	$\boldsymbol{x}^\top \boldsymbol{w}^r + (\boldsymbol{x} \odot \boldsymbol{x})^\top \boldsymbol{w}^b + c$	0.742	0.573	0.453	0.514	0.511	0.541	0.923	0.608
	Origin	**0.745**	**0.682**	**0.481**	**0.528**	0.512	**0.547**	**0.959**	**0.636**

4.6.3 Network depth and width

(1) As shown in Table 4.13, we mark different structural parameters of hidden layers as V1 to V4. Where V1 represents hidden neurons as [*dim(x)*/2-*dim(x)*/4], V2 is [*dim(x)*/2-*dim(x)*/3-*dim(x)*/4], V3 is [*dim(x)*/2-*dim(x)*/3-*dim(x)*/4-*dim(x)*/4], and V4 is [*dim(x)*/2-*dim(x)*/3-*dim(x)*/3-*dim(x)*/4-*dim(x)*/4]. We investigate the performance of HAEs in response to different depths on the optidigits dataset. Note that increasing network layers will not tend to a better performance when the scale of the neural network is large enough. It may lead to over-fitting. Therefore, all autoencoders are based on V1 ([*dim(x)*/2-*dim(x)*/4]) structure in our follows experiments.

(2) How to find the best width arrangement for autoencoders is still an open question. If the width is large, the compressing effect is compromised, which causes the encoder to fail to learn the most representative features. If the width is small, the learned features are insufficient to completely represent the original data, which makes the detection based on these features inaccurate. We conduct an experiment to evaluate the validity of different width arrangements. The results are presented

in Table 4.14. While our recommended structure exhibits superior performance in the majority of cases, we also note that the results are quite similar. We believe that, in the context of anomaly detection where large-scale datasets are often unavailable, the number of neurons employed does not significantly impact results.

Table 4.13: Comparison of AUC for different network structures on optidigits. Bold-faced numbers are the best results compared in every column.

	AE	QAE	HAE-X	HAE-Y	HAE-I
V1	**0.652±0.017**	**0.599±0.023**	**0.680±0.054**	**0.681±0.056**	0.617±0.031
V2	0.613±0.010	0.528±0.045	0.669±0.043	0.653±0.027	**0.636±0.023**
V3	0.582±0.028	0.447±0.054	0.671±0.036	0.666±0.068	0.625±0.026
V4	0.572±0.008	0.316±0.052	0.647±0.029	0.653±0.040	0.613±0.041

Table 4.14: Comparison of HAEs using different width arrangements in several datasets. Bold-faced numbers are better.

Dataset		Arrhythmia		Shuffle		XJTU	
Method	Structure	AUC	PRE	AUC	PRE	AUC	PRE
HAE-X	d-d/2-d/4	**0.832**	**0.758**	**0.473**	0.513	**0.957**	**0.956**
	d-d/4-d/8	0.820	0.744	0.435	0.511	0.946	0.925
	d-d-d/2	0.828	0.741	0.452	**0.520**	0.945	0.913
HAE-Y	d-d/2-d/4	0.816	0.740	**0.500**	**0.512**	**0.958**	**0.957**
	d-d/4-d/8	0.816	0.732	0.449	0.511	0.948	0.935
	d-d-d/2	**0.827**	**0.741**	0.421	0.509	0.945	0.908
HAE-I	d-d/2-d/4	**0.817**	**0.745**	**0.495**	**0.512**	**0.961**	**0.959**
	d-d/4-d/8	**0.817**	0.738	0.454	0.511	0.962	0.937
	d-d-d/2	0.814	0.737	0.445	0.508	0.958	0.947

4.7 Summary

In this chapter, we have theoretically shown that a heterogeneous network is more powerful and efficient by constructing a function that is easy to approximate by a one-hidden-layer heterogeneous network but difficult to approximate by a one-hidden-layer conventional or quadratic network. Moreover, we have proposed a novel heterogeneous autoencoder using quadratic and conventional neurons. Lastly, anomaly detection experiments have confirmed that a heterogeneous autoencoder delivers competitive performance relative to homogeneous autoencoders and other baselines. However, the interpretability of heterogeneous networks remains unverified. Future work will conduct a layer-

wise interpretability analysis via frequency spectrum transformation methods to elucidate the feature extraction capability in bearing fault detection tasks.

Chapter 5

Logarithmic Cumulative Transformation: A Simple Yet Effective Approach for Bearing Remaining Useful Life Prediction

5.1 Introduction

BEARINGS, as integral components in rotating machinery, are indispensable in a wide range of production and manufacturing activities (Shao et al., 2023; Sun et al., 2019). Their continuous exposure to high loads, variable speeds, and elevated temperatures, however, renders them susceptible to failures, leading to substantial economic losses in critical sectors (Y. Cao et al., 2023). Consequently, there is an urgent need to develop prognostics and health management (PHM) methods to facilitate smart manufacturing (Lei et al., 2020; Ni, Ji, & Feng, 2023). Generally speaking, PHM consists of three primary tasks: fault detection, fault diagnosis, and remaining useful life (RUL) estimation (M. Ma & Mao, 2020). Among them, RUL prediction that estimates the remaining time until the end of the useful life is particularly crucial (Nemani et al., 2023). Accurate RUL estimation helps to streamline the maintenance operations, minimize the downtime of the machinery and offers a reliable solution to devise cost-effective production plan. However, due to limited historical data and frequent variations in the working conditions, bearing RUL prediction is a challenging task, and timely and accurate RUL prediction is listed as the most formidable task in PHM (X. Li, Zhang, & Ding, 2019; W. Peng, Ye, & Chen, 2019; Zhu et al., 2022).

Currently, RUL prediction methods can be roughly grouped into three categories: physics model-

based, statistical model-based, and data-driven methods (J. Chen et al., 2023). The first two schemes characterize the bearing degradation and failure mechanisms with physical models (Paris model (Paris, 1963)) or statistical processes (Weiner process (W. Cheng et al., 2024)). Despite the sound theoretical foundations, they face several fundamental shortcomings, including the specification of data distributions, domain knowledge for choosing the right statistical model, and poor generalization to varying working conditions. With the emergence of artificial intelligence (AI) over the past decades, deep learning have been extensively leveraged for RUL prediction. Essentially, deep learning-based methods construct end-to-end RUL models without the need of prior knowledge, but they heavily depend on a large volume of high-quality data to train the model (Ren et al., 2023). Some deep learning models, like temporal convolutional network (TCN) (Bai, Kolter, & Koltun, 2018), long short-term memory (LSTM) (C.-G. Huang, Huang, & Li, 2019), and gated recurrent unit (GRU) (Bahdanau, Cho, & Bengio, 2014; Y. Chen et al., 2020), are dedicated to handling time-series data. However, the diverse failure modes with each exhibiting some unique characteristics and the strong noise inherent in the vibration signals can easily dilute the trend associated with bearing degradation. These factors together renders deep learning-based methods ineffective for predicting bearing RUL in practice.

To overcome these problems, several feature extraction methods have been developed to reveal discerning degradation information concealed within the raw vibration data. Generally speaking, vibration signals are converted to their time, frequency, and time-frequency statistical features for building prediction models (Javed et al., 2015). For instance, M. Yan et al. (2020) proposed two novel time-domain statistical features – relative root mean square (RRMS) and inertial relative root mean square (IRRMS) – to amplify the degradation trends and minimize random fluctuations. In a similar vein, T. Yan et al. (2022) proposed a composite health index that fused spectrum amplitudes to provide a monotonically degradation trend. Javed et al. (2015) proposed a feature extraction method using discrete wavelet transform (DWT), followed by trigonometric functions and cumulative transformation. L. Liu et al. (2021) validated the effectiveness of Javed's method by incorporating an attention-based encoder-decoder framework for RUL prediction. Numerous experiments demonstrate that extracted features with superior trendability and monotonicity can effectively improve the RUL prediction performance (K. He et al., 2022; Javed et al., 2015; Lei et al., 2018).

However, existing feature extraction methods primarily concentrate on enhancing the trendability and monotonicity associated with derived features, but they overlook an important and pervasive is-

sue: feature misalignment. Specifically, identical types of bearings may exhibit different degradation trends even under the same operating conditions. Despite that the extracted features possess desirable monotonicity and trendability, they remain misaligned in scale and shape, particularly between training and test sets. Such misalignment has a profound impact on the generalizability of downstream model development, especially when the training samples are limited (X. Li et al., 2023; Qin et al., 2023). In this chapter, we ascertain that the misalignment primarily stems from the diverse shape and scale of the extracted features. Based on this observation, we are motivated to devise a feature extraction approach to standardize the shape and scale of different bearing features without leaking the information. By mitigating the misalignment of extracted features between training data and testing data, we thus enhance the generalizability of the constructed deep learning models.

Drawing inspiration from the cumulative function for calculating the cumulative sum of statistical features to enhance their trendability and monotonicity (Javed et al., 2015; L. Liu et al., 2021), we develop a novel approach coined as Logarithmic Cumulative Transformation (LCT). The overarching goal of LCT is to align the shape and scale of extracted features across disparate data. The LCT operator has three steps: an initial cumulative transformation, a subsequent logarithmic transformation, and a final cumulative transformation. Furthermore, LCT is engineered to convert raw vibration signal into an approximately straight line. Considering that the bearing RUL value is linearly correlated with time, the LCT operator makes it quite useful for establishing the linear health indicator (HI). The LCT operation thus provides a new perspective on improving the generalizability of RUL models and can be readily integrated into other data-driven models as a signal preprocessing step.

In addition, ensuring the reliability of RUL predictions is an important consideration for making informed maintenance decisions in real-world scenarios (T. Zhou, Han, & Droguett, 2022). Some methods in the literature have attempted to quantify the uncertainty of RUL models using Monte Carlo (MC) dropout (L. Cao et al., 2023) or Wiener process (W. Cheng et al., 2024), but they incur additional computational costs due to multiple runs of the model or extra parameter estimation. In this study, we propose a straightforward method to estimate the reliability associated with each RUL prediction. First, we fit a linear regression to the point predictions of deep learning model to reduce fluctuations in the case of no historical data and inject constraints to ensure the validity of RUL estimations by the fitted linear regression model. Next, we fit auxiliary exponential models to the differential slopes of the linear regression models on the training sets to estimate their upper and lower bounds. Finally, we

design a reliability metric based on the estimated upper and lower bounds to measure the reliability on the test sets. This metric offers a fast and effective way to quantify the reliability pertaining to each RUL prediction, thus improving the model's trustworthiness.

Compared to the state-of-the-art literature, our contributions in this chapter are summarized as follows:

1. We develop a novel feature extraction approach termed the Logarithmic Cumulative Transformation (LCT). The LCT operator effectively fixes feature misalignment in the bearing RUL prediction by aligning the shape and scale of extracted features across different bearings. The outcome of LCT operator leads to RUL models with enhanced generalizability.

2. We propose a novel method to estimate the reliability associated with each RUL prediction. By constructing auxiliary exponential models and a reliability metric, this method provides a fast and accurate means to estimate the reliability associated with each RUL estimation in the online setting.

3. We establish an RUL prediction framework that comprises the LCT, an attention-based GRU encoder-decoder, and the reliability estimation method. Computational results on the FEMTO-ST dataset demonstrate the superiority of our approach over other state-of-the-art methods.

5.2 Methodology

As shown in Figure 5.1, the proposed framework follows a three-step procedure: i) Extract time statistical features from the raw vibration signals using the Root Mean Square (RMS), and apply the logarithmic cumulative transformation to convert features into shape and scale-aligned features; ii) Train an attention-based encoder-decoder Gated Recurrent Unit (GRU) backbone using "teacher forcing" strategy [1] for point RUL predictions; iii) Calibrate model prediction and then evaluate model's prediction reliability using an auxiliary exponential model.

[1] https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html

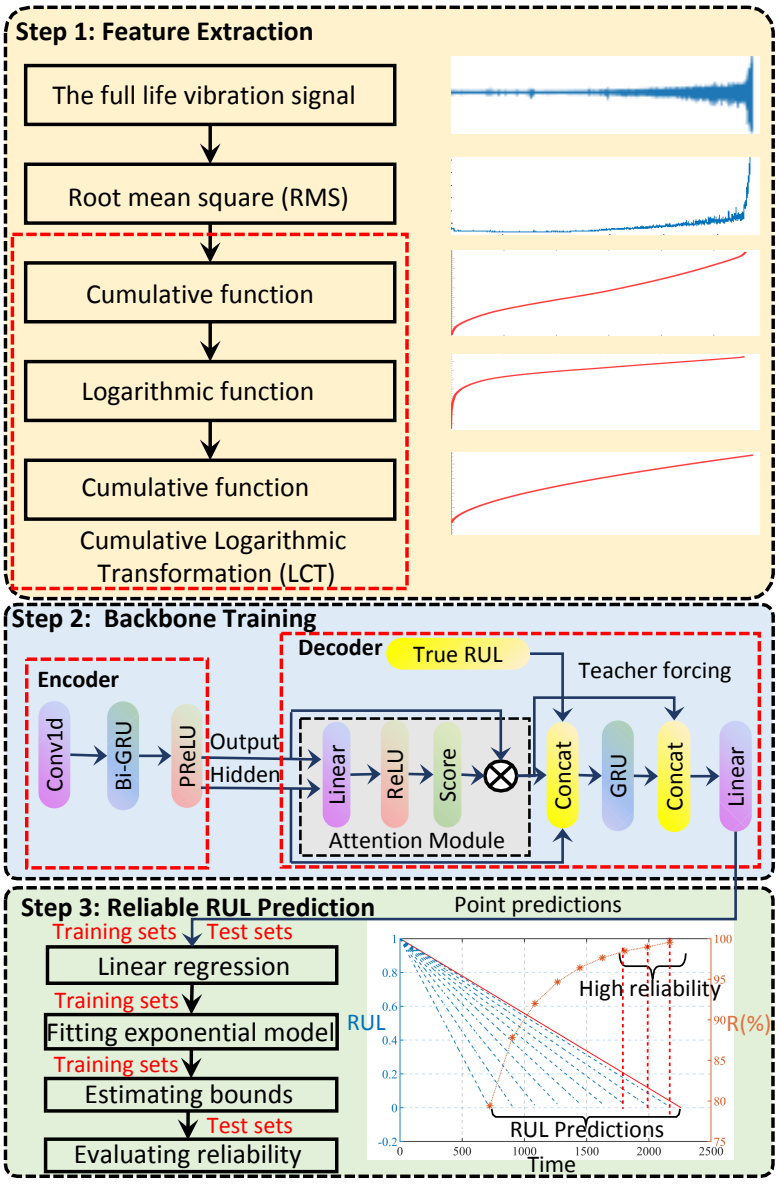
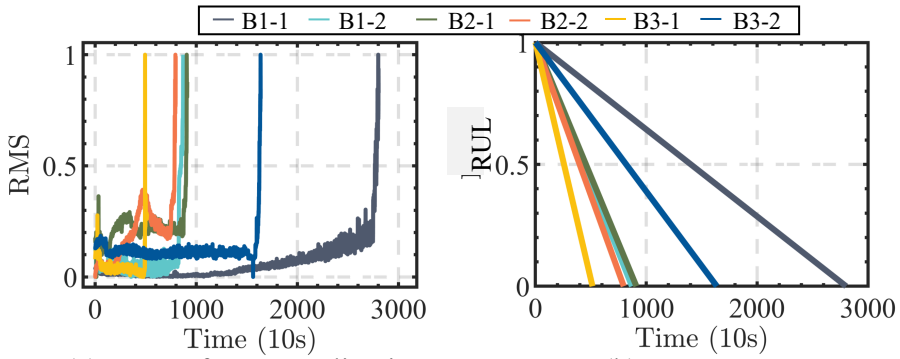


Figure 5.1: Flowchart of the proposed framework.

5.2.1 Feature misalignment in RUL prediction

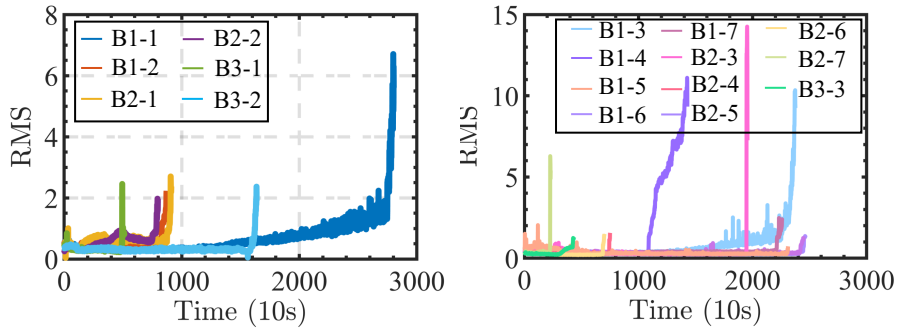
The high variability of failures among bearings poses a significant challenge for the construction of data-driven models (H. Cheng et al., 2022). Even if preprocessing methods are applied during feature extraction, feature misalignment still pervasively exists between the training and test data. As illustrated in Figure 5.2, the RMS values of various bearings exhibit substantial differences in shape and scale. In the case that the amount of training data is scarce, the difference in scale and shape severely compromises the generalizability of the developed RUL model.



(a) RMS after normalization (b) True RUL

Figure 5.2: Root mean square (RMS) of samples in the training and test sets of the FEMTO-ST dataset.

Traditionally, normalization has been proven to be an effective solution, as it lowers the variations in scale. Empirical studies have substantiated that the model can achieve a diminished error in RUL predictions by aligning the shape (L. Cao et al., 2023). The enhanced performance can be attributed to the scale alignment of the feature with the degradation trend. As shown in Figure 5.3(a), the RMS values of all the bearings are mapped within the range $[0, 1]$. Likewise, the corresponding normalized true RULs also vary within the same range. The normalization and scaling operation reduce the difficulty for the machine learning model to map the RMS value to the corresponding RUL value.



(a) RMS of the training set (b) RMS of the test set

Figure 5.3: RMS and the corresponding RUL indicator on the FEMTO-ST training set.

However, such a pragmatic method is infeasible in the online learning setting due to information leakage. Specifically, normalization necessitates the full life cycle signals to determine the extremums and converts the features to the same scale. Nonetheless, in the online learning setting, only the current and historical data are accessible, thereby precluding the acquisition of the global extremums. Based on these observations, we posit that the crux of enhancing model generalizability lies in how to align the shape and scale of extracted features without information leakage.

Remark 1. There are three primary types of RUL indicators: piece-wise linear, exponential, and

linear (D. Chen et al., 2021; W. Cheng et al., 2024; G. Jiang et al., 2022; Kumaraswamidhas, Laha, et al., 2021). In this study, we employ the linear one as our health indicator based on the assumption that bearing failures accumulate progressively over time. Such an approach offers several advantages, including eliminating the need for determining First Prediction Time (FPT) and facilitating the construction of end-to-end prediction models.

Conversely, the piece-wise linear indicator necessitates FPT determination. We elaborate on the limitations of FPT by employing the classical 3σ principle (X. Li, Zhang, & Ding, 2019). As illustrated in Figure 5.4, two bearings of the same type, tested under identical conditions, could still exhibit divergent degradation patterns. Bearing 1-1 follows a standard degradation trend while Bearing 1-2 shows an abrupt degradation. This suggests that even identical types of bearings operating under the same conditions can result in different FPT intervals, thus leading to prediction errors in downstream tasks. Moreover, the lack of consensus on the appropriate features and methods for calculating FPT impedes the objective evaluation of RUL predictors.

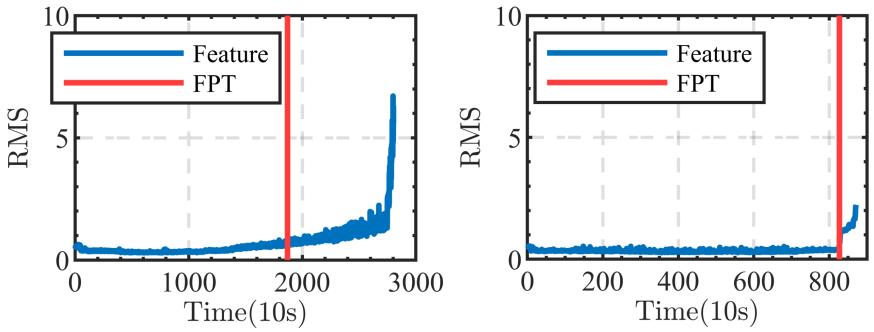


Figure 5.4: The first prediction time (FPT) of two full life cycle bearings on the FEMTO-ST dataset. Where these two bearings are tested under the same condition.

While we do not completely disregard the natural exponential trend in bearing degradation, we exploit this principle to construct the reliability metric subsequently.

5.2.2 Logarithmic cumulative transformation

A multitude of signal processing techniques, such as time-frequency transformation and statistical metrics, have been adopted for extracting feature useful for modeling the degradation behavior. These methods aims to extract features with augmented trendability and monotonicity and facilitate to align the extracted features with the degradation of bearings (Javed et al., 2015; L. Liu et al., 2021). However, to the best of our knowledge, methods are still lacking in the literature to mitigate the feature

misalignment across different bearings. Motivated by this finding, we develop a novel feature extraction approach Logarithmic Cumulative Transformation (LCT). The LCT operator not only guarantees trendability and monotonicity, but also induces feature alignment in both scale and shape.

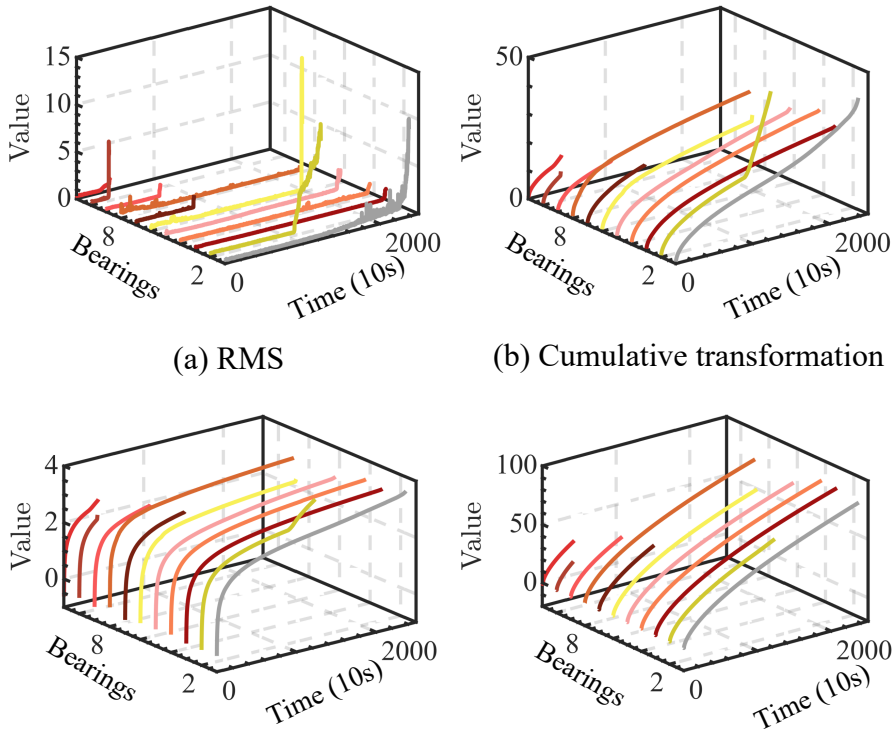
In the LCT operator, the time domain statistical feature of vibration signals is extracted with the widely used RMS (W. Cheng et al., 2024),

$$\text{RMS}\left(\boldsymbol{S}_t\right) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} s_i^2}, \tag{5.1}$$

where $\boldsymbol{S}_t = [s_1, s_2, \cdots, s_n]_t$ indicates the raw vibration signal and n is the length of the signals at the t-th moment. In other words, each moment has a RMS value denoted as:

$$f_t = \text{RMS}(\boldsymbol{S}_t) \quad t = 1, 2, \cdots, N, \tag{5.2}$$

where N indicates the length of the feature over the entire life cycle. For the sake of illustration, Figure 5.5(a) shows the RMS feature values of different bearings.



(a) RMS (b) Cumulative transformation

(c) Logarithmic transformation (d) Cumulative transformation

Figure 5.5: Feature transformation of FEMTO-ST test set by the LCT operator.

Next, the derived LCT feature is segmented into three stages. A cumulative function is first em-

ployed to convert the RMS feature into its cumulative form (Javed et al., 2015). Mathematically, we have:

$$C_t = \frac{\sum_{i=1}^{t} f_i}{\sqrt{|\sum_{i=1}^{t} f_i|}}, \quad t = 1, 2, \cdots, N, \tag{5.3}$$

where the numerator is the total sum of RMS values up to N-th point, while the denominator serves to constrain the upward trend of the cumulative values. Assuming that the trend of bearing failure is progressive, the cumulative function effectively mitigates the fluctuating and nonlinear characteristics of the RMS values. Such a transformation serves to enhance the trendability and monotonicity of the extracted features, thereby making them more amenable for the subsequent HI construction (M. Yan et al., 2020). However, the cumulative function does not completely fix the misalignment issue. While the transformed feature exhibits a smooth shape, as depicted in Figure 5.5(b), a significant disparity in scale arises.

As the logarithmic function is a convenient way for narrowing the scale difference, we propose to adopt the logarithmic transformation as follows:

$$LC_t = \log(C_t), \quad t = 1, 2, \cdots, N. \tag{5.4}$$

As illustrated in Figure 5.5(c), the logarithmic transformation weakens some sharply rising feature values. After the transformation, majority of the features display a good alignment in both scale and shape. However, the feature curves tend to follow a logarithmic trend. The nonlinear changing trend in the feature values makes it difficult to map the feature values into the corresponding RUL indicator.

As a result, an additional cumulative function is performed to create an approximate linear trend with an enhanced shape and scale alignment.

$$CLC_t = C_t(LC_t) = \frac{\sum_{i=1}^{t} LC_i}{\sqrt{|\sum_{i=1}^{t} LC_i|}}, \quad t = 1, 2, \cdots, N. \tag{5.5}$$

Figure 5.5(d) displays the extracted feature after all the transformations. Compared to the original RMS features, LCT features not only exhibit superior monotonicity and trendability, but also achieve a good alignment in shape and scale across diverse bearing sets without information leakage. Such characteristics significantly benefit the downstream tasks to achieve accurate RUL estimations in a variety of scenarios.

Remark 2. To demonstrate the effectiveness of LCT in mitigating the scale misalignment, we compare the extracted features before and after applying LCT on the training and test sets. As shown in Figure 5.6, the LCT features in both datasets exhibit a comparable scale, whereas the raw RMS features have a large difference in their scale. Specifically, post-LCT, the training set has a maximum value of 92 while the test set has a value of 84, which highlights a slight difference of approximately 10%. However, with respect to the raw RMS features, the relative difference exceeds 100% (13.98 vs 6.7).

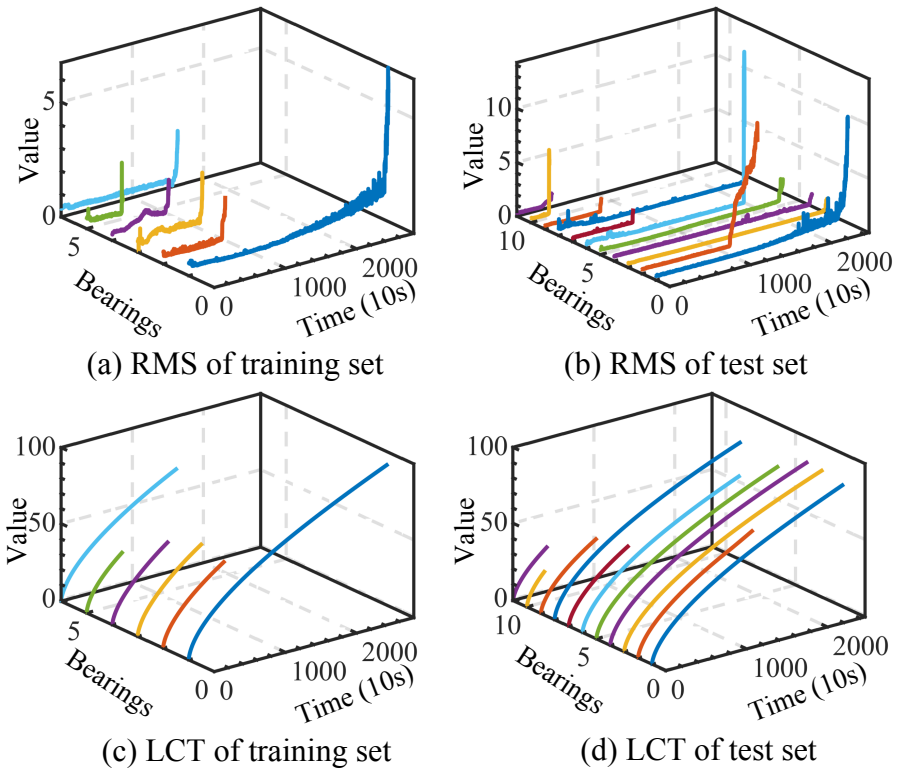


Figure 5.6: Feature comparisons before and after the LCT transformation on the FEMTO-ST dataset.

