DOCTORAL THESIS

Bearing Fault Detection Using Machine Learning on Vibration and Sound Signals

A thesis submitted to Brunel University London
in accordance with the requirements
for award of the degree of Doctor of Philosophy

in

Department of Electronic and Electrical Engineering

Tianhao Wang

September 4, 2025

Declaration of Authorship

I, Tianhao Wang, declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

(Signature of student)

Abstract

A bearing is a machine element that constrains relative motion to only the desired motion and reduces friction between moving parts, especially in high-speed trains, wind turbines, and railway wheelset applications. As the complexity of equipment and the harshness of operational environments in these industries increase, bearings, as critical components, are prone to faults that can lead to significant damage and costly maintenance. Therefore, accurately and in real-time detecting these faults is crucial for ensuring the safe and efficient operation of equipment. This research explores modern signal processing techniques and machine learning algorithms, integrating multimodal data such as vibration and sound, to propose efficient fault diagnosis models, providing valuable insights for future industrial intelligent maintenance systems.

Firstly, for fault diagnosis in high-speed train rolling bearings, this thesis introduces a model based on frequency domain feature extraction and a Bidirectional Long Short-Term Memory (Bi-LSTM) network. High-speed trains, operating under long periods of dynamic load and high-frequency vibrations, are prone to bearing fatigue damage, making early fault detection essential for safety. Traditional fault detection methods mainly focus on time-domain signal analysis, which often fails to capture key frequency domain features. To address this, the study first transforms time-domain signals into the frequency domain using Fast Fourier Transform (FFT), followed by fault classification using the Bi-LSTM model to identify fault types from both directions of sequential data. Experimental results demonstrate that the Bi-LSTM model based on frequency domain signals significantly improves fault classification accuracy compared to traditional methods, validating its effectiveness and superiority.

Secondly, a real-time monitoring system for wind turbine bearing fault detection is designed and implemented, utilizing a simplified neural network model running on a Raspberry Pi. Wind turbine bearings are subjected to high loads and harsh environmental conditions, and bearing faults can lead to turbine shutdowns, affecting energy production and increasing maintenance costs. To achieve real-time monitoring and fault prediction, a simplified neural network algorithm is designed, efficiently implemented on a Raspberry Pi. By segmenting sensor data, the system quickly analyzes each data segment and delivers predictions within milliseconds. This model demonstrates high fault detection efficiency and response speed, significantly reducing maintenance costs and improving operational

efficiency.

Finally, for railway wheelset bearings, this thesis proposes a new fault detection method based on Grey Wolf Optimizer (GWO) and Support Vector Machine (SVM). Wheelset bearings in railway vehicles endure complex loads and environmental pressures, making fault detection critical. Traditional single-source vibration signal analysis often falls short of capturing comprehensive fault characteristics. The study innovatively integrates vibration and sound data, providing a richer set of diagnostic information. The GWO is employed to optimize the SVM model's hyperparameters, resulting in a highly efficient fault detection method. The experimental results show that the proposed approach significantly improves fault detection performance compared to traditional models, achieving rapid and accurate predictions, thereby minimizing downtime and maintenance costs.

Overall, this research demonstrates the potential of various machine learning and optimization algorithms in bearing fault detection across different industrial applications. By integrating frequency domain signal processing, neural networks, and optimization algorithms, the thesis overcomes the limitations of traditional time-domain methods, significantly enhancing the accuracy and real-time capabilities of fault diagnosis. Specifically, the high-speed train bearing fault diagnosis method, based on FFT and Bi-LSTM, showcases superior performance in handling complex frequency domain signals. The real-time monitoring system for wind turbine bearings leverages lightweight neural networks on embedded devices, offering high efficiency and low latency. The vibration-sound fusion method for wheelset bearings, optimized by GWO, achieves better detection precision and speed. Future research could explore the integration of additional sensor modalities and combine deep learning algorithms with optimization techniques to further improve diagnostic accuracy and application breadth. Overall, the three methods proposed in this thesis offer effective solutions for intelligent maintenance systems and provide essential technological support for the safe operation and maintenance of industrial equipment in the future.

Acknowledgements

First, I'd like to thank my supervision team, who gives me great help in my PhD research, especially my principal supervisor, Prof. Hongying Meng, who is very serious on academic and has lots of ideas in research. Every time when I don't know how to continue my work, he always finds a new idea to help me with my work. prof. Hongying Meng has not only helped me on knowledge things like mathematics methods and programming, but also taught me how to do research and how to think. My second supervisor Dr. Lu Gan and Research Development Advisor Dr. Nikolaos Boulgouris also gives me lots of help on researches and life. It is an honor to have a great supervision team, and I learnt a lot from them.

In the end, I'd like to thank my family, who fully support me on all my decisions and researches. I appreciate their understanding and support on me.

Contents

List of Figures

List of Tables

List of Acronyms

ADCNN	Adaptive Deep Convolutional Neural Network
ANN	Artificial Neural Network
Bi-LSTM	Bidirectional Long Short-Term Memory
CMF	Combined Mode Functions
CNN	Convolution Neural Network
DB-LSTM	Deep Bi-Directional Long Short-Term Memory
DBN	Deep Belief Networks
DCNN	Deep Convolutional Neural Network
DFIG	Doubly-Fed Induction Generator
DFT	Discrete Fourier Transform
DWAE	Deep Wavelet Auto-Encoder
EDAES	Ensemble Deep Auto-Encoders
EEMD	Ensemble Empirical Mode Decomposition
ELM	Extreme Learning Machine
EMD	Empirical Mode Decomposition
F-MSCNN	Fusion Multi-Scale Convolutional Neural Network
GWO	Grey Wolf Optimizer
HDN	Hierarchical Diagnosis Network
IEA	International Energy Agency
IMF	Intrinsic Modal Functions
KPCA	Kernel Principal Component Analysis
LDA	Linear Discriminant Analysis
LE	Laplacian Eigenmaps
LLTSA	Linear Local Tangent Space Alignment
LLE	Locally Linear Embedding
LPP	Locality Preservation Projection
LSTM	Long Short-Term Memory Network
LTSA	Local Tangent Space Alignment
MFCC	Mel-Frequency Cepstral Coefficients
MLP	Multi-Layer Perceptron
MSCNN-BiLSTM	Multi-Scale Convolutional Neural Network with Bidirectional Long Short-Term Memory
MS-DCNN	Multi-Scale Deep CNN
MSI	Modified Smoothness Index
NEM	Non-negative EMD Manifold
OFDM	Orthogonal Frequency-Division Multiplexing

PCA	Principal Component Analysis
PFGI2	Power Function-based Gini Indices II
PFGI3	Power Function-based Gini Indices III
PNN	Probabilistic Neural Network
PSD	Power Spectral Density
RBF	Radial Basis Function
RNN	Recurrent Neural Network
SATLBO-MLP	Self-Adaptive Teaching-Learning-Based Optimization-Multi-Layer Perceptron
SCADA	Supervisory Control And Data Acquisition
SFAM	Fuzzy Auto Adaptive Resonance Theory MAP
SPL	Sound Pressure Level
SST	Synchro Squeezing Transform
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
VMD-DCNNs	Variational Mode Decomposition with Deep Convolutional Neural Networks

Chapter 1

Introduction

1.1 Background

In contemporary industrial and transportation networks, the dependability and safety of mechanical apparatus are essential. Bearings, as fundamental components of this equipment, provide the essential function of conveying motion and load. Bearing failure is a primary cause of equipment downtime and production disruption throughout high-speed railroads, wind power generation, aerospace, and diverse manufacturing gear. With increasing equipment operating speed, load, and environmental complexity, diagnosing and preventing bearing failures have become critically important. Timely and accurately identifying bearing issues can reduce maintenance costs, prevent unexpected equipment failures, and improve overall system stability.

The working environment of bearings is usually harsh, and they are often affected by high stress, vibration, temperature changes, and contaminant intrusion from the outside. This makes bearings prone to failures such as fatigue damage, wear, and corrosion during long-term operation. If these problems are not discovered and repaired in time, the bearing may fail seriously, leading to downtime for the entire equipment or system. Therefore, effectively detecting and diagnosing bearing failures, especially in discovering potential problems in the early stages, has become one of the key challenges in mechanical equipment maintenance.

In recent decades, bearing fault detection methods have evolved from traditional experience-based manual detection to automation and intelligence. Traditional detection methods mainly rely on analyzing vibration signals, especially time domain and frequency domain

analysis techniques. These methods identify anomalies by analyzing the vibration characteristics of bearings in normal and faulty states. However, traditional methods have the following limitations:

- Limitations of time domain signals: Although time domain analysis methods can capture the overall vibration characteristics of bearings, they cannot fully reflect the frequency components in the signal, especially since some types of faults are not obvious in the time domain.

- Singleness of frequency domain analysis: Although frequency domain analysis can reveal the frequency characteristics of signals through Fourier transform (FFT), it ignores the time dependence of signals and makes it difficult to accurately locate faults in dynamic environments.

- Limitations in data processing capabilities: Traditional signal analysis techniques often cannot cope with complex and diverse working conditions, and it is difficult to capture all fault modes in complex systems. In addition, as the complexity of modern industrial equipment increases, the collected signals are becoming more and more diverse, and traditional analysis methods make it difficult to cope with the real-time processing requirements of large amounts of data.

In order to address the limitations of traditional methods, intelligent algorithms such as machine learning and deep learning have been gradually applied to the field of fault detection in recent years. These methods can achieve more accurate prediction and diagnosis of faults by extracting features and patterns from a large amount of historical data. For example, deep learning methods such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks (LSTMs) have been widely used to process time series signals and have made significant progress in the accuracy of fault detection. However, there are still some challenges for existing intelligent algorithms, including:

- Limitations of single-modal signals: Many existing intelligent algorithms are still limited to processing a single type of signal (such as vibration signals or sound signals), and cannot fully reflect the various fault characteristics in complex systems.

- Real-time requirements: Although intelligent algorithms have improved detection accuracy, their high computational complexity leads to certain difficulties in real-

time detection in embedded or low-resource environments.

- Complexity of data fusion: Complex equipment faults usually involve multi-dimensional data, such as vibration, sound, temperature, etc. How to effectively fuse multiple data to improve the detection capability and robustness of the model is still an important research issue.

In complex industrial environments, a single signal type is usually unable to fully characterize the working state of bearings. For example, vibration signals can reflect the dynamic characteristics of mechanical motion, while sound signals can provide sound changes in the internal operation of the equipment. By fusing these different types of signals, the working state of the equipment can be comprehensively analyzed from multiple angles to obtain more accurate fault diagnosis results. Multi-modal data fusion technology can better capture the characteristics of different fault types by integrating data from multiple sensors, making the detection model more robust and flexible.

In recent years, many studies have begun to explore fault detection methods that combine multi-modal data fusion with intelligent optimization algorithms to solve the above problems. This type of method can not only comprehensively utilize the advantages of multiple signals such as vibration and sound, but also improve the performance of the model through optimization algorithms (such as genetic algorithms, grey wolf optimization algorithms, etc.), thereby achieving significant breakthroughs in the accuracy, real-time and applicability of fault detection. This trend represents an important direction for the development of fault detection technology from single signal processing to multi-dimensional data fusion, and promotes the further evolution of intelligent maintenance systems.

In summary, with the increasing complexity of industrial equipment and the demand for efficient and stable equipment operation, bearing fault detection technology is gradually moving from traditional single signal analysis to combining multi-modal data fusion and intelligent algorithms. This research direction not only has broad application prospects, but can also significantly improve the operating safety, reliability and maintenance efficiency of mechanical equipment.

1.2 Motivation

Bearings are core components in mechanical equipment, and their reliability directly affects the safety and operational efficiency of the equipment. However, due to the high load and complex working environment that bearings endure during operation, long-term vibration, wear, and fatigue may lead to bearing failure, thereby affecting the normal functioning of the entire system. Quickly and accurately detecting bearing faults can not only reduce downtime but also lower maintenance costs and extend the equipment's service life. Nevertheless, current fault detection technologies still face certain challenges and limitations, forming the primary motivation for further research.

Firstly, traditional fault detection methods mainly rely on vibration signal analysis, particularly single-signal processing methods in either the time domain or the frequency domain. However, these methods show limitations when handling complex operating conditions and diverse fault modes.

- Limitations of single signal type: Many traditional methods rely on a single time-domain or frequency-domain signal, but these signals often fail to fully reflect the fault characteristics of bearings under complex environments. For example, while time-domain signals can capture certain vibration patterns, they lack the resolution in the frequency domain. Conversely, frequency-domain signals may overlook some dynamic features over time. This results in low detection accuracy, especially for early fault diagnosis.

- Unmet real-time requirements: With the increasing demand for real-time monitoring of industrial equipment, achieving rapid, real-time fault diagnosis while ensuring high detection accuracy has become an urgent issue. Although traditional methods have shown good results in laboratory settings, they often fail to deliver sufficient computational efficiency in embedded or resource-constrained environments, making it difficult to meet real-time fault detection requirements.

- Insufficient utilization of multimodal data: Complex mechanical equipment faults usually exhibit multi-dimensional characteristics, such as changes in vibration, sound, and temperature signals. Traditional detection methods are mostly limited to single-signal processing and cannot effectively utilize multimodal data (e.g., combining vibration and sound signals), thus restricting the accuracy and robustness of fault detection.

Secondly, with the rapid development of Industry 4.0 and the Internet of Things, the need for intelligent monitoring and equipment maintenance is becoming increasingly intense. Companies aim to use intelligent fault detection systems to accurately diagnose and provide early warnings at the early stages of equipment failure, preventing major losses caused by bearing failure. To achieve this, considerable research is increasingly focused on two main areas:

- Application of intelligent algorithms: Machine learning and deep learning algorithms can automatically extract features and learn fault patterns through large-scale data training, promising to significantly improve the accuracy of fault detection. However, existing intelligent algorithms are computationally complex, making them difficult to apply in embedded systems. Additionally, their capability to process multimodal data has not been fully explored.

- Real-time optimization in embedded devices: Real-time detection systems for industrial equipment are often deployed in embedded environments with limited computational resources. Therefore, it is essential to design lightweight, real-time fault detection algorithms to meet the low-latency requirements of practical applications.

Additionally, analyzing single signals is insufficient to fully reflect the condition of bearing faults, especially in complex industrial environments where bearing failures often manifest through comprehensive changes across multiple dimensions. For example, the fusion of vibration and sound signals can provide more comprehensive fault information. Multimodal data fusion techniques can enhance the accuracy and robustness of fault detection by integrating signals from different sensors. Using intelligent optimization algorithms to further improve the performance of multimodal data fusion models has become an effective approach to enhancing fault detection capabilities.

Lastly, based on the above issues and requirements, the motivations of this thesis can be summarized as follows:

- Overcoming limitations in signal processing: The accuracy and robustness of fault detection can be enhanced by improving existing signal processing methods, particularly by converting time-domain signals into frequency-domain signals and integrating intelligent algorithms.

- Achieving real-time fault detection: Developing lightweight intelligent detection models to fit the computational capacity of embedded devices and meet real-time requirements, providing technical support for online fault diagnosis in industrial settings.

- Utilizing multimodal data fusion and intelligent optimization algorithms: By combining various signal data such as vibration and sound, and integrating intelligent optimization algorithms, the accuracy and efficiency of fault detection can be further improved, providing more comprehensive solutions for equipment monitoring in complex industrial environments.

In summary, the motivation of this thesis is to address the limitations of existing bearing fault detection methods by improving detection accuracy, real-time performance, and multimodal data fusion capabilities to meet the diverse demands of complex industrial environments. These improvements will not only enhance the reliability of equipment operation but also effectively reduce maintenance costs, advancing the application of intelligent fault detection technology in industrial practice.

1.3 Aim and objectives

The aim of this thesis is to design and implement a bearing fault detection system based on intelligent algorithms, capable of achieving high accuracy, real-time performance, and multimodal data fusion in complex industrial environments. By improving signal processing techniques, optimizing detection models, and integrating multiple fault signals, this thesis aims to address the limitations of existing fault detection methods in terms of accuracy, real-time capabilities, and adaptability. The specific thesis objectives are as follows:

- The first objective is to enhance fault feature extraction by improving existing signal processing methods, specifically by converting time-domain signals into frequency-domain signals. While time-domain signals can reflect overall trends of bearing faults, frequency-domain signals are more capable of capturing specific fault features in complex industrial environments. This thesis will employ the FFT to convert time-domain signals into frequency-domain signals and utilize a Bidirectional Long Short-Term Memory (Bi-LSTM) model to process these signals. Bi-LSTM has unique advantages in handling sequential data and can predict fault trends in both forward and backward directions. This combination will not only improve feature

extraction but also significantly enhance fault detection accuracy in complex working environments, overcoming the limitations of current detection methods.

- Real-time monitoring has become a core requirement in industrial equipment management. Therefore, this is necessary to design a fault detection model that can efficiently operate on embedded devices, ensuring low-latency and high-accuracy real-time fault detection in industrial applications. Given the computational constraints of embedded systems, it is essential to design a lightweight neural network model with minimal computational complexity. The second objective is simplifying the network structure and integrating efficient algorithm optimization techniques to enable fast model execution on embedded devices. This method allows the system to analyze sensor data in real-time and immediately trigger alerts upon detecting anomalies, minimizing potential downtime. This approach improves equipment efficiency and reduces unplanned maintenance costs caused by failures.

- A single signal type is insufficient to fully capture bearing fault characteristics, especially in complex industrial environments where signals such as vibration and sound provide multidimensional information. Therefore, multimodal data fusion becomes an essential approach for improving detection accuracy. The third objective is to apply multimodal data fusion techniques by combining data from various sensors, such as vibration and sound signals, to enhance the model's sensitivity to complex fault patterns. This thesis will utilize the GWO to optimize the hyperparameters of a Support Vector Machine (SVM) to improve the overall performance of the multimodal data fusion model. The GWO's global search capability allows it to find the optimal model configuration quickly, further enhancing fault detection accuracy and efficiency. Through collaborative multimodal data processing, the system can more accurately identify different types of faults, significantly improving detection robustness and adaptability.

Achieving these objectives will provide innovative solutions to the challenges in bearing fault detection, particularly addressing bottlenecks in signal processing, real-time embedded detection, and multimodal data fusion. The system developed through this thesis will not only improve fault detection accuracy but also achieve real-time performance on embedded devices while enhancing its adaptability in complex environments, offering robust technical support for the safe operation and maintenance of industrial equipment.

1.4 Thesis contribution

The main contributions of this thesis are as follows:

- A novel diagnostic approach that combines FFT-based frequency domain analysis with Bi-LSTM neural networks for detecting rolling bearing faults in high-speed trains. By transforming vibration signals into the frequency domain, the method overcomes the limitations of time-domain analysis, providing more effective fault detection. Using Bi-LSTM improves the ability to classify different types of faults, and validation on public datasets shows that the approach offers significant improvements over traditional methods. This contribution enhances the safety and reliability of high-speed trains by reducing the likelihood of bearing-related failures and improving operational safety.

- Real-time fault monitoring through the efficient processing of vibration data, both in wind turbines and transportation systems, ensuring minimal system downtime and enabling proactive maintenance strategies. The deployment of a real-time monitoring system for wind turbine bearings using a simplified Multi-Layer Perceptron (MLP) neural network on a Raspberry Pi. This system segments sensor data into smaller chunks, enabling fast and accurate fault detection while running efficiently on low-complexity edge devices. The MLP model is designed to work in real-world environments with limited computational resources, offering a practical, cost-effective, and scalable solution for wind turbine maintenance. This method supports the reliability of wind turbine operations, helping to reduce downtime improving system reliability, and extending the lifespan of equipment, and maintenance costs while promoting sustainable energy solutions.

- A multi-modal data fusion method based on the GWO algorithm and SVM was proposed, which successfully combined vibration and sound signals, significantly improved the performance of fault detection, and optimized the algorithm hyperparameters. The thesis highlights the importance of data fusion techniques, particularly the combination of vibration and sound signals, to enhance fault detection by leveraging complementary information from both sources. This fusion provides more robust and reliable fault identification compared to analyzing individual signals in isolation.

  The approach is validated through real-time analysis, showing that it can effectively detect faults with minimal latency, making it highly applicable to transportation systems. This contribution reduces maintenance costs and enhances operational efficiency by enabling timely fault detection, thus improving the reliability and safety of transportation infrastructure.

1.5 Thesis structure

This thesis has six chapters. The first chapter is an introduction which is this chapter, which includes the background, research questions, motivations, aims and objectives of my thesis.

Chapter 2 is a literature review, reviews the existing research in the field of bearing fault detection, analyzes the advantages and disadvantages of traditional methods and intelligent algorithms, and proposes the research framework of this thesis.

In chapter 3, the fault detection method based on frequency domain signal and Bi-LSTM, introduces the technical details and experimental results of converting time domain signals into frequency domain signals and using the Bi-LSTM model for fault detection.

In chapter 4, the real-time fault detection on embedded devices, describes how to implement efficient fault detection models on resource-limited embedded systems, as well as related system optimization and experimental evaluation.

In chapter 5, the multimodal data fusion method based on GWO-SVM, introduces the multimodal data fusion method combining grey wolf optimization algorithm and support vector machine, and its application in fault detection and experimental results.

Finally, the last chapter is the conclusion and future prospects, summarizes the contributions of this thesis, and looks forward to possible future research directions.

1.6 List of publications

[1] **Wang, T.**, Meng, H., Zhang, F. and Qin, R. Fault Detection of Wheelset Bearings through Vibration-Sound Fusion Data Based on Grey Wolf Optimizer and Support Vector Machine. Technologies 2024, 12, 144. https://doi.org/10.3390/technologies12090144

[2] **Wang, T.**, Meng, H., Qin, R., Zhang, F. and Nandi, A.K. Real-Time Monitoring of Wind Turbine Bearing Using Simple Neural Network on Raspberry Pi. Appl. Sci. 2024, 14, 3129. https://doi.org/10.3390/app14073129

[3] **Wang, T.**, Qin, R., Meng, H., Li, M., Cheng, M. and Liu, Y. (2022, October). Frequency Domain Feature Extraction and Long Short-Term Memory for Rolling Bearing Fault

Diagnosis. In 2022 International Conference on Machine Learning, Control, and Robotics (MLCR) (pp. 72-77). IEEE.

Chapter 2

Literature Review

In this chapter, I will systematically explore the key components and monitoring technologies of two modern engineering systems: high-speed trains and wind turbines. First, I will provide a brief introduction to the core components of high-speed trains and wind turbines, analyzing their design and operational characteristics. The focus will then shift to one of their critical elements—bearings—examining their structure, functions, and performance under various operating conditions.

Subsequently, I will delve into the sensor technologies used for bearing monitoring, including vibration, temperature, and acoustic sensors.

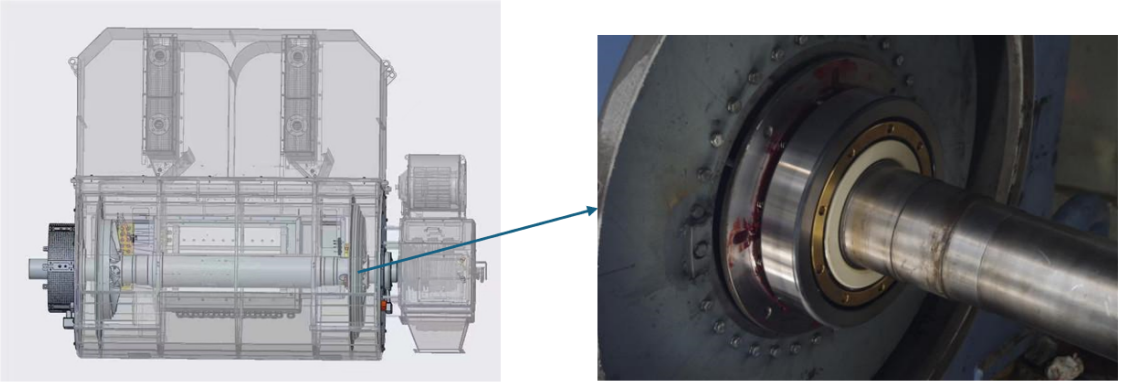
Building on this foundation, I will further analyze signal preprocessing techniques. Following this, feature extraction methods will be introduced, ranging from traditional time-domain and frequency-domain features to data-driven automatic feature extraction techniques. Finally, I will examine various classification methods, including traditional machine learning algorithms such as SVM, and deep learning models such as Bi-LSTM, evaluating their effectiveness in bearing fault diagnosis within the context of high-speed trains and wind turbines based on application scenarios and performance metrics.

2.1 Bearing

2.1.1 Introduction of bearing

High-speed trains and wind turbines, though operating in different domains, share a common requirement: the need for efficient mechanical systems to convert energy into motion or electrical power. Within these complex mechanical systems, there is an indispensable core component—the bearing. Whether it is the smooth rotation of high-speed train wheels or the efficient operation of wind turbine blades, bearings play a critical role.

As shown in Figure 2.1.



(a) Generator (b) Bearing

Figure 2.1: Generator bearing. (**a**) Generator; (**b**) bearing.

The basic functions of a bearing include support, friction reduction, load transmission, and efficiency improvement. Bearings support rotating or moving components, ensuring they move in a fixed direction; reduce friction between components through rolling or sliding mechanisms; withstand radial and axial loads (or a combination of both) and transfer them to other mechanical parts; and minimize energy loss to enhance mechanical efficiency.

2.1.2 Sensor of bearing

Introduction

Sensors play an important role in both industrial production and scientific research. In modern scientific research, there are many things to observe: the vast universe, invisible particles, long celestial evolution, and instants in seconds. In addition, there is research on high-tech materials, the development of new energy sources, and the deepening understanding of matter, which are not directly accessible to humans. Therefore, sensors

are indispensable. Many scientific researches are based on the acquisition of external information. Some high-quality sensors often make breakthroughs in some science and technology. The development of sensors can also be considered the core of some scientific research [6].

More importantly, sensors are important in a variety of industries, such as industrial production, space development, ocean exploration, environmental protection, resource surveys, medical diagnostics, bioengineering, and even the protection of cultural relics. In modern society, almost every project is inseparable from sensors. Sensors have many characteristics, such as miniaturization, digitization, intelligence, multifunctionality, systematization, and networking. It is the main support point for economic growth in the 21st century [1].

Therefore, sensors in bearings are used for real-time condition monitoring, supporting predictive maintenance, optimizing operational efficiency, ensuring safety, and providing data analysis to enhance equipment performance and life [7].
The sensor composition is shown in Fig. 2.2.

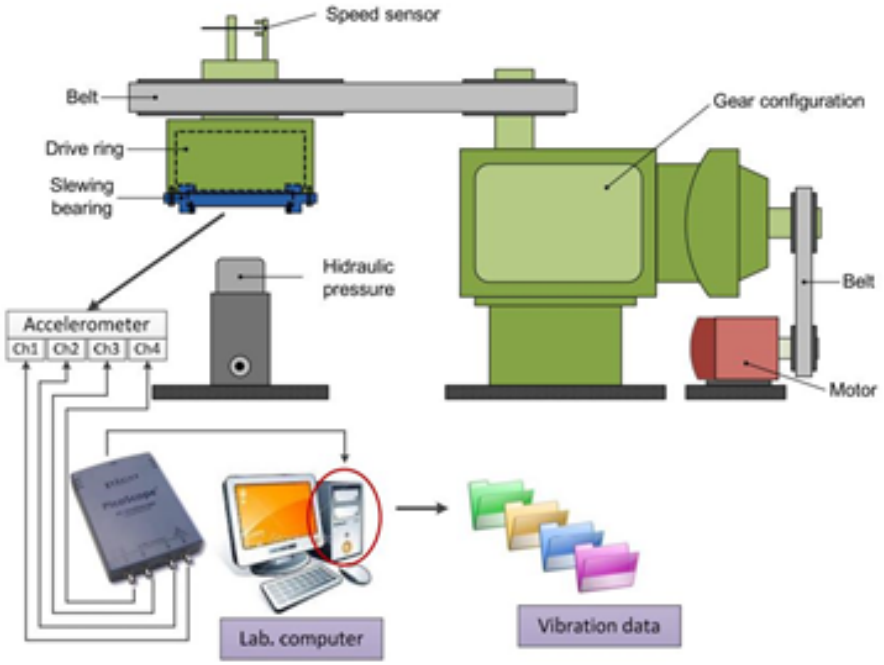


Figure 2.2: Sensor composition [1], is a sensor-based system for monitoring and analyzing mechanical equipment's performance through vibration data acquisition and processing.

Mainly sensor in detection system

Vibration, sound, and temperature are the most commonly used key indicators in bearing monitoring. Monitoring these indicators comprehensively helps identify problems early,

prevent downtime, and enhance the efficiency and reliability of the equipment.

Acoustic sensor

The acoustic sensor functions the same as a microphone. It is used to receive sound waves and record the oscillation pattern of sound [8]. Generally, the acoustic sensor is used to detect the position, speed, etc. of the vehicle [9]. In this thesis, an acoustic sensor and a vibration sensor are combined to use the acoustic sensor to detect the location of a train failure.

Vibration sensor

A vibration sensor is a sensor that can sense the parameters (vibration speed, frequency, acceleration, etc.) of mechanical motion vibration and convert it into a usable output signal [8]. When some parts of the train are abnormal, its vibration mode will change slightly. However, the driver finds it hard to notice the vibration during the driving. The vibration sensor can detect these abnormalities, hence the vibration sensor is significant for all vehicles [10].

Temperature Sensor

Temperature sensors are usually electronic components that convert temperature into electronic data. A temperature sensor is made of a conductor whose resistance changes with temperature. The element most commonly used is platinum, which has a resistance of 100 ohms (Pt100) at 0 ° C. Semiconductor temperature sensors are usually integrated with amplification and regulation circuits. The oscillation frequency of the crystal oscillator changes with temperature, so the temperature can be measured very accurately [11].

For thermocouples that use the thermoelectric effect to measure temperature, the surface charge density of the thermoelectric material changes with temperature, so the surface charge intensity can be used to measure temperature.

From the early 17th century, people began to use them to make some measurements, modern temperature sensors are minuscule, making them widely used in various fields of production practices, including high-speed railway monitoring.

2.1.3 Application scenarios of bearing

High-speed trains

As an important component of the bogie, the high-speed train transmission system is composed of traction motors, gearboxes, axle boxes, bearings and other key components. It is mainly responsible for power driving and power transmission. The power transmission route is as follows: the torque output of the traction motor is through the meshing gear. It is transmitted to the wheelset to drive the train [12].

The operating environment of the high-speed train transmission system is harsh. When the train is running at high speed, it has to not only generate and transmit the driving force and the driving torque but also bear the disturbance of the track irregularity. Long-term high-frequency vibration makes it easy for the key components of the high-speed train transmission system to appear. Fatigue damage affects the safe operation of the train.

Motor

The traction motor is the power unit of the high-speed train and plays an important role in the transmission system. Motor bearing failures account for 40% of motor failures [13]. As shown in Figure 2.3 [2]. Common motor bearing fault types are wear, fatigue peeling, fracture, corrosion, and bonding.

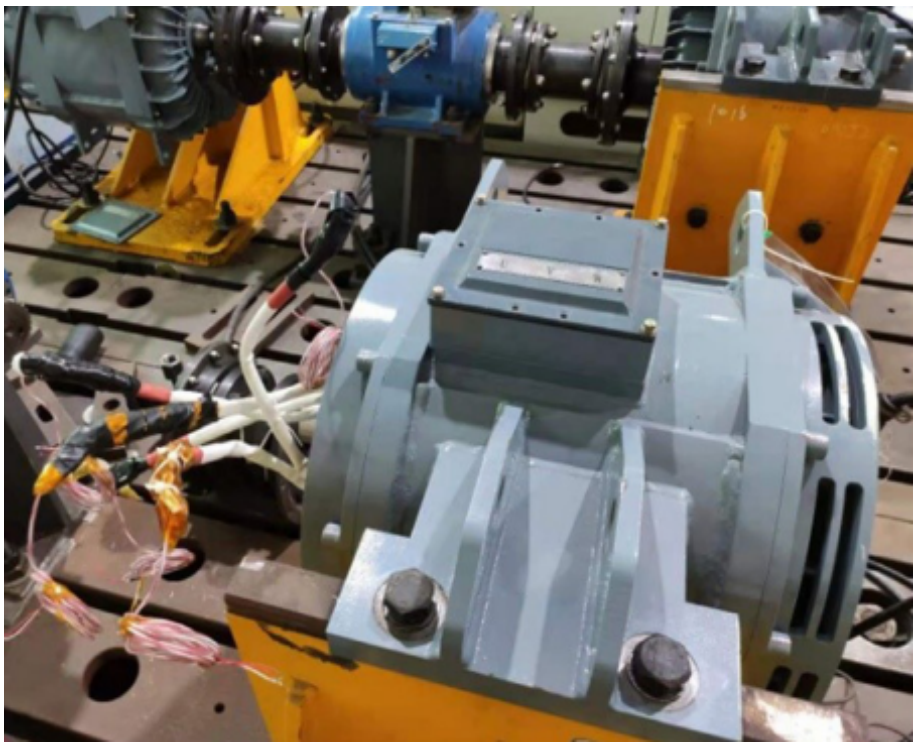


Figure 2.3: Traction motor [2], is the power unit of the high-speed train.

Bearing wear failure is mainly caused by improper installation or poor lubrication. The surface of the bearing element is rough and wears during operation.

Fatigue spalling is the most typical fault of motor bearings, which is mainly manifested as partial damage or fall off on the inner ring, outer ring, rolling element and other surfaces of the bearing. The main cause of bearing fatigue spalling is the fatigue stress under alternating loads. When the bearing has fatigue spalling failure, it generates shock pulses at specific frequencies.

Fracture is one of the more severe forms of bearing failure, primarily caused by factors such as poor machining, excessive load, prolonged fatigue damage, and inadequate lubrication.

Bearing corrosion failures are mainly divided into chemical corrosion and electrical corrosion. Chemical corrosion is mainly caused by the presence of moisture in the lubricating oil. Electrochemical corrosion refers to corrosion induced by electrochemical reactions. Bearings affected by chemical corrosion develop rust on their surfaces, while those subjected to electrochemical corrosion exhibit surface pitting, resembling the characteristics of fatigue spalling.

Glueing failure refers to the occurrence of adhesion on the bearing surface. Glueing failure is mainly caused by the rise in temperature caused by a too-high speed or poor lubrication.

Gearbox

The gearbox is a key component of the high-speed train transmission system, as shown in Figure 2.4 [3]. It consists of gear pairs, box bodies, bearings, transmission shafts and other components. The main function is to reduce speed, increase torque and transmit power. In the process of gear meshing, internal excitation and external excitation are generated. The internal excitation refers mainly to the excitation of the gear mesh. When the train is running at high speed, the frequency of alternating loads of the mesh increases, the excitation of the mesh and the external load received by the gears increase, and it is easy to cause gear failures under the action of high frequency alternating loads for a long time [14]. In addition, the gearbox experiences vibrations from internal excitations due to gear meshing and external excitations from track irregularities transmitted through the transmission shaft and bearings. This results in the bearings and gearbox casing being subjected to random alternating loads over time, making them susceptible to failure. Common failures in high-speed train gearboxes include gear failure, bearing failure, and casing failure. [15].

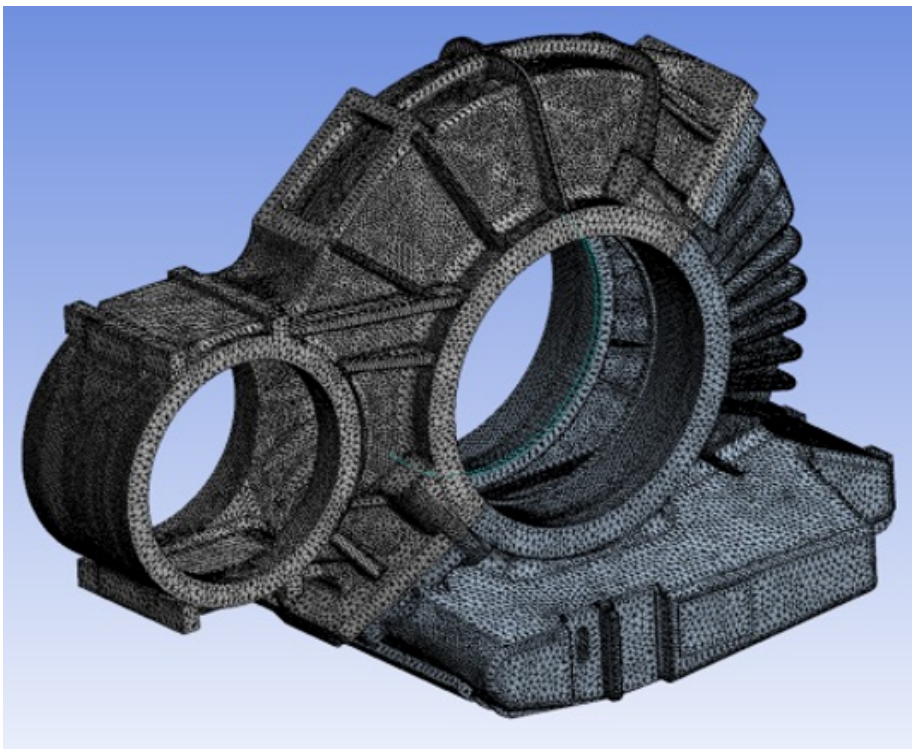


Figure 2.4: Gearbox [3], is a key component of the high-speed train transmission system, composed of gear pairs, box bodies, bearings, transmission shafts and other components.

- Common types of gear failure mainly include tooth surface wear, pitting, spalling, fatigue cracks, and tooth breakage [16–19].

  Gear wear is caused by the friction between the gear surfaces when the gears are meshed. Normal wear will not cause impact signals and modulation signals. However, when the abrasion is severe, an impact signal will be generated, and the meshing frequency and its higher harmonic frequencies will appear when reflected in the frequency domain spectrum of the vibration signal. Tooth surface wear is mainly related to poor lubrication [20–22].

  Gear pitting failure is primarily caused by excessive contact stress and repeated loading. When gear pitting occurs, periodic impact signals appear in the time-domain waveform of the vibration signal. In the frequency domain, this manifests as a modulation phenomenon, where the rotational frequency of the shaft modulates the meshing frequency on either side of the faulty gear [17, 23].

  Gear tooth breakage is a more serious type of gear failure. Gear tooth breakage are usually caused by two reasons, one is that it receives an impact or load that exceeds their carrying capacity, and the other is that the alternating stress during meshing

exceeds the fatigue limit of the tooth surface. When the gear is broken, the time-domain waveform will have regular impact components, and the frequency-domain spectrum will have a modulation phenomenon centred on the meshing frequency and the rotation frequency of the shaft where the gear is located [24].

- The common faults in high-speed train gearboxes include cracks, oil leakage, and excessive oil temperature. Gearbox cracks are primarily caused by cyclic reciprocating stress that exceeds the material's stress limit. The causes of cracks include internal issues, such as casting process problems that lead to uneven wall thickness and stress distribution, and external factors, such as wheel irregularities and resonance from the alignment of the gearbox's natural frequency with external forces. These combined factors increase the likelihood of cracks, affecting the gearbox's reliability and performance [25, 26].

  The gearbox oil leakage failure will cause poor lubrication of the gearbox and accelerate the wear of the gear pair. The main reason for the gearbox oil leakage may be related to the lack of sealing. The high oil temperature of the gearbox may be related to the quality of the lubricating oil, or it may be related to the wear of gears, bearings and other components [27].

- Gearbox bearing failure types are similar to motor bearing failure types, with a key difference being that the vibration frequencies of various gearbox components can interact with each other. For instance, the frequencies generated by the contact between the rolling elements and the inner and outer rings of the gearbox bearing may interfere with one another, especially when operating at high speeds. This interaction can propagate vibrations throughout the gearbox casing, potentially impacting the entire system's stability and amplifying wear or damage [28].

Axle box

The axle box and the axle box bearing together form the axle box device, which is responsible for transmitting the weight and load of the car body to the wheelset, as shown in Figure 2.5 [4]. This part of the work is mainly completed by the axle box bearing. The axle box bearing runs at high speed and heavy load for a long time. Easy to cause fatigue damage. The fault types of axle box bearings are the same as those of motor bearings, including fatigue spalling, pitting, wear, and fracture. According to the statistical analysis of the failure of the axle box bearing in the literature [29], the axle box bearing is very prone to raceway peeling failure, especially the outer raceway fault, which accounts most

of the axle box failure, which is the most frequent type of failure of the axle box bearing. When the bearing peels off, it will produce periodic impact components, and the main vibration characteristics shown are:

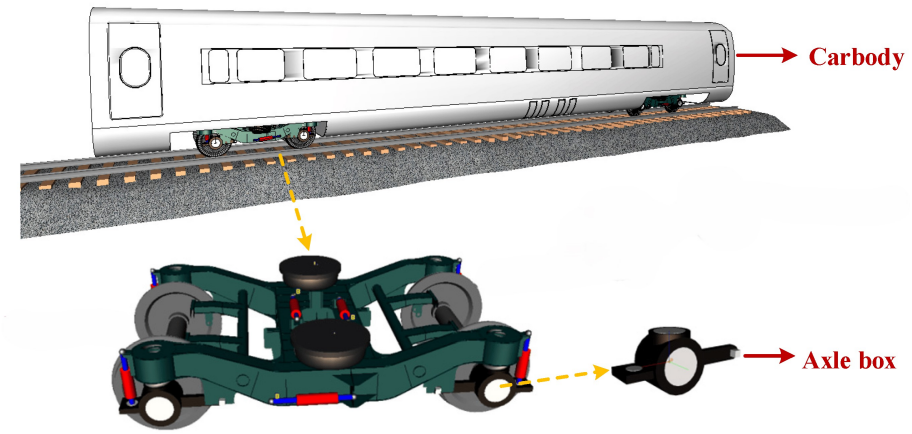


Figure 2.5: Alexbox [4], is responsible for transmitting the weight and load of the car body to the wheelset.

The bearing's outer ring experiences spalling. When this occurs, periodic pulses with the characteristic frequency of the outer ring fault appear in the time domain. In the frequency spectrum, these pulses manifest as spectral lines centered around the natural frequency of the outer ring, with clear separation at the fault frequency of the outer ring [30].

The inner ring of the bearing is peeled off. Since the inner ring of the axle box bearing is connected to the axle, the characteristic frequency of the inner ring will be modulated by the rotation frequency of the axle. In the spectrogram, there will be a spectrum centered on the natural frequency of the inner ring of the bearing, a spectrum separated by the fault characteristic frequency of the inner race, and a spectrum separated by the rotation frequency [30, 31].

For roller failure, there will be a spectrum centered on the natural frequency of the rolling element, and a spectrum separated by the frequency of the rolling element failure, and a spectrum separated by the revolution frequency of the rolling element [32].

Wheelset

The wheelset is the part of the rolling stock that is in contact with the steel rail. It consists of two wheels on the left and right firmly pressed on the same axle, as shown in Figure 2.6. The role of the wheelset is to ensure the running and steering of the rolling stock on the rail, bearing all the static and dynamic loads from the rolling stock, transmitting it to the rail, and transmitting the load caused by the uneven track to the parts of the rolling stock. In

addition, the driving and braking of rolling stock are also affected by wheel sets. There are strict requirements on the assembly pressure and press-fitting process of axles and wheels. The distance between the inner sides of the wheels must be within 1353 mm [33]. In order to ensure the smooth running of locomotives and vehicles, reduce the wheel-rail interaction force and running resistance, the machining ovality and eccentricity of axle journals and wheel treads, as well as the journal taper, shall not exceed the prescribed limits.

Figure 2.6: Wheelset, is the part of the rolling stock, consists of two wheels on the left and right firmly pressed on the same axle.