5.2.3 Attention GRU-based encoder-decoder

In this chapter, a commonly used attention GRU encoder-decoder is adopted as the backbone of the neural network (Bahdanau, Cho, & Bengio, 2014; Y. Chen et al., 2020). We utilize the attention-based GRU encoder-decoder framework as outlined in Ref. (Y. Chen et al., 2020). The encoder comprises a single 1D convolutional layer and a bidirectional GRU (Bi-GRU) layer. In contrast, the decoder consists of an attention module composed of a linear layer followed by a ReLU activation function, a GRU layer, and a final linear layer that maps the hidden state to a single-point prediction. The specification of this structure is given in Table 5.1. Moreover, the key components of the backbone

network are described as follows.

Table 5.1: Structural parameters of attention GRU-based encoder-decoder.

	Layer	Kernel (Neuron) size	Channel	Stride	Hidden size (GRU)
Encoder	Conv1D	8X1	64	5	-
	PReLU	-	-	-	-
	Bi-GRU	-	64	-	200
Decoder	Linear (Attention)	(400, 200)	-	-	-
	ReLU (Attention)	-	-	-	-
	GRU	-	201	-	200
	Linear	(400,1)	-	-	-

5.2.3.1 Gated recurrent unit (GRU)

GRU is a specific variant of RNN designed to address the long-term memory forgetting problem inherent in conventional RNNs (Cho et al., 2014). GRU can be conceptualized as a streamlined version of LSTM as it discards the separate memory cell. Mathematically, GRU is formulated as:

$$\begin{aligned} r_t &= \sigma(W_r x_t + b_r + W_{hr} h_{(t-1)} + b_{hr}) \\ z_t &= \sigma(W_z x_t + b_z + W_{hz} h_{(t-1)} + b_{hz}) \\ n_t &= \tanh(W_n x_t + b_n + r_t \odot (W_{hn} h_{(t-1)} + b_{hn})) \\ h_t &= (1 - z_t) \odot n_t + z_t \odot h_{(t-1)}, \end{aligned} \tag{5.6}$$

where $\odot$ denotes the Hadamard product, W, b represent the weight and bias parameters, $\sigma(\cdot)$ denotes the activation function, and h_t is the hidden output at the last time step t.

A Bi-GRU incorporates both forward and backward GRUs. The forward hidden states are represented as $[\overrightarrow{h_1}, \cdots, \overrightarrow{h_t}]$, while the backward hidden states are denoted as $[\overleftarrow{h_t}, \cdots, \overleftarrow{h_1}]$. Consequently, the output of Bi-GRU is given by:

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_1}]. \tag{5.7}$$

5.2.3.2 Attention module

Given that the conventional GRU encoder only carries the information contained in the last hidden state, but the input information of the previous moments is still discarded. The attention module is introduced in the decoder to utilize the previous hidden encoded information (Luong, Pham, &

Manning, 2015). Each previous hidden output is assigned a weight α and the attention output is

$$c_t = \sum_{i=1}^{t} \alpha_{it} h_i, \tag{5.8}$$

where

$$\alpha_{it} = \frac{\exp(e_{it})}{\sum_{k=1}^{t} \exp(e_{kt})} \tag{5.9}$$

and the score model e is used to evaluate the match performance of the GRU input s at $i-1$ and the output at h_t

$$e_{it} = \boldsymbol{v}^\top \sigma(\boldsymbol{W}[s_{i-1}; h_t] + b), \tag{5.10}$$

with $\boldsymbol{v}, \boldsymbol{W}$ and b being weight parameters.

In summary, given the post-LCT feature at time t as the input x_t, an encoder-decoder GRU generates a point prediction as:

$$\hat{y_t} = D(E(x_t, h_{t-1}), \boldsymbol{c}), \tag{5.11}$$

where $E(\cdot)$ and $D(\cdot)$ are the encoder and decoder networks, respectively; $h_{t-1} = E(x_t, h_{t-2})$ denotes the hidden state of GRU at time $t-1$, with the initial state $h_0 = 0$; $\boldsymbol{c} = [c_1, c_2, \cdots, c_t]$ represents the context vector that incorporates previous information through attention module. The network recursively uses current feature x_t and historical data h_{t-1} to generate the prediction $\hat{y_t}$.

Currently, there are many deep learning methods for RUL prediction, i.e., temporal convolutional network (TCN) (Bai, Kolter, & Koltun, 2018), multiscale CNN (X. Li, Zhang, & Ding, 2019), long short-term memory (LSTM) (W. Li & Deng, 2023), etc. Compared to these methods, the advantage of encoder-decoder GRU network is that it can keep the long-distance input for predicting the current point, improving the accuracy of the prediction.

5.2.4 Reliability estimation of RUL prediction

The encoder-decoder network generates a point estimation as the RUL prediction, however, the reliability associated with each point RUL estimation during online learning setting remains unknown. To address this problem, we develop a straightforward method to estimate the reliability associated with each RUL prediction.

5.2.4.1 Linear regression

We employ a linear regression model due to the adoption of the linear health indicator. This makes the RUL point predictions from the deep learning backbone have lower errors (Y. Chen et al., 2020; L. Liu et al., 2021; M. Zhao, Tang, & Tan, 2016). At $t = 0$, as there is no historical data available, the model prediction at $t = 0$ totally depends on the initialization of neural network. As parameters of neural network are often initialized randomly, the model's prediction at $t = 0$ is random in nature because no information is available to guide the model. This randomness affects the model predictions over the subsequent time steps. As depicted in Figure 5.7, the bias in the RUL predictions at the early stage gets translated into discrepancy in the RUL estimations over the remaining prediction horizon.

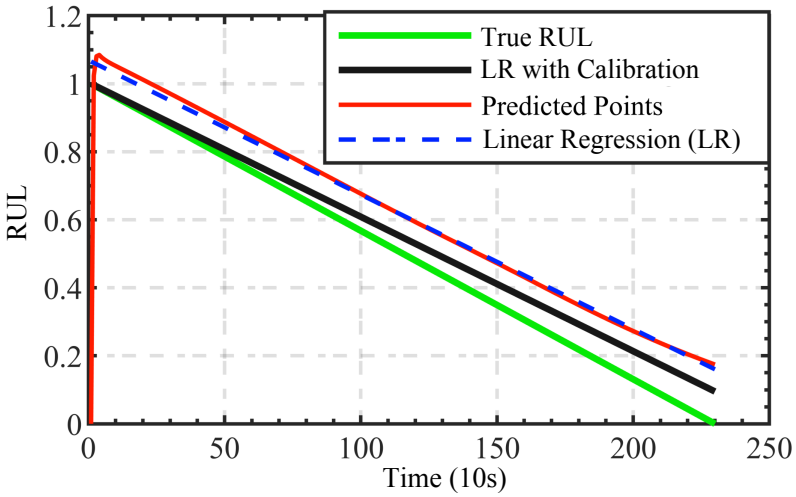


Figure 5.7: Fitted linear regression model for Bearing 2-7 on the FEMTO-ST dataset.

As the change of normalized RUL follows a straight line with value decreasing from 1 to 0, it is reasonable to use linear regression to characterize the behavior of normalized RUL. Consequently, least squares is employed to estimate parameters a, b in the linear function $\hat{y}_t = at + b$ modeling the relationship between time t and the point RUL estimation $\hat{y}_t$.

As the value of normalized RUL falls within the range $[0, 1]$, we inject the following two constraints to ensure that the RUL values estimated by the linear function are valid:

$$\hat{y}_t = \begin{cases} at + b - (\hat{y}_1 - 1) & \text{if } \hat{y}_1 > 1 \\ at + b + (1 - \hat{y}_1) & \text{if } \hat{y}_1 \leq 1, \end{cases} \tag{5.12}$$

where $\hat{y}_1$ denote the first point of the linear fitting model.

In principle, we adjust the bias of the fitted linear regression model to ensure that the RUL estima-

tion always starts from the value 1. In doing this, we guarantee that the RUL predictions produced by the linear model are physically meaningful. The effectiveness of linear regression is demonstrated in Figure 5.7. Clearly, it largely reduces the prediction error, particularly when the initial RUL estimation is biased.

5.2.4.2 Reliability estimation

The reliability of encoder-decoder paradigm is contingent on the amount of available historical data (Y. Chen et al., 2020). As shown in Figure 5.8, when the amount of the data used as the input to predict the RUL increases, we observe that the model predictions and true RULs show better and better consistency. However, in the online setting, we have no idea about when the model predictions will align well with the actual RUL, thus rendering doubt on the credibility of model predictions. Therefore, it becomes vital to estimate the reliability associated with each RUL estimation. To tackle this problem, we construct a novel approach to quantify the reliability associated with each RUL prediction by analyzing the intrinsic property of the RUL predictor.

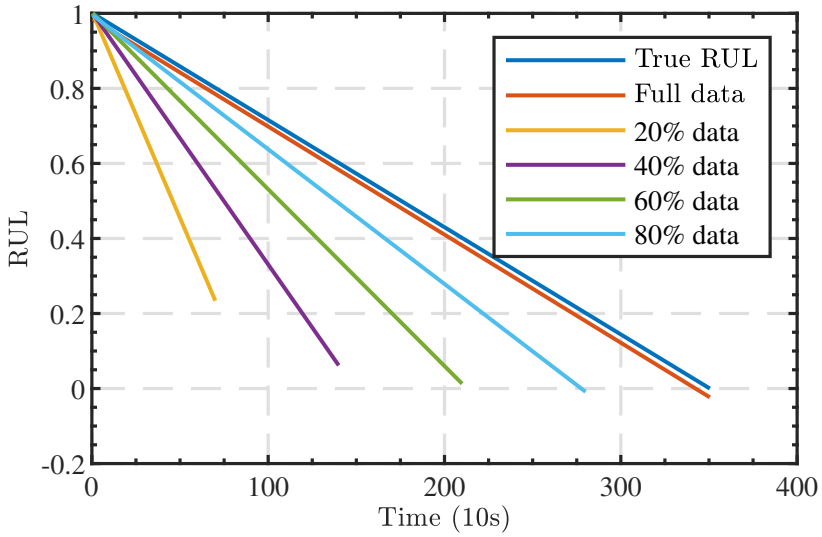


Figure 5.8: RUL predictions using different amounts of historical data for Bearing 1-3 in the FEMTO-ST dataset.

As mentioned earlier, the slope of the fitted linear regression model is closely dependent on the length of the input data. As illustrated in Figure 5.8, at the early stage, model predictions are severely biased as there is a large gap between the model's predictions and the ground truth RUL values. When more and more data becomes available, the RUL estimations get closer and closer to the actual RUL values. However, due to the uncertain failure times, it is impossible to ascertain a universal parameter to compensate the model prediction so as to match it with the true RUL value. Fortunately, there is

a consistent pattern pertaining to the RUL estimations across different bearing failure data. Thus, we propose to exploit this pattern to estimate the reliability associated with each RUL prediction.

Suppose the slope matrix is denoted as $\boldsymbol{a} = [a_1, a_2, \cdots, a_n]$, where a_n denotes the slope associated with the linear regression model fitted with RUL predictions up to time n, we derive the discrete differentiation of the slope matrix as:

$$\boldsymbol{a}' = [(a_2 - a_1), (a_3 - a_2), \cdots, (a_n - a_{n-1})]. \tag{5.13}$$

After applying Eq. (5.13), we observe that the trend of $\boldsymbol{a}'$ across various bearings exhibits a similar exponential degradation pattern as shown in Figure 5.9. Based on such finding, we leverage this pattern as a prior knowledge to build an exponential model:

$$\hat{a}'(t) = c\exp(d \cdot t), \tag{5.14}$$

where c and d are two parameters estimated with the nonlinear least squares algorithm.

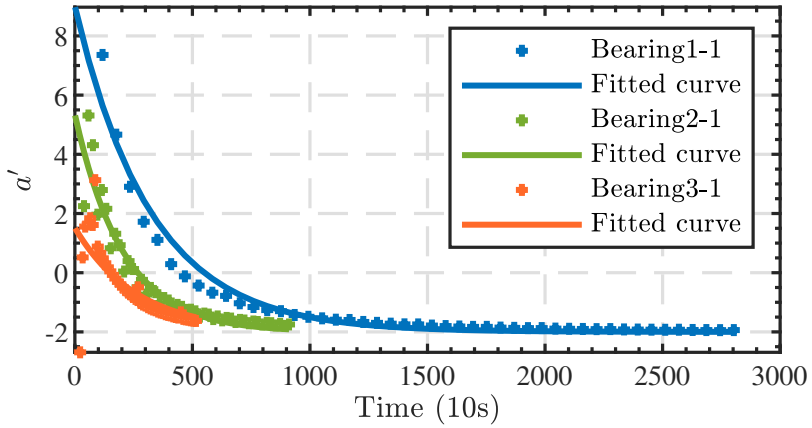


Figure 5.9: Variation of differential slopes and corresponding fitted curves on the FEMTO-ST training sets, where each point is the calculated $a'(t)$ at time t.

From Figure 5.9, it can be observed that all the fitted curves converge to a similar lower bound. When the differential slopes get closer to this lower bound, the RUL estimations tend to align well with the actual RUL value. Consequently, we devise a reliability metric established upon the bound of the exponential fitting models derived from the training set by following a simple two-step process. First, the training sets are used to approximate the upper B_u and lower bounds B_l of the differential

slopes:

$$B_l = \frac{1}{M}\sum_{i=1}^{M} \hat{a}'_i(n)$$
$$B_u = \frac{1}{M}\sum_{i=1}^{M} \hat{a}'_i(1), \tag{5.15}$$

where M indicates the number of fitted curves using the training sets, $\hat{a}'_i(1)$ and $\hat{a}'_i(n)$ are the start and end points of the i-th fitted curve respectively.

Second, we use B_u and B_l to construct a reliability indicator R to indicate the reliability associated with each RUL prediction in the online setting:

$$R_t = 1 - \frac{a'_t - B_l}{B_u - B_l} \times 100\%, \tag{5.16}$$

where a'_t is the differential slope on the test sets at time t.

Remark 3. Eqs. (5.15) and (5.16) defines the transformation of a scaler. This transformation aims to exploit the exponential degradation trend and establish a reliability indicator within the range $[0, 1]$. The point-wise reliability level on the test sets are depicted in Figure 5.10. As time progresses, the reliability of all bearings converges to 100% indicating that the RUL prediction becomes more and more reliable.

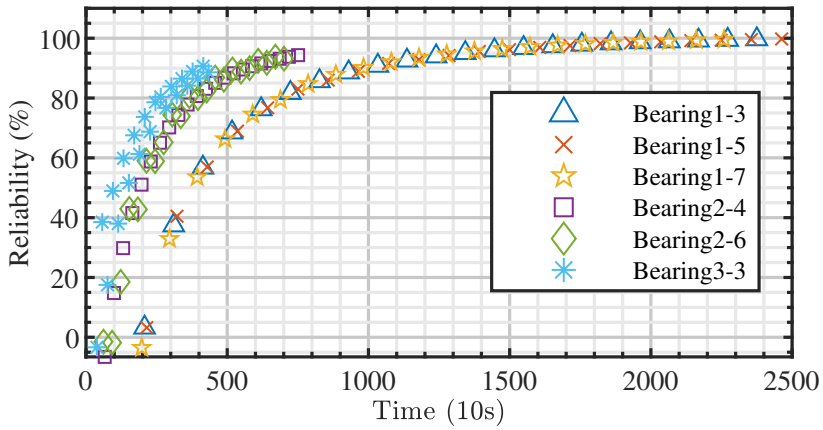


Figure 5.10: Reliability associated with RUL predictions on the FEMTO-ST test sets.

Remark 4. The proposed reliability metric is a simplified version of a physical failure model – the Weibull distribution model (Weibull, 1951). The Weibull cumulative distribution function (CDF) $F(t)$ describes the cumulative probability of failure, which is defined as (von Hahn & Mechefske, 2022):

$$F(t) = 1 - e^{-(t/\eta)^{\beta}}, \tag{5.17}$$

where t denotes the running time, β and η are the shape and characteristic life parameters respectively. By comparing the CDF of Weibull distribution (see Eq. (5.17)) and Eqs. (5.14)- (5.16), we conclude that the proposed reliability metric can be interpreted as a normalized Weibull CDF with $\beta = 1$ and $c = -\frac{1}{\eta}$. However, unlike previous studies that estimate the parameters of the Weibull CDF using Monte Carlo simulation (Hall & Strutt, 2003) or Bayesian estimation (Kaminskiy & Krivtsov, 2005), our method use the normalization to constrain the scale of the Weibull CDF, thereby facilitating a rapid estimation of the reliability pertaing to the model's RUL prediction.

5.3 Experiments

5.3.1 Experimental configurations

5.3.1.1 Dataset description

To validate the performance of the proposed method, we utilize the FEMTO-ST dataset in the literature. The FEMTO-ST dataset of the IEEE PHM 2012 data challenge comprises data from bearings that underwent accelerated degradation on a laboratory test rig (PRONOSTIA) under varying operating conditions (Nectoux et al., 2012), including rotating speed and load force. A total of 17 bearings were subjected to run-to-failure experiments under three distinct operating conditions: 1800rpm and 4000N, 1650rpm and 4200N, and 1500rpm and 5000N. The vibration signals were sampled at a rate of 25.6kHz with 2560 points (equivalent to 1/10s) recorded every 10 seconds until failure occurrence. We follow the official scheme to partition the entire data into training and test sets as outlined in Table 5.2. Both horizontal and vertical signals were utilized as training and test data.

Table 5.2: Dataset segmentation on FEMTO-ST.

	Condition1	Condition2	Condition3
Training set	Bearing 1-1 Bearing 1-2	Bearing 2-1 Bearing 2-2	Bearing 3-1 Bearing 3-2
Test set	Bearing 1-3 Bearing 1-4 Bearing 1-5 Bearing 1-6 Bearing 1-7	Bearing 2-3 Bearing 2-4 Bearing 2-5 Bearing 2-6 Bearing 2-7	Bearing 3-3

5.3.1.2 Baseline methods

Several SOTA RUL methods are used as baselines: i) Attention GRU, an encoder-decoder model using frequency domain statistical features (Y. Chen et al., 2020); ii) P-DA-GRU, parallel GRUs with dual-stage attention mechanism using RMS (L. Cao et al., 2023); iii) Multi-Scale, a multi-scale CNN using short-time Fourier transform (STFT) (X. Li, Zhang, & Ding, 2019); iv) TCN, a conventional temporal convolutional network (Bai, Kolter, & Koltun, 2018) using time domain statistical features in Ref. (Y. Chen et al., 2020); v) CNN-LSTM, a CNN encoder followed by an LSTM decoder using continuous wavelet transform (CWT) (W. Li & Deng, 2023).

All the experiments are conducted on a Windows 10 server with Intel i9-10900K @ 3.70GHz CPU, Nvidia RTX 3080 Ti 12GB GPU, and Python 3.10, Pytorch 2.1 environment.

5.3.1.3 Training settings

Given that the RUL task exhibits low sensitivity to hyperparameters, we opt to train the model using fixed hyperparameters. For the baselines, we either adopt the suggested hyperparameters or used hyperparameters identical to ours. The initial hyperparameters are set as follows: learning rate = 4×10^{-3}, training epochs = 600. Furthermore, we employ the teacher-forcing strategy (Algorithm 1)[2] for training the GRU network, with the corresponding hyperparameters being set to $\alpha = 0.7$. AdamW is chosen as the optimizer. The batch size is configured to match the length of features of each bearing.

Algorithm 1 Teacher Forcing

Require: Encoder output $\boldsymbol{x} = [x_1, x_2, \cdots, x_N]$, ground truth $\boldsymbol{y} = [y_1, y_2, \cdots, y_N]$,
teacher forcing ratio $0 < \alpha < 1$

```
Initial hidden h ← 1
Initial output ŷ ← 0
for t = 1 to N do
    if random.random() < α then
        ŷ[t], h[t] = Decoder(x[t], h[t])
    else
        ŷ[t], h[t] = Decoder(y[t], h[t])
    end if
end for
return ŷ
```

To evaluate the performance of these models, we consider two metrics, root mean square error

[2] https://machinelearningmastery.com/teacher-forcing-for-recurrent-neural-networks/

(RMSE) and mean absolute error (MAE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$
$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|.$$

5.3.2 Experimental results

Computational results are reported in Table 5.3 with several key findings. Firstly, our proposed method demonstrates a superior performance over baseline models in most instances and reduces the RMSE and MAE by 65.8% and 67.5% compared to the second-best method P-DA-GRU.

Table 5.3: RUL predictions by different methods on FEMTO-ST dataset.

Method	Attention GRU		P-DA-GRU		Multi-scale		TCN		CNN-LSTM		Ours	
Metric	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Bearing1-3	0.045	0.036	0.024	**0.014**	0.090	0.080	0.270	0.220	0.361	0.291	**0.017**	0.016
Bearing1-4	0.027	0.003	0.026	0.022	0.244	0.220	0.340	0.290	0.513	0.434	**0.012**	**0.010**
Bearing1-5	0.143	0.125	0.027	0.020	0.222	0.190	0.290	0.230	0.422	0.334	**0.008**	**0.007**
Bearing1-6	0.191	0.168	0.022	0.025	0.229	0.200	0.290	0.240	0.469	0.405	**0.005**	**0.005**
Bearing1-7	0.113	0.097	0.061	0.061	0.103	0.080	0.310	0.260	0.299	0.240	**0.000**	**0.000**
Bearing2-3	0.169	0.149	0.016	0.014	0.402	0.330	0.270	0.220	0.300	0.228	**0.004**	**0.004**
Bearing2-4	0.128	0.109	0.066	0.059	0.089	0.060	0.340	0.260	0.317	0.245	**0.013**	**0.012**
Bearing2-5	0.144	0.125	0.024	0.020	0.316	0.240	0.330	0.270	0.444	0.362	**0.014**	**0.012**
Bearing2-6	0.039	0.009	0.068	0.066	0.257	0.210	0.220	0.180	0.206	0.163	**0.019**	**0.019**
Bearing2-7	0.216	0.177	**0.044**	**0.040**	0.306	0.270	0.310	0.240	0.285	0.223	0.055	0.047
Bearing3-3	0.113	0.088	0.070	0.068	0.357	0.360	0.210	0.180	0.371	0.306	**0.004**	**0.004**
Average	0.121	0.099	0.041	0.037	0.238	0.203	0.288	0.237	0.363	0.294	**0.014**	**0.012**

Figure 5.11 shows the RUL predictions of the proposed method over some bearing instances. It is evident that our method exhibits exceptional prediction performance in most cases, as reflected by the high consistency between model predictions and the ground truth RULs.

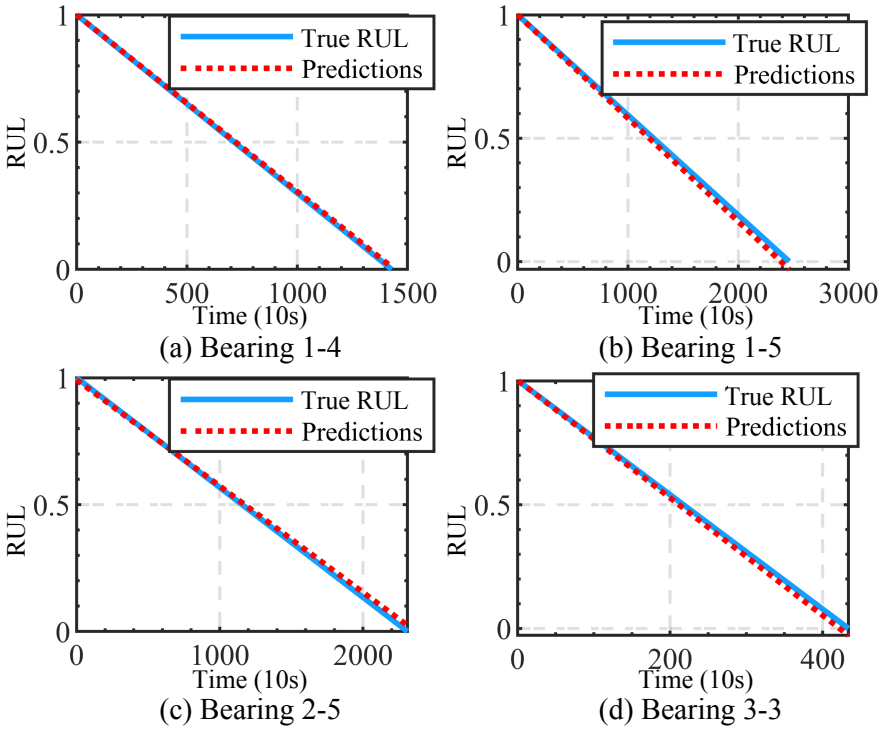


Figure 5.11: RUL predictions by the proposed method with respect to the FEMTO-ST dataset.

5.3.3 Analysis experiments

5.3.3.1 Ablation study

An ablation study is performed to evaluate the individual contribution of two key components in the developed method: LCT and linear regression (LR). Some combinations of these components are tested using the FEMTO-ST dataset and experimental results are reported in Table 5.4.

Table 5.4: RMSE of different models on the FEMTO-ST dataset. CT stands for cumulative transformation, LR stands for linear regression, LCT w/o LR means LCT without linear regression.

	RMS+LR	CT+LR	LCT+LR	LCT w/o LR
Bearing1-3	0.043	0.007	0.017	0.028
Bearing1-4	0.117	0.033	0.012	0.029
Bearing1-5	0.327	0.009	0.008	0.023
Bearing1-6	0.316	0.024	0.005	0.022
Bearing1-7	0.212	0.008	0.000	0.022
Bearing2-3	0.206	0.016	0.004	0.024
Bearing2-4	0.435	0.019	0.013	0.039
Bearing2-5	0.512	0.070	0.014	0.026
Bearing2-6	0.175	0.011	0.019	0.043
Bearing2-7	0.303	0.024	0.055	0.136
Bearing3-3	0.260	0.022	0.004	0.056
Average	0.264	0.022	**0.014**	0.041

Firstly, the results reveal that the combination of LCT and LR yields the best performance, thus demonstrating the effectiveness of our framework. By comparing RMS, CT and LCT, cumulative transformation significantly enhances performance, emphasizing the importance of smooth, monotone features for accurate RUL estimation. LCT further improves the prediction performance by aligning training and test set features, as evidenced by the sub 1×10^{-3} error rate for bearing 1-7.

Secondly, the component of linear regression also contributes to improving the prediction performance by reducing the impact of initial fluctuations. As shown in Table 5.4, the RUL prediction of Bearing 2-7 and Bearing 3-3 without LR exhibits a large RMSE with a value of 0.136 and 0.056, respectively. After adding the LR, the RMSE drops down to 0.055 and 0.004. These results clearly reveal the essential role of LR in improving the accuracy of RUL prediction.

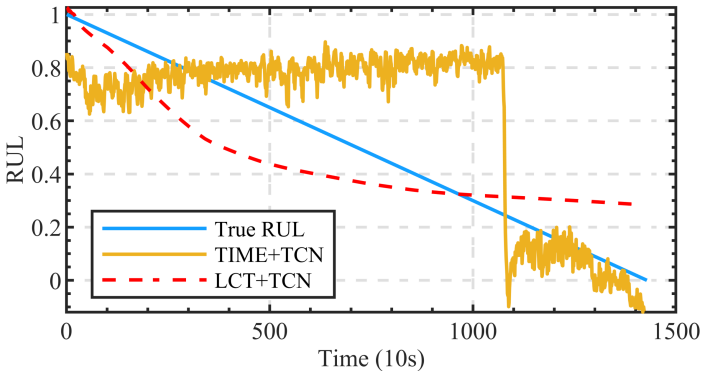
5.3.3.2 Employing LCT to other baseline models

As LCT is a feature extraction method, it is crucial to evaluate its performance if LCT is used with other RUL models. Here, we employed three mainstream RUL prediction methods, P-DA-GRU, TCN, and CNN-LSTM, to demonstrate the efficacy of LCT. It is noteworthy to say that these methods originally used different feature extraction methods. P-DA-GRU employed normalized RMS and isotonic regression (RMS+IR), TCN adopted time domain features (TIME), and CNN-LSTM utilized continuous wavelet transform (CWT).

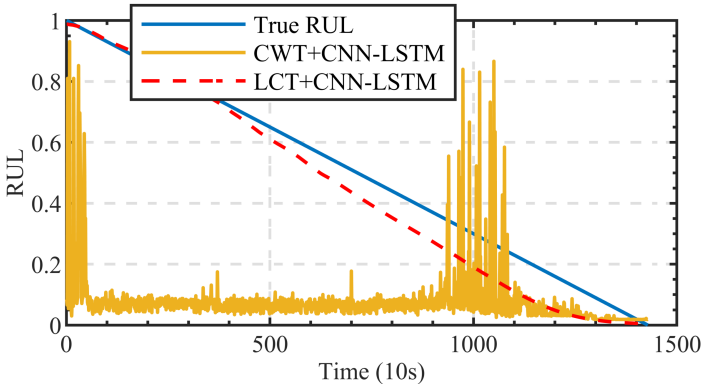
The results are illustrated in Table 5.5. LCT substantially improves the performance of all baseline networks compared to using other pre-processing methods. The average RMSEs decrease by 23% for TCN, 33% for CNN-LSTM, and 20% for P-DA-GRU, which highlights the simplicity and effectiveness of LCT. Moreover, Figure 5.12 compares the predictions of LCT and other preprocessing methods on Bearing 1-4. The predictions made with LCT highly align with the true RUL, whereas the adoptions of other pre-processing features result in significant discrepancies. This study highlights the huge impact of the signal pre-processing method on the performance of underlying models and the effectiveness of LCT in improving RUL prediction accuracy.