Wheelset bearings have large axial and lateral clearances, which aggravate the vibration of the bearing rolling elements, and at the same time cause the impact of the bearing rolling elements and raceways to increase, and it is easy to cause uneven load distribution between the inner ring and the rolling elements. When the bearing is subjected to excessive load or impact, the bearing will undergo plastic deformation. If it is coupled with poor lubrication and high-speed operation, the temperature of the bearing will rise sharply, and the surface metal will be glued, and the bearing will appear indentation or spalling area. Damage or fracture of the bearing cage will increase the friction between the cage and the rolling elements, and may also cause the rolling elements to jam and cause bearing damage. As the vehicle runs at high speed, most of the falling materials or impurities in the external environment roll with the rolling elements, and the change in the gap between the cage and the rolling elements will accelerate the wear of the rolling bearings [34]. Under the action of alternating loads, the internal structural parts of the bearing move relative to each other, which may cause structural internal cracks. The expansion of the cracks to the contact surface will increase the impact load, vibration and noise of the bearing during operation [35, 36]. It can be seen that any damage and failure of the wheelset bearing will result in the reduction of bearing operation accuracy, intensified shock and vibration, and a sharp rise in temperature. There are hidden safety hazards such as hot shafts, short shafts, shaft cutting, and motion interruptions, which will seriously affect the train. Ride comfort and driving safety. At present, high-speed trains run at a high speed, and the vibration

state of various structural components is aggravated, and the deterioration rate is rapid. Wheelset bearings have been used in harsh vibration time-varying environments for a long time, and they are prone to burn, corrosion, peeling, pitting, code skins, and dents. Faults such as marks, cracks, bruises, scratches, deformations, etc.

Wind turbine

A wind turbine is a device that generates electricity by harnessing wind energy. It converts the kinetic energy of the wind into mechanical energy, which is then further converted into electrical energy. The main components of a wind turbine typically include a rotor, generator, pitch control system, nacelle, and tower [37–39].

The working principle of a wind turbine is based on aerodynamics. The wind drives the rotation of the rotor blades, which capture the wind's kinetic energy and generate mechanical power. This power is transmitted through a drivetrain to the generator, which produces electric current by cutting through a magnetic field, ultimately generating electricity.

Wind power has several advantages, including being a clean, environmentally friendly energy source that produces no greenhouse gases or waste emissions. It utilizes renewable wind resources, reducing reliance on fossil fuels, and has relatively low operating and maintenance costs once installed. However, it also has drawbacks, such as intermittency due to variable wind speeds, which can lead to inconsistent power output. Additionally, wind turbines may produce noise and have a visual impact on the landscape, and large-scale wind farms require substantial land, which can affect local ecosystems [40–44].

Wind power technology is widely applied around the world, particularly in areas with abundant wind resources. In many countries, offshore wind farms are being developed to harness the strong, steady winds over the sea [45, 46].

Rotor

The rotor is the main component responsible for capturing wind energy in a wind turbine, consisting of multiple blades designed to convert the wind's kinetic energy into mechanical energy. The design of the rotor blades is especially critical, as they must efficiently initiate rotation at low wind speeds while maximizing energy capture across a wide range of conditions. Blades are typically designed with aerodynamic profiles similar to airplane wings, creating lift as wind flows over them. This lift force drives the rotation of the rotor, which is connected to the turbine's generator. The number, length, and shape of the blades are carefully optimized to balance power output, durability, and cost, making rotors one of

the most technically demanding components of wind turbine design. Advanced materials, such as carbon or glass fiber composites, are often used to ensure strength, reduce weight, and improve overall efficiency. Larger rotors can capture more energy, which is especially advantageous for low-wind sites or offshore installations where stable, high wind speeds are prevalent [37, 39, 47]. As shown in Figure 2.7.

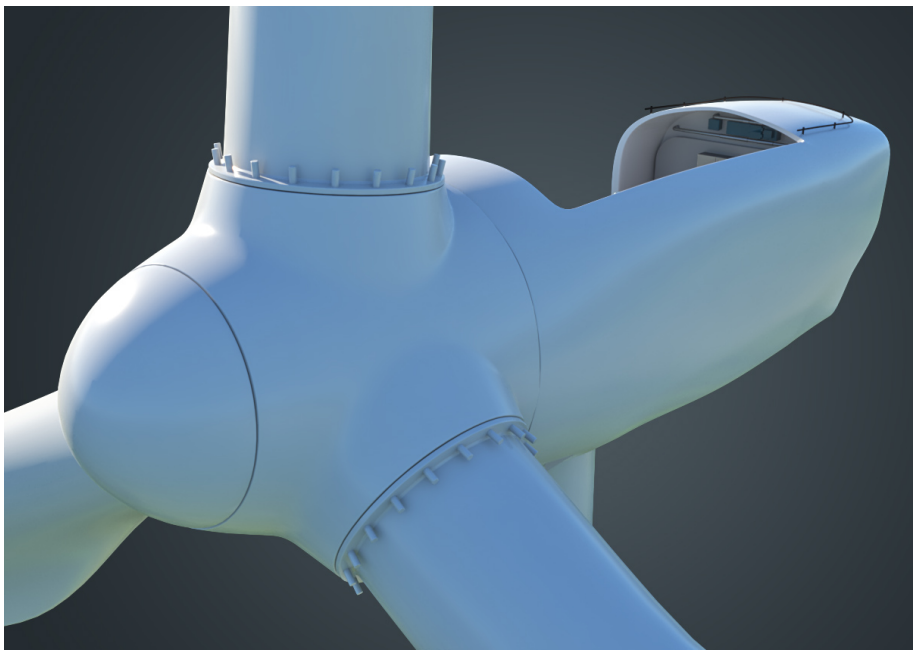


Figure 2.7: Wind turbine rotor, is the key component responsible for capturing wind energy in a wind turbine, consisting of multiple blades.

Generator

The generator in a wind turbine is the core component responsible for converting the mechanical energy captured by the rotor into electrical energy. Its performance directly determines the efficiency, stability, and cost-effectiveness of a wind power system.
A wind turbine generator typically comprises the following components:

- Stator: The stationary part of the generator, usually consisting of an iron core and windings. The induced current in the windings determines the output voltage and power. The design of the stator is crucial for minimizing losses and enhancing efficiency.
- Rotor: The rotating part, which can include permanent magnets (as in permanent magnet synchronous generators) or an excitation winding to create a magnetic field (as in electrically excited synchronous generators and induction generators). The rotor design affects the generator's magnetic field strength and stability.

- Bearings and cooling system: Bearings support stable rotation, while the cooling system dissipates heat to prevent overheating during prolonged operation [48].
- Power electronics equipment: Modern wind generators often include converters and rectifiers to adjust the frequency and voltage of the output current, meeting grid requirements.

For generator, Doubly-Fed Induction Generator (DFIG): DFIGs integrate power electronics with the rotor, allowing variable-speed operation. This enables optimal efficiency across a range of wind speeds. As shown in Figure 2.8, DFIGs offer high efficiency, lower maintenance costs, and are a popular choice in modern wind power systems [49–51].

Figure 2.8: Wind turbine generator is the core component responsible for converting the mechanical energy captured by the rotor into electrical energy.

Pitch control system

The pitch control system is a crucial component of a wind turbine, used to adjust the angle of the turbine blades to control the rotor speed and power output. By adapting blade angles to varying wind speeds, the pitch control system optimizes power generation and protects the turbine. This system not only enhances the efficiency of the wind turbine but also positions the blades safely during high wind speeds to prevent equipment damage [52, 53]. As shown in Figure 2.9.

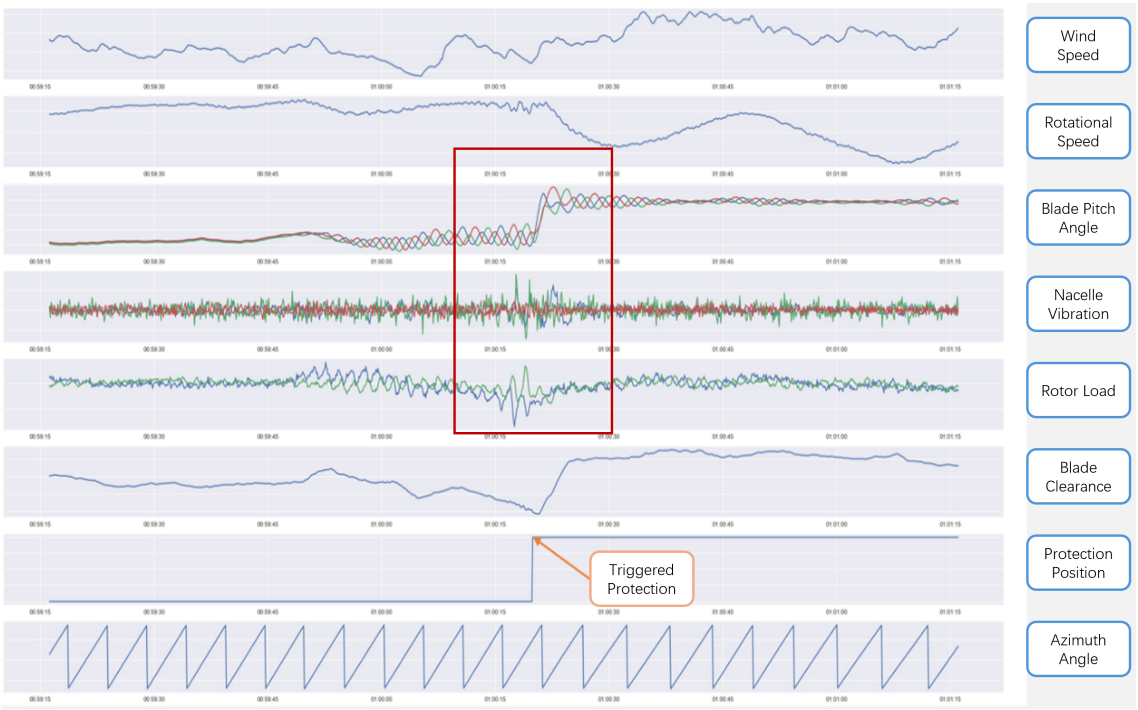


Figure 2.9: Pitch control system, is a crucial component of a wind turbine, used to adjust the angle of the turbine blades to control the rotor speed and power output.

Working Principles and Functions of the Pitch Control System:

- At lower wind speeds, the pitch control system adjusts the blade angle to capture as much wind energy as possible, ensuring maximum power output. As wind speed increases, the system gradually changes the blade angle to prevent the rotor from overspeeding, maintaining stability in the power generation system [54].
- When wind speeds become too high, the pitch system can position the blades to a "stall" angle, reducing their wind-facing area to significantly decrease rotor load. This "stall protection" prevents excessive wind forces from damaging the generator and mechanical components, ensuring the turbine's safe and stable operation in extreme weather conditions [55].
- The pitch system also helps minimize noise and mechanical load. By adjusting the blade pitch angle appropriately, the system can reduce noise output from high-speed rotation, which is especially beneficial for wind farms near residential areas. Additionally, pitch control reduces blade vibration in turbulent airflows, extending the turbine's operational life [38, 56, 57].
- Active pitch control uses hydraulic or electric servo systems to adjust the blade angle in real-time with high precision. Though it offers greater accuracy, it also requires higher costs and maintenance.

Passive pitch control relies on aerodynamic or mechanical mechanisms to automatically adjust blade angles. It is simpler and lower-cost, though less precise, and is typically used in smaller wind turbines [58].

Nacelle

The nacelle is one of the core structures of a wind turbine, located at the top and typically mounted above the tower. Its primary function is to protect and support the turbine's essential components, ensuring the smooth operation of the wind turbine. The nacelle houses complex and vital systems, including the generator, gearbox, control system, cooling system, and braking system, all of which contribute to its intricate structure and diverse functions [37]. As shown in Figure 2.10.

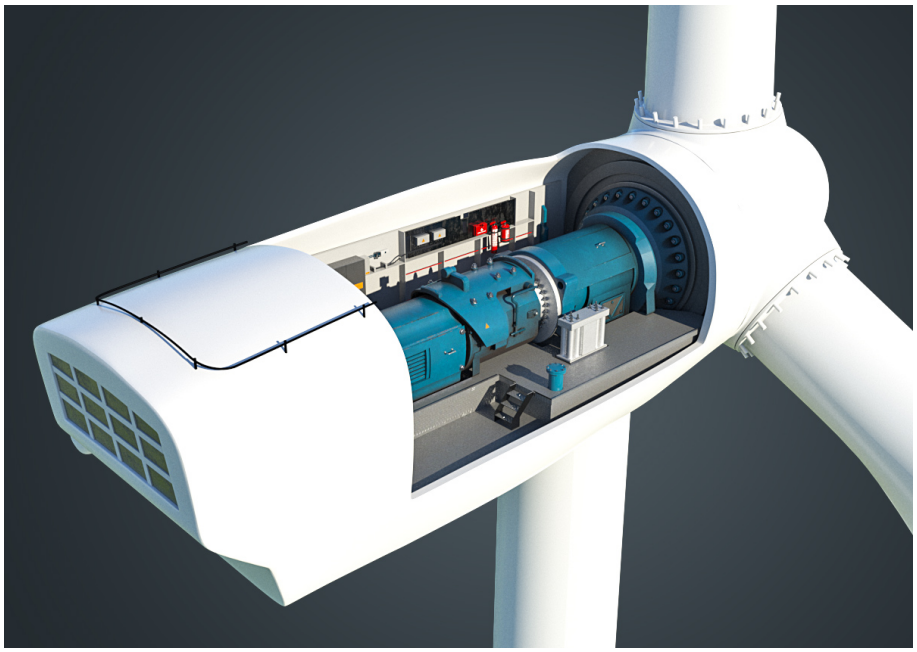


Figure 2.10: Nacelle, is located at the top and typically mounted above the tower to protect and support the turbine's essential components.

The nacelle faces several technical challenges in design, given its need to operate continuously under harsh weather conditions:

- The nacelle requires robust sealing to prevent rain, dust, and corrosive air from entering. This is especially crucial for offshore wind farms, where nacelles must have enhanced sealing and anti-corrosion features to withstand marine environments [39].
- The weight of the nacelle directly affects the structural requirements of the tower and foundation, making lightweight design essential. The use of composite materials and lightweight metal alloys is a growing trend in modern nacelle design, allowing for reduced weight without compromising strength.

- Despite limited space within the nacelle, it must accommodate a variety of complex equipment. The design must prioritize ease of access for repairs and component replacements, reducing maintenance time and costs.
- With high-speed rotating components inside the nacelle, controlling vibration and noise is a critical design factor. This is typically achieved through damping systems and soundproofing materials to minimize vibration and noise levels within the nacelle [38, 59].

Tower

The tower is a critical component of a wind turbine, supporting the entire turbine structure and elevating the rotor and nacelle to an optimal height to capture more wind energy. The design height and structural type of the tower directly impact the efficiency and stability of the wind turbine. Generally, the higher the tower, the greater the wind speeds it can capture, thus increasing energy output. However, taller towers also present challenges in terms of materials, manufacturing, and transportation [38, 60]. As shown in Figure 2.11.

Figure 2.11: Tower, supports the entire turbine structure and elevates the rotor and nacelle to an optimal height to capture more wind energy.

Key Design Considerations for the Tower:

- Tower height has a direct impact on power generation efficiency. Generally, the taller the tower, the higher the wind speeds it can capture, which can significantly increase power output with each additional meter in height. However, increased height also raises manufacturing and maintenance costs, necessitating a balance between height and cost.
- Tower materials are typically high-strength, corrosion-resistant steel or concrete. For offshore wind farms, materials must also be resistant to corrosion and wave

impact. Recently, high-strength composite materials and lightweight metals have been introduced in tower design to reduce weight and enhance strength [61].

- The tower must withstand environmental stresses from strong winds and seismic activity, making wind and seismic resistance critical. Design solutions often include dampening devices or thicker materials to mitigate the impact of wind and seismic forces on the tower [62, 63].
- Tower design must also address vibration and noise. Due to the tower's height and the rotational forces from the rotor, vibration and low-frequency noise can be generated. To reduce environmental impact, towers are commonly equipped with dampers or damping materials to minimize vibration [64, 65].
- Transporting tall towers to wind farm sites and assembling them can be costly and complex. For remote locations, segmented transport and on-site assembly methods are often used to reduce transportation costs and logistical challenges.

2.2 Fault detection of Bearing

In the previous section, I introduced the various sensors commonly used in modern engineering systems, such as vibration, temperature, and acoustic sensors. These sensors play a critical role in capturing raw data that reflect the operational state of key components like bearings. Faults in bearings often lead to changes in these sensor readings, as shown in Figure 2.12.

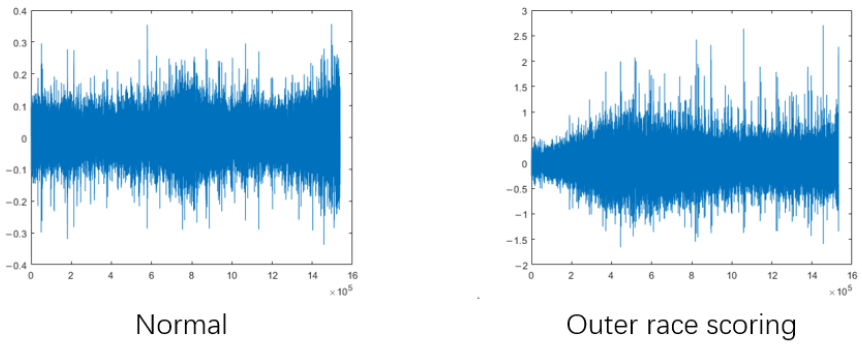


Figure 2.12: Sensor response: normal and fault bearing.

This section builds on the foundation of sensor technologies to explore how these data changes can be effectively analyzed to detect and diagnose bearing faults. A bearing fault detection system typically consists of three main components: signal preprocessing,

feature extraction, and classification methods. Each of these components works in synergy to transform raw sensor data into actionable insights for fault detection and maintenance decision-making, as shown in the following flowchart 2.13:

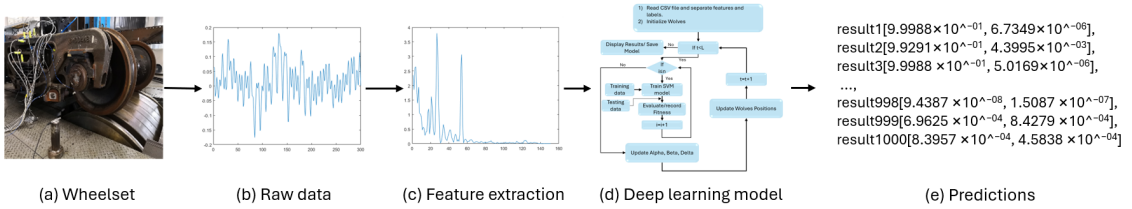


Figure 2.13: Example of flowchart, describes the progress of the system.

Figure 2.13a shows the wheelset bearing setup used in the experiment. Sensors are mounted to capture raw vibration signals during operation, providing critical data for fault diagnosis. Figure 2.13b illustrates the raw vibration signals collected from the sensor. These time-domain signals represent the baseline data that will undergo further analysis to extract meaningful features. However, these signals often include noise, trends, and irrelevant components that can obscure meaningful patterns. To address this, data preprocessing is applied as the first step in the analysis pipeline such as noise reduction, normalization and segmentation. Figure 2.13c shows the application of feature extraction techniques, specifically using FFT to convert the raw signals into the frequency domain. This transformation highlights important frequency components that may indicate faults in the bearing. Figure 2.13d outlines the machine learning workflow. The process begins by splitting the dataset into training and testing subsets. A deep learning model is trained iteratively, with each iteration evaluating the fitness of the model. Optimization is achieved by updating key parameters to enhance the model's performance. Figure 2.13e represents the final prediction phase. The trained model predicts the condition of each segment of the dataset, enabling accurate detection and classification of potential faults in the wheelset bearing system.

2.2.1 Signal pre-processing

Signal preprocessing is an essential step in bearing monitoring, aimed at improving signal quality, removing noise and interference, and laying the groundwork for subsequent analysis and feature extraction. Common signal preprocessing methods include time-domain analysis, time-frequency analysis and denoising techniques.

The primary goal of signal preprocessing is to transform complex, noisy raw signals into clearer and more structured representations, providing reliable data for subsequent feature extraction and classification.

2.2.2 Feature extraction

Feature extraction is the process of extracting representative features or information from data to facilitate subsequent machine learning or data analysis tasks. It is commonly applied to various types of data, such as text, images, audio, and time-series data, and is a critical step in machine learning and data mining. By focusing on key features, feature extraction helps to reduce data dimensionality, improve model performance, and enable more efficient data processing [66, 67].

Fast Fourier Transform (FFT)

FFT is an efficient algorithm for computing the Discrete Fourier Transform (DFT), transforming time-domain signals into the frequency domain. FFT significantly reduces the computational load of DFT, accelerating calculation speed, and is widely used in fields like signal processing, image processing, communication systems, and audio analysis [68].

DFT is the core algorithm for frequency domain analysis of discrete signals, converting signals from the time domain to the frequency domain to obtain amplitude and phase at different frequency components. Given a sequence $x(n)$ of length N, its DFT is expressed as:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi kn/N} \quad \text{for } k = 0, 1, \ldots, N-1 \tag{2.1}$$

Directly computing the DFT has a time complexity of $O(N^2)$, which becomes computationally intensive when N is large.

FFT leverages the symmetry and periodicity properties of DFT, reducing the computational complexity to $O(N \log N)$ [68]. The classic FFT algorithm is the Cooley-Tukey algorithm, which recursively decomposes the DFT into smaller DFTs, minimizing redundant calculations. FFT typically employs a butterfly structure, dividing the original sequence into odd and even parts, calculating the DFTs of these subsequences, and then combining the results. This decomposition avoids redundant operations, making the algorithm more efficient.

FFT has extensive applications, especially where frequency analysis is essential. In signal processing, FFT is used for tasks such as filtering and spectrum analysis to extract key frequency components from signals. In image processing, it enables frequency domain analysis for applications like image compression and enhancement. In audio processing, FFT aids in speech recognition and audio feature extraction by analyzing the frequency spectrum of audio signals. In communication systems, FFT plays a critical role in technolo-

gies such as OFDM (Orthogonal Frequency-Division Multiplexing) for signal modulation and demodulation. Additionally, in vibration and fault diagnosis, FFT is used to analyze the vibration signals of mechanical equipment, helping to identify different frequency components and detect potential faults. These diverse applications showcase FFT's ability to reveal essential frequency characteristics across a variety of fields [69–71].