Table 5.5: The RMSE of RUL prediction on the FEMTO-ST dataset. Where TIME denotes the time domain statistical features, CWT is continuous wavelet transform, RMS+IR represents normalized RMS and isotonic regression.

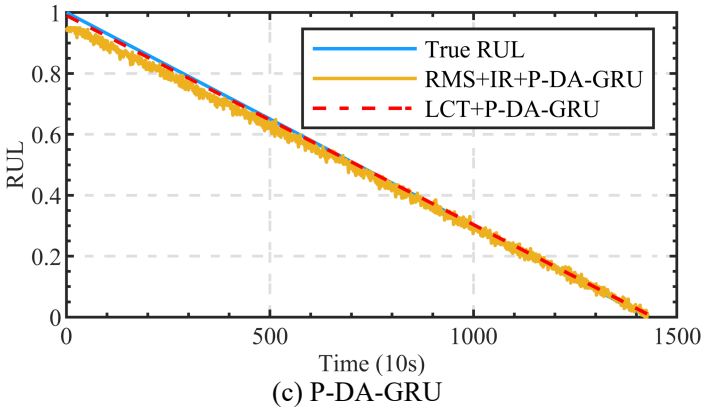
	TIME+TCN	LCT+TCN	CWT+CNN-LSTM	LCT+CNN-LSTM	RMS+IR+P-DA-GRU	LCT+P-DA-GRU
Bearing1-3	0.268	0.223	0.361	0.150	0.024	0.019
Bearing1-4	0.335	0.160	0.513	0.055	0.026	0.022
Bearing1-5	0.292	0.232	0.422	0.251	0.027	0.017
Bearing1-6	0.288	0.231	0.469	0.254	0.024	0.024
Bearing1-7	0.312	0.218	0.299	0.080	0.061	0.056
Bearing2-3	0.272	0.212	0.300	0.044	0.016	0.011
Bearing2-4	0.337	0.142	0.317	0.324	0.066	0.050
Bearing2-5	0.326	0.220	0.444	0.207	0.024	0.021
Bearing2-6	0.221	0.138	0.206	0.352	0.068	0.053
Bearing2-7	0.312	0.402	0.285	0.511	0.044	0.034
Bearing3-3	0.210	0.250	0.371	0.451	0.070	0.054
Average	0.288	**0.221**	0.363	**0.244**	0.041	**0.033**



(a) TCN



(b) CNN-LSTM



(c) P-DA-GRU

Figure 5.12: The predicted RUL of TCN and CNN-LSTM using different preprocessing on the Bearing 1-4. Where time denotes the time domain statistical features and CWT denotes continuous wavelet transform.

5.3.3.3 Comparing LCT with other feature extraction methods

We undertake experiments to validate the performance of LCT in comparison with other feature extraction methods. All these methods adopt an identical encoder-decoder backbone. We employ four classical feature extraction methods for comparison purposes: i) Energy values from five sub-band frequency spectra are extracted as frequency features (FRE) (Y. Chen et al., 2020). ii) Several time domain features such as RMS, standard deviation, and impulse factor, among others, are referred to (L. Liu et al., 2021). iii) Inertial relative root mean square (IRRMS) values are derived from raw vibration signals (M. Yan et al., 2020). iv) Trigonometric functions (arcsin, arctan) are utilized to transform the raw vibration signals (TRI) (Javed et al., 2015).

The results of these experiments are summarized in Table 5.6. It is evident that LCT outperforms the other methods by an order of magnitude. This underscores the significant usefulness of the LCT. Additionally, we also compare the model predictions using LCT and other feature extraction methods. As shown in Figure 5.13, the LCT-based predictions closely align with the true RUL, while other methods result in different levels of discrepancy. Such an experimental study clearly indicates that the LCT is a simple yet effective method for RUL feature extraction.

Table 5.6: Performance comparison of different feature extraction methods on the FEMTO-ST dataset, where FRE denotes frequency features, TIME denotes time features, IRRMS refers to the inertial relative root mean square, and TRI stands for trigonometric features.

	FRE	TIME	IRRMS	TRI	LCT
Bearing1-3	0.043	0.099	0.233	0.273	0.017
Bearing1-4	0.065	0.104	0.014	0.307	0.012
Bearing1-5	0.105	0.055	0.015	0.038	0.008
Bearing1-6	0.155	0.067	0.035	0.009	0.005
Bearing1-7	0.120	0.095	0.022	0.038	0.000
Bearing2-3	0.143	0.162	0.015	0.342	0.004
Bearing2-4	0.015	0.166	0.152	0.076	0.013
Bearing2-5	0.030	0.050	0.258	0.479	0.014
Bearing2-6	0.016	0.203	0.050	0.186	0.019
Bearing2-7	0.214	0.377	0.356	0.484	0.055
Bearing3-3	0.194	0.215	0.287	0.230	0.004
Average	0.100	0.145	0.131	0.224	**0.014**

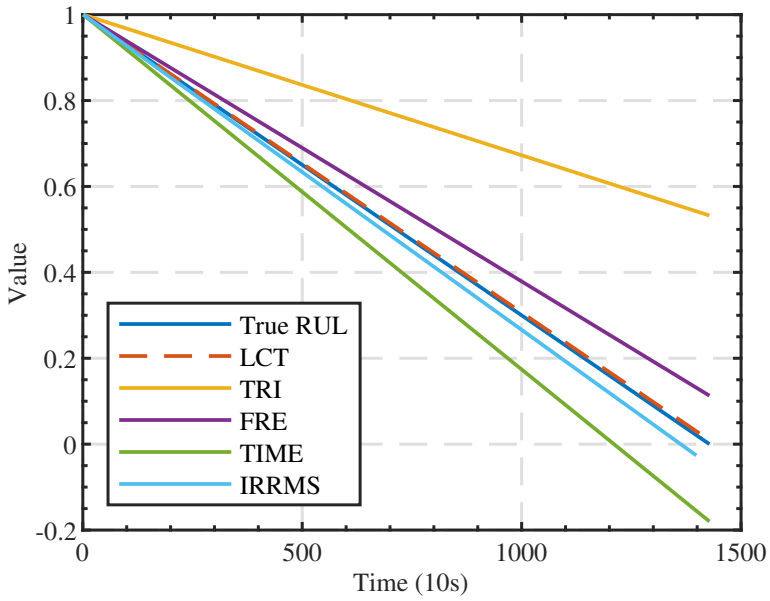


Figure 5.13: RUL predictions of different feature extraction methods on the Bearing 1-4.

5.3.3.4 The relationship between reliability metric and predictions

To further substantiate the plausibility of the proposed reliability metric, we present the RUL prediction associated with high reliability R. Note that some bearings in the test set possess a relatively short sample length. As reliability metrics are intrinsically correlated with the data length, these data do not attain a reliability exceeding 95%. The corresponding results are illustrated in Table 5.7.

Table 5.7: The RUL prediction results with high reliability.

	Reliability (%)	RMSE	MAE
Bearing1-3	99.656	0.033	0.028
Bearing1-4	97.618	0.012	0.011
Bearing1-5	99.755	0.028	0.024
Bearing1-6	99.714	0.025	0.022
Bearing1-7	99.562	0.020	0.017
Bearing2-3	99.024	0.024	0.021
Bearing2-4	94.341	0.019	0.017
Bearing2-5	99.889	0.004	0.004
Bearing2-6	93.830	0.050	0.044
Bearing2-7	89.889	0.013	0.011
Bearing3-3	90.662	0.038	0.033
Average	96.722	0.024	0.021

Generally, RUL predictions have an average RMSE and MAE of 0.024 and 0.021 when the average reliability surpasses 96%. Owing to the diverse degradation characteristics inherent to different bearings, it is infeasible to ensure a uniform correlation between reliability level and error across all

the scenarios. Nevertheless, a high reliability (over 90%) implies that the remaining useful life of the bearing is close to its end. Our reliability metric provides a fast evaluation for online scenarios.

5.3.3.5 Feeding LCT with various statistical features

In this study, we evaluate the performance of LCT by employing various statistical features as inputs. Specifically, we incorporate both time-domain and frequency-domain features as inputs to the LCT. Some of these features are consistent with those used in the main chapter, while others, such as the crest factor (CRE) and peak value, are introduced for comparison purposes. The IRRMS, which generates negative values that are incompatible with the LCT method, is excluded from this experimental study.

Computational results are summarized in Table 5.8. Compared to the frequency-domain features, time-domain features (e.g., Peak, TRI, CRE, RMS) are more effective in implementing the LCT method. Furthermore, the performance of different time-domain features varies depending on the specific data. Specifically, RMS-LCT and CRE-LCT yield the lowest and second-lowest RMSE across different bearings. Although the choice of features influences the results, the use of LCT leads to a consistent increase in the model performance. We therefore recommend using RMS-LCT and CRE-LCT for feature extraction.

Table 5.8: RMSE of LCT when different statistical features are used as input, where Peak is the peak value, TRI represents trigonometric features, CRE is the crest factor, and FRE denotes the frequency features.

	Peak-LCT	TRI-LCT	CRE-LCT	FRE-LCT	RMS-LCT
Bearing1-3	0.002	**0.016**	**0.016**	0.092	0.017
Bearing1-4	0.002	0.025	**0.000**	0.036	0.012
Bearing1-5	0.003	0.009	0.017	0.061	**0.008**
Bearing1-6	0.004	0.007	0.005	0.100	**0.005**
Bearing1-7	0.001	0.013	0.007	0.097	**0.000**
Bearing2-3	0.011	0.011	**0.003**	0.085	0.004
Bearing2-4	0.017	0.080	0.014	0.015	**0.013**
Bearing2-5	**0.005**	0.100	0.007	0.113	0.014
Bearing2-6	0.016	0.046	0.027	**0.001**	0.019
Bearing2-7	0.071	0.131	**0.045**	0.294	0.055
Bearing3-3	0.043	0.008	0.013	0.153	**0.004**
Average	0.016	0.041	**0.014**	0.095	**0.014**

Figure 5.14 illustrates the features transformed by LCT. LCT serves as a potent tool for aligning morphological differences among various time-domain features. Despite the differences in these features, they all exhibit a similar linear trend, albeit with inconsistent growth. Frequency-domain

features, on the other hand, do not conform to this monotonically increasing pattern, thus resulting in unstable fluctuations in their LCT. Given that our RUL indicator is linear, time-domain features yield desirable results. At last, it is noteworthy that TRI-LCT and RMS-LCT yield similar values and rank first and second (See Table 5.8) respectively, for Bearing 3-3. These methods have the smallest growth rate, suggesting that features with smooth variations are more beneficial for RUL prediction.

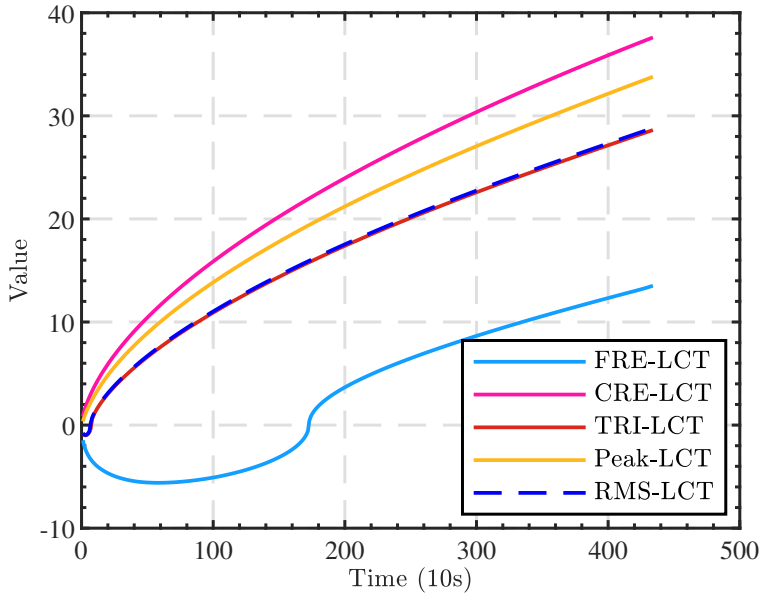


Figure 5.14: The statistical features of different methods followed by LCT on the Bearing 3-3.

5.3.3.6 Iteration endpoints of LCT

Considering the iterative nature of LCT, it is intriguing to consider the outcome of repeating LCT transformations. We continue executing the LCT following the initial transformation. However, this process must cease after the second cumulative transformation is performed. This rationale can be elucidated from two perspectives.

Firstly, the continuation of the LCT is impeded as the logarithmic function is incapable of handling negative numbers. This phenomenon is clarified in Table 5.9. As the transformation progresses, there is a clear increase in the appearance of negative values. This is attributed to the logarithmic function's ability to map the input to negative values, and the cumulative function's propensity to pass further negative numbers to subsequent values.

Table 5.9: The first five values of the feature after transformation. Where CT denotes cumulative transformation and LT denotes logarithmic transformation. Steps 1 to 3 complete an LCT operation.

Step 1-CT	Step 2-LT	Step 3-CT	Step 4-CT
0.75	-0.29	-0.54	-0.73
1.05	0.05	-0.49	-1.01
1.28	0.24	0.04	-0.99
1.48	0.39	0.63	-0.60
1.66	0.51	0.95	0.76

Secondly, as illustrated in Figure 5.15, the cumulative function incessantly accumulates preceding inputs, resulting in an output range significantly larger than the inputs. This is not conducive to confining the features of disparate bearings within the same interval. Consequently, the execution of a single LCT emerges as a judicious choice.

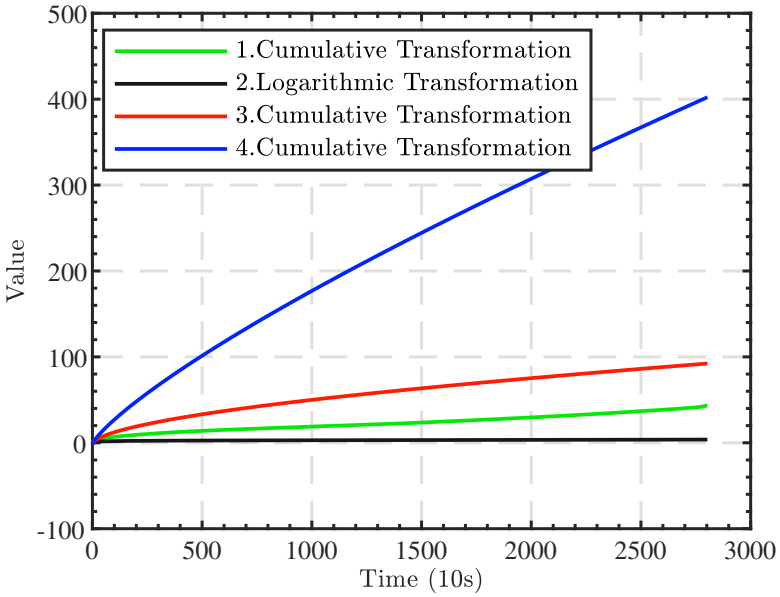


Figure 5.15: The feature transformed by continuous using LCT.

5.3.3.7 The power of normalization

We conduct an experiment to elucidate the role of normalization on the RUL prediction. In this study, we employ the temporal convolutional network (TCN) as the backbone of the neural network. Given that TCNs are trained using a short period of features at once, the effect of normalization is significant. Subsequently, we implement two feature extraction methods for comparison purposes: cumulative transform (CT) and frequency features (FRE), as shown in Table 5.10.

The results indicate that the normalized feature has a substantial impact on the model performance. Specifically, normalization greatly improves the performance of RUL prediction using frequency do-

main features. With normalization, the average RMSE diminishes by an order of magnitude. Similarly, normalization is also beneficial to CT and it results in a 47% improvement in the average RMSE.

Table 5.10: Comparing the RMSEs of TCN using normalization. Where CT denotes cumulative transform and FRE denotes frequency domain features.

	CT-TCN	CT (Norm)-TCN	FRE-TCN	FRE (Norm)-TCN
Bearing1-3	0.131	0.155	0.287	0.104
Bearing1-4	0.110	0.296	0.205	0.040
Bearing1-5	0.151	0.116	0.328	0.014
Bearing1-6	0.170	0.109	0.272	0.014
Bearing1-7	0.152	0.058	0.311	0.026
Bearing2-3	0.160	0.120	0.311	0.052
Bearing2-4	0.246	0.096	0.314	0.026
Bearing2-5	0.171	0.150	0.357	0.026
Bearing2-6	0.278	0.086	0.309	0.021
Bearing2-7	0.445	0.031	0.285	0.046
Bearing3-3	0.373	0.049	0.248	0.039
Average	0.217	**0.115**	0.293	**0.037**

Furthermore, we visualize the RUL predictions of these methods in Figure 5.16. Obviously, normalization regularizes the predicted intervals and leads to a better alignment between model prediction and the ground truth RUL value.

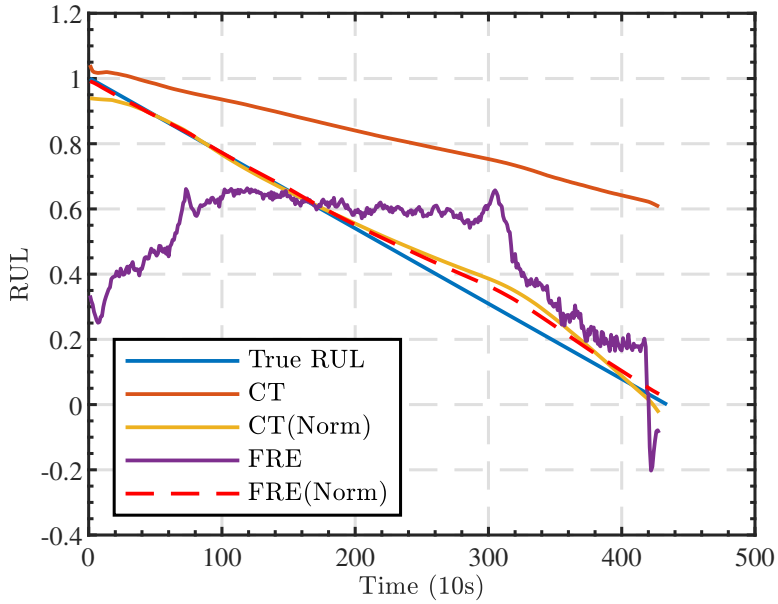
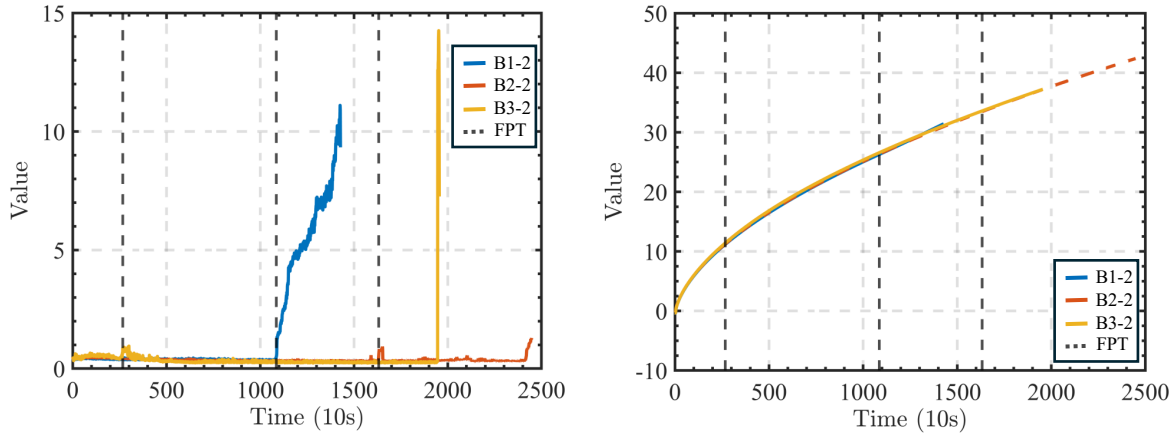


Figure 5.16: The predicted RUL of TCN with and without normalization on the Bearing 3-3.

5.3.3.8 Influence of FPT

We employ the 3σ principle (X. Li, Zhang, & Ding, 2019) to determine the First Prediction Time (FPT) and validate the performance of the LCT under piece-wise linear Health Indicators (HI). Since the LCT smooths the exponential trends, we adopt the RMS value to calculate the FPT; subsequently, only the features obtained after the FPT are used to train the model, as shown in Figure 5.17. Note that signal fluctuations can lead to erroneous FPT determination. However, despite this error, the shape and scale of the LCT remain invariant. This is because the cumulative computation initiates at the beginning of running, ensuring alignment across different bearings.



(a) RMS of Bearings (b) LCT of Bearings

Figure 5.17: Calculating the FPT of three bearings on the FEMTO-ST training set using the RMS value.

Table 5.11 compares the results of using FPT for model training and RUL prediction on the test set. While the average performance notably improves with the application of FPT, individual results exhibit some variance. Specifically, instability in FPT determination—caused by RMS fluctuations in the bearing signal—can lead to incorrect RUL initialization and a subsequent increase in RMSE. Conversely, the LCT's ability to maintain shape and scale alignment ensures that model features remain robust. This indicates that our method effectively integrates FPT for piece-wise linear HI without requiring additional modifications.

Table 5.11: The RMSE of using FPT in RUL models.

Method	RMS	RMS+FPT	LCT	LCT+FPT
Bearing1-3	0.043	0.037	0.017	0.017
Bearing1-4	0.117	0.253	0.012	0.011
Bearing1-5	0.327	0.337	0.008	0.010
Bearing1-6	0.316	0.125	0.005	0.004
Bearing1-7	0.212	0.208	0.000	0.001
Bearing2-3	0.206	0.185	0.004	0.005
Bearing2-4	0.435	0.354	0.013	0.012
Bearing2-5	0.512	0.237	0.014	0.011
Bearing2-6	0.175	0.255	0.019	0.020
Bearing2-7	0.303	0.443	0.055	0.066
Bearing3-3	0.260	0.321	0.004	0.002
Average	0.264	0.250	0.014	**0.013**

5.3.4 Reliability evaluation in online test

In this section, we first explore the correlation between the proposed reliability indicator and predicted results to validate the soundness of the developed reliability estimator. Subsequently, we illustrate how to employ the proposed approach to predict the bearing RUL and estimate its associated reliability in an online setting.

Firstly, the full-time LCT features on the test set are divided into 25 segments varying from 4% to 100% of the total data for prediction. Subsequently, the encoder-decoder network and linear regression are employed to ascertain the slopes of each RUL prediction. The reliability R of each slope is then computed using Eq. (5.16). Note that the lower and upper bounds are determined using training sets. The correlation curves of R and MAE are illustrated in Figure 5.18. These curves demonstrate that R exhibits a monotonically decreasing trend with MAE. In most cases, when R's value exceeds 98%, the MAE is less than 0.03, which indicates that the reliability indicator R clearly reflects the magnitude of error in RUL prediction.

Next, an instance of utilizing the proposed framework for online test is depicted in Figure 5.19. In the online setting, the volume of historical data incrementally expands, yielding multiple sets of RUL predictions over time. It is evident that the accuracy of the prediction is directly proportional to its reliability; the more reliable the prediction, the closer it is to the actual RUL value. The reliability metric evaluates each prediction based on the available data, thereby facilitating a rapid assessment. This metric can effectively indicate when the prediction is reliable, offering valuable guidance for maintenance decision-making.

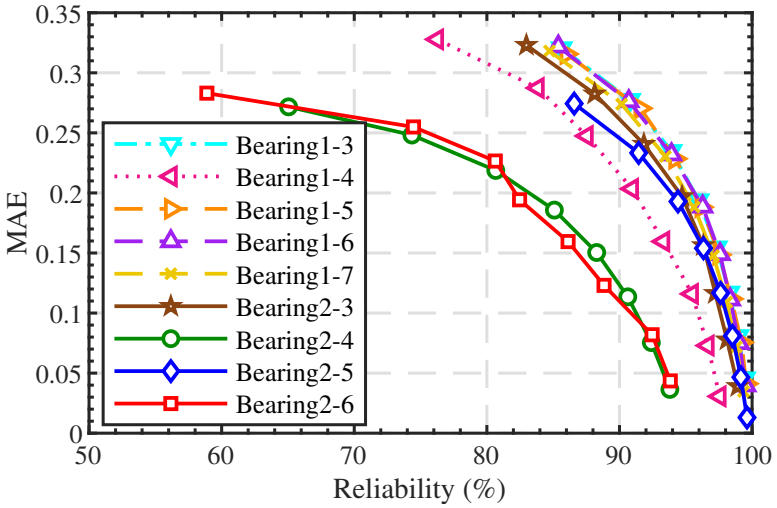


Figure 5.18: Relationship between the reliability indicator R and MAE of RUL estimations on the FEMTO-ST test sets.

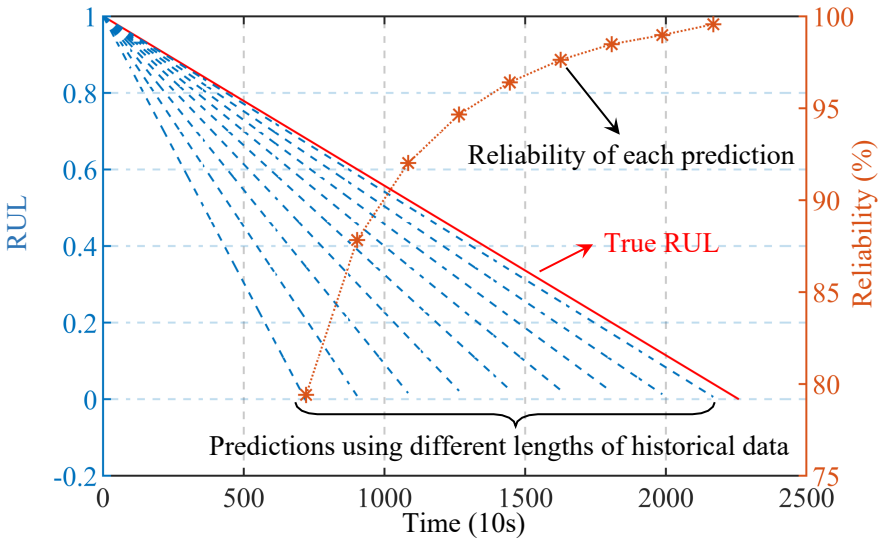


Figure 5.19: RUL predictions and corresponding reliability estimations using different lengths of historical data on the Bearing2-3.

5.4 Summary

While the LCT is highly effective in bearing RUL prediction, there are several limitations that need to be future investigated. Firstly, we only validate the performance of the developed method using the FEMETO-ST dataset. Experimental results suggest that the LCT can enhance generalizability across different bearings and operating conditions on the same test rig. However, the generalization capability of LCT needs to be further examined using other bearing datasets (B. Wang et al., 2018). Finally, the proposed model does not account for uncertainty. Future work could address this by developing an uncertainty quantification framework. For instance, task uncertainty could be modeled via an additional noise term appended to the linear regression model, followed by Bayesian updating

of parameter posteriors. Extrapolation would then yield both RUL point estimates and confidence intervals, enabling uncertainty quantification.

In this chapter, we propose a novel and effective feature extraction approach – the Logarithmic Cumulative Transformation (LCT) – to address the feature misalignment issue across different bearings in RUL prediction. The LCT, composed of three straightforward transformations, yields features with good alignment in shape and scale, thereby enhancing the generalizability of downstream RUL estimations. Furthermore, we have devised a novel method to estimate the reliability associated with each RUL prediction. This method employs the differential slopes of the linear regression model to derive the reliability indicator. Computational results demonstrate strong generalizability and reliability of our method across diverse samples, achieving a lower 0.03 MAE in RUL prediction when the Reliability score is over 98%.

Chapter 6

BearingPGA-Net: A Lightweight and Deployable Bearing Fault Diagnosis Network via Decoupled Knowledge Distillation and FPGA Acceleration

6.1 Introduction

ROTATING machines, including turbines, pumps, compressors, fans, etc., are widely used in industrial fields (Boudiaf et al., 2016; X. Chen et al., 2023; Qiao & Lu, 2015a). Rolling bearings, whose primary roles are to support the whole machine and reduce friction, are crucial components in rotating machines. Statistics from authoritative institutes reveal that bearing failure accounts for approximately 45%–70% of all mechanical faults (Benbouzid, 2000; Bonnett & Yung, 2008). Therefore, detecting bearing failure is an essential task in industry and civil fields. A timely and accurate detection can greatly enhance the reliability and efficiency of bearings, thereby reducing the enormous economic loss.

Since bearing failures often exhibit unique abnormal vibration (Qiao & Lu, 2015b; Rauber, de Assis Boldt, & Varejao, 2014; Smith & Randall, 2015), the most common approach to diagnosing bearing faults is to install an acceleration sensor on mechanical surfaces to measure vibration signals, and then a diagnosis algorithm is applied to detect faulty signals. Over the past decade, deep learning methods, represented by convolutional neural networks (CNNs) and attention-based models, have

been dominating the problem of bearing fault diagnosis (L. Jia et al., 2022; J.-X. Liao, Dong, Sun, et al., 2023; Tran et al., 2022; H. Wang et al., 2022). Despite the significant success achieved by deep learning methods, the growing model size renders them impractical in industrial settings, since deploying deep learning models often demands high-performance computers. But a factory usually has multiple rotating machines. For instance, in nuclear power plants, there are numerous rotating machines such as seawater booster pumps and vacuum pumps that require monitoring of their vibration signals (W. Cheng et al., 2024). Thus, installing computers for each machine will not only undesirably occupy space but also impede production due to excessive circuit connections.

In fact, the field programmable gate array (FPGA), a class of low-power consumption, high-speed, programmable logic devices, is widely used in industrial fields. FPGA has high-speed signal processing capabilities such as time-frequency analysis, which can enhance the accuracy and real-time performance of fault detection (Choi et al., 2003; Dick & Harris, 2000; Jamshidpour et al., 2014). By leveraging its parallel computing capabilities, FPGA dramatically increases the speed and energy efficiency, achieving over a thousand times improvement than a digital signal processor (DSP) in fault detection execution (Q. Liu et al., 2019). However, the FPGA suffers from limited storage space and computational resources, which brings huge trouble in bearing fault diagnosis, as it needs to process long signal sequences in a timely manner. In brief, to deploy fault diagnosis in practical scenarios, we need to address two challenges: designing lightweight but well-performed deep learning models and deploying these models on embedded hardware such as FPGAs without compromising the performance too much.

Along this line, researchers have put forward lightweight models for bearing fault diagnosis, aiming to strike a balance between diagnostic accuracy and model complexity (H. Fang, Deng, Bai, et al., 2021; W. Li et al., 2023; D. Yao et al., 2020; Zhong et al., 2022). Nevertheless, these lightweight models have not been really deployed on the embedded hardware, and whether or not the good performance can be delivered as expected remains uncertain. On the other hand, two types of hardware have been utilized to deploy CNNs: System-on-Chip (SoC) FPGA and Virtex series FPGA. The former, exemplified by ZYNQ series chips, which integrate an ARM processor with an FPGA chip, have been employed to deploy a limited number of bearing fault diagnosis methods (Ji et al., 2022; Toumi et al., 2022). These approaches rely on the PYNQ framework, which translates the Python code into FPGA-executed Verilog through the ZYNQ board. While the PYNQ framework reduces development

time, the FPGA resources available for CNN acceleration are limited, reducing the processing time compared to FPGA boards. As for the latter, some prior researchers have designed FPGA-based accelerators for CNNs and successfully deployed CNNs on Virtex series FPGA (Mittal, 2020; Qiu et al., 2016; C. Zhang et al., 2015). However, these FPGAs, capable of deploying multi-layer CNNs, cost ten times more than commonly used mid-range FPGAs (Kintex series). Consequently, widespread deployment is hindered due to excessive costs. To resolve the tension between model compactness and deployment, here we propose BearingPGA-Net: a lightweight and deployable bearing fault diagnosis network via decoupled knowledge distillation and FPGA acceleration.

First, we apply the decoupled knowledge distillation (DKD) to build a lightweight yet well-performed network, named BearingPGA-Net. Knowledge distillation (KD) is a form of model compression that transfers knowledge from a large network (teacher) to a smaller one (student) (Gou et al., 2021; G. Hinton, Vinyals, & Dean, 2015), which is deemed as more effective than directly prototyping a small network. Knowledge distillation can be categorized into response-based KD (logit KD) and feature-based KD. Albeit previous studies have demonstrated the superiority of feature-based KD over logit KD for various tasks (Romero et al., 2014; L. Wang & Yoon, 2021), we select the logit KD. Our student network comprises only a single convolutional layer for deployment, and there are no hidden features for feature-based KD to distill. Recently, a novel approach called decoupled knowledge distillation has been proposed, which reformulates the traditional logit distillation by decomposing the knowledge distillation loss function into two separate functions. The training of these functions can be flexibly balanced based on the task (B. Zhao et al., 2022). The decoupling is a strong booster for the performance of logit KD, and we favorably translate it into our task.

Second, we utilize Verilog to implement neural network accelerators. By devising parallel computing and a module reuse, we deploy BearingPGA-Net and enhance its computing speed in a Kintex-7 FPGA. Specifically, we construct basic arithmetic modules with basic multiplication and addition units for the convolutional and fully-connected layers. To cater to the requirements of BearingPGA-Net, we fuse a ReLU and Max-pooling layer to further reduce computation. These computational modules are also translatable to other CNN-based tasks. Moreover, we design a tailored layer-by-layer fixed-point quantization scheme for neural networks, ensuring minimal loss of parameter accuracy in FPGA computations while also cutting the number of parameters by half. Notably, our FPGA framework fully leverages the computational resources of the Kintex-7 FPGA, which is a widely used mid-range

FPGA in industrial fields. Compared to previous implementations via SoC FPGA and Virtex FPGA, BearingPGA-Net+Kintex FPGA achieves preferable power consumption, model compactness, and high performance, which is highly scalable in real-world fault diagnosis scenarios. In summary, our contributions are threefold:

1. We build BearingPGA-Net, a lightweight neural network tailored for bearing fault diagnosis. This network is characterized by a single convolutional layer and is trained via decoupled knowledge distillation.

2. We propose a CNN acceleration scheme for BearingPGA-Net, where the general convolutional operators are implemented by the basic multiplication-accumulation unit, and a ReLU and max-pooling fusion layer is designed to further save computational sources. Furthermore, we utilize parallel computing and module reuse techniques to fully leverage the computational resources of the Kintex-7 FPGA. To the best of our knowledge, it is the first time that a bearing fault diagnosis model is directly deployed into a pure FPGA platform, which is cheaper and more efficient.

3. Compared to lightweight competitors, our proposed method demonstrates exceptional performance in noisy environments, achieving an average F1 score of over 98% on CWRU datasets. Moreover, it offers a smaller model size, with only 2.83K parameters. Notably, our FPGA deployment solution is also translatable to other FPGA boards.

6.2 The proposed method

As depicted in Figure 6.1, prototyping BearingPGA-Net follows a two-step pipeline: i) training a lightweight BearingPGA-Net via decoupled knowledge distillation; ii) deploying the BearingPGA-Net into an FPGA. Notably, we devise a layer-by-layer fixed-point quantization method to convert the parameters (weights and biases) of PyTorch's CNN, which are originally in 32-bit floating format, into a 16-bit fixed-point format. Additionally, for online diagnosis, the measured signal is amplified by a signal conditioner and then converted into a digital signal by an Analog-Digital (AD) converter. Subsequently, the FIFO (first in first out) performs clock synchronization and buffering, followed by executing convolutional neural network operations. Finally, the diagnostic result is displayed using four LEDs.

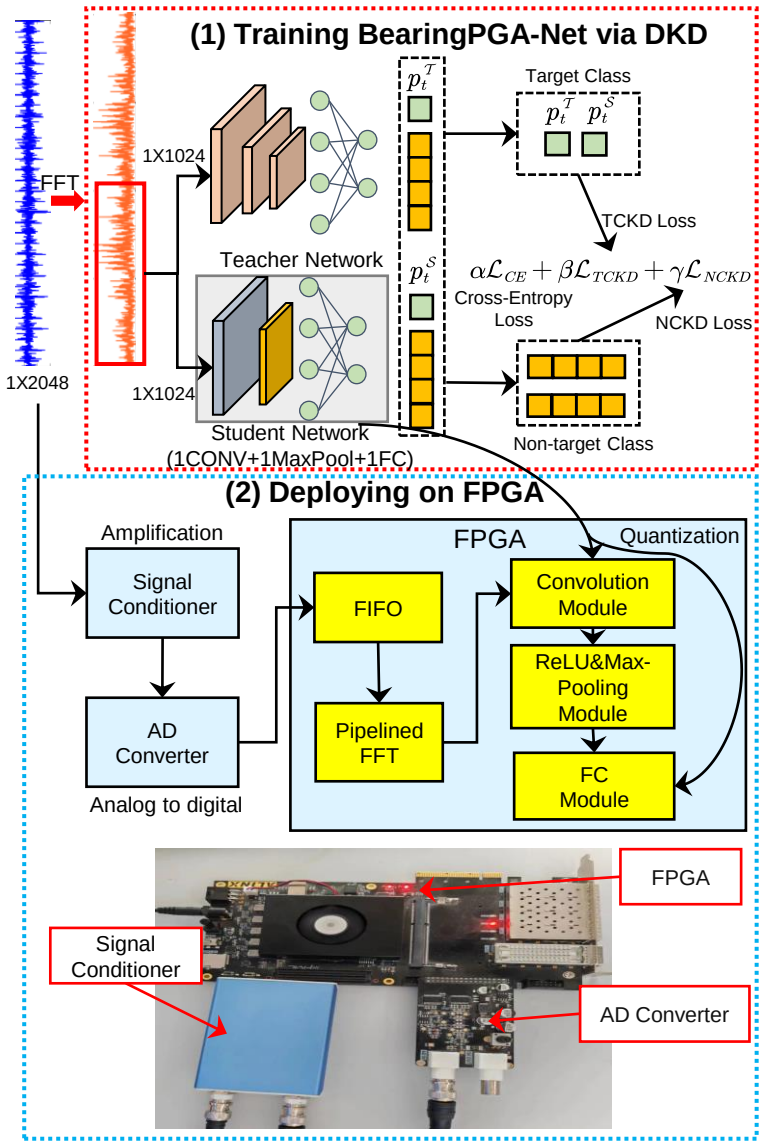


Figure 6.1: The overall framework for prototyping and deploying BearingPGA-Net.

6.2.1 Training BearingPGA-Net

The BearingPGA-Net is trained via decoupled knowledge distillation (DKD), which is an approach to training a small model (student) with a well-trained large model (teacher). It forces the student to emulate the output of the teacher. During the training process, the teacher model is trained first. Subsequently, its parameters are frozen to generate the outputs as new labels to train the student model.

6.2.1.1 Constructing teacher and student networks

For the teacher network, we adopt a widely-used 1D-CNN architecture known as the WDCNN (W. Zhang et al., 2017), which consists of six CNN blocks and one fully-connected layer. In our implementation, since the input size of our network is 1×1024, which is different from the size in the

original WDCNN (1×2048), we need to adjust the structure of WDCNN. Specifically, convolution operations do not care about the dimension of the input, and only fully-connected layers are revised. Since the lengths of feature vectors after convolution are decreased, we lower the widths of two fully connected layers to (64, 48) accordingly.

Additionally, we design a one-layer student network specifically for FPGA deployment (BearingPGA-Net). This student network comprises a single 1D-CNN layer, a ReLU activation layer, and a Max-Pooling layer, followed by a fully-connected layer mapping the latent features to the logits. The structure information of BearingPGA-Net is shown in Table 6.1.

Table 6.1: The structure parameters of BearingPGA-Net.

Layer	Kernel size	Channel	Output size	Padding	Stride
Conv1d	64×1	4	128×4	28	8
ReLU	-	-	128×4	-	-
MaxPool	2×1	4	64×4	0	2
Linear	(256,10)	-	10	-	-

6.2.1.2 Decoupled knowledge distillation

Despite numerous forms of knowledge distillation, our method adopts the response-based KD, which utilizes the teacher model's logits for knowledge transfer. This is because the limited hardware resources in a low-cost FPGA cannot hold two or more convolutional layers. Then, there are no hidden features for feature-based KD to distill.

In classical knowledge distillation, soft labels, which are the logits produced by the teacher, are deemed as distilled knowledge (G. Hinton, Vinyals, & Dean, 2015). Soft labels are obtained by the softmax function converting the logits z_i of a neural network into the probability p_i of the i-th class:

$$p_i = \frac{\exp\left(z_i/T\right)}{\sum_{j=1}^{C}\exp(z_j/T)}, \tag{6.1}$$

where C represents the number of classes, and $T \in \mathbb{R}^+$ serves as the temperature factor. When the temperature T is higher, the probability distribution over classes becomes smoother. A lower value of T (where $T < 1$) sharpens the output, increasing the disparity in probability values of different classes.

For classification, the cross-entropy (CE) loss is adopted to measure the difference between the

probability of the predicted label p and the ground truth y:

$$\mathcal{L}_{CE}(y, p(z,T)) = -\sum_{i=1}^{C} y_i \log p_i, \tag{6.2}$$

and KL-Divergence measures the similarity between the probability labels of the teacher $p^{\mathcal{T}}$ and the student $p^{\mathcal{S}}$,

$$\mathcal{L}_{KL}(p^{\mathcal{T}}(z,T), p^{\mathcal{S}}(z,T)) = -\sum_{i=1}^{C} p_i^{\mathcal{T}} \log \frac{p_i^{\mathcal{S}}}{p_i^{\mathcal{T}}}. \tag{6.3}$$

KD combines two loss functions:

$$\mathcal{L}_{KD} = (1-\alpha)\mathcal{L}_{CE}(y, p^{\mathcal{S}}) + \alpha T^2 \mathcal{L}_{KL}(p^{\mathcal{T}}(z,T), p^{\mathcal{S}}(z,T)), \tag{6.4}$$

where α is the scale factor to reconcile the weights of two loss functions, and T^2 keeps two loss functions at the same level of magnitude. Combining $\mathcal{L}_{KL}$ and $\mathcal{L}_{KD}$ helps the student get the guidance of the teacher and the feedback from the ground truth. This ensures that the student model learns effectively and reduces the risk of being misled.

However, $\mathcal{L}_{KL}$ implies that all logits from the teacher are transferred to the student on an equal booting. Intuitively, student models should have the ability to filter the knowledge they receive. In other words, knowledge relevant to the current target category should be reinforced, while knowledge far from the target category should be attenuated. The coupling in classical KD harms the effectiveness and flexibility across various tasks. To address this, decoupled knowledge distillation, which factorizes $\mathcal{L}_{KL}$ into a weighted sum of two terms (the target class and non-target class), was proposed (B. Zhao et al., 2022).

Let p_t and $p_{/t}$ be probabilities of the target and non-target classes, respectively. Then, we have

$$p_t = \frac{\exp(z_t/T)}{\sum_{j=1}^{C} \exp(z_j/T)},\ p_{/t} = \frac{\sum_{i=1, i\neq t}^{C} \exp(z_i/T)}{\sum_{j=1}^{C} \exp(z_j/T)}, \tag{6.5}$$

and

$$\hat{p}_t = \frac{p_t}{p_{/t}} = \frac{\exp(z_t/T)}{\sum_{i=1, i\neq t}^{C} \exp(z_i/T)}. \tag{6.6}$$

Then, $\mathcal{L}_{KL}$ can be factorized as

$$\begin{aligned}
&\mathcal{L}_{KL}(p^{\mathcal{T}}(z,T), p^{S}(z,T)) \\
=& -p_t^{\mathcal{T}} \log \frac{p_t^{\mathcal{S}}}{p_t^{\mathcal{T}}} - \sum_{i=1, i\neq t}^{C} p_i^{\mathcal{T}} \log \frac{p_i^{\mathcal{S}}}{p_i^{\mathcal{T}}} \\
=& \underbrace{-p_t^{\mathcal{T}} \log \frac{p_t^{\mathcal{S}}}{p_t^{\mathcal{T}}} - p_{/t}^{\mathcal{T}} \log \frac{p_{/t}^{\mathcal{S}}}{p_{/t}^{\mathcal{T}}}}_{\mathcal{L}_{KL}^{TCKD}} - p_{/t}^{\mathcal{T}} \underbrace{\sum_{i=1, i\neq t}^{C} \hat{p}_i^{\mathcal{T}} \log \frac{\hat{p}_i^{\mathcal{S}}}{\hat{p}_i^{\mathcal{T}}}}_{\mathcal{L}_{KL}^{NCKD}},
\end{aligned} \tag{6.7}$$

where $\mathcal{L}_{KL}^{TCKD}$ denotes the KL loss between teacher and student's probabilities of the target class, named target class knowledge distillation (TCKD), while $\mathcal{L}_{KL}^{NCKD}$ denotes the KL loss between teacher and student's probabilities of the non-target classes, named non-target class knowledge distillation (NCKD). The derivation can also be found in (B. Zhao et al., 2022).

Thus, the entire DKD loss function is

$$\mathcal{L}_{DKD} = (1-\alpha)\mathcal{L}_{CE}(y, p^{\mathcal{S}}) + \alpha T^2(\beta \mathcal{L}_{KL}^{TCKD} + \gamma \mathcal{L}_{KL}^{NCKD}), \tag{6.8}$$

where p_t and $p_{/t}$ are merged into two hyperparameters β and γ to coordinate the contributions of two terms. In bearing fault diagnosis, encouraging TCKD while suppressing NCKD ($\beta > \gamma$) often leads to performance improvement. By optimizing this decoupled loss, the knowledge acquired by the teacher model is more easily imparted into the student model, thereby improving the student network's performance.

6.2.2 FPGA development workflow

The PYNQ framework can directly convert the Python code into FPGA bitstream files, but it is specifically designed for the architecture of a System-on-Chip (SoC) FPGA. This type of FPGA integrates an ARM processor and an FPGA chip into a single package. However, the FPGA in such an SoC has only 50k look-up table (LUT), while other FPGAs have over 200k LUT. As a result, the computation speed of the SOC FPGA is much slower (Ji et al., 2022; Toumi et al., 2022) than other FPGAs.

In contrast, without bells and whistles, we design a simple yet pragmatic CNN acceleration scheme that directly translates the network computation modules into logic circuitry, which directly deploys the module into FPGA chips, without the need for ARM chip translation, thereby fully leveraging the

FPGA's high-speed computing capabilities. Specifically, we convert the operations of BearingPGA-Net into hardware description language in Verilog and optimize the place and route (P&R) of the logical connection netlist, such as logic gates, RAM, and flip-flops, for hardware acceleration. The FPGA simulation and deployment are performed using Vivado 2018.3, an FPGA design suite developed by Xilinx. The entire process includes register transfer level (RTL) design, simulation and verification, synthesis, implementation (P&R), editing timing constraints, generation of bitstream files, and the FPGA configuration. Notably, our scheme is also scalable to other FPGA chips such as Virtex-7 series.

6.2.2.1 Fixed-point quantization

The original network in Python employs the 32-bit floating-point numbers for high-accuracy calculations. However, these 32-bit numbers consume a significant amount of memory, which cannot allow the deployment of single-layer networks. To address this, we employ fixed-point quantization, which converts the numbers to a 16-bit fixed-point format. In this format, the bit-width remains intact for integers, decimals, and symbols in a floating-point number. The reduction in precision is up to half of the original, despite a minor decrease in performance. The fixed-point number is expressed as

$$Q_{X,Y} = \pm \sum_{i=0}^{X-1} b_i \cdot 2^i + \sum_{j=1}^{Y} b_{X+j} \cdot 2^{-j}. \tag{6.9}$$

where $Q_{X,Y}$ is a fixed-point number that contains X-bit integers and Y-bit decimals, b_i and b_{X+j} are the value on the binary bit. The storage format of fixed-point numbers in an FPGA is $(\pm, b_i, \cdot, b_{X+j})$.

Furthermore, when considering a fixed bit-width, it is crucial to determine the trade-off between the bit-widths of integers and decimals. It is important to have a sufficiently large range to minimize the probability of overflow while ensuring a small enough resolution to reduce quantization error (D. Lin, Talathi, & Annapureddy, 2016). In this study, a dynamic fixed-point quantization is assigned to each layer of the BearingPGA-Net to preserve the overall accuracy (Abdelouahab et al., 2018; Williamson, 1991). The specific bit-width for integers and decimals is determined based on the maximum and minimum values of parameters in each layer, so different models may have various bit-width. The allocation of bit-width of integers and decimals in each layer is illustrated in Figure 6.2.

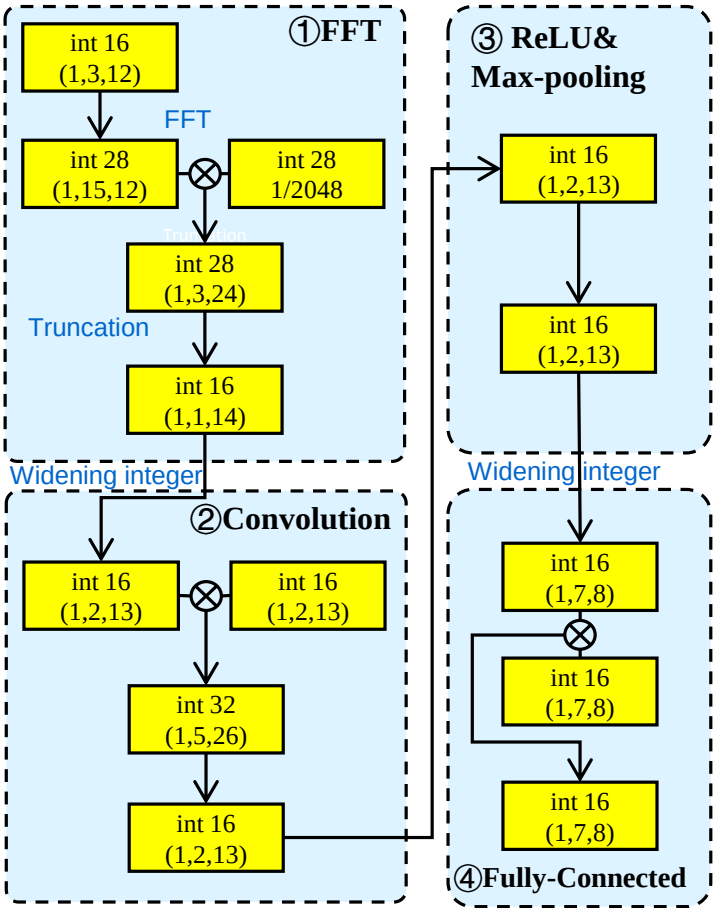


Figure 6.2: An example of bit-width of integers and decimals in each layer of BearingPGA-Net on FPGA computation, where (S, X, Y) denotes the number of symbol bit, integer bits, and decimal bits, respectively.

6.2.2.2 The overall workflow

As shown in Figure 6.3. Parameters are stored in the ROM after quantization, and we implement the BearingPGA-Net accelerator (FFT, convolutional layer, max-pooling conjugated with ReLU, fully-connected layer) using Verilog. The FFT operation is performed by the IP core, and we carefully design logic circuit modules for all operations of the BearingPGA-Net to carry out. Moreover, some specific modules used in FPGAs are described below:

(1) Receiving filter (RF) selector. It splits the input signal into 128 segments of 64 points with a stride of 8, which is equivalent to sliding a convolutional kernel of 1×64 with a stride of 8 over the input signal.

(2) ROM control (Ctrl). It retrieves weight parameters in ROM for the convolutional and fully-connected layers. For the convolutional layer, it reads 256 convolutional weight parameters (4×64 kernels) once a time and 10 weight parameters per cycle to update the input of the multiplication-accumulation module for the fully-connected layer. In addition, the FPGA directly stores bias param-

eters (4 for the convolutional layer and 10 for the fully-connected layer) in registers.

(3) Shift module. It shifts the position of the decimal point, which converts the bit-width of integers and decimals from (2,13) to (7,8) to adjust the fixed-point numbers in the downstream fully-connected layer.

(4) Classification module. It selects the maximum value among 10 outputs of the fully-connected layer as the failure category of the bearing. Softmax is waived in FPGA deployment.

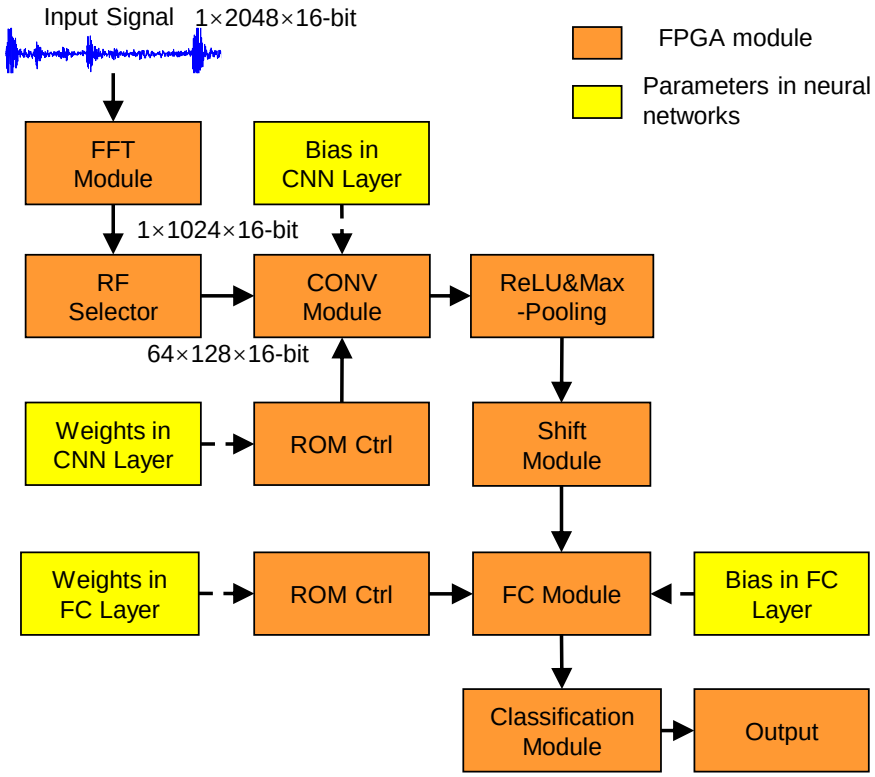


Figure 6.3: The overall diagram for deploying the BearingPGA-Net into FPGA.

6.2.2.3 BearingPGA-Net acceleration scheme

CNN accelerator designs can be generally classified into two groups: computing acceleration and memory system optimization (Qiu et al., 2016). Since our BearingPGA-Net consists of only one convolutional layer and one fully-connected layer, memory management does not require additional optimization. We focus on optimizing the FPGA acceleration of each layer in BearingPGA-Net.

(1) Multiplication-accumulation unit. As FPGA registers cannot directly perform matrix calculations, we design the multiplication-accumulation unit beforehand. The logic circuit diagram of the multiplication-accumulation unit is shown in Figure 6.4, where the accumulation operation is executed in N cycles, and the result register is used to store the temporary results in each cycle. The expression

of the multiplication-accumulation unit is

$$y = \sum_i^N (p_i \times q_i), \tag{6.10}$$

where p_i and q_i are 16-bit fixed-point numbers.

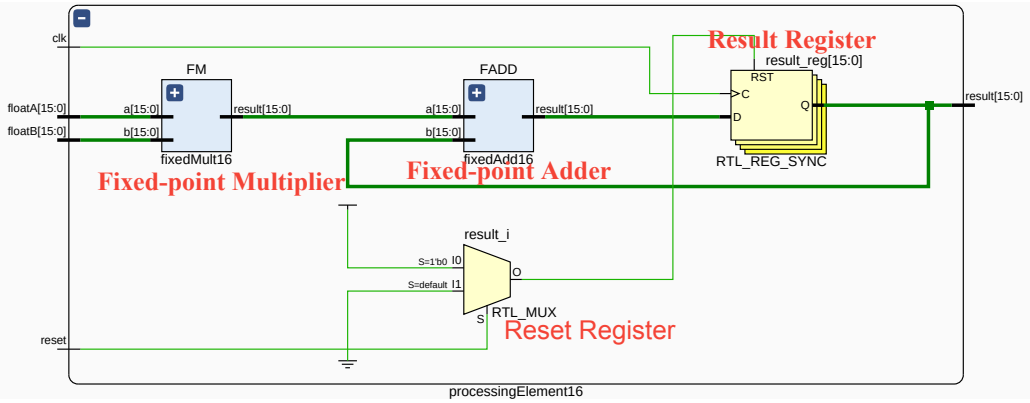


Figure 6.4: The circuit diagram of a multiplication-accumulation unit, which includes a fixed-point multiplier, a fixed-point adder, a reset register, and a result register.