FFT offers the advantage of fast computation, making it well-suited for processing large-scale data, and it efficiently reveals the frequency information of signals, facilitating analysis and understanding of frequency structures. However, FFT has limitations: it is less effective for non-stationary signals since it assumes signal stability over the entire analysis period, and its resolution is limited by the sampling rate. In certain cases, additional window functions or methods like the Short-Time Fourier Transform (STFT) are necessary for more precise time-frequency analysis [72, 73].

The efficiency and widespread application of FFT makes it an indispensable tool for frequency domain analysis and numerous engineering applications.

Mel-Frequency Cepstral Coefficients (MFCC)

Mel-Frequency Cepstral Coefficients (MFCC) is a widely used feature extraction method in speech processing and audio signal analysis, especially suitable for applications like speech recognition and music classification. MFCC leverages human auditory characteristics to transform audio signals into a set of cepstral coefficients, effectively capturing speech features within audio [74].

Steps in MFCC Calculation [75, 76]:

- The audio signal is preprocessed, often removing silence, and then divided into short-time frames (typically 20–40 ms) to capture the transient characteristics of speech. Each frame contains hundreds of sampling points.
- To minimize spectral distortion due to framing, a Hamming window or other window function is applied to each frame. This windowing reduces edge effects and ensures continuity across frames.
- FFT is applied to each windowed frame to convert the time-domain signal into the frequency domain, extracting the frequency components of each frame to obtain spectral information.
- Mel frequency simulates the human auditory scale. A triangular filter bank (usually 20 to 40 filters) is designed based on this scale, with filters more densely placed in

the low-frequency range to reflect human sensitivity to lower frequencies.

The energy output for each filter is computed to form the Mel spectrum.

- The logarithm of each filter's energy output is taken, reducing the dynamic range of amplitude and emphasizing contributions from lower energy components, making them comparable to higher energy components.
- DCT is applied to the Mel spectrum, converting it to the cepstral domain. This compresses the frequency information into fewer dimensions, reducing the correlation between adjacent frequency bands to produce cepstral coefficients.

  Typically, the first 12-13 coefficients are selected as MFCC features, capturing the essential characteristics of the speech signal.

MFCC is widely applied across various fields, particularly in speech recognition, where it captures the core frequency characteristics of human speech, making it essential for automatic speech recognition systems. In music classification, MFCC analyzes spectral features of music signals, aiding in the categorization of different music genres or timbres. It also supports speaker recognition by representing the unique frequency attributes of an individual's voice, facilitating speaker identification. Additionally, in emotion analysis, MFCC extracts speech features to assess emotional states, as emotions influence the speech spectrum [75–78].

MFCC has notable advantages, including its alignment with human auditory characteristics, which allows it to effectively capture essential frequency features in speech, and its high computational efficiency with low-dimensional features, making it suitable for real-time applications. However, MFCC is sensitive to background noise, which can significantly affect its accuracy, and it is less effective at representing temporal dynamics in speech, often necessitating supplementary features like first-order and second-order derivatives to capture transient changes [74–76, 78].

MFCC, combining frequency characteristics of audio signals with human auditory perception, is a highly effective and computationally efficient feature extraction method, widely used in various audio and speech processing systems.

Empirical Mode Decomposition (EMD)

Under normal conditions, the high-speed rail wheelset runs smoothly, and its sound signal, temperature, vibration signal, etc. Fluctuate smoothly within a certain range, and the fluctuations obey a certain distribution [79, 80]. When the high-speed rail wheelset fails, the sound signal, temperature curve and vibration signal light will all produce serious

deformation, which can be used as a basis for fault diagnosis [81].

Norden E. Professor Huang proposed a method of adaptively decomposing non-stationary signals into a series of zero-mean Intrinsic Modal Functions (IMF), which was called Empirical Mode Decomposition (EMD) [82]. The EMD method is not limited by the Fourier analysis and is an adaptive time-frequency local refinement analysis method. EMD is widely used in various fields, including voice recognition [83], image processing [84], and fault diagnosis. Due to the adaptability of EMD and its advantages in non-stationary signal processing, many people innovate on the basis of EMD.

To address the limitations of the classic EMD method, such as modal aliasing and boundary effects, NE Hung mitigated modal aliasing by introducing Gaussian white noise and innovatively developed the Ensemble Empirical Mode Decomposition (EEMD). Generally, time-domain and frequency-domain features exhibit varying sensitivities to faults and are easily influenced by operating conditions. Time-frequency domain features, on the other hand, can simultaneously capture both time-domain and frequency-domain information of a signal, though each time-frequency analysis method comes with its own strengths and limitations. Wavelet decomposition, with its high time-frequency resolution, demonstrates significant advantages in processing non-stationary signals. Nevertheless, both wavelet decomposition and wavelet packet decomposition face the challenge of selecting an optimal wavelet basis. EEMD effectively addresses the modal aliasing issue in EMD by incorporating Gaussian white noise during the decomposition process, enabling adaptive decomposition of signals into intrinsic modal functions across different frequency bands, making it particularly suitable for handling nonlinear and non-stationary signals. However, individual time-domain, frequency-domain, and time-frequency domain features are typically sensitive to specific faults. Therefore, the integration of multi-domain features can be considered to provide a more comprehensive representation of the mechanical operating state.

Lotfi Saidi et al. [85] used EMD to dissect the non-stationary signal into several IMFs according to the local characteristic time scale of the signal. The bispectrum of H-order statistics was used to determine the phase coherence effect. For Gaussian noise, the bispectrum analysis theoretical value is zero, and it is constant for non-Gaussian white noise, so the bispectrum is insensitive to random noise. The author combines the advantages of EMD and bispectrum and proposes the bispectrum EMD method.

Jaouher Ben Ali et al. [86] used the IMF energy bribe generated by empirical mode decomposition to describe seven different bearing states. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) methods are used for feature reduction

processing of bearing vibration signals. The extracted IMF, PCA, and LDA are used as feature vectors to perform Probabilistic Neural Network (PNN) and simplified Fuzzy Auto Adaptive Resonance Theory MAP (SFAM) neural network analysis. The calculation results show that the combined analysis of IMF-LDA-PNN-SFAM is an effective method to improve the accuracy of bearing fault classification, and it has good application capabilities and can be used for bearing detection in different mechanical systems.

Jacek Dybala et al. [87] proposed an early damage detection method for rolling bearings based on EMD. Several IMFs obtained by decomposition are used to aggregate into three Combined Mode Functions (CMF), and the vibration signal is divided into H parts; pure noise part, pure signal part and pure trend part. The local amplitude of the frequency spectrum is used to further extract the eigenvalues related to bearing faults. This method can effectively identify the early failure of the bearing and its development trend.

Wang et al. [88] proposed a new Non-negative EMD Manifold (NEM) bearing failure feature extraction method. Manifold features are mainly extracted from IMF related to the fault. First, the non-negative EMD matrix is decomposed through correlation analysis, the relevant IMF is selected, and the non-negative EMD is extracted through the optimization algorithm; then, the internal law of the non-negative EMD feature is further explored, and redundant information is removed to obtain the inherent stability feature. The stability characteristics associated with a large amount of vibration data can characterize the characteristics of the best value in the failure mode.

Feature selection refers to the selection of several effective features from a set of high-dimensional features. The principle of selection is separability between classes. Traditional feature dimensionality reduction methods mainly include PCA and LDA. Both of these methods are linear dimensionality reduction methods, but the features extracted in engineering practice are generally non-linear, so the linear dimensionality reduction method is limited in its application. In order to reduce the dimensionality of nonlinear features, the Kernel Principal Component Analysis (KPCA) method is developed, which maps the features to the nonlinear feature space through the kernel function to achieve dimensionality reduction [89].

Popular learning is a more popular dimensionality reduction method in recent years. It has been widely used in the field of fault diagnosis. Among them, Laplacian Eigenmaps (LE) [90], Locality Preservation Projection (LPP) [91], Locally Linear Embedding (LLE) [92], Local Tangent Space Alignment (LTSA) [93] and improved algorithms of these algorithms. Qingbo He et al. [94] used LLE to extract the popular features of wavelet packet energy and effectively distinguished bearing and gear failures with different failure

degrees. Benwei Li et al. [95] used the supervised LLE algorithm to map the features from the high-dimensional space to the embedding space and performed bearing fault classification in the embedding space.

2.2.3 Classification methods

Neural networks

The Artificial Neural Network (ANN) is simply referred to as the neural network [96]. It is a complex network formed by a large number of interconnected neural processing units. It is a simplified abstraction of the human brain from the microscopic structure and function. It features large-scale parallel analogue processing, continuous-time dynamics, and a global network, which can greatly increase the speed of work. The storage of information is reflected in the distribution of connections between neurons. The neural network has strong adaptability and learning ability, robustness and fault tolerance, which can replace complex traditional algorithms and make signal processing closer to human thinking activities. The study of neural networks involves a wide range of subject areas that combine, infiltrate and promote each other.

The neural network is a mathematical model that uses information similar to the structure of brain synaptic connections for information processing. A neural network is an operational model consisting of a large number of nodes (or neurons) and interconnected. Each node represents a specific output function called an activation function. The connection between every two nodes represents a weighting value for passing the connection signal, called weight, which is equivalent to the memory of the artificial neural network. The output of the network varies depending on the network's connection method, weight value, and excitation function. The network itself is usually an approximation of an algorithm or function in nature, or it may be an expression of a logic strategy.

Its construction philosophy is inspired by the operation of biological (human or other animal) neural network functions [97]. Neural networks are usually optimized through a learning method based on mathematical statistics types, so neural networks are also a practical application of mathematical-statistical methods [98]. We can get a large number of usable functions through statistical standard mathematical methods. To express the local structural space, on the other hand, in the field of artificial sensing of artificial intelligence, we can use the application of mathematical statistics to make decisions about artificial perception (that is, through statistical methods, neural networks can be similar to humans). It has the same simple decision-making ability and simple judgment ability. This method

is more advantageous than the formal logical reasoning calculation [99].

Neural networks advantages

First, it has an independent learning function. For example, when identifying an image, only a plurality of different image templates and corresponding recognition results are input into the neural network, and the network slowly learns to recognize similar images through the self-learning function.

Second, this associative capability can be achieved using the feedback mechanism within the neural network.

Third, the ability to find optimized solutions at high speed. Finding an optimal solution to a complex problem often requires a large amount of computation. By using a feedback neural network designed for a problem, the computer's high-speed computing power can be used to find an optimal solution [100].

Neural network basic feature

Neural networks possess four fundamental characteristics: nonlinearity, boundlessness, variability, and non-convexity. Their widespread recognition and application are primarily attributed to these distinct features: learning capability, distributed architecture, parallel processing, and nonlinearity.

The learning capability of neural networks is a critical indicator of their intelligence. It allows the network to abstract the core features of training samples through the learning process, demonstrating a high degree of adaptability.

In conventional serial computing systems, data is stored in discrete memory units, making any corruption in a specific unit potentially render the entire dataset unusable. In contrast, neural networks distribute information across the interconnections of neurons. While individual neuron connections or weights hold limited significance on their own, collectively they encapsulate specific informational patterns. Damage to individual neurons or weights has a minimal impact on the overall data representation, highlighting the robustness (self-stabilization under disturbances) and fault tolerance of neural networks. Even when subjected to input disturbances, the network's output remains largely unaffected. This distributed information storage also endows neural networks with powerful associative capabilities.

Neural networks are inspired by the structure and functionality of the human brain, with a primary focus on structural simulation. Each neuron independently processes its input and produces an output. This parallel computation mechanism enables neural networks to process information efficiently in real-time, laying the foundation for the development of next-generation intelligent computing systems.

The neural network can effectively realize the nonlinear mapping of the input space to the output space. Seeking a nonlinear relationship model between input and output is a common problem in the engineering community. For most model-free nonlinear systems, neural networks are well simulated. Therefore, neural networks have become an important tool for the study of nonlinear systems.

Neural network classification

According to the classification of algorithms, neural networks can be broadly categorized into supervised learning networks, unsupervised learning networks, hybrid learning networks, associative learning networks, and optimization application networks. Among these, the primary focus is on supervised learning networks, which serve as the foundation for this classification [101].

There is a machine learning method called supervised learning, and supervised learning can judge its function based on specific training data. In supervised learning, each training instance consists of input and output. The supervised learning algorithm analyzes these training data and infers the function, and after observing some training examples, the supervised learning algorithm can predict all the outputs corresponding to this type of data input value. Therefore, to achieve this goal, the supervised learning algorithm needs to correctly determine the label of the training data through a ”reasonable” method.

According to the connectionist classification, neural networks are primarily divided into feed-forward networks, recurrent neural networks (RNNs), and reinforcement networks. Among these, RNNs are the main focus of this classification [102].

Next, I will mainly introduce CNN, RNN and LSTM.

Convolutional neural networks (CNNs)

CNNs have been used for machine learning operations and computer vision. Deep 2D CNNs have many hidden layers and millions of parameters, enabling us to learn complex objects and patterns and a visual database trained with ground-truth labels. 1D has

been proven successful in several applications such as health monitoring, biomedical data classification, motor-fault detection and anomaly detection and identification in power electronics [103].

Recurrent neural networks (RNNs)

Unlike other traditional neural networks, RNNs focus on transmitting information from the start to each neural network layer. RNNs are designed to create loops that allow information to flow from one layer to subsequent or even later layers. Unlike traditional neural networks, which typically operate under the assumption that inputs and outputs are relatively independent, RNNs account for the dependencies between them. In real-world problems, outputs often rely not only on the current inputs but also on information from previous states or memories, making RNNs particularly effective for tasks involving sequential or temporal data [104, 105].

x_t is the input of layer t, which could be a one-hot or distributed representation vector. s_t is the hidden layer of layer t and it will be the memory of whole network. s_t is depending on the output of the last layer $t-1$ and input of current layer t. $s_t = f(U_{x_t} + W_{s_{ti1}})$ where f is a nonlinear activation function, usually will be $tanh$ or ReLU. In theory, s_t should be able to capture all the information in each layer before as the s_t will be keeping pass back.

o_t is the output of layer t. Like in the experiment will predict the sequence of movements, o_t is a vector with the length of V, and V is all the states of the movements. $o_t[i]$ is the possibility of the next statement w_i. The softmax function could be used to normalize these possibilities, $o_t = softmax(V_{s_t})$. Where the parameters U, V, W in every layer are shared to reduce the parameter space. And the outputs and inputs are not necessary for every layer, some RNNs may only need output in the last layer. The hidden layers are most important in RNN.

RNN learns the parameters with backpropagation and gradient descent algorithm, so the error function in layer t is related to o_t. And o_t dependent with all the x_i and s_i, $i < t$ in the layers before which is so-called BackPropagation Through Time(BPTT). In these BPTT networks, the vanishing gradient problem and the exploding gradient problem may appear because the network is too deep. A simple modification could be applied to avoid these problems by shortening the dependence of o_t. Let o_t only decided by x_i and s_i, $t-1 \leq i \leq t$. Or a unique method called LSTM could be applied to solve these problems.

Long short-term memory (LSTM)

LSTM is a special RNN that can learn long-term dependencies [106].

As mentioned in the last subsection, vanishing gradient problems and exploding gradient problems are hard to avoid in traditional RNNs. LSTM learned the long-term dependence on the network with passed these problems. The hidden layer of traditional RNN is usually a *tanh* function or ReLU. A typical LSTM unit will conclude 3 *sigmoid* layers and 1 *tanh* layer.

LSTM consists of three gate variables as follows:

Input gate: Indicates whether information is allowed to be added to the memory unit. If the value is 1 (door open), the input is allowed. If it is 0 (door closed), it is not allowed, so some useless input information can be discarded.

Forgetting Gate: Indicates whether to retain the historical information stored in the current hidden layer node. If the value is 1 (door open), it is reserved. If it is 0 (gate closed), the historical information stored by the current node is cleared.

Output gate: indicates whether the current node output value is output to the next layer (the next hidden layer or output layer). If the value is 1 (door open), the output value of the current node will be applied to the next layer. If it is 0 (gate closed), the information is discarded.

A cell state C is applied in LSTM with only a few linear operating on it which could retain information easily. The first gate in LSTM is the forget gate which decides what information should be discarded. x_t will be sent to a sigmoid function with h_{t-1} and get a value between 0 and 1 which is multiplied by the cell state C_{t-1}. The output of the sigmoid function will decide how much information to remain. Part of the information in the last layer $t-1$ has been forgotten in the cell state C_{t-1} and the new information in the current layer will be added by a *tanh* function and a sigmoid function. This sigmoid function is called an input gate, and the output of it will multiply by a *tanh* function. When the value of it is 0, the cell state doesn't need to be updated.

Then the last cell state C_{t-1} multiplies with forget gate f_t to discard part of the information and update the information from $i_t \times C_t$. The information in the updated cell state C_t will output after a *tanh* function and a sigmoid function and this is called the output gate. A brief figure of LSTM is shown in Fig. 3.3

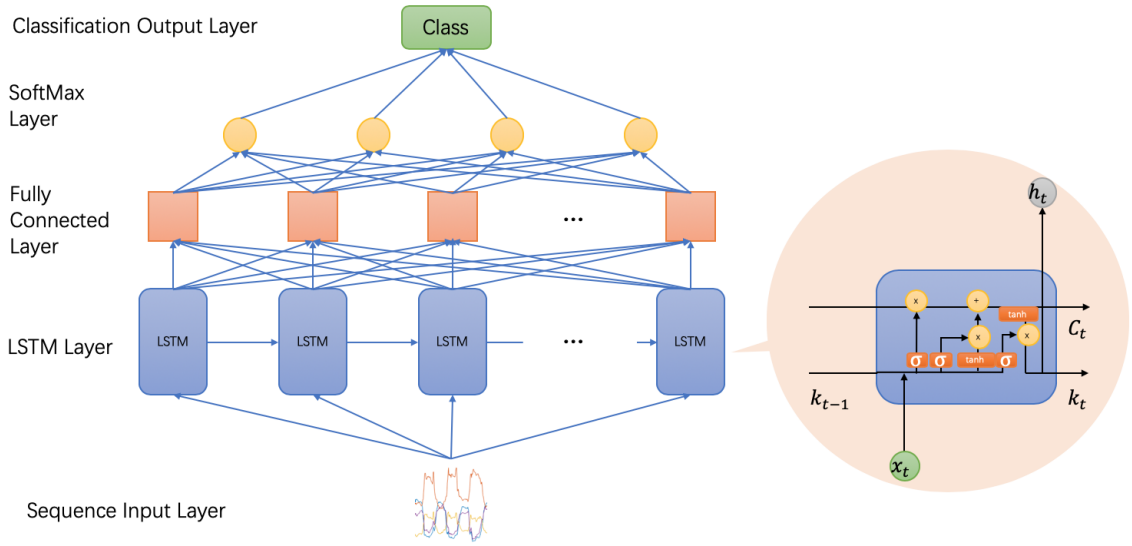


Figure 2.14: LSTM network. (a) RNN with LSTM unit; (b) the structure of LSTM unit.

LSTM has the advantages of long-term trajectory memory and transient memory unification, simulation of selective brain forgetting, and more accurate trajectory modelling. Therefore, the multi-layer structure can be mixed to solve the efficiency and stability problems of massive data training

SVM

SVM (Support Vector Machine) is a supervised learning model and algorithm widely used for classification and regression tasks in machine learning and data science. The primary goal of SVM is to find an optimal hyperplane that separates different classes of data as widely as possible, providing strong classification and predictive capabilities [107]. In a simple binary classification scenario, SVM aims to find an optimal hyperplane (a linear separating plane) that divides the dataset into two classes while maximizing the distance between the hyperplane and the nearest points of each class (called support vectors). This maximum margin approach enhances the model's generalization, allowing it to classify new data accurately. Support vectors are the data points closest to the decision boundary, playing a crucial role in defining the optimal hyperplane. Since SVM relies only on these support vectors to determine the boundary, it is less sensitive to noise from data points far from the boundary. For data that is not linearly separable, SVM uses kernel functions (such as polynomial, radial basis, or Gaussian kernels) to map the data into a higher-dimensional space where a linear separating plane can be found. Kernel functions enable SVM to handle non-linear problems effectively, achieving strong classification results even with complex data structures.

SVM Advantages

SVM, as a classic machine learning algorithm, offers several advantages:

SVM performs well in high-dimensional spaces, particularly suitable for datasets with a large number of features. In high dimensions, SVM identifies a hyperplane that maximizes the margin between classes, which enhances model generalization and reduces the risk of overfitting as data dimensionality increases [107–109].

SVM utilizes kernel functions (such as linear, polynomial, and Gaussian kernels) to map data into a higher-dimensional feature space, enabling it to effectively classify nonlinearly separable data. This use of kernels allows SVM to achieve strong results with complex data distributions [107, 109].

By maximizing the margin between classes to select the optimal decision boundary, SVM inherently reduces model complexity and increases generalization, making it highly resistant to overfitting. This is particularly advantageous when working with small datasets [108, 109].

SVM relies on support vectors—points closest to the decision boundary—to determine the hyperplane, meaning data points far from the support vectors have little influence on the model. As a result, SVM is relatively robust to noise and outliers in the data [110].

SVM does not require large datasets to perform well, which gives it an advantage over small datasets. This makes SVM especially useful in scenarios where data availability is limited or labeling is costly [107, 108].

SVM provides a clear geometric explanation by maximizing the margin between two classes to achieve the optimal classification boundary. This geometric interpretation not only improves model interpretability but also aids in understanding the decision-making process [110].

These advantages make SVM particularly effective for applications involving high-dimensional, nonlinear, and limited-sample data, such as text classification, image classification, and bioinformatics.

2.3 Summary

Existing high-speed train bearing fault diagnosis methods primarily focus on time-domain signal analysis. While time-domain features are relatively simple to extract and compute, they often fail to capture critical fault-related characteristics more apparent in the frequency domain, especially under the influence of high-frequency vibrations and dynamic loads in high-speed trains. Many traditional approaches also overlook the sequential nature of vibration data, which contains temporal dependencies crucial for understanding progressive fault development. Additionally, fault diagnosis systems may struggle with early detection, leading to delayed responses and potential safety risks in high-speed train operations. My thesis applies frequency-domain feature extraction using FFT and integrates it with a Bi-LSTM model to effectively capture both the frequency characteristics and sequential dependencies, enabling more accurate and timely fault detection.

Wind turbine bearings operate in harsh environments and are subject to high loads, making them prone to faults that require immediate detection to prevent costly downtime. However, traditional machine learning models often require high computational resources and offline processing, making them unsuitable for real-time applications. The deployment of computationally heavy algorithms on embedded devices, such as those used in wind turbines, is particularly challenging due to resource constraints (e.g., limited processing power, memory, and energy). More importantly, current fault detection systems also suffer from latency issues, delaying predictions and reducing the ability to implement real-time corrective actions. My thesis addresses these limitations by developing a lightweight neural network optimized for deployment on a Raspberry Pi, capable of segmenting and analyzing data in milliseconds for efficient real-time fault detection.

Most existing fault diagnosis methods for railway wheelset bearings rely exclusively on vibration data. While effective in many scenarios, vibration signals alone may fail to capture the full spectrum of fault characteristics, especially in complex loading and environmental conditions commonly experienced by railway systems. Acoustic data, which often contains complementary fault-related information, is frequently ignored, leading to incomplete diagnostics and reduced fault detection accuracy. Additionally, traditional diagnostic models often fail to optimize their parameters effectively, resulting in suboptimal performance, especially in noisy or highly variable conditions. My thesis integrates vibration and acoustic data for a more comprehensive fault detection approach and employs the Grey Wolf Optimizer (GWO) to optimize the hyperparameters of the SVM model, significantly enhancing detection precision and robustness.

By addressing these three critical challenges, my thesis introduces innovative solutions

tailored to the unique requirements of high-speed trains, wind turbines, and railway wheelsets, significantly improving the accuracy, efficiency, and practicality of bearing fault diagnosis systems.

Chapter 3

Frequency Domain Feature Extraction and Long Short-Term Memory for Rolling Bearing Fault Diagnosis

3.1 Introduction

A few essential components ensure that the train operates safely and efficiently, and the development of high-speed rail has considerably improved traffic conditions in many countries. The transmission system is one of them and is crucial to the efficient running of high-speed trains. In the majority of high-speed train transmission systems, the traction motor serves as the power source and is a critical component. Therefore, it is important to recognize traction motor breakdowns. The most frequent type of motor failure is motor bearing failure [13]. For instance, according to the research, 40% of motor failures are caused by motor bearing failures. Fatigue spalling, which includes partial damage or fall-off on the inner ring, outer ring, rolling element, and other bearing surfaces, is one of the most common defects in motor bearings. Fatigue stress from alternating loads is the main factor in bearing fatigue spalling [111]. When a bearing experiences a fatigue spalling failure, a particular frequency of shock pulse will manifest. As a result, rolling bearing maintenance is quite expensive and very important for every country. For instance, the US spends hundreds of billions of dollars every year on maintaining machinery and routinely replacing vital components [112].

However, if the crucial components of the equipment are not updated in a timely manner, catastrophic tragedies could happen. For instance, on June 3, 1998, a high-speed train in Germany's elastic wheel burst due to prolonged use, resulting in 101 fatalities and 194 serious injuries [113]. 72 people lost their lives and 416 were hurt when 9 carriages of the NO.T195 train from Beijing to Qingdao derailed and crashed with the NO.5034 train from Yantai to Xuzhou on April 28, 2008, in China. [114].

Consequently, rolling bearing detection and recognition for health monitoring has emerged as one of the key research fields in order to lower the cost of rolling bearing maintenance and maintain the safety of operation for high-speed trains. In recent years, some laboratories used accelerometers located at the driving end of the motor housing to adopt a significant amount of vibration data. The sampling frequency is usually 12000 samples per second, 25600 samples per second and 48000 samples per second. These vibration data lengths are generally more than 120000, and many subcategories of vibration signals exist.

Hence, signal processing and status recognition have taken center stage in rolling bearings research. For signal processing, He and Partha [115] proposed Locality Preservation Projection (LPP), Zhang et al. [116] proposed Linear Local Tangent Space Alignment (LLTSA). More importantly, status recognition achieves much success as well. For instance, Shao et al. [117] proposed a Deep Wavelet Auto-Encoder (DWAE) with an Extreme Learning Machine (ELM). They used the wavelet function to design a wavelet autoencoder, to get data features and improve the ability to study unsupervised features. ELM is a classifier. The result is 95.2%. Shao et al. [118] proposed Ensemble Deep Auto-Encoders (EDAEs). Use the Unsupervised feature learning from the raw vibration data and design a strategy to ensure accuracy and stability. The result is 97.18%. Tao et al. [119] proposed Deep Belief Networks (DBN). DBN can reduce energy loss between the output and input vibration signals. The result is 96.67%. Gan and Wang [120] proposed Hierarchical Diagnosis Network (HDN) can achieve 99.03%. Zhuang and Qin [121] proposed a Multi-Scale Deep CNN (MS-DCNN) model that can reach 99.27%. Guo et al. [122] constructed a hierarchical Adaptive Deep Convolutional Neural Network (ADCNN), the accuracy is 97.7%.

In these recognition methods, the ability to reinforce learning is the most similar to manual detection. Among them, Wang et al. [123] proposed a reinforcement neural architecture search method to achieve success. The article suggested and validated the neural network architecture automatic search method. The framework of the article includes two parts: the controller model and the child model.

The controller model has 2 Nascell layers, and the output of this model are convolutional

kernel size and kernel number and a pooling kernel size of each layer. They formed the CNN [124]. The child models are CNNs. The model consists of an input layer. The two groups of the same convolutional layer, the pooling layer, take turns to each other. The complete connection layer.

However, the time domain is the main emphasis of these methods. The time domain analysis is unable to observe the frequency-dependent signal properties for the vibration signal. The frequency domain analysis is more succinct than the time domain. Following the signal in the frequency domain provides a deeper and more practical analysis of the issue.

This chapter proposed feature extraction and recognition for rolling bearing fault diagnosis based on frequency domain and LSTM to overcome the shortcomings mentioned before. In this method, uses the FFT to alter the bearing's time-domain signal before it is transmitted to the network. I only need to fine-tune the maxepochs and hidden units in the process.

3.2 Related works

In recent years, numerous neural network-based approaches have been proposed for rolling bearing fault detection. These methods often incorporate advanced feature extraction techniques and deep learning architectures, aiming to improve diagnostic accuracy and robustness under complex industrial conditions. Among them, Reinforced Neural Architectures—enhanced with hybrid layers, attention mechanisms, or domain-specific transformations—have demonstrated significant potential.

To better position my proposed method, we summarize and compare several representative works in Table 3.1. These methods are evaluated in terms of their feature processing strategies, network design, and classification performance.

From the comparison in Table 3.1, it is evident that frequency-domain analysis, when combined with deep learning models such as ADCNN, tends to yield higher accuracy and better robustness. However, such methods often come at the cost of increased model complexity and training requirements. Conversely, models like EDAEs offer efficient learning but may suffer from sensitivity to hyperparameters.

The comparison focuses on Reinforcement neural architecture because it represents one of the most advanced methods among the listed approaches, with key advantages such

as automated architecture design through reinforcement learning, elimination of manual network tuning, and strong generalization performance. Compared to traditional CNN, DBN, or AE-based models, Reinforcement neural architecture significantly reduces human effort and achieves high accuracy.

Overall, while many of the existing methods demonstrate promising results, they either rely heavily on handcrafted features or require intensive computation. These limitations motivate the development of a more balanced approach, as proposed in this work, combining frequency domain feature extraction with Bi-LSTM for enhanced fault detection.

Table 3.1: Comparative Summary of State-of-the-Art Methods for Rolling Bearing Fault Diagnosis

Methods	**Time Domain**	**Frequency Domain**	**Deep Learning**	**Advantages**	**Shortcoming**
DWAE+ELM [117]	✓	✓	✓	No manual feature, strong unsupervised learning	Relies on wavelet basis, limited to shallow classifier
DBN [119]	✓	✗	✓	Extracts features layer-wise, handles nonlinearity	Difficult to tune, slow training
EDAEs [118]	✓	✗	✓	Ensemble improves robustness and accuracy	High computational cost, less interpretable
ADCNN [122]	✓	✗	✓	Automatically extracts hierarchical features	Requires tuning, sensitive to overfitting
HDN [120]	✓	✓	✓	Two-level diagnosis, high precision	Needs handcrafted features (WPT), more parameters
MS-DCNN [121]	✓	✗	✓	Multiscale filters, end-to-end learning	Model design complexity, data-sensitive
Reinforcement neural architecture [123]	✓	✗	✓	Auto architecture search, high adaptability	Large training cost, requires RL expertise
FFT+BiLSTM	✗	✓	✓	Low variance and high stability, Simple two-step pipeline: FFT + Bi-LSTM	Requires FFT preprocessing

3.3 System

3.3.1 Overview of system

The structure of the proposed method is shown in Fig. 3.1, the vibration signal is collected by accelerometers which are located at the drive end of the motor housing, and the signal is measured at 1750 RPM in each working state. Then, the vibration signal is divided into several overlapping samples, and each sample is window processed and transformed with FFT. And then input these data into networks, the neural network trains these data, gets outputs and calculates the accuracy. Adjust the number of the hidden units and MaxEpochs until the result reaches the best.

The layers in the neural network system include the input layer, Bi-LSTM layer, fully connection layer, softmax Layer and classification Layer.

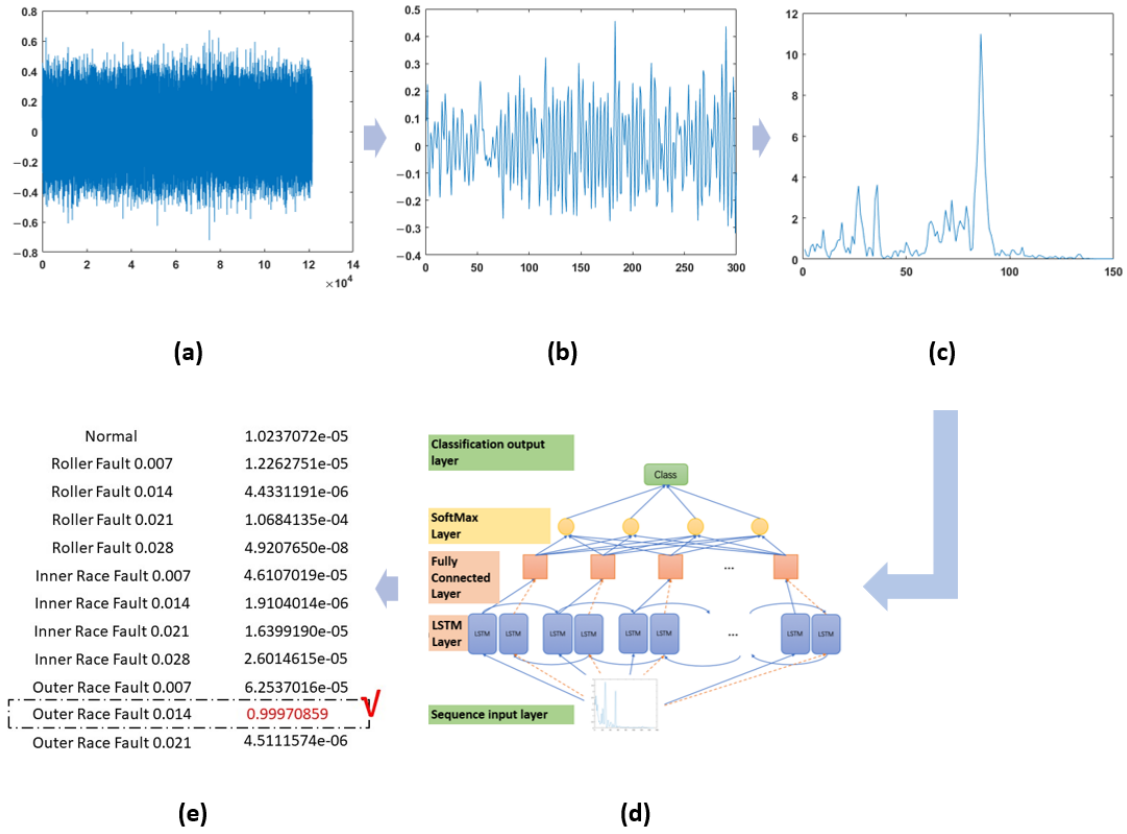


Figure 3.1: Overview of the system: (a) raw data example from CWRU dataset [5]; (b) segmented data with 300 points; (c) segmented data using FFT; (d) training data sent to the network; and (e) score of 12 classes.

3.3.2 Frequency domain analysis

The existence of FFT makes DFT play a central role in algorithms in digital signal processing. The calculation formula of Discrete Fourier Transform is [125]:

$$X(k) = \sum_{n=0}^{N-1} x(n) W^{nk}, \quad \text{for } (0 \leq k \leq N-1) \tag{3.1}$$

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) W^{-nk}, \quad \text{for } (0 \leq k \leq N-1) \tag{3.2}$$

Where x stands a limited long sequence, X stands data after Discrete Fourier transformation, N is sampled n points in a sinusoidal cycle, and $W = e^{\frac{-j2p}{N}}$ is the Fourier factor. For a discrete signal x, FFT will transform the signal in frequency domain. Fig 3.2 shows that the signal changes before and after FFT.

To justify the use of frequency-domain feature extraction and sequence modeling with Bi-LSTM, it is essential to understand the internal structure of the vibration signal data. As illustrated in Figure 3.2, the left column shows the original time-domain signals for different bearing states, while the right column shows the corresponding signals after applying the Fast Fourier Transform (FFT). In the time domain, the signals appear complex and visually indistinct between different fault types. However, once transformed into the frequency domain, each condition reveals unique frequency patterns—peaks and spectral energy concentrations—that are closely associated with specific types and severities of faults.

This clear separation in frequency space provides a strong rationale for choosing FFT as a preprocessing step. Furthermore, the frequency-domain data can be treated as sequential patterns of spectral magnitudes. Bi-LSTM, which models sequences in both forward and backward directions, is particularly well-suited for capturing dependencies across the frequency components, even though they do not follow a strict time progression. This structural insight supports the combined use of FFT and Bi-LSTM in our method, aligning the algorithm design with the characteristics of the data itself.

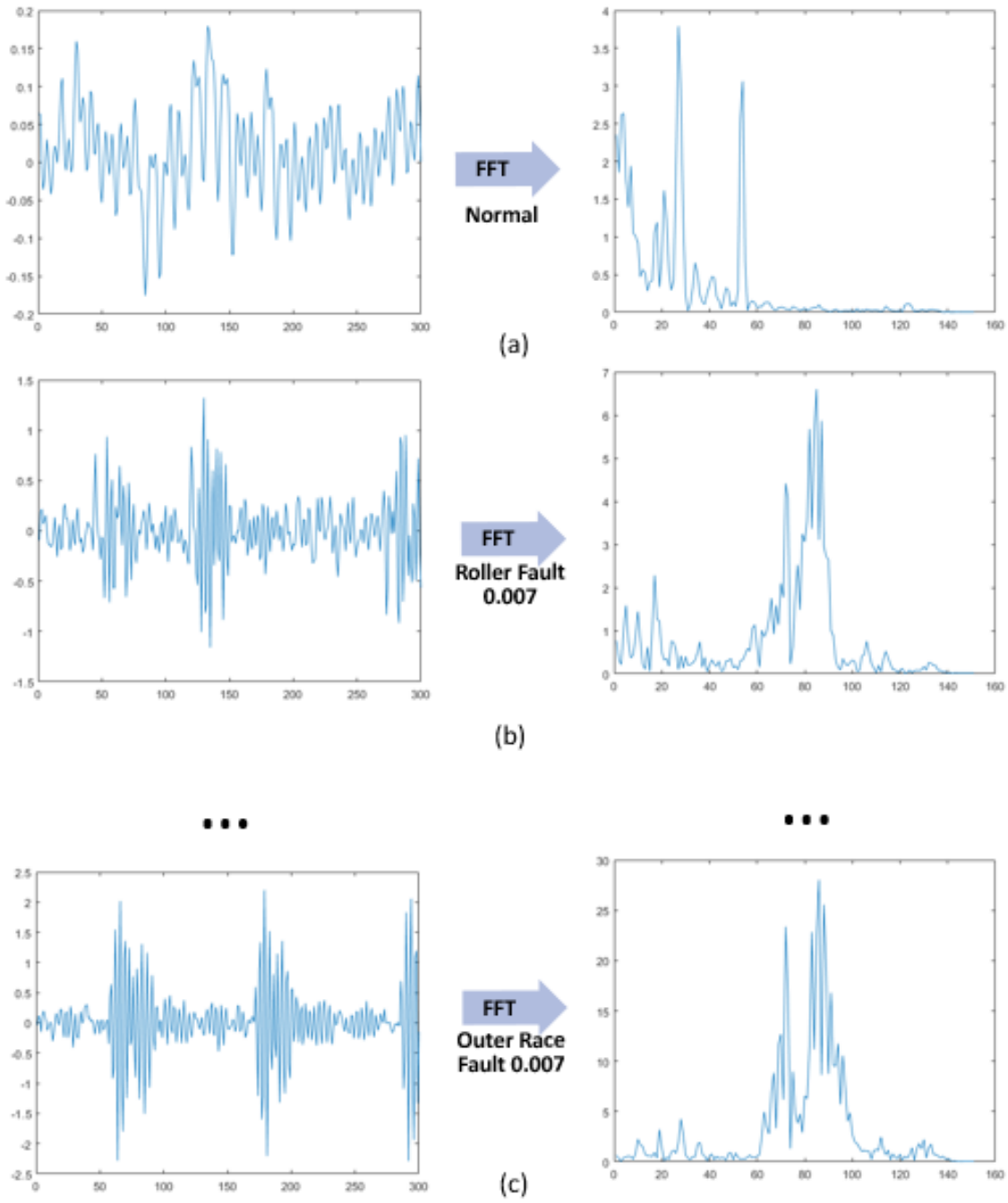


Figure 3.2: Signal transformed by FFT, left part is raw data, each of them has 300 points, the right part is data with FFT, each of them has 150 points. (a) An example of normal signal; (b) an example of ball fault size 0.007 inches signal; and (c) an example of outer fault size 0.007 inches signal.

3.3.3 Bi-LSTM

LSTM is a special RNN that can learn long-term dependencies [126]. Vanishing and exploding gradient problems are hard to avoid in traditional RNNs. LSTM learned the long-term dependence on the network with passed these problems. The hidden layer of

traditional RNN is usually a *tanh* function or ReLU. A typical LSTM unit will conclude 3 *sigmoid* layer and 1 *tanh* layer.

LSTM consists of three gate variables: Input gate, Forgetting Gate and Output gate. A cell state C is applied in LSTM with only a few linear operating on it, which could retain information easily. The first gate in LSTM is forget gate, which decides what information should be discarded. x_t will be send to a sigmoid function with h_{t-1} and get a value between 0 and 1 which multiplied with the cell state C_{t-1}. The output of the sigmoid function will decide how much information remains. Part of the information in the last layer $t-1$ has been forgotten in the cell state C_{t-1}, and the new information in the current layer will be added by a *tanh* function and a sigmoid function. This sigmoid function is called input gate and the output of it will multiply by a *tanh* function. When the value of it is 0, the cell state doesn't need to update.

Then the last cell state C_{t-1} multiply with forget gate f_t to discard part of information and update the information from $i_t \times C_t$. The output gate concludes the information in updated cell state C_t and the output after a *tanh* function and a sigmoid function. A brief figure of LSTM is shown in Fig. 3.3.

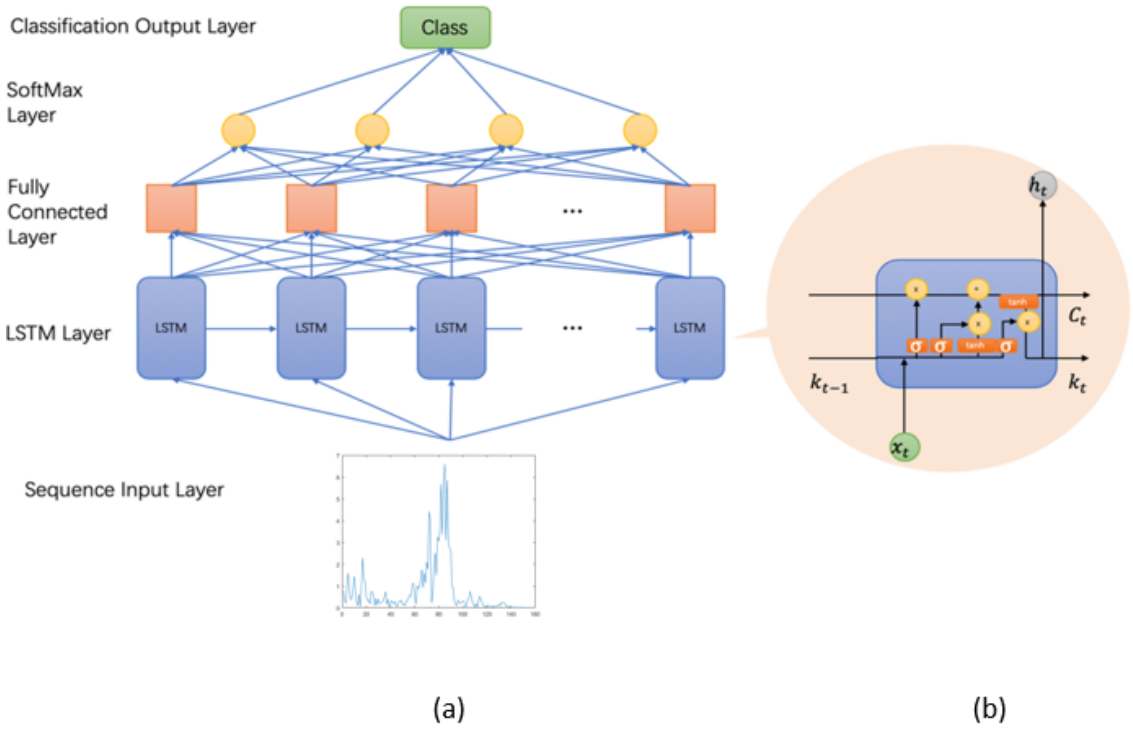


Figure 3.3: LSTM. (a) The network structure of LSTM; (b) LSTM unit. The repetitive module in LSTM has four interaction layers, three sigmoid and one *tanh*, and they interact in a unique way.