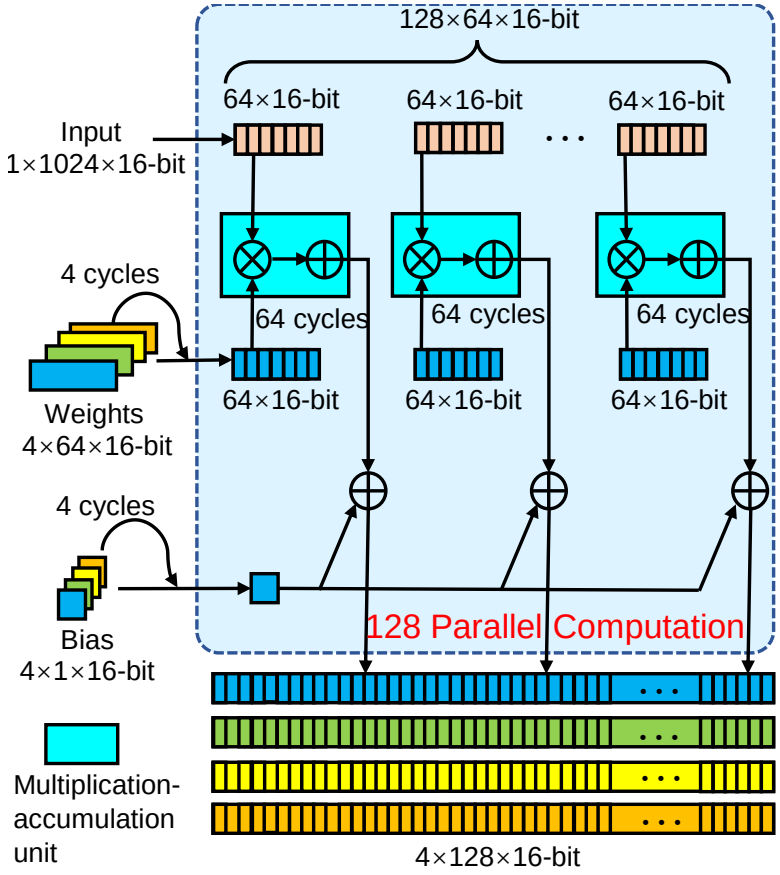


Figure 6.5: The implementation diagram of the convolutional layer.

(2) Convolutional layer. Figure 6.5 illustrates the implementation of the convolution layer. The signal after passing through the RF selector is divided into segments of 128×64. Subsequently, the

multiplication-accumulation unit uses weights $w \in \mathbb{R}^{1\times 64}$ stored in the ROM to multiply the signal $s_i \in \mathbb{R}^{1\times 64}$. Each multiplication-accumulation unit runs for 64 cycles and adds the bias b. The equation for this operation is $z_i = \sum_{j=1}^{64}(s_{i,j} \times w_j) + b$. Moreover, hardware acceleration is achieved using 128 parallel multiplication-accumulation units. This enables each convolutional kernel to complete the full convolution in just 64 cycles. The parallel output is merged into $\boldsymbol{z} = \{z_1, z_2, \cdots, z_{128}\}$. With 4 convolutional kernels, the layer requires 4 external cycles, totaling 256 cycles. The speedup is 128-fold relative to conventional implementations.

(3) ReLU and max-pooling layer. ReLU activation and max-pooling are back-to-back in the BearingPGA-Net. Also, both ReLU and max-pooling functions attempt to find the maximum value between two numbers. Therefore, we fuse them into a single module to save inference time. Suppose that the input signal is $\boldsymbol{x} \in \mathbb{R}^{1\times n}$, the ReLU activation function is

$$p_i = \max(0, x_i),\ i = 1, 2, \cdots, n, \tag{6.11}$$

followed by the max-pooling layer, where the window size is k and the stride is s. The output of this layer is

$$q_i = \max(\{p_j\}_{j=i\times s}^{i\times s+k-1}),\ i = 1, 2, \cdots, \lfloor\frac{n-k}{s}\rfloor + 1, \tag{6.12}$$

where $\lfloor\cdot\rfloor$ is the floor function to ensure the last pooling window is fully contained within the signal length.

We devise a strategy to reduce the amount of computation, as Algorithm 1 shows. Two 16-bit numbers, denoted as x_1 and x_2, are first compared in terms of their symbol bit. If both numbers are smaller than 0, the output is set to 0, thus executing the ReLU function directly. On the other hand, if the numbers have opposite signs, the negative number is killed. These operations can save a maximum comparison. Numerical comparisons are only performed when both numbers are positive, in which case the complete ReLU and max-pooling operations are executed. By adopting our method, we are able to save approximately $2/3$ of LUT resources.

Algorithm 1 ReLU and Max-pooling

Require: x_1, x_2 ($x[15]$ symbol bit, $x[14:0]$ number bits)

if $x_1[15] > 0$ AND $x_2[15] < 0$ **then**

  return x_1

else if $x_1[15] < 0$ AND $x_2[15] > 0$ **then**

  return x_2

else if $x_1[15] < 0$ AND $x_2[15] < 0$ **then**

  return 0

else if $x_1[15] > 0$ AND $x_2[15] > 0$ **then**

  if $x_1[14:0] >= x_2[14:0]$ **then**

    return x_1

  else if $x_1[14:0] < x_2[14:0]$ **then**

    return x_2

  end if

end if

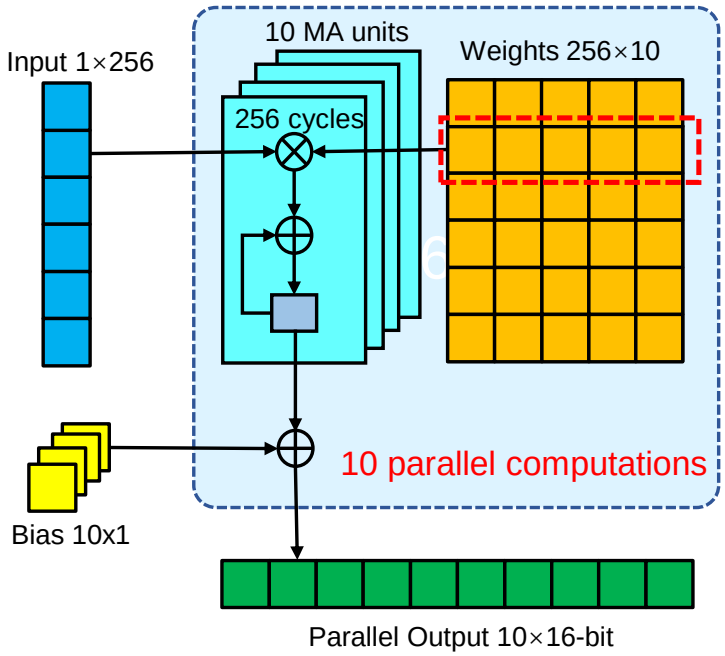


Figure 6.6: The implementation diagram of the fully-connected layer.

(4) Fully-connected layer. The implementation for this design is illustrated in Figure 6.6. The fully-connected layer establishes connections between the 256 outputs of the ReLU and max-pooling fusion layers and the final 10 outputs. Unlike on computers, the fully-connected layer in FPGAs consumes fewer resources, as there is no need to slide windows. For the fully-connected layer, we reuse multiplication-accumulation units. Initially, the size of weights in the fully-connected layer is

256×10. To simplify the computations, we divide this matrix into 256 segments, each consisting of 10 weights. Consequently, 10 multiplication-accumulation units perform parallel computation over 256 cycles to generate the output.

In summary, our FPGA accelerator offers two main advantages: i) Parallelism. Multiplication-accumulation units are simultaneously calculated, which greatly boosts computational efficiency. ii) Module reuse. Due to the constraint of limited FPGA computing resources that cannot parallelize 1024 units concurrently, multiplication-accumulation units are reused to strike a balance between resources and speed.

6.3 Experiments

6.3.1 Experimental configuration

6.3.1.1 Datasets descriptions

Three datasets, CWRU, PU, and HIT datasets, are used in the experiments. Among them, descriptions of HIT and PU datasets are summarized in Chapter 2. The CWRU dataset and the specific settings of the PU dataset are illustrated as follows.

(1) Case Western Reserve University (CWRU) dataset. This widely-used dataset is curated by Case Western Reserve University Bearing Data Center, which consists of two deep groove ball bearings mounted on the fan-end (FE) and drive-end (DE) of the electric motor. Electro-discharge machining is used to inject single-point defects with diameters of 7, 14, and 21 mils into the outer race, inner race, and ball of both bearings, respectively. As a result, there are ten categories in this dataset, including nine types of faulty bearings and one healthy bearing. Specifically, the motor shaft is subjected to four levels of load (0HP, 1HP, 2HP, 3HP, where HP denotes horsepower), which slightly affects the motor speed (1797 r/min, 1772 r/min, 1750 r/min, 1730 r/min). Vibration data are collected at 12 kHz and 48 kHz, respectively. In this study, we analyze the vibration signals collected at 12 kHz on the DE side of the motor.

(2) Paderborn University (PU) dataset. In our experiments, we select the single point faults that resulted from accelerated lifetime tests as our datasets (refer to Table 6.2). It's important to note that this dataset includes only three classes: healthy, faulty at the outer race, and faulty at the inner race. Consequently, the outputs of all models are adjusted to three classes to fit this dataset.

Table 6.2: The real damage PU datasets used in our experiments.

Class	Dataset Code
Healthy	K001,K002,K003,K004,K005
Single point faulty at the outer race	KA04,KA15,KA22
Single point faulty at the inner race	KI16,KI18,KI21

6.3.1.2 Signal preprocessing

First, the raw long signal is divided into segments of 2,048 points and standardized. Next, the Fast Fourier Transform (FFT) is applied to convert the signal from the time domain to the frequency domain. Previous research has experimentally demonstrated that employing signal processing techniques such as FFT and wavelet transform, can enhance the performance of shallow neural networks (Z. Zhao et al., 2020). Because the characteristics of non-stationary vibration signals are more pronounced in the frequency or time-frequency domain than the time domain, this compensates for the constrained feature extraction capability of shallow networks.

Specifically, the starting point of the segment of 2,048 points is randomly chosen, and the stride is set to 28 to resample the raw signal. All classes of signals are sampled 1,000 times, resulting in a total of 10,000 samples. Then, the dataset is randomly divided into training, validation, and testing sets with a ratio of 2:1:1. Next, the Gaussian noise is added to the signals to simulate noise in the industrial fields and also evaluate different models' performance in noisy experiments. The signal-to-noise ratio (SNR) is calculated as $\text{SNR} = 10 \log 10(P_s/P_n)$, where P_s and P_n are the average power of the signal and noise, respectively. Lastly, the signals are standardized using z-score standardization. Notably, each signal $\boldsymbol{x}$ is transformed as $\boldsymbol{x}' = (\boldsymbol{x} - \mu)/\sigma$, where μ and σ are the mean and standard deviation of the signal $\boldsymbol{x}$, respectively. For the CWRU dataset, the SNR ranges from -6dB to 2dB in 2dB intervals. For our dataset, the SNR ranges from 0dB to 8dB in 2dB intervals.

6.3.1.3 Baseline methods

We compare our method against four state-of-the-art lightweight baselines: wide deep convolutional neural networks (WDCNN) (W. Zhang et al., 2017), lightweight efficient feature extraction networks (LEFE-Net) (H. Fang, Deng, Zhao, et al., 2021), lightweight transformers with convolutional embeddings and linear self-attention (CLFormer) (H. Fang, Deng, Bai, et al., 2021), and knowledge distillation-based student convolutional neural networks (KDSCNN) (Ji et al., 2022). All these mod-

els have been published in flagship journals of this field in recent years. As summarized in Table 6.3, our model enjoys the smallest model size, relatively low computational complexity, and the second shortest inference time compared to its competitors. Although our model has slightly higher computational complexity and inference time than KDSCNN due to the introduced FFT, it has a substantially smaller number of parameters (2.83K vs 5.89K), while the model size is the first priority for deployment on FPGAs.

Table 6.3: The properties of compared models. #FLOPs denotes floating point operations. Time is the elapsed time to infer 1000 samples on an NVIDIA Geforce GTX 3080 GPU.

Method	#Parameters	#FLOPs	Inference time
LEFE-Net	73.95K	14.7M	22.4540s
WDCNN	66.79K	1.61M	1.6327s
CLFormer	4.98K	0.63M	0.7697s
KDSCNN	5.89K	70.66K	0.2219s
BearingPGA-Net	2.83K	78.34K	0.6431s

6.3.1.4 Implementation settings

All experiments are conducted in Windows 10 with Intel(R) Core(TM) 11th Gen i5-1135G7 at 2.4GHz CPU and one NVIDIA RTX 3080 8GB GPU. Our code is written in Python 3.10 with PyTorch 2.1.0.

For all compared models, we use stochastic gradient descent (SGD) (Bottou, 2012) as the optimizer with momentum=0.9 and Cosine Annealing LR (Loshchilov & Hutter, 2016) as the learning rate scheduler. The training epochs are set to 75, and we use grid search to find the best hyperparameters. $[32, 64, 128, 256]$ is for the batch size, and $[10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 0.01, 0.03, 0.1, 0.3]$ is for the learning rate. In KD-based methods, we search T from $[1, 2, 3, 4, 5, 6, 7, 8, 9]$, and α from $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$, while in our model we have to additionally search β and γ. We configure $[0, 0.2, 0.5, 1, 2, 4]$ for β and $[1, 2, 4, 8, 10]$ for γ, respectively.

6.3.1.5 Evaluation metrics

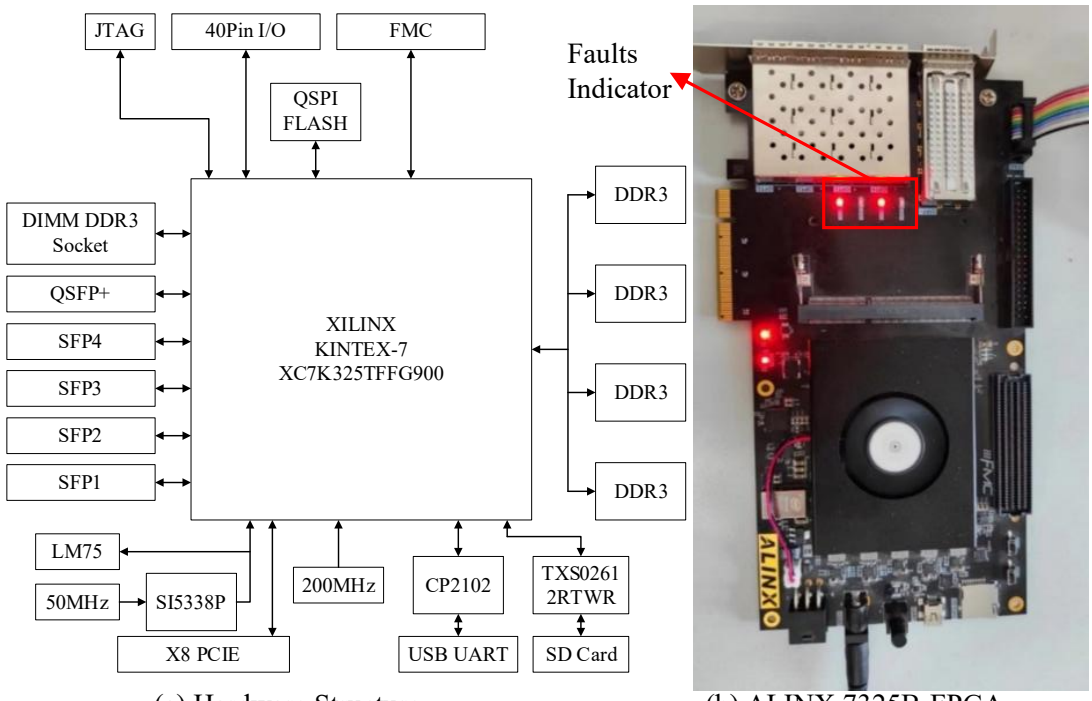
We use the F1 score to validate the performance of the proposed method. The metric is defined as

$$\text{F1 score} = \frac{1}{\text{C}} \sum_{i=1}^{\text{C}} \frac{2\text{TP}_i}{2\text{TP}_i + \text{FP}_i + \text{FN}_i},$$

where TP_i, FP_i, and FN_i are the numbers of true positives, false positives, and false negatives for the i-th class. C denotes the number of classes. All results are the average of ten runs.

6.3.1.6 FPGA description

We utilize the ALINX 7325B-FPGA as the hardware deployment board, as shown in Figure 6.7. This board incorporates the Kintex-7 XC7K325T FPGA chip, which is widely used in the industrial field due to its optimal balance between price and performance-per-watt. The clock frequency is set to 100MHz, and we utilize four LEDs as indicators of the fault diagnosis result, where ten categories are encoded as 4-bit binary numbers.



(a) Hardware Structure (b) ALINX 7325B-FPGA

Figure 6.7: The hardware architecture and physical diagram of the FPGA where BearingPGA is deployed.

6.3.2 Computational experiments

6.3.2.1 Experiments on three datasets

On the CWRU dataset, as shown in Table 6.4, BearingPGA-Net achieves the top F1 scores in 0HP and 1HP conditions and the second-best scores in 2HP and 3HP conditions. Moreover, our method maintains over 95% F1 score at the -6dB noise level, whereas the KDSCNN performs the worst on the CWRU-0HP dataset at -6dB noise (79.95%). Although WDCNN performs comparably, its number

of parameters is approximately 22 times larger than ours. These results demonstrate BearingPGA's strong diagnosis performance despite its small parameter size.

Table 6.4: The F1 score (%) of all models on the CWRU dataset. The bold-faced number is the best, and the underline number is the second best.

Datasets	Model	-6dB	-4dB	-2dB	0dB	2dB	Clean	Average
CWRU-0HP	LEFE-Net	86.62±3.99	94.34±2.42	98.69±1.33	98.89±1.14	98.96±1.05	99.87±0.10	98.15
	WDCNN	95.44±2.21	**98.33±1.11**	98.52±1.05	99.04±0.63	99.00±0.32	**100.00±0.00**	98.39
	CLFormer	85.27±3.71	89.84±3.13	96.11±3.31	98.75±1.32	99.32±0.23	99.46±0.13	94.79
	KDSCNN	79.95±4.35	84.90±4.01	87.57±3.53	93.55±2.36	96.76±2.41	98.25±1.70	90.16
	BearingPGA-Net	**97.20+2.16**	98.23±2.07	**99.20±0.01**	**99.89±0.11**	**99.98±0.01**	**100.00±0.00**	**99.08**
CWRU-1HP	LEFE-Net	75.95±3.41	82.42±2.33	92.40±2.01	97.62±2.36	97.51±2.43	99.24±0.45	90.86
	WDCNN	92.24±2.35	93.74±2.56	**97.88±2.10**	97.64±2.07	97.48±2.33	98.47±1.29	96.24
	CLFormer	65.74±3.75	70.94±3.11	88.22±3.14	90.16±1.65	95.90±1.01	99.02±0.89	85.00
	KDSCNN	85.50±3.36	87.76±3.51	89.43±2.27	91.99±2.33	96.52±1.11	97.65±1.38	91.48
	BearingPGA-Net	**96.85±1.11**	**97.56±1.14**	97.75±1.71	**97.96±2.64**	**98.08±1.85**	**99.97±0.01**	**98.03**
CWRU-2HP	LEFE-Net	91.56±1.14	97.68±2.31	99.96±0.04	99.84±0.19	99.94±0.06	99.93±0.02	98.15
	WDCNN	**99.40±0.01**	**99.99±0.01**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**99.90**
	CLFormer	88.47±3.85	89.84±3.98	97.31±1.59	99.84±0.05	99.85±0.14	99.92±0.03	95.87
	KDSCNN	94.68±2.92	97.56±1.41	98.23±1.63	98.60±1.04	98.98±1.01	99.10±0.84	97.86
	BearingPGA-Net	98.66±1.01	99.56±0.33	99.77±0.14	99.94±0.01	**100.00±0.00**	**100.00±0.00**	99.66
CWRU-3HP	LEFE-Net	89.65±3.39	95.13±1.15	96.87±3.11	99.56±0.41	**99.99±0.01**	99.97±0.39	96.86
	WDCNN	**95.83±1.31**	98.65±1.31	**99.84±0.17**	**99.96±0.02**	**99.99±0.01**	**100.00±0.00**	**99.05**
	CLFormer	87.59±3.13	89.53±3.11	97.19±1.11	98.92±0.53	99.28±0.27	99.92±0.01	95.41
	KDSCNN	83.15±3.95	86.58±2.99	95.95±1.33	96.54±1.63	98.17±1.64	99.71±0.14	93.35
	BearingPGA-Net	95.47±1.39	**98.93±1.03**	99.15±0.61	99.68±0.31	99.78±0.22	**100.00±0.00**	98.84

On the HIT dataset, as Table 6.5 shows, our method achieves the highest average F1 score in noisy scenarios. Although LEFE-Net slightly outperforms ours on clean signals, the difference is only 1%, and LEFE-Net has a substantially larger model size. Furthermore, the performance of other models (CLFormer: 76.37%, KDSCNN: 74.98%) falls behind ours (83.30%) by a large margin. Overall, our model is the best performer on this dataset.

Table 6.5: The F1 score (%) of all compared methods on the HIT dataset. The clean denotes the raw signal without noise, the bold-face number is the best.

SNR	LEFE-Net	WDCNN	CLFormer	KDSCNN	Ours
0	58.71	58.53	54.61	64.78	**70.51**
2	71.27	68.92	64.88	70.02	**72.64**
4	79.30	78.96	77.26	73.64	**82.46**
6	83.31	84.60	78.38	76.62	**85.53**
8	86.28	90.92	89.91	79.10	**91.54**
Clean	**98.96**	94.35	93.15	85.72	97.39
Average	79.64	79.38	76.37	74.98	**83.35**

On the PU dataset, as presented in Table 6.6. Our method delivers the best and second-best outcomes on the Paderborn dataset, with a difference of roughly 1% compared to WDCNN. Notably, our

method outperforms other lightweight methods. For instance, compared to the same type of KD-based method (KDSCNN), our method shows around 1.5% improvements in these four conditions, and our structure is only half the size (2.83K vs 5.89K). The results indicate that BearingPGA can achieve commendable performance in diagnosing real bearing damage.

Table 6.6: The F1 scores (%) of all models on the PU dataset. The bold-faced number is the best, and the underlined number is the second best.

Method	N15M01F10	N09M07F10	N15M07F10	N15M07F04
LEFE-Net	96.48+1.35	95.45±1.12	97.42±0.98	97.66±1.22
WDCNN	**98.80±0.55**	**97.45±1.30**	98.15±0.46	97.99±1.21
CLFormer	96.12±2.25	95.94±2.14	96.87±1.67	98.12±1.02
KDSCNN	94.44±2.43	95.37±2.32	97.12±2.13	96.98±1.78
BearingPGA-Net	97.42±1.05	96.09±1.53	**98.15±0.26**	**98.02±0.38**

6.3.2.2 Model compression results

(1) Compression performance. We compare the compression performance of DKD. Firstly, Table 6.7 shows the properties of teacher and student models. DKD effectively compresses the teacher model, reducing model parameters by 17 times, FLOPs by 10.6 times, and inference time by 2.45 times.

Table 6.7: The properties of teacher and student models. #FLOPs denotes floating point operations, lower is better. Time is the elapsed time to infer 1000 samples.

Model	#Parameters	#FLOPS	Inference Time
Teacher	50.09K	0.83M	1.5703s
Student	2.83K	78.34K	0.6431s

Secondly, Table 6.8 compares the performance of the models. Clearly, DKD enhances the performance of student model, though still slightly below the teacher (with an average F1 score approximately 1% lower). Comparing the student model with and without DKD, distillation provides considerable improvements, especially in noisy environments. On the CWRU-3HP, DKD improves the average F1 score by 3.7%. Notably, DKD admits 8.23% and 8.46% gains at -6dB on CWRU-2HP and 0dB on the HIT dataset, respectively. These results demonstrate DKD can successfully transfer knowledge to shallow models, despite substantial compression.

Table 6.8: The average F1 score (%) of teacher and student models and student models without decoupled knowledge distillation.

Datasets	SNR	Teacher	Student	Student w/o KD
CWRU-0HP	-6	98.57	97.20	93.20
	-4	99.69	98.23	99.04
	-2	99.79	99.20	99.07
	0	99.53	99.89	99.67
	2	100.00	99.98	99.64
	Avg.	99.52	98.83	98.12
CWRU-1HP	-6	97.21	96.85	88.62
	-4	97.92	97.56	92.62
	-2	97.73	97.75	95.77
	0	98.20	97.96	96.53
	2	99.41	98.08	96.78
	Avg.	98.09	97.64	94.06
CWRU-2HP	-6	99.42	98.66	96.45
	-4	99.89	99.56	98.85
	-2	99.64	99.77	99.85
	0	100.00	99.94	99.78
	2	100.00	100.00	99.99
	Avg.	99.79	99.59	98.98
CWRU-3HP	-6	97.97	95.47	91.62
	-4	98.98	98.93	94.22
	-2	99.62	99.15	95.28
	0	99.72	99.68	96.71
	2	100.00	99.78	97.06
	Avg.	99.26	98.60	94.98
HIT	0	71.36	70.51	62.05
	2	73.77	72.64	72.68
	4	84.15	82.46	80.92
	6	85.66	85.54	85.10
	8	93.12	91.54	90.67
	Avg.	81.61	80.54	78.28

(2) Ablation study. The ablation experiments investigate the roles of TCKD and NCKD in decoupled knowledge distillation. The experiments are divided into classical KD, TCKD performed independently, NCKD performed independently, and DKD. Note that DKD involves two loss functions, with $\beta = 4$ and $\gamma = 1$, while for the independent TCKD or NCKD, one hyperparameter is assigned a value of 1 and the other is set to 0.

As shown in Table 6.9, the performance of classical KD is superior to that of two independent loss functions, and the F1 score of independent TCKD is comparable to that of classical KD. In contrast, independent NCKD performs the worst. This phenomenon suggests that TCKD plays a major role

in the knowledge distillation loss function. Moreover, the DKD loss demonstrates the best results. It is noteworthy that the weight of TCKD is set to four times that of NCKD, indicating that focusing the attention of neural networks more on TCKD can further unleash the potential of logit knowledge distillation.

Table 6.9: The F1 score of different KD functions.

TCKD	NCKD	0HP(-6dB)	1HP(-6dB)	2HP(-6dB)	3HP(-6dB)	HIT(8dB)
—	—	94.32	91.00	97.62	92.14	82.15
✓	✗	94.31	90.22	97.57	92.11	82.15
✗	✓	88.80	91.38	96.03	89.36	75.81
✓	✓	**97.20**	**96.85**	**98.66**	**95.47**	**93.12**

(3) Hyperparameter sensitivity. Experiments are conducted to verify the hyperparameters in DKD, namely T, α, β, and γ. In this experiment, we set the learning rate to 0.1 and the batch size to 64. DKD's default hyperparameters are $T = 2.5, \alpha = 0.2, \beta = 4, \gamma = 1$. We then test the performance on the CWRU-0HP dataset at -6dB noise by changing one of these hyperparameters.

The results are presented in Table 6.10. Firstly, the parameter of temperature (T) does not appear to be highly influential. While a larger value of T leads to slightly better performance, the improvement is only about 1%. Secondly, the analysis shows that all three scale parameters have a significant impact. In particular, setting α to 0.2 yields superior results. Once α surpasses 0.3, there is a noticeable decline in performance, with approximately 64% drop in F1 score. Unlike α, β exhibits less sensitivity, but increasing its value still leads to improved results. Conversely, a lower value (below 2) is recommended for the parameter γ.

Table 6.10: The hyperparameter sensitivity in DKD on the CWRU-0HP dataset with -6dB noise.

T	1.5	2	2.5	3	3.5
F1(%)	95.28	94.85	95.20	**96.19**	96.18
α	0.1	0.2	0.3	0.4	0.5
F1(%)	92.49	**96.19**	32.13	32.25	32.19
β	0.2	0.5	1	2	4
F1(%)	89.49	91.08	92.84	95.07	**96.20**
γ	0.2	0.5	1	2	4
F1(%)	**96.41**	95.68	96.19	96.30	32.09

In addition, we observe an intriguing phenomenon that the performance of BearingPGA-Net im-

proves as γ decreases. However, when $\gamma = 0$ (in the absence of NCKD), the performance appears to deteriorate. To investigate this further, we conduct a series of experiments with other hyperparameters keeping the status quo ($\alpha = 0.2, \beta = 5$, and $T = 3$) while varying the value of γ from 1 to 0.0002. The results of these experiments are depicted in Figure 6.8. The performance exhibits a significant degradation when γ falls within the intervals [0.001,0.05] and [0.0005,0.0008]. This suggests that although there is no absolute cut-off boundary for γ, the performance becomes notably unstable when γ is less than 0.1. We leave explaining the behavior of DKD at the paradigm of $\gamma < 0.1$ as our future work.

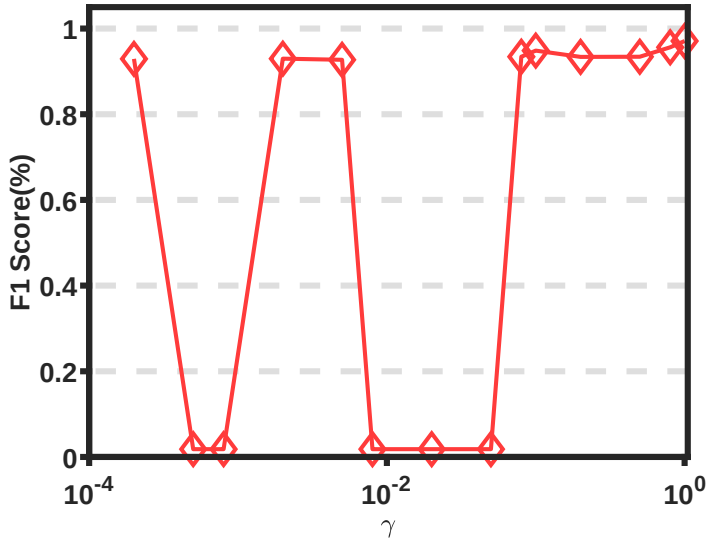


Figure 6.8: The F1 score(%) obtained by varying γ and fixing other hyperparameters in DKD.