LSTM has the advantages of long-term trajectory memory and short-memory unification,

simulation of selective brain forgetting, and more accurate trajectory modeling. Therefore, the multi-layer structure can be mixed to solve the efficiency and stability problems of massive data training.

Bi-LSTM is an RNN with LSTM unit and will predict the sequence from both directions [127]. It could perform better in a sequence without directionality [128]. Actually the sequences are sent to two LSTM unit indenpently with different directions. The structure of Bi-LSTM is shown in (a) of Fig. 3.4.

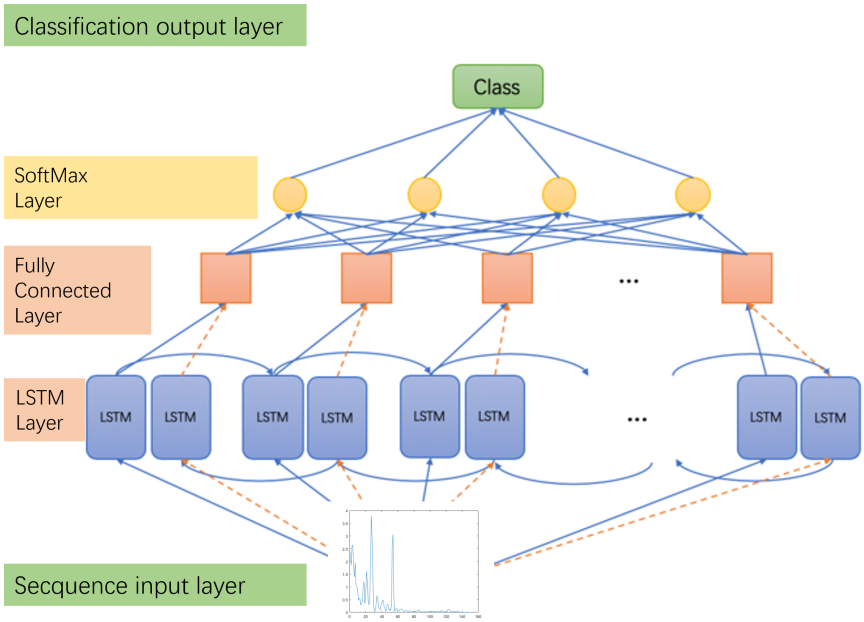


Figure 3.4: Bi-LSTM networks. The LSTM framework is used to merge the input sequence's front and backward directions. The two LSTM layers' vectors can be added to, averaged out, or connected.

3.4 Case Western Reserve University dataset experiment

3.4.1 48,000 samples/second dataset

The data are collected from the Electrical Engineering Laboratory at Case Western Reserve University [5]. The data were collected at 48,000 samples/second and have four different fault inches: 0.007, 0.014 and 0.021.

Siamese network

A siamese network is a type of neural network architecture that is designed for comparing two inputs and determining their similarity. Unlike standard neural networks, siamese networks consist of two identical subnetworks (hence the name ”Siamese”) that share the same weights and parameters. The primary objective of a siamese network is to learn a meaningful comparison function that can differentiate between similar and dissimilar input pairs [129].
As shown in Figure 3.5,

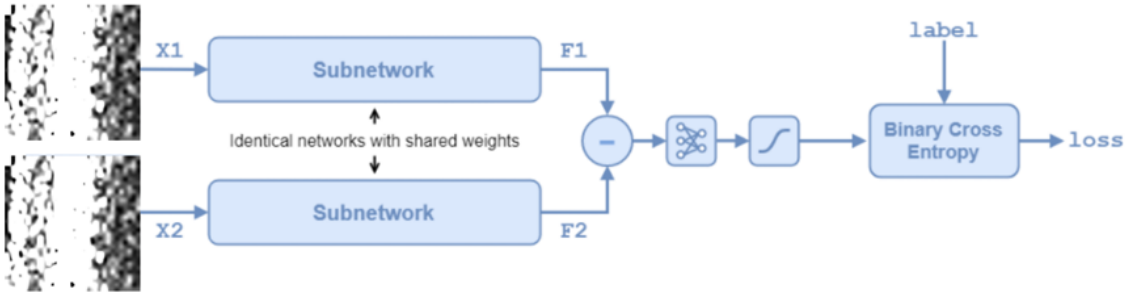


Figure 3.5: Saimese network, is a type of neural network architecture that is designed for comparing two inputs and determining their similarity.

For an image-based task, a typical siamese network could consist of convolutional layers that extract features from both input images, followed by a fully connected layer to map these features to a fixed-dimensional embedding space. Then, the distance between the two embeddings is computed, and the network learns to minimize the distance for similar image pairs and maximize it for dissimilar pairs [130].

Experiment setup

In this experiment, the subset data related to 36 faulty types are used, each of which has 7 samples, 3 samples for training and 4 samples for testing, totally 252 samples and the length of each sample is 2048 because of the limitations length of some data. The parameter information is shown in Table 3.2.

Table 3.2: Description of 36 states.

Data no.	Fault type	Fault size/inches	Motor speed (r/min)
1	Inner Race Fault	0.007	1730
2		0.014	1730
3		0.021	1730
4		0.007	1750
5		0.014	1750
6		0.021	1750
7		0.007	1772
8		0.014	1772
9		0.021	1772
10		0.007	1797
11		0.014	1797
12		0.021	1797
13	Ball Fault	0.007	1730
14		0.014	1730
15		0.021	1730
16		0.007	1750
17		0.014	1750
18		0.021	1750
19		0.007	1772
20		0.014	1772
21		0.021	1772
22		0.007	1797
23		0.014	1797
24		0.021	1797
25	Outer Race Fault-Center@6:00	0.007	1730
26		0.014	1730
27		0.021	1730
28		0.007	1750
29		0.014	1750
30		0.021	1750
31		0.007	1772
32		0.014	1772
33		0.021	1772
34		0.007	1797
35		0.014	1797
36		0.021	1797

As shown in Figure 3.6, there are three states: Inner ring fault 0.007 inches, ball fault 0.007 inches, and outer ring fault 0.007 inches. The length of each sample of raw data was 2048, each sample was window processed and transformed with FFT and then reshaped and resized to 105 by 105 images.

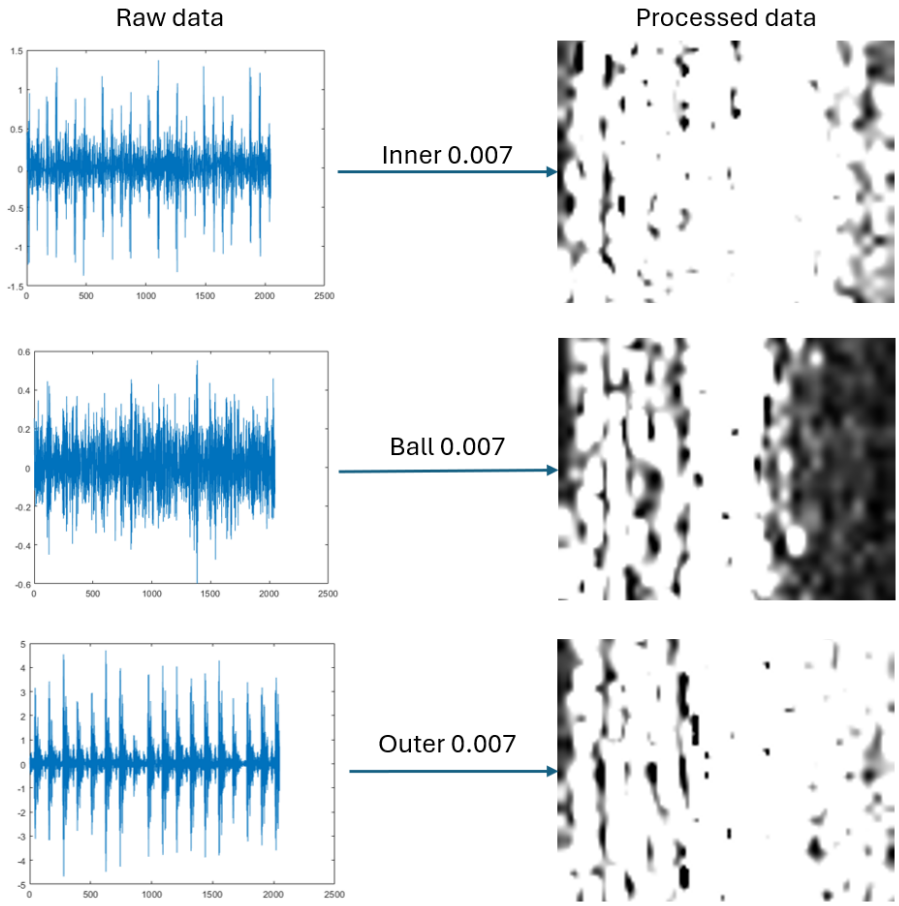


Figure 3.6: Data processing, the left part is raw data, each of them has 2048 points, the right part is data with FFT, each of them has 1024 points, reshaped and resized to 105 by 105 images.

Results of siamese network

In this part, I tested the data with different parameters, as shown in Table 3.3 and Table 3.4, the best test result was 86.8%.

Table 3.3: Siamese network result of 32 by 32 images.

NO.convolution Layers and Stride	Convolution kernel	Result%
Con3-Stride2	7-5-3	27.1
	7-3-5	29.9
	5-3-5	35.4
	5-3-3	56.3
	3-3-3	50.7
Con4-Stride1	5-3-5-3	**61.1**
	3-3-3-3	59

Table 3.4: Siamese network result of 105 by 105 images.

NO.Iterations	Learning Rate	Result%
1000	0.005	14.6
	0.00006	84
	0.00001	**86.8**
2000	0.00001	85.4

In Table 3.3, the length of data after FFT is 1024, hence, the size of the images is 32 by 32, for 3 convolution2d layers with Stride 2, the best result is 56.3%, and for 4 convolution2d layers with Stride 1, the best result is 61.1%. 61.1% corresponds to 5-3-5-3, which refers to the kernel sizes of the convolutional layers in the network.

5-3-5-3 indicates the sizes of the convolution kernels (filters) in four convolutional layers of the network, as follows: The first convolutional layer uses a kernel of size 5×5, the second convolutional layer uses a kernel of size 3×3, the third convolutional layer again uses a kernel of size 5×5, and the fourth convolutional layer uses a kernel of size 3×3.

Due to the accuracy, I reshaped and resized images into 105×105 images, the image of 105 by 105 can offer a balance between enough detail for classification and computational feasibility for training, This results in a higher-resolution image that retains the spatial structure from the 32×32 matrix but provides finer detail.

In Table 3.4, I adopted 4 convolution2d layers with Stride 1 and the size of the convolution kernels was 5-3-5-3. Then, I tested different learning rates and iterations. The best result was 86.8%, the learning rate was 0.00001 and iterations were 1000, the training loss is shown in Fig. 3.7.

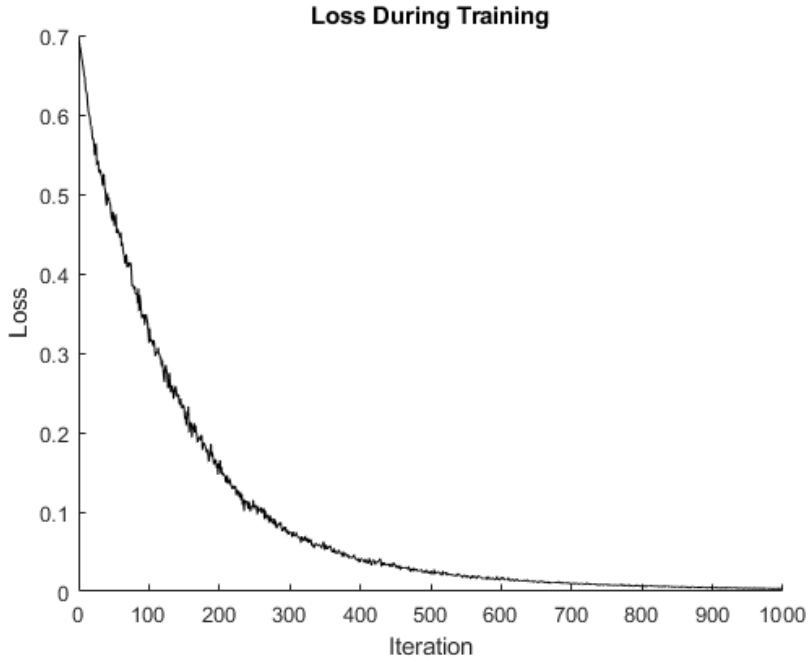


Figure 3.7: The loss curve of changes with training progress in siamese network.

I was not satisfied with this result, and due to the limitations of the data length, I selected 12k sampling dataset.

3.4.2 12,000 samples/second dataset

The data are collected from the Electrical Engineering Laboratory at Case Western Reserve University [5]. The data was collected at 12,000 samples/second and has four different fault inches: 0.007, 0.014, 0.021 and 0.028 inches.

LSTM sequence-to-sequence networks

Unlike directly using classes as the cell state of LSTM, LSTM sequence-to-sequence networks use another sequence as the cell state, which will output each frame in this cell state sequence. The whole sequence will be predicted, and the classification will only be made at the final step after analysing the whole sequence. The structure of it is shown in (c) of Fig. 3.8.

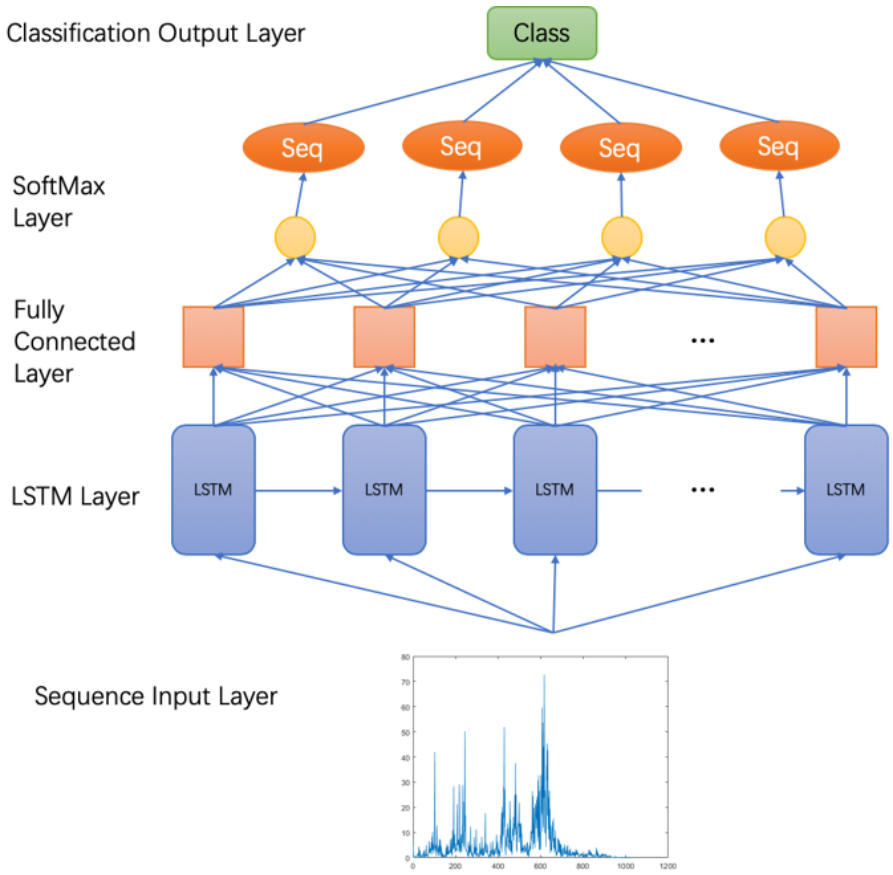


Figure 3.8: LSTM sequence-to-sequence networks, are a type of neural architecture that uses LSTM units to map input sequences to output sequences, commonly applied in tasks like machine translation and time-series prediction.

Experimental setup

In this experiment, I selected eight types of data in different states, the motor speed is 1750 (r/min), and the sampling frequency is 12 (kHz). To facilitate comparison, a sample is taken every 2048 data points, 50 samples of each non-overlapping type, 8 types, and 400 samples in total. Then, FFT will be used for feature extraction of the sample, and abs will be used for absolute value processing of data. The parameter information is shown in Table 3.5.

Table 3.5: Details of subjects in dataset.

Data no.	Fault type	Fault size/inches
1	Normal	0
2	Inner Race Fault	0.007
3	Ball Fault	0.007
4	Outer Race Fault	0.007
5	Inner Race Fault	0.014
6	Inner Race Fault	0.021
7	Ball Fault	0.021
8	Outer Race Fault	0.021

Results of LSTM sequence-to-sequence

The dataset has been divided into train set and test set with subject independent. The first 320 samples are used as train set and the last 80 samples are used as test set, first 250 samples are used as train set and the last 150 are used as test set, first 80 samples are used as train set and the last 320 are used as test set. 400 samples have been disrupted in advance.

Table 3.6: Comparison of Bi-LSTM and sequence-to-sequence (%).

Train Numbers	**320**		**250**		**80**	
	Bi-LSTM	Seq2Seq	Bi-LSTM	Seq2Seq	Bi-LSTM	Seq2Seq
1	96.3	20	92	15	76.9	15.6
2	97.5	12.5	95	19	81.9	15.6
3	100	21.3	97	23	84.2	30
4	100	23.8	100	24	88.1	35
5	100	31.3	100	11.5	90.3	26.3
Average	98.76	21.78	96.8	18.5	84.28	24.5

Table 3.6 shows that in the Bi-LSTM method, the more training sets, the higher the accuracy obtained. The highest single training accuracy reaches 100%, and the highest accuracy average is 98.76%. The sequence-to-sequence method has low accuracy in various situations, and the average value does not exceed 25%.
Hence, the Bi-LSTM network was the best choice.

Bi-LSTM networks

Experiment 1

In this experiment, I tested the accuracy for each kind of fault and normal state. The parameter information is shown in Table 3.7. 64 datasets in different states include one

normal state and 63 fault states. Each of these has 50 samples. Among them, the first 40 samples are training samples, and the left 10 samples are test samples, a total of 3200 samples, and the length of each sample is 2048.

Table 3.7: Motor speed and fault size classification.

Category	Motor Speed (rpm)	Fault Size (inches)
Normal	1730	-
	1750	-
	1772	-
	1797	-
Inner fault	1730	0.007, 0.014, 0.021, 0.028
	1750	0.007, 0.014, 0.021, 0.028
	1772	0.007, 0.014, 0.021, 0.028
	1797	0.007, 0.014, 0.021, 0.028
Ball fault	1730	0.007, 0.014, 0.021, 0.028
	1750	0.007, 0.014, 0.021, 0.028
	1772	0.007, 0.014, 0.021, 0.028
	1797	0.007, 0.014, 0.021, 0.028
Outer center fault	1730	0.007, 0.014, 0.021
	1750	0.007, 0.014, 0.021
	1772	0.007, 0.014, 0.021
	1797	0.007, 0.014, 0.021
Outer Orthogonal fault	1730	0.007, 0.021
	1750	0.007, 0.021
	1772	0.007, 0.021
	1797	0.007, 0.021
Outer Opposite fault	1730	0.007, 0.021
	1750	0.007, 0.021
	1772	0.007, 0.021
	1797	0.007, 0.021

I divided the data into 4 parts; the first part was the comparison of normal and inner faults, totally 20 classes. The second part was the comparison of normal and Ball faults, totally 20 classes. The third part was the comparison of normal and outer faults, totally 32 classes. The last part was the comparison of normal and all faults, totally 64 classes.

As shown in Table 3.8, for raw data, the best result was outer faults with normal with the accuracy achieved 30.9% and 17.5% for all faults with normal.
As shown in Table 3.9, the best result was outer faults with normal which the accuracy achieved 94.6% after FFT, and 92.4% for all faults with normal.

Table 3.8: Comparison of fault and normal categories without FFT.

Category	**Types**	**Samples**	**3 Times Result(%)**	**Average (%)**
Inner & Normal	20	1000	8, 9, 16.5	11.2
Ball & Normal	20	1000	9, 9, 10.5	9.5
Outer & Normal	32	1600	27.8, 25.9, 39.1	30.9
All Faults & Normal	64	3200	10.2, 18, 24.2	17.5

Table 3.9: Comparison of fault and normal categories with FFT.

Category	**classes**	**Samples**	**3 Times Result(%)**	**Average (%)**
Inner & Normal	20	1000	89.5, 94.5, 96.5	93.5
Ball & Normal	20	1000	82.5, 82.5, 82.5	82.5
Outer & Normal	32	1600	93.4, 96.9, 93.4	94.6
All Faults & Normal	64	3200	91.1, 93.1, 93.1	92.4

These two tables show the excellent performance of Bi-LSTM combined with FFT, integrating Bi-LSTM and FFT enhances signal processing by utilizing the strengths of both temporal and frequency-based feature extraction, improving the accuracy, robustness, and generalization of models in tasks like fault detection and time-series classification.

Experiment 2

The parameter information is shown in Table 3.10. Twelve data in different states include one normal state and eleven fault states, motor speed is 1750 r/min. Each of these has 400 samples. Among them, the first 350 samples are training samples, and the left 50 samples are test samples, a totally of 4800 samples, and the length of each sample is 300.

Table 3.10: Description of 12 states, one normal state and eleven failure states.

Data no.	Fault type	Fault size/inches
1	Normal	0
2	Ball Fault	0.007
3		0.014
4		0.021
5		0.028
6	Inner Race Fault	0.007
7		0.014
8		0.021
9		0.028
10	Outer Race Fault-Center@6:00	0.007
11		0.014
12		0.021

The last vision of network parameters in this chapter is in the following Table 3.11.

Table 3.11: Parameters of network.

Name of Parameters	Size of Parameters
Mini Batch-Size	**20**
Input Size	**1**
Number Hidden Units	**100**
Number Classes	**12**
Max Epochs	**100**

Results of experimental 2

In this experiment, the experimental setting follows the same one in paper [123] for a fair comparison. In this setting, the subset data related to 12 faulty types are used, as shown in Table 3.10. In [123], the result of the recognition accuracy is 98.47%, and the standard deviation obtained by reinforcement neural architecture after 10 experiments are 0.61.

The average classification accuracy rate of proposed method on raw data can reach 94.88%, and the accuracy standard deviation after 10 experiments is 1.90 as shown in Table 3.12. The average classification accuracy rate for data with FFT can reach 99.70±0.23%, improved by 4.82%. Compared wi-th neural reinforcement architecture, enhanced by 1.23%, the classification accuracy of test samples is summarized in Table 3, which is better than reinforcement neural architecture. Therefore, it can be considered that FFT data in Bi-LSTM has better performance.

Table 3.12: Results of comparison paper, Bi-LSTM for raw data and FFT data.

Method	Average accuracy (%) ± standard deviation (%)
Reinforcement neural architecture [123]	98.47 ± 0.61
Raw data+Bi-LSTM	94.88±1.90
FFT+Bi-LSTM	**99.70±0.23**

And the 10 times of results show in table 3.13, the best result for raw data is 97.83% and the worst is 92%. The best result for data with FFT approached 100% and the worst is 99.33%, which improved 2.17% and 7.33% respectively.

Table 3.13: Test results of raw data and data with FFT.

NO. Times	Raw data+Bi-LSTM %	FFT+Bi-LSTM %
1	97.83	100
2	96.67	100
3	96	99.83
4	96.5	99.83
5	95.67	99.83
6	94	99.67
7	93.67	99.5
8	93.67	99.5
9	92.83	99.5
10	92	99.33

It is significantly better than the comparison network structures, it can be seen that the proposed method has better performance. The diagnostic results are summarized in Fig. 3.9.

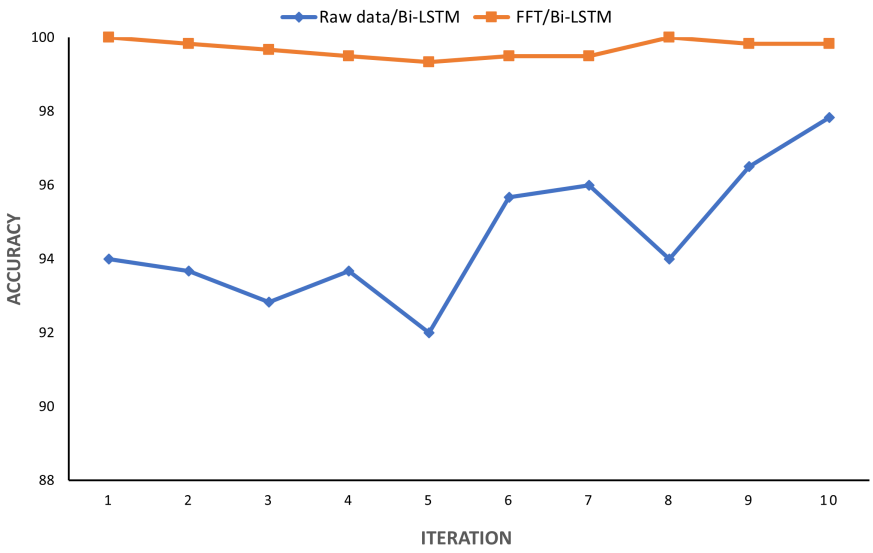


Figure 3.9: Accuracy of the 10 times. The horizontal axis is the number of iterations and Vertical axis is accuracy. The blue folding line is the result of the raw signal combined with the Bi-LSTM network, the orange folding line is the result of the signal after FFT combined with the Bi-LSTM network.

3.5 Summary

Deep learning methods, in particular, have become increasingly popular in recent years. As a result, deep learning-based automated recognition is now utilized in numerous new sectors. The goal of increasingly more automatic recognition systems is to simplify our

lives. In this chapter, the signal is transformed from the time domain to the frequency domain by fully using the FFT. The best result, when combined with Bi-LSTM, is 100%, while the mean value over ten times is 99.70±0.23 %. Future studies on the existing research issue can include a variety of methods, including EMD. Practical engineering challenges will be solved using some methodologies that will be enhanced and employed in this contribution.

Chapter 4

Real-Time Monitoring of Wind Turbine Bearing Using Simple Neural Network on Raspberry Pi

Although Chapter 3 demonstrated that a Bi-LSTM model trained on frequency-domain features can achieve high classification accuracy for bearing fault detection, the method relies on offline computation using desktop-level resources. This limits its applicability in real-time industrial monitoring scenarios.

To overcome this limitation, this chapter focuses on designing and implementing a lightweight fault detection system deployable on embedded hardware. Specifically, the proposed system utilizes FFT for efficient feature extraction and a simplified neural network model optimized for real-time inference on a Raspberry Pi platform. This enables on-site fault classification with minimal latency and without the need for external computation.

4.1 Introduction

Wind energy has now become a component of the energy mix, significantly bolstering its role in renewable energy sources. According to the International Energy Agency (IEA) in 2020 the global installed capacity for wind power amounted to 721GW [131]. Wind power plays a role in minimizing impact and fostering the adoption of low carbon alternatives

over traditional fossil fuels. The growing demand for energy has driven the acceptance of wind turbines as a pivotal technology for generating renewable energy, which is essential for our worldwide shift towards cleaner and more sustainable energy sources. This transition is essential in reducing our reliance on fossil fuels, curbing carbon emissions and promoting progress. The reliability of wind turbines largely determines the consistency of the electricity supply. When wind turbines are operating, they experience loads and stress in key components like bearings. Bearings are essential in converting wind energy into power. However, these components are susceptible to wear and damage, especially when exposed to conditions such as high winds, temperature fluctuations, or salty offshore atmospheres. Any defects in the bearings possibly affect the performance of wind turbines and may lead to unforeseen unexpected downtime and substantial maintenance costs. This ultimately impacts the viability and overall energy output of wind farms.

One of the most common reasons for wind turbine downtime is generator failure, which accounts for 37% of all failure downtimes [132]. Real-time generator defect detection can aid in preventing system shutdowns and mitigating their effects.

There are a number of defect detection systems for wind turbine generators. These methods include spatiotemporal attention-based long-short-term memory auto-encoder networks [132], marker-tracking for immediate rotational speed measurement [133], chaotic system and extension neural network fault diagnostics [134], time-varying models with augmented observers [135], deep learning approaches for sensor data prediction and fault diagnosis [136], sensor selection algorithms for real-time fault detection [137], enhanced variational mode algorithms fault diagnosis [138], image texture analysis for fault detection and classification [139], and cost-sensitive algorithms for online fault detection [140].

The wind turbine bearing is one of its key components and the performance and lifespan of the device are significantly influenced by its normal operation. However, wind turbine bearings frequently sustain damage and malfunction in long-term operation due to high loads and harsh weather conditions. These flaws, which pose major risks to the ability of the wind turbine to operate safely, may include rolling bead fatigue, inadequate lubrication, and unbalanced load [141].

For prompt maintenance actions, shorter stop times, and lower maintenance costs, real-time detection of wind turbine bearing failures is crucial. Preventive maintenance can be accomplished through real-time monitoring quickly and precisely detecting bearings, preventing further degradation of faults, and timely correcting faults to maintain wind turbines' continuous functioning and dependability.

Currently, the detection of bearing faults mainly relies on traditional vibration analysis and oil analysis [141]. These techniques often involve processing amounts of raw data, which makes real-time detection and prediction more complicated. Recently, deep learning techniques such as CNN and LSTM have also been used in bearing fault detection [142–144]. However, their complex model structures and numerous parameters make it challenging to deploy them in real time on edge devices. With Variational Mode Decomposition with Deep Convolutional Neural Networks (VMD-DCNNs) [145], it is possible to diagnose rolling bearing faults that do not require manual labor or human experience. The performance variations of existing approaches in various contexts and situations are obtained by extracting characteristics from each IMF. The Multi-Scale Convolutional Neural Network with Bidirectional Long Short-Term Memory (MSCNN-BiLSTM) model [146], using a weighted majority voting rule, enhance the intelligent fault diagnosis of bearings in wind turbines under complex working and testing environments, achieving improved diagnostic performance compared to existing methods. The deep bi-directional long short-term memory (DB-LSTM) [147] method circumvents feature selection challenges, enhances computational efficiency, and generates simulated data that closely resemble real-world settings, hence improving the approach's applicability. Supervisory Control And Data Acquisition (SCADA) is ensemble method that consists of XGBoost [148], a framework based on ensemble learning and genetic algorithms, utilizing SCADA data to detect faults in wind turbines. The Deep Convolutional Neural Network (DCNN) and Synchro Squeezing Transform (SST) [149], compared to traditional spectral analysis methods, can automatically identify fault features, avoiding misdiagnosis and missed diagnosis that may be caused by manual identification, and its excellent classification effect has been verified through experiments. The Self-Adaptive Teaching-Learning-Based Optimization-Multi-Layer Perceptron (SATLBO-MLP) [150] is a data-based wind turbine bearing fault diagnosis method that is applied to the SATLBO algorithm optimization MLP model. The related experimental results demonstrate the effectiveness of the method.

Some studies have attempted to address this issue by training models in the cloud [151] and then transferring them to edge devices. However, this method relies on a network connection and has challenges in meeting real-time performance demands in practical applications. Currently, research mostly concentrates on the impact of models, while giving less consideration to the practical viability of models, such as optimizing model size and predictive latency. This aspect requires further enhancement.

Overall, real-time defect detection research on wind turbine bearing failure has not been conducted, and most methods need to process raw data, which will cost time in practical engineering applications, resulting in the inability to make strategic adjustments and take appropriate measures in a timely manner, which may lead to irreparable losses. In gen-

eral, the existing approaches for detecting bearing faults still have limitations in terms of real-time capability, user-friendliness, and other factors, making it challenging to fulfill the requirements of real-time monitoring and prediction in industrial settings.

In spite of that, deep learning exhibits significant promise for utilization in wind turbine health monitoring, particularly in the detection of bearing faults. Compared to traditional machine learning techniques, the main benefits of deep learning include the following: Unsupervised learning has the capacity to extract intricate and non-linear patterns from data without the need for manual configuration, making it particularly well suited for complicated datasets [152]; Deep learning is acquiring intricate functional connections by utilizing multi-layer network architectures with strong fitting capabilities [153]; Finally, its implements of end-to-end learning, autonomously capturing the geographical and temporal relationships within the data [154]. Due to the growing processing power and data volume, performance is consistently enhancing and has surpassed manually built algorithms in numerous tasks [155]. Furthermore, the operational data of bearings contains abundant information in both the time and frequency domains. The neural network model has the capability to automatically acquire these intricate characteristics by utilizing its multilayer network architecture. Furthermore, it can proficiently address the difficulties posed by substantial data quantities and the fusion of multiple sensors. Thanks to advancements in edge computing and mobile Internet technology, the neural network model can now be deployed in real-time on edge devices. For example, the neural network model can be deployed on the control box of a wind turbine [136], resulting in a significant improvement in the response speed of bearing fault monitoring. In summary, neural networks have shown huge application potential in the monitoring of wind turbine bearing failure monitoring.

In this study, ”real-time” refers to the system’s ability to process and classify incoming sensor data within one second—the duration of each data acquisition window. Specifically, for the wind turbine dataset, vibration signals were collected at a sampling rate of 25,600 Hz, resulting in 25,600 data points per second. For 64 data points of each segment, the system is considered real-time if it can process this 2.5-ms each segment and provide diagnostic output before the next segment arrives.

In order to achieve the real-time detection of wind turbine bearing failure, this chapter aimed to apply machine learning methods in a Raspberry Pi to achieve high-precision and low-delay real-time monitoring with this portable device, which is of great significance for improving the reliability and utilization rate of wind power equipment. Our model can accurately identify patterns by training the physical characteristics of health and fault bearings. The data were divided into smaller segments so that the model can quickly analyze each segment and generate high-speed predictions. In addition, in order to achieve

a minimal delay treatment, high-efficiency algorithms were developed. The network was trained and, subsequently, the NN algorithm was embedded into the Raspberry Pi. The network has two fully connected layers, and the tested time was 0.06 milliseconds on the Raspberry Pi. The results of this chapter indicate that the model can accurately detect wind turbine bearing faults and provide real-time predictions within milliseconds of the fault occurrence. This model possesses the capability to generate real-time predictions and assess the holistic health condition of bearings, thereby substantially diminishing maintenance expenses and enhancing the accessibility and efficacy of wind turbines. In essence, this chapter demonstrates the potential of AI-driven solutions in optimizing the generation of renewable energy and mitigating reliance on fossil fuels.

4.2 Methodology

4.2.1 Overview of the proposed system

The overview of the proposed method is shown in Figure 4.1. The vibration signal is collected by accelerometers, which are located at the non-drive end in the wind turbine. Wind turbines are usually constructed in areas where wind energy is abundant, such as coastlines, mountains, plains, and deserts.

Figure 4.1a shows wind turbines constructed on grasslands. The data were trained in the model, which was implemented in Python language on PC. The data were divided into small segments; thus, the model can quickly analyze each segment without other processing methods, as shown in Figure 4.1b,c. The trained neural model was embedded in the Raspberry Pi, as is shown in Figure 4.1d. The last step was for the trained model to predict each segment's score, as is shown in Figure 4.1e.

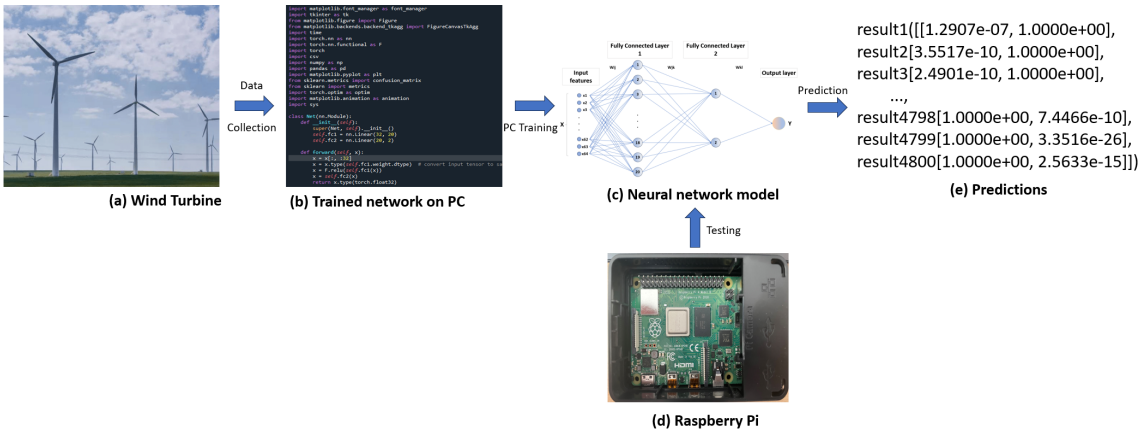


Figure 4.1: Overview of the system: (**a**) wind turbine; (**b**) modeling in python language; (**c**) neural network model; (**d**) Raspberry Pi implementation; and (**e**) predictions.

4.2.2 The description of the neural network model

Many research works use neural networks for fault detection or the monitoring of bearings. Fu et al. [156] monitored wind turbines with deep learning. In their paper, CNN and LSTM are used to analyze variables and monitor wind turbine gears. Gearbox bearing temperature data are processed for AI monitoring and troubleshooting. Asmuth and Korb [157] provide a basic three-dimensional CNN-based wake model to predict wind turbine wake flow fields. The model accurately predicts wake flow characteristics, showing its potential for wind turbine wake forecasts. Xie et al. [158] proposed an attention mechanism-based CNN-LSTM wind turbine fault prediction model. The CNN extracts feature, LSTM captures time sequence correlations, and an attention algorithm gathers fault-related target information from the ontology-annotated Semantic Sensor Network. The model was shown to be precise and generalized.

In this chapter, I adopted a real-time detection neural network model to assess wind turbine bearing failure, as shown in Figure 4.2. The model adopts the architecture of a MLP [159].

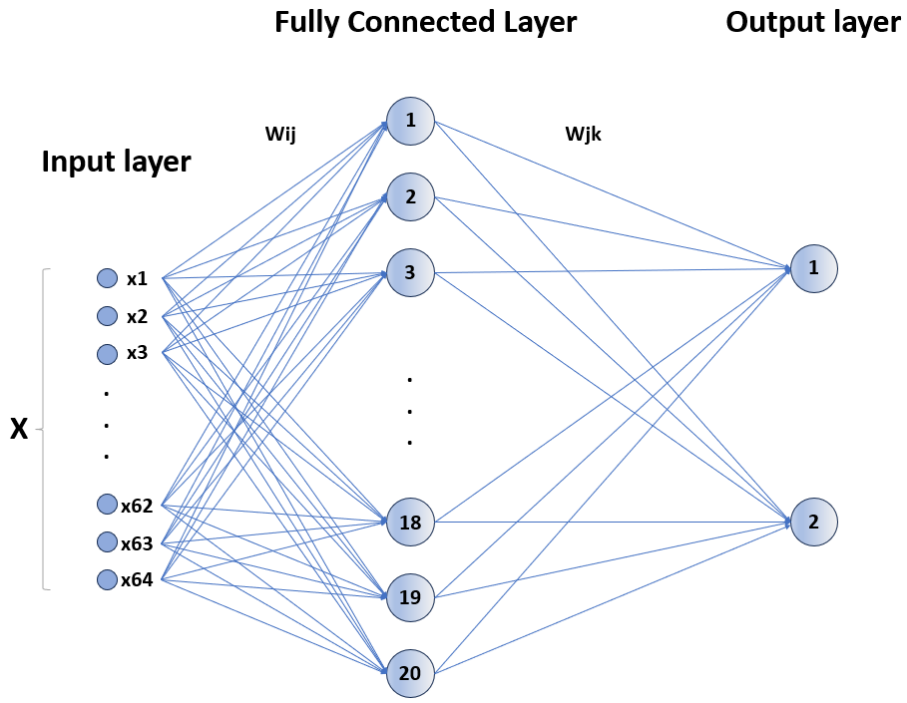


Figure 4.2: Proposed network, it contains two fully connected layers, and the second fully connected layer acts as the output layer.

The neural network model contains two fully connected layers (FC1 and FC2), and the second fully connected layer (FC2) acts as the output layer. The first fully connected layer (FC1) has 64 input features and 20 hidden units, which are used to learn complex features in the input data. The output layer (FC2) is a layer that takes the 20 output features from FC1 and reduces them to two output features, which are used to map the learned features to faults and normal categories.

4.2.3 From input layer to hidden layer

The input vector $x \in \mathbb{R}^{64}$. The first fully connected layer has a weight matrix $W_1 \in \mathbb{R}^{20\times 64}$ and a bias vector $b_1 \in \mathbb{R}^{20}$. The operation performed by this layer can be expressed as:

$$h = W_1 \times x + b_1 \tag{4.1}$$

Then, the ReLU (Rectified Linear Unit) activation function is applied to h. The ReLU function is defined as $f(z) = \max(0,z)$, which introduces non-linearity, allowing the model to learn more complex functions:

$$h' = \max(0,h) \tag{4.2}$$

In the hidden layer of the model, the ReLU (Rectified Linear Unit) [160] is used as the activation function. The ReLU function can enhance the non-linear modeling capabilities of the model and help capture the complex features of the input data.

4.2.4 From hidden layer to output layer

The vector h' is then passed through the output layer ('FC2'), which transforms the 20-dimensional input into a 2-dimensional output suitable for binary classification. Let the weight matrix of the output layer be $W_2 \in \mathbb{R}^{2\times 20}$ and the bias vector be $b_2 \in \mathbb{R}^2$. The final output y can be expressed as:

$$y = W_2 \times h' + b_2 \tag{4.3}$$

During the forward-direction of the model, the input data that first passed the first full connection layer (FC1) and activated the function ReLU are applied to the output of the hidden layer. Then, the final prediction result is output through the output layer (FC2).

In order to enable the model to accurately identify the mode of health and bearing faults, the method of monitoring learning was adopted for the training of model parameters. The labeling dataset was used for training and optimizing the weight and bias of the model by minimizing the loss function, so that the prediction results of the model were as close to the real label as possible.

The structure of neural network models is simple and effective, with less parameters and computing complexity, and it can operate efficiently on edge equipment such as a Raspberry Pi. Through this model, we can realize the real-time detection of wind turbine bearing faults and provide fast and accurate predictive results to provide feasible solutions for reducing maintenance costs and improving the availability and efficiency of wind turbines.

4.2.5 Acceleration sensor and data logger

As shown in Figure 4.3a, the primary purpose of the acceleration sensor located on the bearing is to monitor and assess the condition and functionality of the bearings. These sensors offer an efficient means for detecting and preventing possible malfunctions by monitoring the acceleration of the vibration in the bearing while it is in operation [161–163].

As shown in Figure 4.3b, the data logger obtains an analog signal from the acceleration sensor at the bearing, and uses the A/D converter to convert it to a digital signal. Then, the conversion signal is stored [164, 165].