The aforementioned phenomenon exemplifies the characteristics of DKD. Regarding the parameter α, it is utilized to regulate both $\mathcal{L}_{CE}$ and $\mathcal{L}_{KL}$ loss. A lower value of α implies that the student network predominantly learns from the ground truth, while any knowledge transmitted by the teacher model serves merely as supplementary information. Next, β and γ are employed to control the TCKD and NCKD losses, respectively. Consistent with the findings of the ablation experiments, a larger value for β and a smaller value for γ yield optimal results, as they compel the model to effectively harness the knowledge pertaining to the target classes.

6.3.2.3 Classification results on FPGA

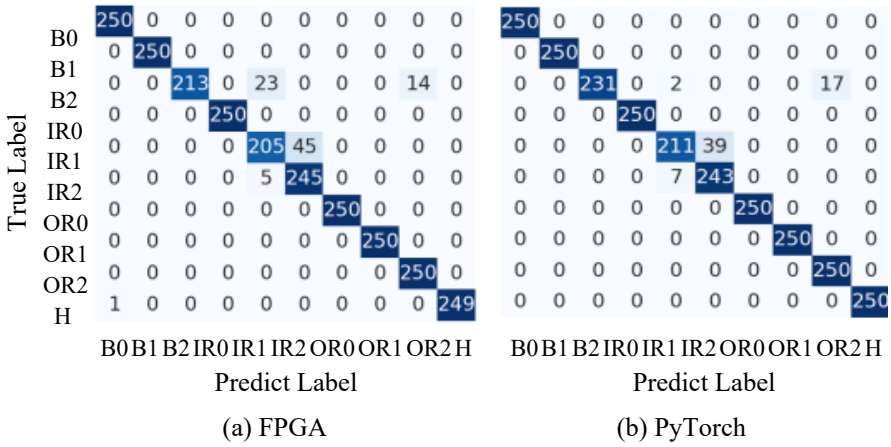
We utilize the HIT dataset as the input to compare the performance of the 32-bit float PyTorch and the 16-bit fixed-point FPGA implementation. Table 6.11 reveals that the quantized model demonstrates a lower-than-0.4% performance drop in terms of F1, Recall, and Precision scores relative to the original

model. Additionally, the parameters in the quantized model are reduced by more than half. These findings highlight the effectiveness of the designed quantization method and convolution architecture for FPGA deployment.

Table 6.11: Comparison of 32-bit PyTorch and 16-bit FPGA models on HIT dataset.

	#Parameters	F1 (%)	Recall (%)	Precision (%)
PyTorch	2.83K	97.39	97.40	97.57
FPGA	1.42K	97.12	97.34	97.12

Furthermore, the confusion matrices between the FPGA and PyTorch models are compared. As illustrated in Figure 6.9, the FPGA model exhibits > 18 errors in classifying ball faults (B0-B2), < 4 errors for inner race faults (IR0-IR2), and 1 additional error in healthy bearings, compared to PyTorch results. The BearingPGA-Net when deployed on FPGA only generates a total of 24 extra errors compared to the original version across 2500 samples.



(a) FPGA (b) PyTorch

Figure 6.9: The confusion matrices of results produced by models before and after quantization on the HIT dataset, where B, IR, OR denote faults at the ball, the inner race, and the outer race, respectively, and H denotes the healthy bearing.

6.3.2.4 Resource analysis

The resource usage for the entire network is summarized in Table 6.12. The LUT resource and block RAM (BRAM) resource exhibit utilization rates of 74.40% and 58.20%, respectively, indicating that BearingPGA-Net makes good use of the FPGA's computational and memory resources. Next, we analyze the resource consumption specifically within the LUT, which is the key computational resource in an FPGA. As illustrated in Figure 6.10, the convolution and FFT modules account for the

majority of resources in the LUT, constituting approximately 50% and 23%, respectively. This observation underscores the limited computational resources available on a Kintex-7 FPGA, *i.e.*, a single convolutional layer alone consumes a half LUT resource.

Table 6.12: The resource report of Kintex-7 XC7K325T FPGA.

Resources	Used resources	Available resources	Resource occupancy
LUT	151637	203800	74.40%
LUTRAM	936	64000	1.46%
FF	180099	407600	44.19%
BRAM	259	445	58.20%
DSP	185	840	22.02%
IO	6	500	1.20%
BUFG	3	32	9.38%
PLL	1	10	10.00%

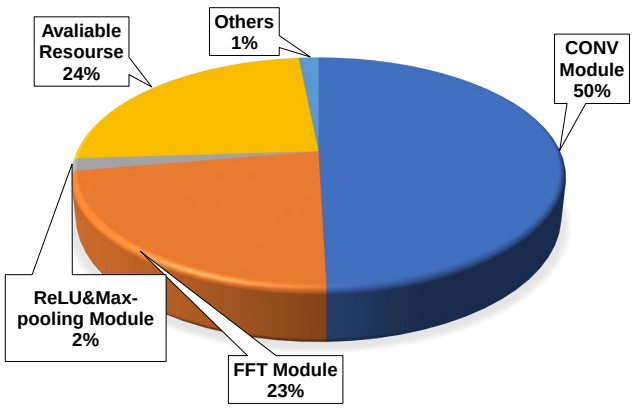


Figure 6.10: The LUT consumption with respect to each module.

6.3.3 Online test

6.3.3.1 Experimental setup

We conduct online bearing fault tests to validate the performance of our FPGA-based bearing fault diagnosis system. As illustrated in Figure 6.11, the entire system attaches the test rig. The bearings are monitored using an acceleration meter, and the collected vibration signals are transmitted to the FPGA through a signal conditioner and AD converter for real-time diagnosis. After the electric motor runs for 1s, we activate the FPGA to process and record the diagnostic results. BearingPGA-Net presents fault diagnosis results via four LEDs.

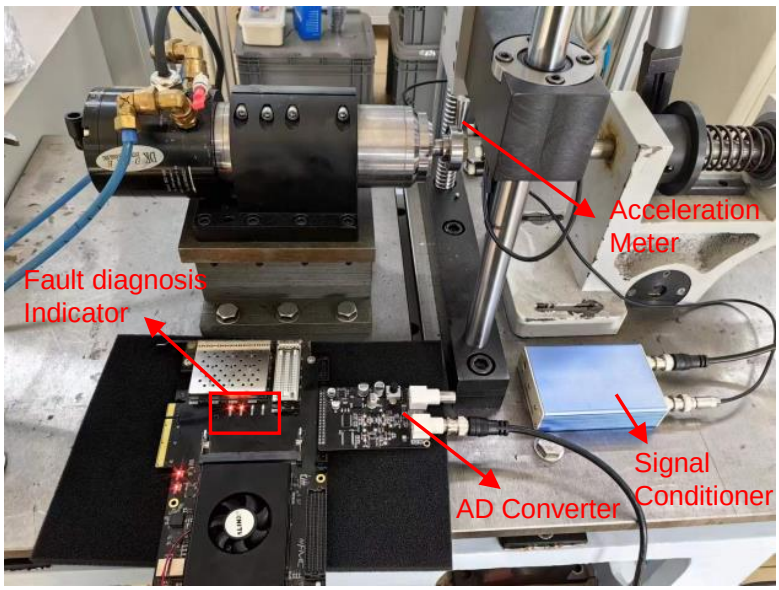


Figure 6.11: The online test of BearingPGA-Net.

6.3.3.2 Online test results

Given each bearing, the test is repeated ten times. The confusion matrix for 100 runs is presented in Figure 6.12. Primarily, it is observed that the majority of faults are accurately identified, confirming the efficacy of our FPGA system in performing online diagnosis. The errors only occur in the diagnosis of ball faults, where two minor ball faults were misclassified as moderate, and one moderate fault was falsely identified as severe. This can be attributed to the fact that ball faults are accompanied by random slips, which do not exhibit strictly periodic fault characteristics, thereby adding a layer of difficulty (Randall & Antoni, 2011). Finally, the macro F1 score, Precision, and Recall for this test are 96.98%, 97.27%, and 97.00%, respectively, demonstrating a high degree of consistency with our test results derived from the HIT dataset.

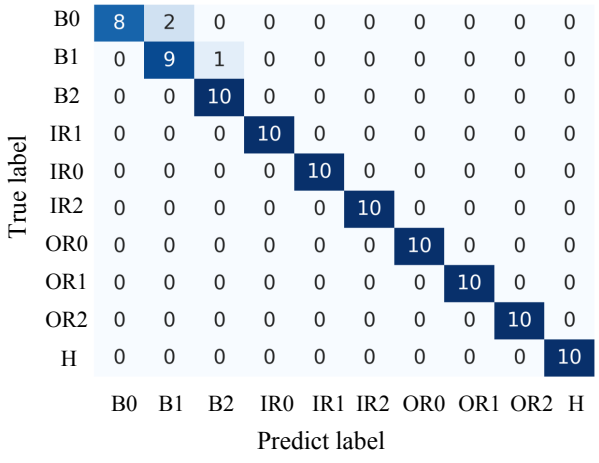


Figure 6.12: The confusion matrix of online test.

6.4 Summary

While BearingPGA-Net demonstrates promising performance on embedded hardware, there are still several limitations that warrant future exploration. The generalization capability of such a simple convolutional network is inherently limited. Enhancing the generalization performance necessitates the incorporation of additional network branches and loss functions to facilitate transfer learning, which in turn requires more complex network structures (Weiss, Khoshgoftaar, & Wang, 2016). Consequently, BearingPGA-Net needs to undergo meticulous fine-tuning for specific machinery applications.

In this chapter, we have proposed a lightweight and deployable CNN, referred to as BearingPGA-Net. By integrating FFT and DKD, BearingPGA-Net outperforms other state-of-the-art lightweight models when signals are noisy. Additionally, we have fully utilized the computational resources of the Kintex-7 FPGA and designed CNN accelerating scheme. The parallelism and model reuse significantly improves the running speed of our method, while the customized parameter quantization greatly alleviates the performance drop. Our deployment scheme has achieved over 200 times faster diagnosis speed compared to CPU and maintained over 97% classification F1 score on the HIT bearing dataset. Finally, our model introduces a new approach to realizing online bearing fault diagnosis.

Chapter 7

Conclusions and Future Works

7.1 Conclusions

This dissertation has endeavored to advance the field of deep learning-based intelligent prognostics and health management (PHM) of rolling bearings by leveraging signal processing techniques. Through a comprehensive review of the limitations of contemporary PHM approaches, we have developed a suite of signal processing-empowered neural networks to address key challenges in PHM, including intepretability, generalizability, reliability, and deployability. The proposed methodologies have demonstrated significant performance enhancements in the realm of machinery PHM, showcasing promising potential for real-world applications.

From the perspective of bearing fault diagnosis, our proposed methodologies, built upon blind deconvolution and quadratic neural networks, have effectively addressed the several challenges of supervised bearing fault diagnosis and unsupervised anomaly detection, both theoretically and practically.

- **Classifier-guided neural blind deconvolution:** We have introduced a novel approach termed as ClassBD for bearing fault diagnosis under heavy noisy conditions. ClassBD is composed of cascaded time and frequency neural BD filters, succeeded by a deep learning classifier. Specifically, the time BD filter incorporates quadratic convolutional neural networks (QCNN), and we have mathematically proved its superior capability in extracting periodic impulse features in the time domain. The frequency BD filter includes a fully-connected linear filter, supplementing the frequency domain filter subsequent to the time filter. Furthermore, a deep learning classifier is directly integrated to empower classification capabilities. We have devised a physics-

informed loss function composed of kurtosis, l_2/l_4 norm, and cross-entropy loss to facilitate the joint learning. This unified framework transforms traditional unsupervised BD into supervised learning, providing interpretability due to its retention of conventional BD operations. Finally, comprehensive experiments conducted on three public and private datasets reveal that ClassBD outperforms other state-of-the-art methods, achieving an average of 95% F1 scores on several bearing datasets under heavy noise.

- **Heterogeneous autoencoder:** We have theoretically shown that a heterogeneous network is more powerful and efficient by constructing a function that is easy to approximate for a one-hidden-layer heterogeneous network but difficult to approximate by a one-hidden-layer conventional or quadratic network. Moreover, we have proposed a novel type of heterogeneous autoencoders using quadratic and conventional neurons. Lastly, anomaly detection experiments have confirmed that heterogeneous autoencoders delivers 80% and 90% F1 scores on XJTU and SBP bearing datasets respectively, outperforming relative to homogeneous autoencoders and other baselines.

From the perspective of bearing RUL prediction, we have proposed a novel feature extraction approach with reliability estimation, which addresses the generalizability and reliability issues of deep learning RUL models.

- **Logarithmic cumulative transformation:** We have proposed a novel and effective feature transformation approach, the logarithmic cumulative transformation (LCT), employed as a signal preprocessing method. The LCT composed of three straightforward transformations yields features with good alignment in shape and scale, thereby enhancing the generalizability of downstream RUL estimations. Furthermore, we have devised a novel method to estimate the reliability associated with each RUL prediction. This method employs the differential slopes of the linear regression model to derive the reliability indicator. Computational results demonstrate strong generalizability and reliability of our method across diverse samples, achieving a lower 0.03 MAE in RUL prediction when the Reliability score is over 98%.

From the perspective of online fault diagnosis, we have constructed a lightweight neural network and have successfully deployed it into an FPGA hardware platform.

- **BearingPGA-Net:** We have proposed a lightweight bearing fault diagnosis network by integrating FFT and decoupled knowledge distillation. BearingPGA-Net outperforms other state-of-the-art lightweight models when signals are noisy. Additionally, we have fully utilized the computational resources of the Kintex-7 FPGA and designed a CNN accelerating scheme. The parallelism and model reuse significantly improve the running speed of our method, while the customized parameter quantization greatly alleviates the performance drop. Our deployment scheme has achieved over 200 times faster diagnosis speed compared to CPU and maintained over 97% classification F1 score on the HIT bearing dataset. Finally, our model introduces a new approach to realizing online bearing fault diagnosis.

7.2 Limitations and future works

Despite our efforts to address various key issues in PHM, there are still limitations to our work. Here, we summarize some of the limitations and highlight potential directions for future research.

Firstly, our methods have only been validated on bearing failures, but bearings are not the only components that can fail in rotating machinery. Diagnosing gears, drive shafts, and other complex components remains an open question. Future research should focus on extending the applicability of our proposed methodology to a wide variety of fault diagnosis scenarios.

Secondly, although the neural blind deconvolution paradigms have demonstrated effectiveness across various scenarios and tasks, the network structure and theoretical mechanism are not fully understood. Future work should aim at investigating the theoretical connection between the failure mechanism and signal processing-empowered neural networks.

Thirdly, for RUL prediction, our proposed method is a signal preprocessing approach, but it has only been validated on limited backbone networks and datasets. The generalizability of this method needs to be further investigated. Moreover, RUL uncertainty should be quantified through Bayesian extrapolation, enabling the determination of more trustworthy predictions in future work.

Lastly, the BearingPGA-Net is a promising step towards embedded deployment, but it is essential to further balance performance, model complexity, and power consumption on edge devices. Future research should focus on investigating more efficient model compression methods, finer hardware acceleration, and power balance strategies to enable deployment on resource-constrained devices.

Chapter 8

References

Abdelouahab, K., Pelcat, M., Serot, J., & Berry, F. (2018). Accelerating CNN inference on FPGAs: A survey. *arXiv preprint arXiv:1806.01683*.

Aggarwal, C. C., & Sathe, S. (2017). *Outlier ensembles: An introduction*. Springer.

Alexandridis, A. K., & Zapranis, A. D. (2013). Wavelet neural networks: A practical guide. *Neural Networks*, *42*, 1–27.

An, Y., Zhang, K., Chai, Y., Liu, Q., & Huang, X. (2023). Domain adaptation network base on contrastive learning for bearings fault diagnosis under variable working conditions. *Expert Systems with Applications*, *212*, 118802.

Antoni, J. (2006). The spectral kurtosis: A useful tool for characterising non-stationary signals. *Mechanical Systems and Signal Processing*, *20*(2), 282–307.

Antoni, J. (2007a). Cyclic spectral analysis in practice. *Mechanical Systems and Signal Processing*, *21*(2), 597–630.

Antoni, J. (2007b). Cyclic spectral analysis of rolling-element bearing signals: Facts and fictions. *Journal of Sound and vibration*, *304*(3-5), 497–529.

Antoni, J. (2016). The infogram: Entropic evidence of the signature of repetitive transients. *Mechanical Systems and Signal Processing*, *74*, 73–94.

Antoni, J., & Hanson, D. (2012). Detection of surface ships from interception of cyclostationary signature with the cyclic modulation coherence. *IEEE Journal of Oceanic Engineering*, *37*(3), 478–493.

Antoni, J., & Randall, R. B. (2006). The spectral kurtosis: Application to the vibratory surveillance and diagnostics of rotating machines. *Mechanical Systems and Signal Processing*, *20*(2), 308–331.

Antoni, J., Xin, G., & Hamzaoui, N. (2017). Fast computation of the spectral correlation. *Mechanical Systems and Signal Processing*, *92*, 248–277.

Antoni, J., Bonnardot, F., Raad, A., & El Badaoui, M. (2004). Cyclostationary modelling of rotating machine vibration signals. *Mechanical systems and signal processing*, *18*(6), 1285–1314.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50–58.

Babiloni, F., Marras, I., Deng, J., Kokkinos, F., Maggioni, M., Chrysos, G., Torr, P., & Zafeiriou, S. (2023). Linear complexity self-attention with 3rd order polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(11), 12726–12737.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Benbouzid, M. E. H. (2000). A review of induction motors signature analysis as a medium for faults detection. *IEEE transactions on industrial electronics*, *47*(5), 984–993.

Bonnardot, F., Randall, R., Antoni, J., & Guillet, F. (2004). Enhanced unsupervised noise cancellation using angular resampling for planetary bearing fault diagnosis. *International Journal of Acoustics and Vibration*, *9*(2), 51–60.

Bonnett, A. H., & Yung, C. (2008). Increased efficiency versus increased reliability. *IEEE Industry Applications Magazine*, *14*(1), 29–36.

Borghesani, P., & Antoni, J. (2018). A faster algorithm for the calculation of the fast spectral correlation. *Mechanical Systems and Signal Processing*, *111*, 113–118.

Borghesani, P. (2015). The envelope-based cyclic periodogram. *Mechanical Systems and Signal Processing*, *58*, 245–270.

Bottou, L. (2012). Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade* (pp. 421–436). Springer.

Boudiaf, A., Moussaoui, A., Dahane, A., & Atoui, I. (2016). A comparative study of various methods of bearing faults diagnosis using the Case Western Reserve University data. *Journal of Failure Analysis and Prevention*, *16*(2), 271–284.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 93–104.

Bu, J., & Karpatne, A. (2021). Quadratic residual networks: A new class of neural networks for solving forward and inverse problems in physics involving pdes. *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, 675–683.

Buzzoni, M., Antoni, J., & d'Elia, G. (2018). Blind deconvolution based on cyclostationarity maximization and its application to fault identification. *Journal of Sound and Vibration*, *432*, 569–601.

Cai, S., Zhang, J., Li, C., He, Z., & Wang, Z. (2024). A RUL prediction method of rolling bearings based on degradation detection and deep BiLSTM. *Electronic Research Archive*, *32*(5).

Campos, G. O., Zimek, A., Sander, J., Campello, R. J., Micenková, B., Schubert, E., Assent, I., & Houle, M. E. (2016). On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, *30*, 891–927.

Cao, L., Zhang, H., Meng, Z., & Wang, X. (2023). A parallel GRU with dual-stage attention mechanism model integrating uncertainty quantification for probabilistic RUL prediction of wind turbine bearings. *Reliability Engineering and System Safety*, *235*, 109197.

Cao, Y., Jia, M., Ding, P., Zhao, X., & Ding, Y. (2023). Incremental learning for remaining useful life prediction via temporal cascade broad learning system with newly acquired data. *IEEE Transactions on Industrial Informatics*, *19*(4), 6234–6245.

Chen, D., Qin, Y., Wang, Y., & Zhou, J. (2021). Health indicator construction by quadratic function-based deep convolutional auto-encoder and its application into bearing RUL prediction. *ISA Transactions*, *114*, 44–56.

Chen, J., Huang, R., Chen, Z., Mao, W., & Li, W. (2023). Transfer learning algorithms for bearing remaining useful life prediction: A comprehensive review from an industrial application perspective. *Mechanical Systems and Signal Processing*, *193*, 110239.

Chen, Q., Dong, X., Tu, G., Wang, D., Cheng, C., Zhao, B., & Peng, Z. (2024). TFN: An interpretable neural network with time-frequency transform embedded for intelligent fault diagnosis. *Mechanical Systems and Signal Processing*, *207*, 110952.

Chen, S., Dong, X., Peng, Z., Zhang, W., & Meng, G. (2017). Nonlinear chirp mode decomposition: A variational method. *IEEE Transactions on Signal Processing*, *65*(22), 6024–6037.

Chen, X., Yang, R., Xue, Y., Huang, M., Ferrero, R., & Wang, Z. (2023). Deep transfer learning for bearing fault diagnosis: A systematic review since 2016. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–21.

Chen, Y., Peng, G., Zhu, Z., & Li, S. (2020). A novel deep learning method based on attention mechanism for bearing remaining useful life prediction. *Applied Soft Computing*, *86*, 105919.

Cheng, H., Kong, X., Wang, Q., Ma, H., & Yang, S. (2022). The two-stage rul prediction across operation conditions using deep transfer learning and insufficient degradation data. *Reliability Engineering and System Safety*, *225*, 108581.

Cheng, W., Xie, S., Xing, J., Nie, Z., Chen, X., Liu, Y., Liu, X., Huang, Q., & Zhang, R. (2024). Interactive hybrid model for remaining useful life prediction with uncertainty quantification of bearing in nuclear circulating water pump. *IEEE Transactions on Industrial Informatics*, *20*(2), 2154–2166.

Cheng, Y., Chen, B., Mei, G., Wang, Z., & Zhang, W. (2019). A novel blind deconvolution method and its application to fault identification. *Journal of Sound and Vibration*, *460*, 114900.

Cheng, Y., Chen, B., & Zhang, W. (2019). Adaptive multipoint optimal minimum entropy deconvolution adjusted and application to fault diagnosis of rolling element bearings. *IEEE Sensors Journal*, *19*(24), 12153–12164.

Cheng, Y., Wang, Z., Zhang, W., & Huang, G. (2019). Particle swarm optimization algorithm to solve the deconvolution problem for rolling element bearing fault diagnosis. *ISA Transactions*, *90*, 244–267.

Cheng, Y., Zhou, N., Zhang, W., & Wang, Z. (2018). Application of an improved minimum entropy deconvolution method for railway rolling element bearing fault diagnosis. *Journal of Sound and Vibration*, *425*, 53–69.

Cheng, Y., Chrysos, G., Georgopoulos, M., & Cevher, V. (2024). Multilinear operator networks. *The Twelfth International Conference on Learning Representations*.

Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., & Jitsev, J. (2023). Reproducible scaling laws for contrastive language-image learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2818–2829.

Chi, L., Jiang, B., & Mu, Y. (2020). Fast Fourier convolution. *Advances in Neural Information Processing Systems*, *33*, 4479–4488.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.10*.

Choi, S., Scrofano, R., Prasanna, V. K., & Jang, J.-W. (2003). Energy-efficient signal processing using FPGAs. *Proceedings of the 2003 ACM/SIGDA Eleventh International Symposium on Field Programmable Gate Arrays*, 225–234.

Chrysos, G. G., Georgopoulos, M., Deng, J., Kossaifi, J., Panagakis, Y., & Anandkumar, A. (2022). Augmenting deep classifiers with polynomial neural networks. *European Conference on Computer Vision*, 692–716.

Chrysos, G. G., Wang, B., Deng, J., & Cevher, V. (2023). Regularization of polynomial networks for image recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16123–16132.

Compare, M., Baraldi, P., & Zio, E. (2020). Challenges to IoT-enabled predictive maintenance for industry 4.0. *IEEE Internet of Things Journal*, *7*(5), 4585–4597.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*, 273–297.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *20*(2), 215–232.

Da Costa, K. A., Papa, J. P., Lisboa, C. O., Munoz, R., & De Albuquerque, V. H. C. (2019). Internet of Things: A survey on machine learning-based intrusion detection approaches. *Computer Networks*, *151*, 147–157.

Debao, C. (1993). Degree of approximation by superpositions of a sigmoidal function. *Approximation Theory and Its Applications*, *9*(3), 17–28.

Dewangan, G., & Maurya, S. (2022). Fault diagnosis of machines using deep convolutional beta-variational autoencoder. *IEEE Transactions on Artificial Intelligence*, *3*(2), 287–296.

Dick, C., & Harris, F. (2000). FPGA signal processing using sigma-delta modulation. *IEEE Signal Processing Magazine*, *17*(1), 20–35.

Ding, A., Qin, Y., Wang, B., Guo, L., Jia, L., & Cheng, X. (2024). Evolvable graph neural network for system-level incremental fault diagnosis of train transmission systems. *Mechanical Systems and Signal Processing*, *210*, 111175.

Du, Z., Chen, X., & Zhang, H. (2018). Convolutional sparse learning for blind deconvolution and application on impulsive feature detection. *IEEE Transactions on Instrumentation and Measurement*, *67*(2), 338–349.

Eldan, R., & Shamir, O. (2016). The power of depth for feedforward neural networks. *Conference on Learning Theory*, 907–940.

Fan, F., Cong, W., & Wang, G. (2018). A new type of neurons for machine learning. *International Journal for Numerical Methods in Biomedical Engineering*, *34*(2), e2920.

Fan, F., Xiong, J., & Wang, G. (2020). Universal approximation with quadratic deep networks. *Neural Networks*, *124*, 383–392.

Fan, F.-L., Xiong, J., Li, M., & Wang, G. (2021). On interpretability of artificial neural networks: A survey. *IEEE Transactions on Radiation and Plasma Medical Sciences*, *5*(6), 741–760.

Fan, F.-L., Wang, D., Guo, H., Zhu, Q., Yan, P., Wang, G., & Yu, H. (2022). On a sparse shortcut topology of artificial neural networks. *IEEE Transactions on Artificial Intelligence*, *3*(4), 595–608.

Fan, F.-L., Li, M., Wang, F., Lai, R., & Wang, G. (2023). On expressivity and trainability of quadratic networks. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15.

Fan, F.-L., Dong, H.-C., Wu, Z., Ruan, L., Zeng, T., Cui, Y., & Liao, J.-X. (2025). One neuron saved is one neuron earned: On parametric efficiency of quadratic networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *47*(11), 9702–9717.

Fan, F., Shan, H., Kalra, M. K., Singh, R., Qian, G., Getzin, M., Teng, Y., Hahn, J., & Wang, G. (2019). Quadratic autoencoder (Q-AE) for low-dose CT denoising. *IEEE Transactions on Medical Imaging*, *39*(6), 2035–2050.

Fan, H., Liu, X., Fuh, J. Y. H., Lu, W. F., & Li, B. (2024). Embodied intelligence in manufacturing: Leveraging large language models for autonomous industrial robotics. *Journal of Intelligent Manufacturing*, 1–17.

Fang, B., Hu, J., Yang, C., Cao, Y., & Jia, M. (2021). A blind deconvolution algorithm based on backward automatic differentiation and its application to rolling bearing fault diagnosis. *Measurement Science and Technology*, *33*(2), 025009.

Fang, B., Hu, J., Yang, C., & Chen, X. (2022). Minimum noise amplitude deconvolution and its application in repetitive impact detection. *Structural Health Monitoring*, 14759217221114527.

Fang, H., Deng, J., Bai, Y., Feng, B., Li, S., Shao, S., & Chen, D. (2021). CLFormer: A lightweight transformer based on convolutional embedding and linear self-attention with strong robustness for bearing fault diagnosis under limited sample conditions. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–8.

Fang, H., Deng, J., Zhao, B., Shi, Y., Zhou, J., & Shao, S. (2021). LEFE-Net: A lightweight efficient feature extraction network with strong robustness for bearing fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–11.

Fefferman, C., Mitter, S., & Narayanan, H. (2016). Testing the manifold hypothesis. *Journal of the American Mathematical Society*, *29*(4), 983–1049.

Frusque, G., & Fink, O. (2022). Learnable wavelet packet transform for data-adapted spectrograms. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3119–3123.

Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Yang, J., Zheng, X., Li, K., Sun, X., et al. (2023). MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., & Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, *17*(59), 1–35.

Gao, Q., Huang, T., Zhao, K., Shao, H., & Jin, B. (2024). Multi-source weighted source-free domain transfer method for rotating machinery fault diagnosis. *Expert Systems with Applications*, *237*, 121585.

Gong, S., Li, S., Zhang, Z., & Xia, M. (2024). Nonlinear blind deconvolution based on generalized normalized lp/lq norm for early fault detection. *IEEE Transactions on Instrumentation and Measurement*, *73*, 1–12.

Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, *129*(6), 1789–1819.

Goyal, M., Goyal, R., & Lall, B. (2020). Improved polynomial neural networks with normalised activations. *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Gu, X., Yang, S., Liu, Y., Hao, R., & Liu, Z. (2022). Multi-sparsity-based blind deconvolution and its application to wheelset bearing fault detection. *Measurement*, *199*, 111449.

Gu, Z., Zhu, B., Zhu, G., Chen, Y., Tang, M., & Wang, J. (2024). AnomalyGPT: Detecting industrial anomalies using large vision-language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(3), 1932–1940.

Gunerkar, R. S., Jalan, A. K., & Belgamwar, S. U. (2019). Fault diagnosis of rolling element bearing based on artificial neural network. *Journal of Mechanical Science and Technology*, *33*, 505–511.

Hall, P., & Strutt, J. (2003). Probabilistic physics-of-failure models for component reliabilities using monte carlo simulation and weibull analysis: A parametric study. *Reliability Engineering and System Safety*, *80*(3), 233–242.

Han, S., Hu, X., Huang, H., Jiang, M., & Zhao, Y. (2022). Adbench: Anomaly detection benchmark. *Advances in Neural Information Processing Systems*, *35*, 32142–32159.

Haykin, S. (1986). *Adaptive filter theory*. Prentice-Hall, Inc.

He, C., Shi, H., & Li, J. (2023). IDSN: A one-stage interpretable and differentiable STFT domain adaptation network for traction motor of high-speed trains cross-machine diagnosis. *Mechanical Systems and Signal Processing*, *205*, 110846.

He, C., Shi, H., Si, J., & Li, J. (2023). Physics-informed interpretable wavelet weight initialization and balanced dynamic adaptive threshold for intelligent fault diagnosis of rolling bearings. *Journal of Manufacturing Systems*, *70*, 579–592.

He, C., Shi, H., Li, R., Li, J., & Yu, Z. (2024). Interpretable modulated differentiable STFT and physics-informed balanced spectrum metric for freight train wheelset bearing cross-machine transfer fault diagnosis under speed fluctuations. *Advanced Engineering Informatics*, *62*, 102568.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

He, K., Su, Z., Tian, X., Yu, H., & Luo, M. (2022). RUL prediction of wind turbine gearbox bearings based on self-calibration temporal convolutional network. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–12.

He, L., Wang, D., Yi, C., Zhou, Q., & Lin, J. (2021). Extracting cyclo-stationarity of repetitive transients from envelope spectrum based on prior-unknown blind deconvolution technique. *Signal Processing*, *183*, 107997.

He, L., Yi, C., Wang, D., Wang, F., & Lin, J.-h. (2021). Optimized minimum generalized lp/lq deconvolution for recovering repetitive impacts from a vibration mixture. *Measurement*, *168*, 108329.

Hinton, G. E. (2008). Visualizing high-dimensional data using t-SNE. *Vigiliae Christianae*, *9*, 2579–2605.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Ho, T. K. (1995). Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, *1*, 278–282.

Hojjati, H., & Armanfard, N. (2023). DASVDD: Deep autoencoding support vector data descriptor for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 1–12.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Hou, B., Wang, D., Chen, Y., Wang, H., Peng, Z., & Tsui, K.-L. (2022). Interpretable online updated weights: Optimized square envelope spectrum for machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, *169*, 108779.

Hou, B., Xie, M., Yan, H., & Wang, D. (2024). Impulsive mode decomposition. *Mechanical Systems and Signal Processing*, *211*, 111227.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobileNetV3. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1314–1324.

Huang, C.-G., Huang, H.-Z., & Li, Y.-F. (2019). A bidirectional LSTM prognostics method under multiple operational conditions. *IEEE Transactions on Industrial Electronics*, *66*(11), 8792–8802.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4700–4708.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., Yen, N.-C., Tung, C. C., & Liu, H. H. (1998). The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, *454*(1971), 903–995.

Huang, Q., Han, Y., Zhang, X., Sheng, J., Zhang, Y., & Xie, H. (2023). FFKD-CGhostNet: A novel lightweight network for fault diagnosis in edge computing scenarios. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–10.

Huang, X., Wen, G., Dong, S., Zhou, H., Lei, Z., Zhang, Z., & Chen, X. (2021). Memory residual regression autoencoder for bearing fault detection. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–12.

Ivakhnenko, A. G. (1971). Polynomial theory of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics*, (4), 364–378.

Jamshidpour, E., Shahbazi, M., Saadate, S., Poure, P., & Gholipour, E. (2014). FPGA based fault detection and fault tolerance operation in DC-DC converters. *2014 International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, 37–42.

Javed, K., Gouriveau, R., Zerhouni, N., & Nectoux, P. (2015). Enabling health monitoring approach based on vibration data for accurate prognostics. *IEEE Transactions on Industrial Electronics*, *62*(1), 647–656.

Ji, M., Peng, G., Li, S., Cheng, F., Chen, Z., Li, Z., & Du, H. (2022). A neural network compression method based on knowledge-distillation and parameter quantization for the bearing fault diagnosis. *Applied Soft Computing*, *127*, 109331.

Jia, L., Chow, T. W., & Yuan, Y. (2023). GTFE-Net: A Gramian time frequency enhancement CNN for bearing fault diagnosis. *Engineering Applications of Artificial Intelligence*, *119*, 105794.

Jia, L., Chow, T. W. S., Wang, Y., & Yuan, Y. (2022). Multiscale residual attention convolutional neural network for bearing fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–13.

Jia, X., Zhao, M., Buzza, M., Di, Y., & Lee, J. (2017). A geometrical investigation on the generalized lp/lq norm for blind deconvolution. *Signal Processing*, *134*, 63–69.

Jia, X., Zhao, M., Di, Y., Jin, C., & Lee, J. (2017). Investigation on the kurtosis filter and the derivation of convolutional sparse filter for impulsive signature enhancement. *Journal of Sound and Vibration*, *386*, 433–448.

Jia, X., Zhao, M., Di, Y., Li, P., & Lee, J. (2018). Sparse filtering with the generalized lp/lq norm and its applications to the condition monitoring of rotating machinery. *Mechanical Systems and Signal Processing*, *102*, 198–213.

Jiang, G., Zhou, W., Chen, Q., He, Q., & Xie, P. (2022). Dual residual attention network for remaining useful life prediction of bearings. *Measurement*, *199*, 111424.

Jiang, L., Xuan, J., & Shi, T. (2013). Feature extraction based on semi-supervised kernel marginal fisher analysis and its application in bearing fault diagnosis. *Mechanical Systems and Signal Processing*, *41*(1-2), 113–126.

Jiang, X., Cheng, X., Shi, J., Huang, W., Shen, C., & Zhu, Z. (2018). A new l0-norm embedded med method for roller element bearing fault diagnosis at early stage of damage. *Measurement*, *127*, 414–424.

Jiang, Y., Yang, F., Zhu, H., Zhou, D., & Zeng, X. (2020). Nonlinear CNN: Improving CNNs with quadratic convolutions. *Neural Computing and Applications*, *32*(12), 8507–8516.

Kaminskiy, M., & Krivtsov, V. (2005). A simple procedure for Bayesian estimation of the Weibull distribution. *IEEE Transactions on Reliability*, *54*(4), 612–616.

Kang, M., Kim, J., & Kim, J.-M. (2014). An FPGA-based multicore system for real-time bearing fault diagnosis using ultrasampling rate AE signals. *IEEE Transactions on Industrial Electronics*, *62*(4), 2319–2329.

Kankar, P. K., Sharma, S. C., & Harsha, S. P. (2011). Rolling element bearing fault diagnosis using wavelet transform. *Neurocomputing*, *74*(10), 1638–1645.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *30*.

Kendall, A., Gal, Y., & Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7482–7491.

Kumaraswamidhas, L., Laha, S., et al. (2021). Bearing degradation assessment and remaining useful life estimation based on Kullback-Leibler divergence and Gaussian processes regression. *Measurement*, *174*, 108948.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*, 436–444.

Lee, J.-Y., & Nandi, A. (2000). Extraction of impacting signals using blind deconvolution. *Journal of Sound and Vibration*, *232*(5), 945–962.

Lei, Y., He, Z., Zi, Y., & Chen, X. (2008). New clustering algorithm-based fault diagnosis using compensation distance evaluation technique. *Mechanical Systems and Signal Processing*, *22*(2), 419–435.

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, *104*, 799–834.

Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., & Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, *138*, 106587.

Lessmeier, C., Kimotho, J. K., Zimmer, D., & Sextro, W. (2016). Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. *PHM Society European Conference*, *3*(1).

Levin, A., Weiss, Y., Durand, F., & Freeman, W. T. (2011). Understanding blind deconvolution algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(12), 2354–2367.

Li, K., Ping, X., Wang, H., Chen, P., & Cao, Y. (2013). Sequential fuzzy diagnosis method for motor roller bearing in variable operating conditions based on vibration analysis. *Sensors*, *13*(6), 8013–8041.

Li, L. (2014). Sparsity-promoted blind deconvolution of ground-penetrating radar (GPR) data. *IEEE Geoscience and Remote Sensing Letters*, *11*(8), 1330–1334.

Li, T., Zhao, Z., Sun, C., Cheng, L., Chen, X., Yan, R., & Gao, R. X. (2022). WaveletKernelNet: An interpretable deep neural network for industrial intelligent diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *52*(4), 2302–2312.

Li, T., Ma, Y., Xu, J., Stenger, B., Liu, C., & Hirate, Y. (2018). Deep heterogeneous autoencoders for collaborative filtering. *IEEE International Conference on Data Mining*, 1164–1169.

Li, W., & Deng, L. (2023). A hybrid model-based prognostics approach for estimating remaining useful life of rolling bearings. *Measurement Science and Technology*, *34*(10), 105012.

Li, W., He, J., Lin, H., Huang, R., He, G., & Chen, Z. (2023). A LightGBM-based multi-scale weighted ensemble model for few-shot fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–1.

Li, X., Zhang, W., & Ding, Q. (2019). Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. *Reliability Engineering and System Safety*, *182*, 208–218.

Li, X., Teng, W., Peng, D., Ma, T., Wu, X., & Liu, Y. (2023). Feature fusion model based health indicator construction and self-constraint state-space estimator for remaining useful life prediction of bearings in wind turbines. *Reliability Engineering and System Safety*, *233*, 109124.

Li, Y., Sun, Y., Horoshenkov, K., & Naqvi, S. M. (2022). Domain adaptation and autoencoder-based unsupervised speech enhancement. *IEEE Transactions on Artificial Intelligence*, *3*(1), 43–52.

Li, Z., Zhao, Y., Hu, X., Botta, N., Ionescu, C., & Chen, G. (2022). ECOD: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*.

Liang, K., Zhao, M., Lin, J., Jiao, J., & Ding, C. (2021). Maximum average kurtosis deconvolution and its application for the impulsive fault feature enhancement of rotating machinery. *Mechanical Systems and Signal Processing*, *149*, 107323.

Liao, J.-X., Dong, H.-C., Luo, L., Sun, J., & Zhang, S. (2023). Multi-task neural network blind deconvolution and its application to bearing fault feature extraction. *Measurement Science and Technology*, *34*(7), 075017.

Liao, J.-X., Dong, H.-C., Sun, Z.-Q., Sun, J., Zhang, S., & Fan, F.-L. (2023). Attention-embedded quadratic network (qttention) for effective and interpretable bearing fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–13.

Liao, Z., Song, X., Jia, B., & Chen, P. (2021). Bearing fault feature enhancement and diagnosis based on statistical filtering and 1.5-dimensional symmetric difference analytic energy spectrum. *IEEE Sensors Journal*, *21*(8), 9959–9968.

Liao, Z., Song, X., Wang, H., Song, W., Jia, B., & Chen, P. (2022). Bearing fault diagnosis using reconstruction adaptive determinate stationary subspace filtering and enhanced third-order spectrum. *IEEE Sensors Journal*, *22*(11), 10764–10773.

Lin, B., & Zhang, Y. (2023). LibMTL: A Python library for multi-task learning. *Journal of Machine Learning Research*, *24*(209), 1–7.

Lin, D., Talathi, S., & Annapureddy, S. (2016). Fixed point quantization of deep convolutional networks. *International Conference on Machine Learning*, 2849–2858.

Lin, S., Zhang, M., Cheng, X., Shi, L., Gamba, P., & Wang, H. (2024). Dynamic low-rank and sparse priors constrained deep autoencoders for hyperspectral anomaly detection. *IEEE Transactions on Instrumentation and Measurement*, *73*, 1–18.

Liu, B., Wang, D., Lin, K., Tan, P.-N., & Zhou, J. (2021). RCA: A deep collaborative autoencoder approach for anomaly detection. *IJCAI: proceedings of the conference*, *2021*, 1505.

Liu, H., Xu, X., Li, E., Zhang, S., & Li, X. (2021). Anomaly detection with representative neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, 1–11.

Liu, L., Song, X., Chen, K., Hou, B., Chai, X., & Ning, H. (2021). An enhanced encoder–decoder framework for bearing remaining useful life prediction. *Measurement*, *170*, 108753.

Liu, Q., Liang, T., Huang, Z., & Dinavahi, V. (2019). Real-time FPGA-based hardware neural network for fault detection and isolation in more electric aircraft. *IEEE Access*, *7*, 159831–159841.

Liu, R., Yang, B., Zio, E., & Chen, X. (2018). Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, *108*, 33–47.

Liu, S., Jiang, H., Wu, Z., & Li, X. (2022). Data synthesis using deep feature enhanced generative adversarial networks for rolling bearing imbalanced fault diagnosis. *Mechanical Systems and Signal Processing*, *163*, 108139.

Liu, Y., Li, Z., Zhou, C., Jiang, Y., Sun, J., Wang, M., & He, X. (2019). Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, *32*(8), 1517–1528.

Long, M., Cao, Y., Wang, J., & Jordan, M. (2015). Learning transferable features with deep adaptation networks. *International Conference on Machine Learning*, 97–105.

Loshchilov, I., & Hutter, F. (2016). SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Lu, F., Tong, Q., Jiang, X., Du, S., Xu, J., Huo, J., & Zhang, Z. (2024). Envelope spectrum neural network with adaptive domain weight harmonization for intelligent bearing fault diagnosis under cross-machine scenarios. *Advanced Engineering Informatics*, *62*, 102787.

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Ma, M., & Mao, Z. (2020). Deep-convolution-based LSTM network for remaining useful life prediction. *IEEE Transactions on Industrial Informatics*, *17*(3), 1658–1667.

Ma, X., Dai, X., Bai, Y., Wang, Y., & Fu, Y. (2024). Rewrite the stars. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5694–5703.

Majumdar, A. (2019). Blind denoising autoencoder. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(1), 312–317.

Mantini, P., & Shah, S. K. (2021). CQNN: Convolutional quadratic neural networks. *2020 25th International Conference on Pattern Recognition (ICPR)*, 9819–9826.

McDonald, G. L., & Zhao, Q. (2017). Multipoint optimal minimum entropy deconvolution and convolution fix: Application to vibration fault detection. *Mechanical Systems and Signal Processing*, *82*, 461–477.

McDonald, G. L., Zhao, Q., & Zuo, M. J. (2012). Maximum correlated kurtosis deconvolution and application on gear tooth chip fault detection. *Mechanical Systems and Signal Processing*, *33*, 237–255.

McFadden, P., & Smith, J. (1984). Model for the vibration produced by a single point defect in a rolling element bearing. *Journal of Sound and Vibration*, *96*(1), 69–82.

Miao, Y., Zhang, B., Li, C., Lin, J., & Zhang, D. (2022). Feature mode decomposition: New decomposition theory for rotating machinery fault diagnosis. *IEEE Transactions on Industrial Electronics*, *70*(2), 1949–1960.

Miao, Y., Zhang, B., Lin, J., Zhao, M., Liu, H., Liu, Z., & Li, H. (2022). A review on the application of blind deconvolution in machinery fault diagnosis. *Mechanical Systems and Signal Processing*, *163*, 108202.

Miao, Y., Zhao, M., Lin, J., & Xu, X. (2016). Sparse maximum harmonics-to-noise-ratio deconvolution for weak fault signature detection in bearings. *Measurement Science and Technology*, *27*(10), 105004.

Miao, Y., Zhao, M., Lin, J., & Lei, Y. (2017). Application of an improved maximum correlated kurtosis deconvolution method for fault diagnosis of rolling element bearings. *Mechanical Systems and Signal Processing*, *92*, 173–195.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., et al. (2018). Mixed precision training. *International Conference on Learning Representations*.

Mittal, S. (2020). A survey of FPGA-based accelerators for convolutional neural networks. *Neural Computing and Applications*, *32*(4), 1109–1139.

Mucherino, A., Papajorgji, P. J., Pardalos, P. M., Mucherino, A., Papajorgji, P. J., & Pardalos, P. M. (2009). K-nearest neighbor classification. *Data Mining in Agriculture*, 83–106.

Napolitano, A. (2016). Cyclostationarity: New trends and applications. *Signal Processing*, *120*, 385–408.

Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-Morello, B., Zerhouni, N., & Varnier, C. (2012). Pronostia: An experimental platform for bearings accelerated degradation tests. *IEEE International Conference on Prognostics and Health Management, PHM'12.*, 1–8.

Nemani, V., Biggio, L., Huan, X., Hu, Z., Fink, O., Tran, A., Wang, Y., Zhang, X., & Hu, C. (2023). Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial. *Mechanical Systems and Signal Processing*, *205*, 110796.

Ng, A. (2011). Sparse autoencoder. *CS294A Lecture notes*, *72*(2011), 1–19.

Ni, Q., Ji, J. C., & Feng, K. (2023). Data-driven prognostic scheme for bearings based on a novel health indicator and gated recurrent unit network. *IEEE Transactions on Industrial Informatics*, *19*(2), 1301–1311.

Ni, Q., Ji, J., Halkon, B., Feng, K., & Nandi, A. K. (2023). Physics-informed residual network (PIRes-Net) for rolling element bearing fault diagnostics. *Mechanical Systems and Signal Processing*, *200*, 110544.

Ojo, A. O., & Bouguila, N. (2024). A topic modeling and image classification framework: The generalized dirichlet variational autoencoder. *Pattern Recognition*, *146*, 110037.

Pan, Y., He, F., Yan, X., & Li, H. (2021). A synchronized heterogeneous autoencoder with feature-level and label-level knowledge distillation for the recommendation. *Engineering Applications of Artificial Intelligence*, *106*, 104494.

Panda, M. K., Subudhi, B. N., Veerakumar, T., & Jakhetiya, V. (2023). Modified ResNet-152 network with hybrid pyramidal pooling for local change detection. *IEEE Transactions on Artificial Intelligence*, 1–14.

Pang, G., Shen, C., Cao, L., & Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, *54*(2), 1–38.

Paris, P. (1963). A critical analysis of crack propagation laws. *Journal of Basic Engineering*, *85*(4), 528–533.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, *33*(3), 1065–1076.

Pearson, K. (1905). Das fehlergesetz und seine verallgemeiner-ungen durch fechner und pearson. a rejoinder. *Biometrika*, *4*(1-2), 169–212.

Peeters, C., Antoni, J., & Helsen, J. (2020). Blind filters based on envelope spectrum sparsity indicators for bearing and gear vibration-based condition monitoring. *Mechanical Systems and Signal Processing*, *138*, 106556.

Peng, D., Zhu, X., Teng, W., & Liu, Y. (2023). Use of generalized Gaussian cyclostationarity for blind deconvolution and its application to bearing fault diagnosis under non-Gaussian conditions. *Mechanical Systems and Signal Processing*, *196*, 110351.

Peng, W., Ye, Z.-S., & Chen, N. (2019). Joint online RUL prediction for multivariate deteriorating systems. *IEEE Transactions on Industrial Informatics*, *15*(5), 2870–2878.

Peter, W. T., & Wang, D. (2013). The design of a new sparsogram for fast bearing fault diagnosis: Part 1 of the two related manuscripts that have a joint title as “two automatic vibration-based fault

diagnostic methods using the novel sparsity measurement–parts 1 and 2". *Mechanical Systems and Signal Processing*, *40*(2), 499–519.

Pham, P., Nguyen, L. T., Nguyen, N. T., Kozma, R., & Vo, B. (2023). A hierarchical fused fuzzy deep neural network with heterogeneous network embedding for recommendation. *Information Sciences*, *620*, 105–124.

Philbeck, T., & Davis, N. (2018). The fourth industrial revolution: Shaping a new era. *Journal of International Affairs*, *72*(1), 17–22.

Pitkänen, M. (2006). Negentropy maximization principle. *TGD Inspired Theory of Consciousness. Online book.*

Qi, Y., Yang, Z., Lian, J., Guo, Y., Sun, W., Liu, J., Wang, R., & Ma, Y. (2021). A new heterogeneous neural network model and its application in image enhancement. *Neurocomputing*, *440*, 336–350.

Qiao, W., & Lu, D. (2015a). A survey on wind turbine condition monitoring and fault diagnosis—part i: Components and subsystems. *IEEE Transactions on Industrial Electronics*, *62*(10), 6536–6545.

Qiao, W., & Lu, D. (2015b). A survey on wind turbine condition monitoring and fault diagnosis—part ii: Signals and signal processing methods. *IEEE Transactions on Industrial Electronics*, *62*(10), 6546–6557.

Qin, Y., Yang, J., Zhou, J., Pu, H., & Mao, Y. (2023). A new supervised multi-head self-attention autoencoder for health indicator construction and similarity-based machinery RUL prediction. *Advanced Engineering Informatics*, *56*, 101973.

Qiu, J., Wang, J., Yao, S., Guo, K., Li, B., Zhou, E., Yu, J., Tang, T., Xu, N., Song, S., et al. (2016). Going deeper with embedded FPGA platform for convolutional neural network. *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 26–35.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*, 8748–8763.

Rai, A., & Upadhyay, S. H. (2016). A review on signal processing techniques utilized in the fault diagnosis of rolling element bearings. *Tribology International*, *96*, 289–306.

Randall, R. B., & Monitoring, V.-b. C. (2011). Industrial, aerospace and automotive applications. *Vibration-based Condition Monitoring. West Sussex*, 13–20.

Randall, R. B., & Antoni, J. (2011). Rolling element bearing diagnostics—a tutorial. *Mechanical Systems and Signal Processing*, *25*(2), 485–520.

Randall, R. B., Antoni, J., & Chobsaard, S. (2001). The relationship between spectral correlation and envelope analysis in the diagnostics of bearing faults and other cyclostationary machine signals. *Mechanical Systems and Signal Processing*, *15*(5), 945–962.

Randall, R. B. (2021). *Vibration-based condition monitoring: Industrial, automotive and aerospace applications*. John Wiley; Sons.

Randhawa, R. H., Aslam, N., Alauthman, M., & Rafiq, H. (2023). Evasion generative adversarial network for low data regimes. *IEEE Transactions on Artificial Intelligence*, *4*(5), 1076–1088.

Rauber, T. W., de Assis Boldt, F., & Varejao, F. M. (2014). Heterogeneous feature models and feature selection applied to bearing fault diagnosis. *IEEE Transactions on Industrial Electronics*, *62*(1), 637–646.

Ren, L., Wang, T., Jia, Z., Li, F., & Han, H. (2023). A lightweight and adaptive knowledge distillation framework for remaining useful life prediction. *IEEE Transactions on Industrial Informatics*, *19*(8), 9060–9070.

Reynolds, D. A., et al. (2009). Gaussian mixture models. *Encyclopedia of biometrics*, *741*(659-663).

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., & Bengio, Y. (2014). Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., & Kloft, M. (2018). Deep one-class classification. *International Conference on Machine Learning*, 4393–4402.

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., & Müller, K.-R. (2021). A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, *109*(5), 756–795.

Ruqiang, Y., Zheng, Z., Yuangui, Y., Yasong, L., Chenye, H., Tao, Z., Zhibin, Z., Shibin, W., & Chen, X. (2024). Challenges and opportunities of XAI in industrial intelligent diagnosis: Attribution interpretation. *Journal of Mechanical Engineering*, 1–20.

Ruqiang, Y., Zuogang, S., Zhiying, W., Wengang, X., Zhibin, Z., Shibin, W., & Xuefeng, C. (2024). Challenges and opportunities of XAI in industrial intelligent diagnosis: Priori-empowered. *Journal of Mechanical Engineering*, 1–20.

Sadr, H., Pedram, M. M., & Teshnehlab, M. (2020). Multi-view deep network: A deep model based on learning features from heterogeneous neural networks for sentiment analysis. *IEEE Access*, *8*, 86984–86997.

Sanghvi, Y., Chi, Y., & Chan, S. H. (2024). Kernel diffusion: An alternate approach to blind deconvolution. *European Conference on Computer Vision*, 1–20.

Sarker, I. H. (2021). Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, *2*(6), 1–20.

Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., & Langs, G. (2017). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. *International Conference on Information Processing in Medical Imaging*, 146–157.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, *13*(7), 1443–1471.

Shao, H., Li, W., Cai, B., Wan, J., Xiao, Y., & Yan, S. (2023). Dual-threshold attention-guided GAN and limited infrared thermal images for rotating machinery fault diagnosis under speed fluctuation. *IEEE Transactions on Industrial Informatics*, *19*(9), 9933–9942.

Shao, H., Zhou, X., Lin, J., & Liu, B. (2024). Few-shot cross-domain fault diagnosis of bearing driven by task-supervised anil. *IEEE Internet of Things Journal*, *11*(13), 22892–22902.

Shin, Y., & Ghosh, J. (1991). The pi-sigma network: An efficient higher-order neural network for pattern classification and function approximation. *IJCNN-91-Seattle International Joint Conference on Neural Networks*, *1*, 13–18.

Smith, W. A., & Randall, R. B. (2015). Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study. *Mechanical Systems and Signal Processing*, *64*, 100–131.

Su, Y., Shi, L., Zhou, K., Bai, G., & Wang, Z. (2024). Knowledge-informed deep networks for robust fault diagnosis of rolling bearings. *Reliability Engineering and System Safety*, *244*, 109863.

Sun, C., Ma, M., Zhao, Z., Tian, S., Yan, R., & Chen, X. (2019). Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing. *IEEE Transactions on Industrial Informatics*, *15*(4), 2416–2425.

Takiddin, A., Ismail, M., Atat, R., Davis, K. R., & Serpedin, E. (2023). Robust graph autoencoder-based detection of false data injection attacks against data poisoning in smart grids. *IEEE Transactions on Artificial Intelligence*, *5*(3), 1287–1301.

Tang, Y., Zhang, C., Wu, J., Xie, Y., Shen, W., & Wu, J. (2024). Deep learning-based bearing fault diagnosis using a trusted multiscale quadratic attention-embedded convolutional neural network. *IEEE Transactions on Instrumentation and Measurement*, *73*, 1–15.

Tao, X., Yan, S., Gong, X., & Adak, C. (2022). Learning multi-resolution features for unsupervised anomaly localization on industrial textured surfaces. *IEEE Transactions on Artificial Intelligence*, 1–13.

Tax, D. M., & Duin, R. P. (2004). Support vector data description. *Machine learning*, *54*, 45–66.

Tipping, M. E., & Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *61*(3), 611–622.

Toumi, Y., Bengherbia, B., Lachenani, S., & Ould Zmirli, M. (2022). FPGA implementation of a bearing fault classification system based on an envelope analysis and artificial neural network. *Arabian Journal for Science and Engineering*, *47*(11), 13955–13977.

Tran, M.-Q., Liu, M.-K., Tran, Q.-V., & Nguyen, T.-K. (2022). Effective fault diagnosis based on wavelet and convolutional attention neural network for induction motors. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–13.

Troullinou, E., Tsagkatakis, G., Losonczy, A., Poirazi, P., & Tsakalides, P. (2024). A generative neighborhood-based deep autoencoder for robust imbalanced classification. *IEEE Transactions on Artificial Intelligence*, *5*(1), 80–91.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

von Hahn, T., & Mechefske, C. K. (2022). Knowledge informed machine learning using a Weibull-based loss function. *arXiv preprint arXiv:2201.01769*.

Vrabie, V., Granjon, P., & Serviere, C. (2003). Spectral kurtosis: From definition to application. *6th IEEE International Workshop on Nonlinear Signal and Image Processing (NSIP 2003)*, xx.

Wang, B., Tao, F., Fang, X., Liu, C., Liu, Y., & Freiheit, T. (2021). Smart manufacturing and intelligent manufacturing: A comparative review. *Engineering*, *7*(6), 738–757.

Wang, B., Lei, Y., Li, N., & Li, N. (2018). A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability*, *69*(1), 401–412.

Wang, C.-S., Tang, J., & Zhao, B. (1991). An improvement on D norm deconvolution: A fast algorithm and the related procedure. *Geophysics*, *56*(5), 675–680.

Wang, D., Zhao, Y., Yi, C., Tsui, K.-L., & Lin, J. (2018). Sparsity guided empirical wavelet transform for fault diagnosis of rolling element bearings. *Mechanical Systems and Signal Processing*, *101*, 292–308.

Wang, H., Chen, S., & Zhai, W. (2024). Variational generalized nonlinear mode decomposition: Algorithm and applications. *Mechanical Systems and Signal Processing*, *206*, 110913.

Wang, H., Liu, Z., Peng, D., & Cheng, Z. (2022). Attention-guided joint learning CNN with noise robustness for bearing fault diagnosis and vibration signal denoising. *ISA Transactions*, *128*, 470–484.

Wang, H., Liu, Z., Peng, D., & Zuo, M. J. (2023). Interpretable convolutional neural network with multilayer wavelet for noise-robust machinery fault diagnosis. *Mechanical Systems and Signal Processing*, *195*, 110314.

Wang, L., & Yoon, K.-J. (2021). Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(6), 3048–3068.

Wang, M., Chen, Y., Zhang, X., Chau, T. K., Ching Iu, H. H., Fernando, T., Li, Z., & Ma, M. (2021). Roller bearing fault diagnosis based on integrated fault feature and SVM. *Journal of Vibration Engineering and Technologies*, 1–10.

Wang, S., & Xiang, J. (2020). A minimum entropy deconvolution-enhanced convolutional neural networks for fault diagnosis of axial piston pumps. *Soft Computing*, *24*(4), 2983–2997.

Wang, X., Li, Y., Rui, T., Zhu, H., & Fei, J. (2015). Bearing fault diagnosis method based on Hilbert envelope spectrum and deep belief network. *Journal of Vibroengineering*, *17*(3), 1295–1308.

Wang, X., Du, Y., Lin, S., Cui, P., Shen, Y., & Yang, Y. (2020). adVAE: A self-adversarial variational autoencoder with Gaussian anomaly prior knowledge for anomaly detection. *Knowledge-Based Systems*, *190*, 105187.

Wang, Y., Xiang, J., Markert, R., & Liang, M. (2016). Spectral kurtosis for fault detection, diagnosis and prognostics of rotating machines: A review with applications. *Mechanical Systems and Signal Processing*, *66*, 679–698.

Wang, Y., Yu, Z., Wu, J., Wang, C., Zhou, Q., & Hu, J. (2024). Adaptive knowledge distillation-based lightweight intelligent fault diagnosis framework in IoT edge computing. *IEEE Internet of Things Journal*, *11*(13), 23156–23169.

Wang, Y., Wang, D., Hou, B., Lu, S., & Peng, Z. (2025). Robust optimized weights spectrum: Enhanced interpretable fault feature extraction method by solving frequency fluctuation problem. *Mechanical Systems and Signal Processing*, *222*, 111798.

Wang, Z., Zhou, J., Du, W., Lei, Y., & Wang, J. (2022). Bearing fault diagnosis method based on adaptive maximum cyclostationarity blind deconvolution. *Mechanical Systems and Signal Processing*, *162*, 108018.

Weibull, W. (1951). A statistical distribution function of wide applicability. *Journal of applied mechanics*.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, *3*(1), 1–40.

Wiggins, R. A. (1978). Minimum entropy deconvolution. *Geoexploration*, *16*(1-2), 21–35.

Williamson, D. (1991). Dynamically scaled fixed point arithmetic. *[1991] IEEE Pacific Rim Conference on Communications, Computers and Signal Processing Conference Proceedings*, 315–318.

Xiao, Y., Shao, H., Han, S., Huo, Z., & Wan, J. (2022). Novel joint transfer network for unsupervised bearing fault diagnosis from simulation domain to experimental domain. *IEEE/ASME Transactions On Mechatronics*, *27*(6), 5254–5263.

Xiao, Y., Shao, H., Wang, J., Yan, S., & Liu, B. (2024). Bayesian variational Transformer: A generalizable model for rotating machinery fault diagnosis. *Mechanical Systems and Signal Processing*, *207*, 110936.

Xiong, S., Zhou, H., He, S., Zhang, L., Xia, Q., Xuan, J., & Shi, T. (2020). A novel end-to-end fault diagnosis approach for rolling bearings by integrating wavelet packet transform into convolutional neural network structures. *Sensors*, *20*(17), 4965.

Xu, Y., Deng, Y., Zhao, J., Tian, W., & Ma, C. (2019). A novel rolling bearing fault diagnosis method based on empirical wavelet transform and spectral trend. *IEEE Transactions on Instrumentation and Measurement*, *69*(6), 2891–2904.

Xu, Z., Yu, F., Xiong, J., & Chen, X. (2022). QuadraLib: A performant quadratic neural network library for architecture optimization and design exploration. *Proceedings of Machine Learning and Systems*, *4*, 503–514.

Xue, Z., Song, G., Guo, Q., Liu, B., Zong, Z., Liu, Y., & Luo, P. (2024). Raphael: Text-to-image generation via large mixture of diffusion paths. *Advances in Neural Information Processing Systems*, *36*.

Yan, M., Wang, X., Wang, B., Chang, M., & Muhammad, I. (2020). Bearing remaining useful life prediction using support vector machine and hybrid degradation tracking model. *ISA Transactions*, *98*, 471–482.

Yan, S., Shao, H., Wang, J., Zheng, X., & Liu, B. (2024). LiConvFormer: A lightweight fault diagnosis framework using separable multiscale convolution and broadcast self-attention. *Expert Systems with Applications*, *237*, 121338.

Yan, T., Wang, D., Xia, T., & Xi, L. (2022). A generic framework for degradation modeling based on fusion of spectrum amplitudes. *IEEE Transactions on Automation Science and Engineering*, *19*(1), 308–319.

Yan, T., Wang, D., Xia, T., Peng, Z., & Xi, L. (2025). Theoretical investigation of convex representation based interpretable time-frequency weight optimization for machine health monitoring. *Mechanical Systems and Signal Processing*, *230*, 112625.

Yang, S., Tang, B., Wang, W., Yang, Q., & Hu, C. (2024). Physics-informed multi-state temporal frequency network for RUL prediction of rolling bearings. *Reliability Engineering and System Safety*, *242*, 109716.

Yao, D., Liu, H., Yang, J., & Li, X. (2020). A lightweight neural network with strong robustness for bearing fault diagnosis. *Measurement*, *159*, 107756.

Yao, Y., Ma, J., Feng, S., & Ye, Y. (2024). SVD-AE: An asymmetric autoencoder with SVD regularization for multivariate time series anomaly detection. *Neural Networks*, *170*, 535–547.

Yaqub, M. F., Gondal, I., & Kamruzzaman, J. (2011). Inchoate fault detection framework: Adaptive selection of wavelet nodes and cumulant orders. *IEEE Transactions on instrumentation and Measurement*, *61*(3), 685–695.

Yin, T., Lu, N., Guo, G., Lei, Y., Wang, S., & Guan, X. (2023). Knowledge and data dual-driven transfer network for industrial robot fault diagnosis. *Mechanical Systems and Signal Processing*, *182*, 109597.

You, K., Long, M., Cao, Z., Wang, J., & Jordan, M. I. (2019). Universal domain adaptation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2720–2729.

Yu, H., Huang, J., LI, L., zhou man, m., & Zhao, F. (2023). Deep fractional Fourier transform. *Advances in Neural Information Processing Systems*, *36*, 72761–72773.

Yu, J., Gao, X., Zhai, F., Li, B., Xue, B., Fu, S., Chen, L., & Meng, Z. (2024). An adversarial contrastive autoencoder for robust multivariate time series anomaly detection. *Expert Systems with Applications*, *245*, 123010.

Yu, W.-E., Sun, J., Zhang, S., Zhang, X., & Liao, J.-X. (2023). A class-weighted supervised contrastive learning long-tailed bearing fault diagnosis approach using quadratic neural network. *arXiv preprint arXiv:2309.11717*.

Zhang, B., Miao, Y., Lin, J., & Yi, Y. (2021). Adaptive maximum second-order cyclostationarity blind deconvolution and its application for locomotive bearing fault diagnosis. *Mechanical Systems and Signal Processing*, *158*, 107736.

Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., & Cong, J. (2015). Optimizing FPGA-based accelerator design for deep convolutional neural networks. *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 161–170.

Zhang, H., Chen, X., Du, Z., & Yan, R. (2016). Kurtosis based weighted sparse model with convex optimization technique for bearing fault diagnosis. *Mechanical Systems and Signal Processing*, *80*, 349–376.

Zhang, L., Rao, A., & Agrawala, M. (2023). Adding conditional control to text-to-image diffusion models. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3836–3847.

Zhang, S., Zhang, S., Wang, B., & Habetler, T. G. (2020). Deep learning algorithms for bearing fault diagnostics—a comprehensive review. *IEEE Access*, *8*, 29857–29881.

Zhang, W., Peng, G., Li, C., Chen, Y., & Zhang, Z. (2017). A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors*, *17*(2), 425.

Zhang, X., Zhang, H., & Song, Z. (2023). Feature-aligned stacked autoencoder: A novel semisupervised deep learning model for pattern classification of industrial faults. *IEEE Transactions on Artificial Intelligence*, *4*(4), 592–601.

Zhang, Y., Ren, Z., Feng, K., Yu, K., Beer, M., & Liu, Z. (2023). Universal source-free domain adaptation method for cross-domain fault diagnosis of machines. *Mechanical Systems and Signal Processing*, *191*, 110159.

Zhang, Z., Wang, J., Li, S., Han, B., & Jiang, X. (2023). Fast nonlinear blind deconvolution for rotating machinery fault diagnosis. *Mechanical Systems and Signal Processing*, *187*, 109918.

Zhao, B., Cui, Q., Song, R., Qiu, Y., & Liang, J. (2022). Decoupled knowledge distillation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11953–11962.

Zhao, M., Tang, B., & Tan, Q. (2016). Bearing remaining useful life estimation based on time–frequency representation and supervised dimensionality reduction. *Measurement*, *86*, 41–55.

Zhao, M., Zhong, S., Fu, X., Tang, B., & Pecht, M. (2020). Deep residual shrinkage networks for fault diagnosis. *IEEE Transactions on Industrial Informatics*, *16*(7), 4681–4690.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, *115*, 213–237.

Zhao, Y., Nasrullah, Z., & Li, Z. (2019). PyOD: A Python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, *20*(96), 1–7.

Zhao, Y., Hu, X., Cheng, C., Wang, C., Wan, C., Wang, W., et al. (2021). SUOD: Accelerating large-scale unsupervised heterogeneous outlier detection. *Proceedings of Machine Learning and Systems*, *3*, 463–478.

Zhao, Z., Li, T., Wu, J., Sun, C., Wang, S., Yan, R., & Chen, X. (2020). Deep learning algorithms for rotating machinery intelligent diagnosis: An open source benchmark study. *ISA Transactions*, *107*, 224–255.

Zhong, H., Lv, Y., Yuan, R., & Yang, D. (2022). Bearing fault diagnosis using transfer learning and self-attention ensemble lightweight convolutional neural network. *Neurocomputing*, *501*, 765–777.

Zhou, K., Diehl, E., & Tang, J. (2023). Deep convolutional generative adversarial network with semi-supervised learning enabled physics elucidation for extended gear fault diagnosis under data limitations. *Mechanical Systems and Signal Processing*, *185*, 109772.

Zhou, Q., Zhang, Y., Tang, J., Lin, J., He, L., & Yi, C. (2021). Blind deconvolution technique based on improved correlated generalized lp/lq norm for extracting repetitive transient feature. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–21.

Zhou, T., Han, T., & Droguett, E. L. (2022). Towards trustworthy machine fault diagnosis: A probabilistic Bayesian deep learning framework. *Reliability Engineering and System Safety*, *224*, 108525.

Zhu, R., Chen, Y., Peng, W., & Ye, Z.-S. (2022). Bayesian deep-learning for RUL prediction: An active learning perspective. *Reliability Engineering and System Safety*, *228*, 108758.

Zio, E. (2016). Some challenges and opportunities in reliability engineering. *IEEE Transactions on Reliability*, *65*(4), 1769–1782.

Zio, E. (2022). Prognostics and health management (PHM): Where are we and where do we (need to) go in theory and practice. *Reliability Engineering and Systems Safety*, *218*, 108119.

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., & Chen, H. (2018). Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. *International Conference on Learning Representations*.

Zoumpourlis, G., Doumanoglou, A., Vretos, N., & Daras, P. (2017). Non-linear convolution filters for CNN-based learning. *Proceedings of the IEEE International Conference on Computer Vision*, 4761–4769.

Appendix A

Multi-Task Neural Network Blind Deconvolution and Its Application to Bearing Fault Feature Extraction

This research was conducted during the author's Ph.D. studies at Harbin Institute of Technology:

*"**J.-X. Liao**, H.-C. Dong, L. Luo, J. Sun, S. Zhang*. Multi-task Neural Network Blind Deconvolution and its Application to Bearing Fault Feature Extraction, Measurement Science and Technology, doi: 10.1088/1361-6501/accbdb1."*

A.1 Abstract

Blind deconvolution (BD) is an effective method to extract fault-related characteristics from vibration signals. Previous researches focused on two primary approaches to improve the robustness and effectiveness of BD methods: developing new optimization functions or devising new methods for estimating filter coefficients. However, these methods often suffer from the difficulty of finding the global optimum due to the complex non-convex functions. To address this issue, we propose a novel multi-objective criterion, combining two well-established sparsity criteria: kurtosis and l_1/l_2 norm, that evaluates signal characteristics in both the time and frequency domains. We observe that this criterion, consisting of two sparsity criteria with opposite monotonicity, can mutually constrain and avoid overfitting that occurs with single-domain optimization. Inspired by multi-task convolutional neural networks, we introduce a multi-task 1DCNN with two branches to optimize the criterion in

both domains simultaneously. To our best knowledge, it is the first time a multi-task CNN is used for BD problems. An overview of this method is illustrated in Figure A.1. Experiments show that our method outperforms other state-of-the-art BD methods.

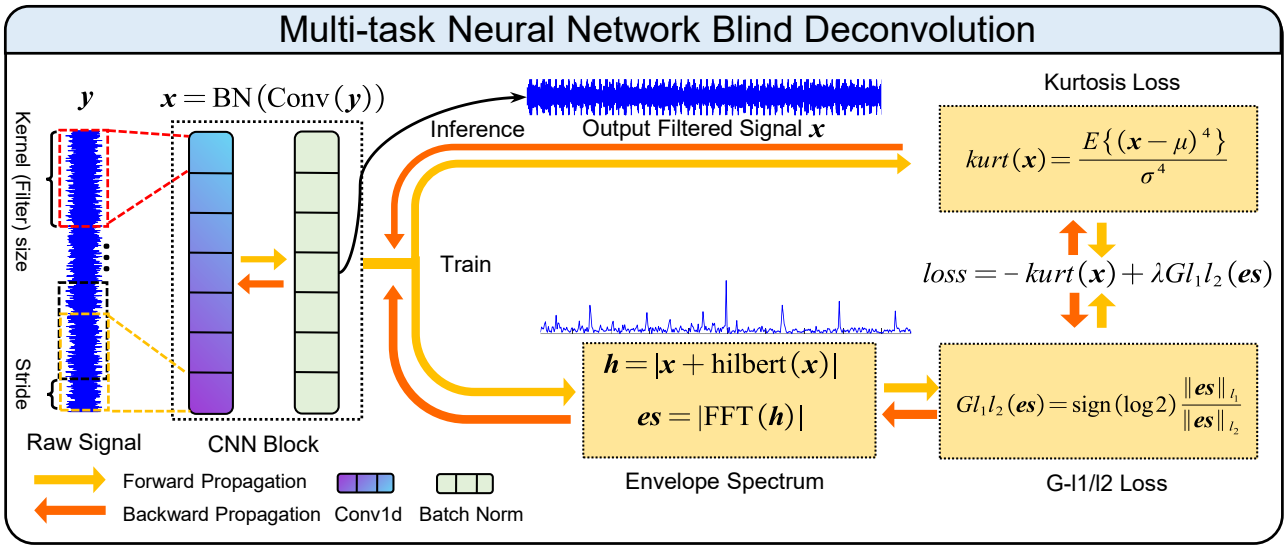


Figure A.1: An overview of the multi-task blind deconvolution neural network. The key idea is to train a convolution neural network that combines the time domain and frequency domain loss functions.

Appendix B

Attention-Embedded Quadratic Network (Qttention) for Effective and Interpretable Bearing Fault Diagnosis

This research was conducted during the author's Ph.D. studies at Harbin Institute of Technology:

*"**J.-X. Liao**, H.-C. Dong, Z.-Q. Sun, J. Sun, S. Zhang*, F.-L. Fan*. Attention-embedded quadratic network (qttention) for effective and interpretable bearing fault diagnosis. IEEE Transactions on Instrumentation and Measurement, doi: 10.1109/TIM.2023.3259031 (ESI High Cited Paper)."*

B.1 Abstract

Bearing fault diagnosis is of great importance to decrease the damage risk of rotating machines and further improve economic profits. Recently, machine learning, represented by deep learning, has made great progress in bearing fault diagnosis. However, applying deep learning to such a task still faces major challenges such as effectiveness and interpretability: i) When bearing signals are highly corrupted by noise, the performance of deep learning models drops dramatically; ii) A deep network is notoriously a black box. It is difficult to know how a model classifies faulty signals from the normal and the physics principle behind the classification. To solve these issues, first, we prototype a convolutional network with recently-invented quadratic neurons. An overview of this method is illustrated in Figure B.1. This quadratic neuron-empowered network can qualify the noisy bearing data due to the strong feature representation ability of quadratic neurons. Moreover, we independently

derive the attention mechanism from a quadratic neuron, referred to as qttention, by factorizing the learned quadratic function in analogue to the attention, making the model made of quadratic neurons inherently interpretable. Experiments on the public and our datasets demonstrate that the proposed network can facilitate effective and interpretable bearing fault diagnosis.

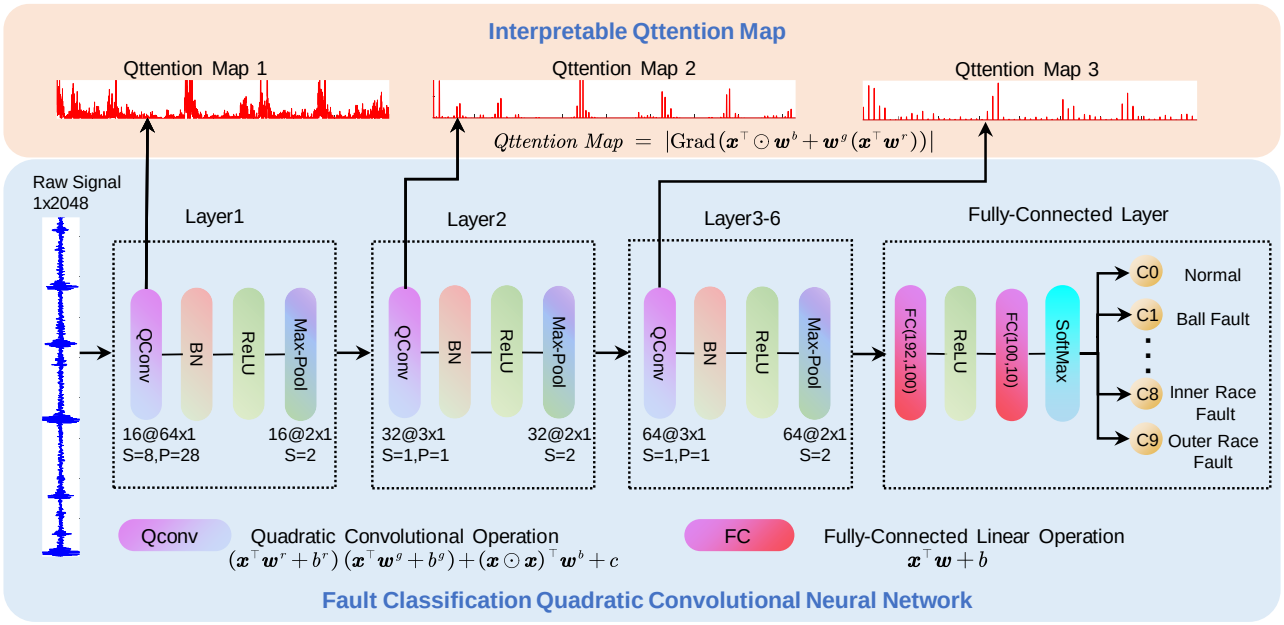


Figure B.1: The proposed quadratic convolutional neural network (QCNN).

Appendix C

A Neural Blind Deconvolution Ensemble Model for Train Transmission Systems Fault Diagnosis

This method, based on the ClassBD model proposed in Chapters 3, ranked 14th in the preliminary round and 23rd in the final round (Competition ID: PDC0026), at the IEEE 2024 Global Reliability & Prognostics and Health Management Conference Data Challenge. [1]. Related conference paper:

*"**J.-X. Liao**, J. Li, M. Zhang*, F.-L. Fan*, X. Zhang*. Uncertainty-Aware Heterogeneous Neural Blind Deconvolution Ensemble Network for Reliable System-Level Fault Diagnosis in Railway Transmission Systems. The sixth International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (Accepted). "*

C.1 Task description

The dataset used in the data challenge was provided by the National Key Laboratory of Advanced Rail Transit Autonomous Operation at Beijing Jiaotong University (Ding et al., 2024). The experimental platform was designed based on a real subway train bogie, with a scale ratio of 1:2. Fault simulation experiments were conducted to acquire multi-source sensor signals corresponding to various health conditions of the subway train transmission system.

The test rig consists of a single transmission chain, which includes a motor, a reduction gearbox,

[1]The competition results are available at https://mp.weixin.qq.com/s/wse2bUeYZBU12w3RE52Nog

and an axle box. The transmission chain is driven by a three-phase asynchronous AC motor, and the load is applied using a hydraulic system. The motor bearing is of type SKF6205-2RSH; the reduction gearbox features helical gears, with the driving gear having 16 teeth and the driven gear having 107 teeth. The support bearing for the driving gear is model 32305 from Harbin Bearing Factory, while the axle box bearing is model 352213, also from Harbin Bearing Factory.

The dataset simulates nine operating conditions by varying motor speed and lateral load. Different motor speeds correspond to various train operating speeds, while different lateral loads simulate straight-line and curve navigation. The experiment collected two types of signals: three-axis vibration and three-phase current from both the traction motor's drive and non-drive ends. Additionally, three-axis vibration signals were recorded from the gearbox's input and output shafts, as well as the left and right axle boxes, totaling 21 signal channels, each sampled at 64 kHz.

The preliminary dataset includes normal conditions and 16 types of single faults, such as bearing and gearbox faults. The final dataset introduces component-level and system-level compound faults, such as simultaneous gear tooth breakage and bearing failure in the gearbox, as well as combined gearbox and axle box bearing faults. The operating conditions of the training and test sets remain unknown, and the test set contains previously unseen fault combinations.

C.2 Model design

Our model adopts the NeuralBD proposed in Chapter 3 as its foundational architecture. To handle multi-channel data, an integrated model is designed to simultaneously process three-axis acceleration signals from different components. The diagnostic results of individual components are then synthesized to obtain the final fault diagnosis outcome, as illustrated in Figure C.1.

First, since this task involves processing fault signals from various components, each exhibiting distinct fault characteristics, multiple fault diagnosis sub-models are designed. Additionally, given that the operating conditions of the test set differ from those of the training set, we employ domain adversarial transfer learning Long et al., 2015 to enhance the generalization capability. The designed framework is well grounded in the domain adversarial neural networks (DANN) (Long et al., 2015). The loss function of it is defined:

$$\mathcal{L}_s = \mathcal{L}_{CE} + \lambda_1 \mathcal{L}_{DA} + \lambda_2 \mathcal{L}_{ADV}, \tag{C.1}$$

where, $\mathcal{L}_{CE}$ denotes cross-entropy loss, $\mathcal{L}_{DA}$ denotes domain adaptation loss, and $\mathcal{L}_{ADV}$ represents adversarial loss.

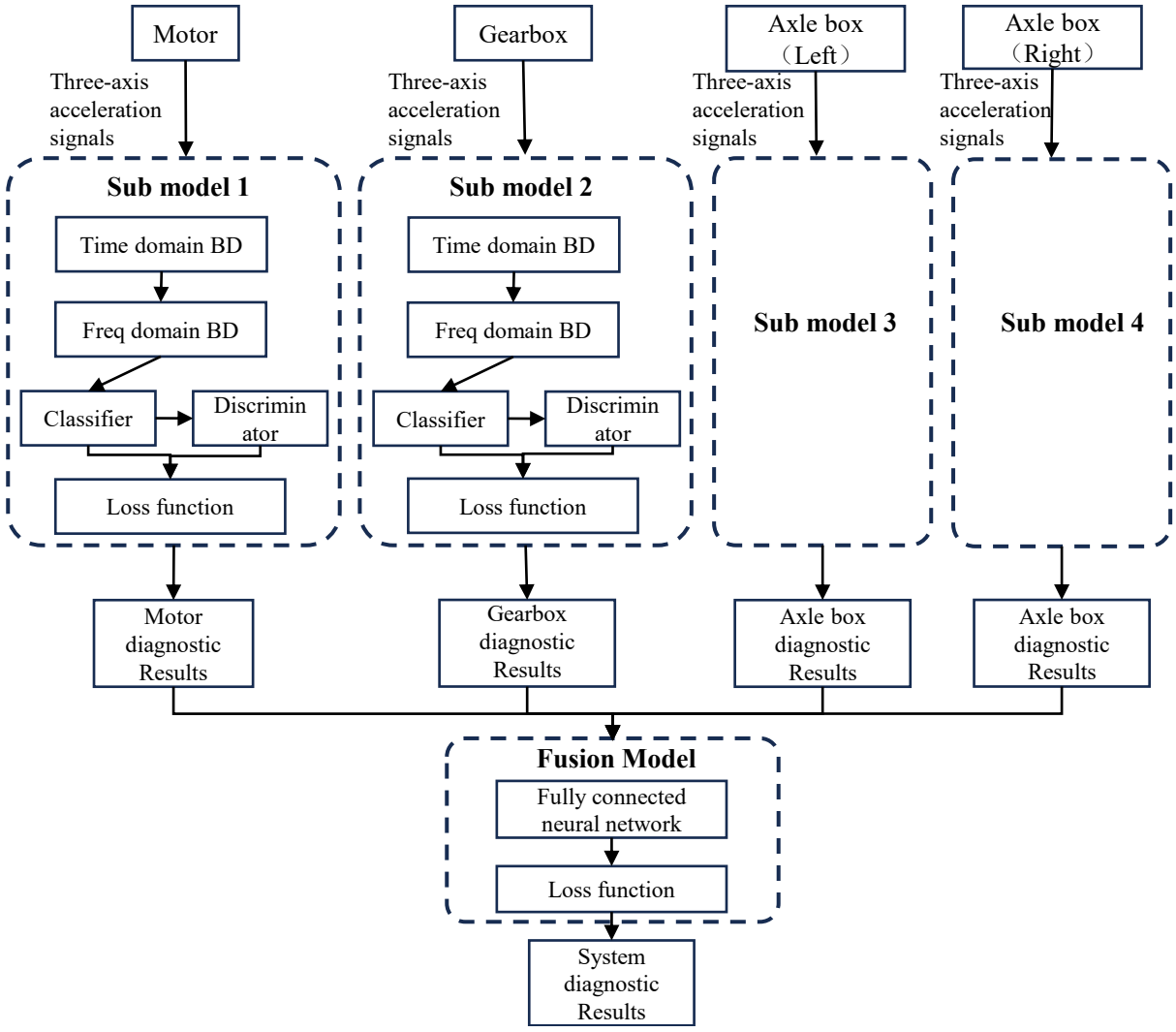


Figure C.1: Blind deconvolution ensemble model.

Second, a fully connected neural network is utilized to construct a result fusion module, which integrates the outputs of individual models to generate a comprehensive diagnosis for the entire mechanical system. This module mitigates inconsistencies between sub-model predictions and actual fault conditions. For instance, in real-world testing, sub-models may output diagnoses such as motor fault type 2, gearbox fault type 3, and axle box faults type 1 and type 0, even though such a fault combination does not exist in reality.

Finally, the integrated model follows an "end-to-en" framework, where the weighted sum of all sub-model loss functions is computed, and sub-model parameters are updated simultaneously. A weight of $\lambda_s = 0.1$ is assigned to ensure the neural network primarily focuses on the final output. The loss function is formulated as follows:

$$\mathcal{L} = \mathcal{L}_m + \lambda s(\mathcal{L}_{s1} + \mathcal{L}_{s2} + \mathcal{L}_{s3} + \mathcal{L}_{s4}), \tag{C.2}$$

where $\mathcal{L}_m$ represents the loss function of the result fusion module, and $\mathcal{L}_{si}$ denotes the loss function of each sub-model, where $i = 1, 2, 3, 4$.