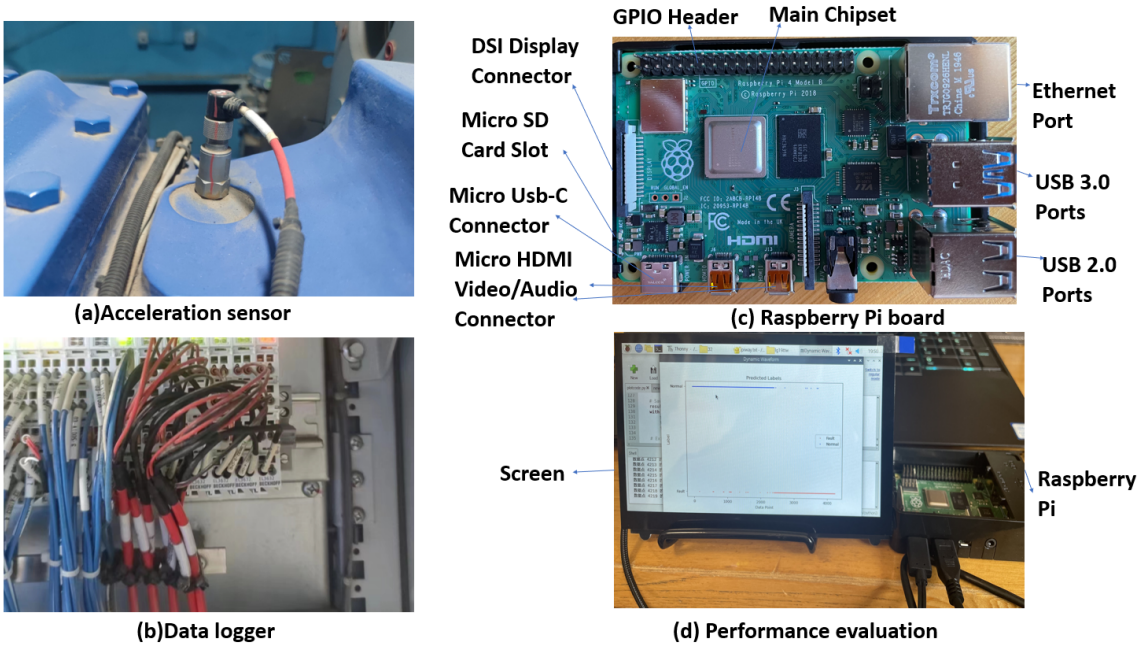


Figure 4.3: Embedded devices. (**a**) Acceleration sensor; (**b**) data logger; (**c**) Raspberry Pi board; and (**d**) performance evaluation.

4.2.6 Raspberry Pi

With Eben Upton serving as the project manager, the "Raspberry Pi Charity Foundation" was established in the UK. The world's smallest desktop computer, sometimes referred to as a card-type computer, was officially introduced in March 2012 by Emben Apoton [166], a Cambridge University research lab.

In this chapter, the network model is trained on the desktop and embedded it into Raspberry Pi 4 [167]. It can be seen as a small and easy to carry computer. As shown in Fig. 4.3 (c), some parts are marked and the touch screen as shown in Fig. 4.3 (d).

The Raspberry Pi was chosen over the Arduino Nano 33 because the latter lacks the computational capacity required for deep learning model inference. Specifically, the Arduino Nano 33 is designed for lightweight sensor interfacing and control tasks, but it does not have sufficient processing power or memory to support neural networks like BiLSTM or CNNs. In contrast, the Raspberry Pi 4B offers a full Linux environment, multi-core CPU, and enough RAM to enable on-device FFT, signal processing, and inference with TensorFlow Lite. Furthermore, the Raspberry Pi ecosystem is rapidly evolving, with increasing support for AI applications, making it a more future-proof platform for real-time fault detection systems.

4.3 Experiments

4.3.1 Dataset

There are many acceleration sensors on the bearing of the wind turbine generator. The data used in the experiment were the signal from one of the acceleration sensors. Data were collected from the turbine generator of a company. The vibration data were collected at 25,600 samples/second and a collection time of 3 s.

The generator is a device that converts mechanical energy into electrical energy. It usually consists of a rotor and a stator, where the rotor is supported by bearings rotating on a shaft. The generator bearing is an important component that supports the rotor, and its proper functioning is critical to the performance and reliability of the generator. It is shown in Figure 4.4a.

In order to monitor the health of the generator bearings, acceleration sensors are installed on the bearings. An acceleration sensor is a device that measures the acceleration of an object. When a generator is in operation, the bearings are subjected to a variety of forces and vibrations that cause the bearings to change in acceleration. The acceleration sensor senses and measures these changes and converts them into a digital signal. It is shown in Figure 4.4b.

(a) Wind Turbine generator (b) Acceleration sensor

Figure 4.4: Data collection platform. (**a**) Wind turbine generator; (**b**) acceleration sensor.

Important information about generator bearing vibration can be obtained by analyzing the data collected by the acceleration sensor. Vibration data can include parameters such as the vibration amplitude, the frequency, and the time variation. These data are critical for monitoring the health of the bearing. Abnormal vibration data may indicate bearing failure

or wear, such as loose bearings, overheating, or damage. By detecting these problems in time, the safety monitoring system can take appropriate repair and maintenance measures to prevent further damage or failure from occurring.

Therefore, acceleration sensor data acquisition is crucial for generator bearing monitoring and maintenance. It provides valuable information to better manage and optimize the generator system, ensuring its proper operation and extended service life.

4.3.2 Experimental baseline

Neural networks are particularly well-suited for handling large-scale datasets and high-dimensional data, especially deep learning network structures that perform exceptionally well on big data. Compared to traditional machine learning algorithms, neural networks are less sensitive to data volume and often achieve better results on large datasets. Hence, my experiment used neural networks for the dataset. The following Table 4.1 is the pre-experiment.

Table 4.1: Accuracy of different samples in each neural network(%).

Sample Size	Trilayered	Wide	Narrow	Bilayered	Medium	Sampling time
2048	83.3	78.9	77.2	76.7	73.3	80 ms
1024	91.4	76.8	87.3	86.8	79.7	40 ms
512	96.4	93.5	94.4	96.4	96	20 ms
256	97.5	99	97.8	98.7	98.2	10 ms
128	97.6	99.3	97.3	98.3	98.3	5 ms
64	98.8	98.9	98.6	98.8	99	2.5 ms
32	99.2	99.7	99.2	99.2	99.6	1.25 ms
16	99	99.4	99	99.1	99.4	0.625 ms

As sample length decreases, the accuracy of all networks gradually improves. With shorter sample lengths (such as 64), the accuracy is consistently close to 99%, suggesting that this length is sufficient to capture essential features and provide high accuracy. The sample length of 16 is too short and may yield high accuracy due to sample bias rather than generalization capability. At such a short length, differences in network structure have minimal impact, making it difficult to meaningfully distinguish the effects of network architecture. Therefore, starting from a sample length of 64 for further experiments will provide a more practical evaluation of the model's performance and ensure sufficient feature capture.

4.3.3 Experimental setup

Data analysis

Two different states are Normal and Fault. Each state has 9600 segments in testing, 2400 segments in training, and each segment has 64 samples; details are shown in Table 4.2.

Table 4.2: Description of data.

Fault Type	Training Segments	Testing Segments	Samples in Each Segment
Normal	9600	2400	64
Fault	9600	2400	64

As shown in Figure 4.5, the length of each sample of raw data was 64, and the raw data produced irregular waveform diagrams. The difference in the amplitude value was readily apparent. After FFT, the difference between health and faults at the peak value was also obvious.

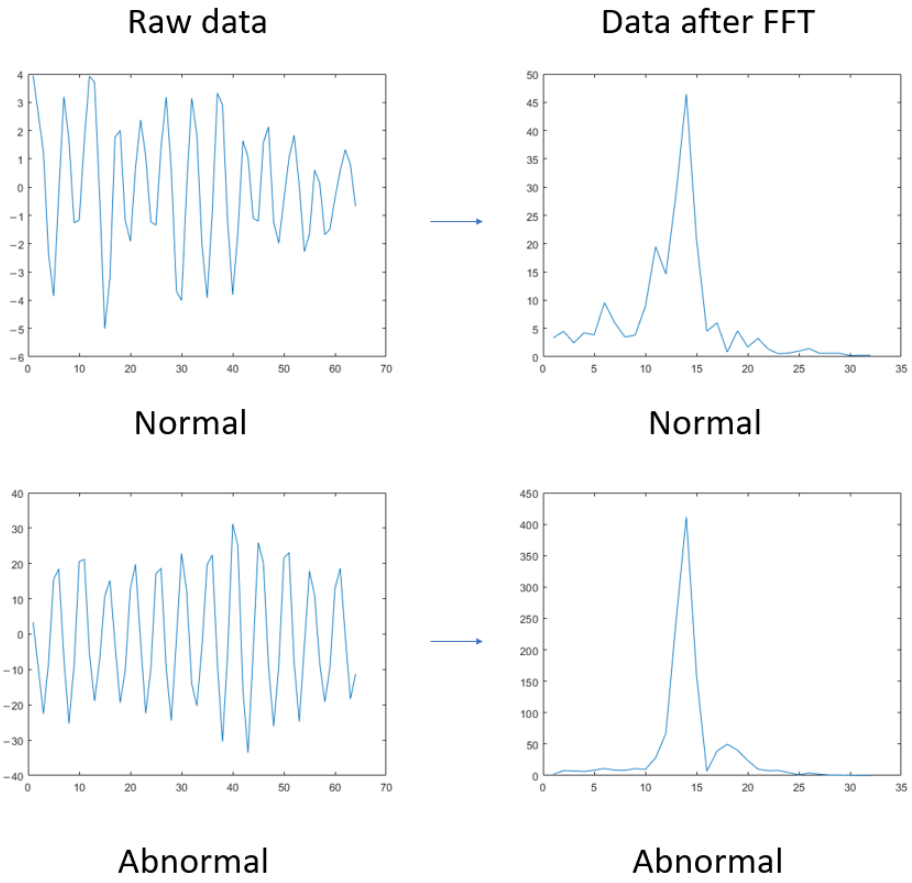


Figure 4.5: Data spectrum, the left part is raw data, each of them has 64 points, the right part is data with FFT, each of them has 32 points.

Data groups

The parameter information is shown in Table 4.3. The data were evenly divided into 10 groups of data with different states, including five normal states (Group 1–5) and five fault states (Group 6–10). In the experiment, in order to validate the performance of the methodology, comparison models were introduced, i.e., a medium neural network, wide neural network, bi-layer neural network, tri-layer neural network, and narrow neural network. The validation of the method accuracy was determined under 5-fold cross-validation.

Table 4.3: Dataset was split into training and testing subsets.

Training Group		**Testing Group**	
Subset 1	2, 7, 3, 8, 4, 9, 5, 10	Subset 1	1, 6
Subset 2	1, 6, 3, 8, 4, 9, 5, 10	Subset 2	2, 7
Subset 3	1, 6, 2, 7, 4, 9, 5, 10	Subset 3	3, 8
Subset 4	1, 6, 2, 7, 3, 8, 5, 10	Subset 4	4, 9
Subset 5	1, 6, 2, 7, 3, 8, 4, 9	Subset 5	5, 10

4.3.4 Experimental results

In this experiment, to verify the performance, five comparison networks were introduced. The first was used to confirm the accuracy of each method and the second was used to confirm the testing time of each method.

Comparison performance

In this part, the raw data were trained and tested with a few neural network methods [168]. For testing the comparison methods, the best method was the medium neural network, whose accuracy was 99.0%. The poorest one was the narrow neural network, whose accuracy was 98.6%. For the proposed method, the accuracy achieved was 99.8%. The accuracy data are shown in Figure 4.6.

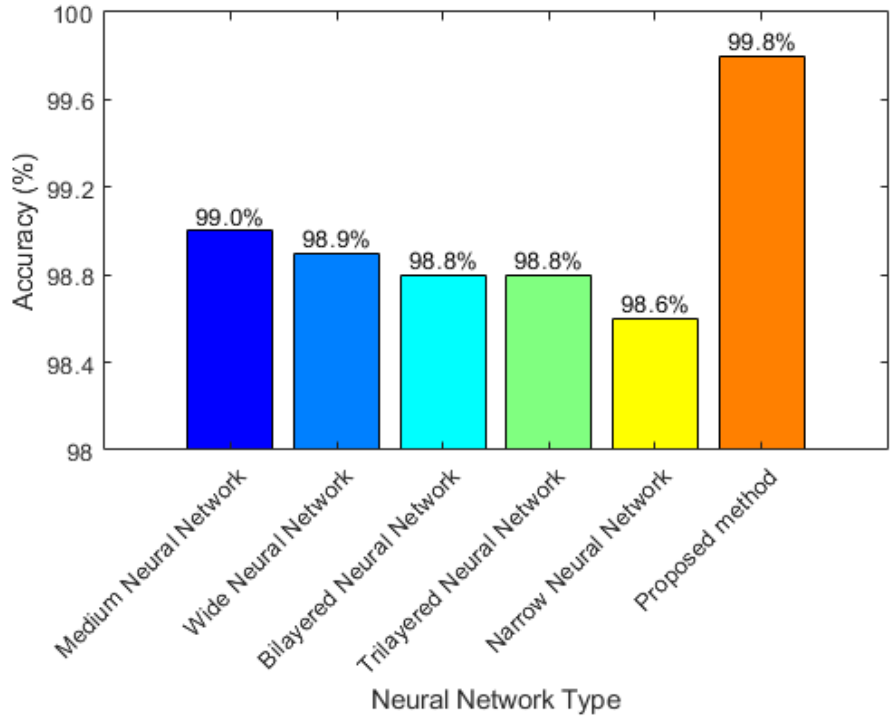


Figure 4.6: Accuracy comparison of raw data, the proposed method and other five methods.

For the proposed method, the network output was used to classify tasks (using 'CrossEntropLoss'). The loss curve of changes with training progress shows that the training converged, as is shown in Figure 4.7.

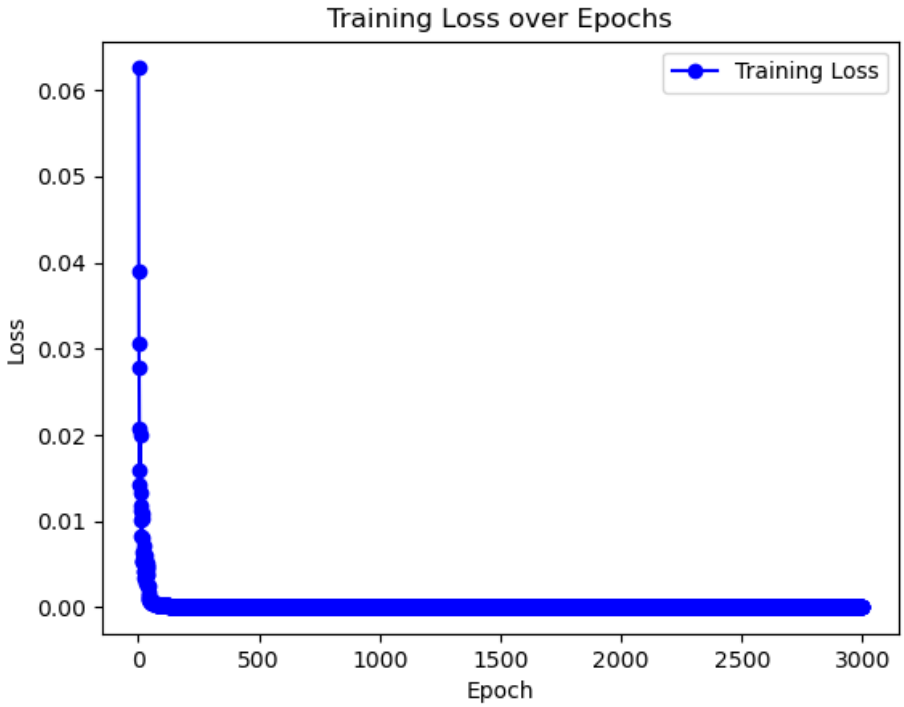


Figure 4.7: The loss curve of changes with training progress in proposed simple neural network.

Speed

In this part, the tested times are shown in Figure 4.8. From the five comparison methods, the fastest was the narrow neural network, whose testing time was 0.72 ms. The slowest was the wide neural network, whose testing time was 3.67 ms. For the proposed method, the testing time achieved was 0.06 ms.

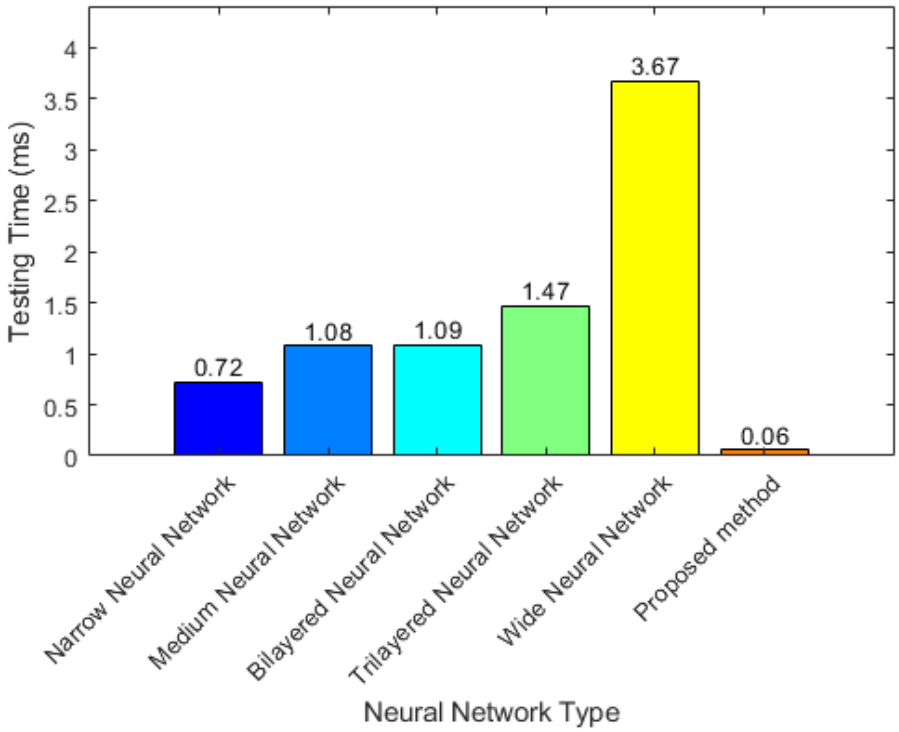


Figure 4.8: Testing time comparison of raw data, the proposed method and other five methods.

For the compared methods, they had a similar accuracy at around 99.0%. However, the testing time of the wide neural network was greater than that of the other methods at 3.67 ms. The main reason for this is that this network has a wide layer. The fastest comparison method was the narrow neural network because of its narrow layer. The average accuracy of the proposed method was 99.8% and the testing time was within 0.06 ms under 5-fold cross-validation. Compared with the other methods, these values represent improvements of 0.8% and 0.66 ms, respectively. It is obvious from the figure 4.8 that the proposed method had a better performance than the other methods.

4.4 Analysis

Achieving a practical model requires a small model, with a high accuracy that can function in real time.

4.4.1 Effect of different segments and epochs

There are four types of segments: the first kind of segment has 8 samples, the second segment has 16 samples, the third segment has 32 samples, and the last segment has 64 samples. In this experiment, different epochs were tested: 20 epochs, 200 epochs, 1000 epochs, 2000 epochs, and 3000 epochs. In addition, the layer size in this experiment was 10. The results are shown in Figure 4.9.

From Figure 4.9, the best performance was observed for 32 sample segments and 64 sample segments. Among them, the best results were seen at 3000 epochs. For 32 sample segments, the result was 96.8% and, for 64 sample segments, it was 98.5%. Hence, the segments of 32 and 64 samples at 3000 epochs were tested in the next experiment.

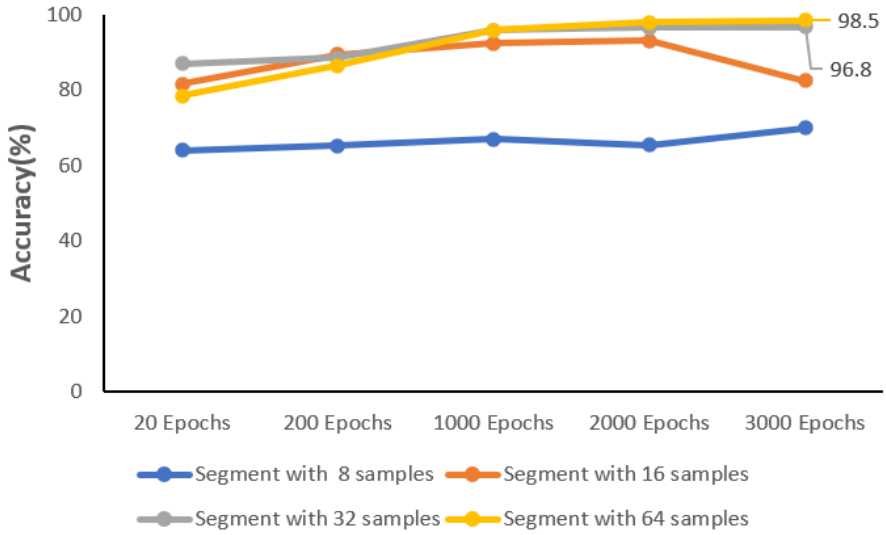


Figure 4.9: Accuracy of different samples in each segment.

4.4.2 Effect of different number of nodes

For testing the effect of a different number of nodes in the network, two ablation experiments were introduced. Details are shown in Table 4.4.

Table 4.4: Description of different segments.

Samples in Each Segment	Fault Type	Training Segments	Testing Segments
32	Normal	19,200	4800
	Fault	19,200	4800
64	Normal	9600	2400
	Fault	9600	2400

In the first experiment, each of these had 2400 segments, and each segment had 32 samples. Among them, 5-fold cross-validation was applied. Moreover, 19,200 segments were training segments, and the remaining 4800 segments were test segments.

For the second experiment, each of these had 1200 segments, and each segment had 64 samples. Among them, 5-fold cross-validation was applied. Moreover, 9600 segments were training segments, and the remaining 2400 segments were test segments.

In this part, testing was performed with different numbers of nodes, i.e., 5, 10, 15, and 20, in the fully connected layer and at 3000 epochs to validate the model. There were five groups. The final model was obtained through training data and was tested with the remaining data from this model.

As shown in Figure 4.10, the best performance was obtained for a 20-layer ANN, which achieved 99.8% for 32 samples and 99.7% for 64 samples with a model size of around 5 kb. Therefore, the next step was to confirm the testing time so that the experiment could achieve real-time detection.

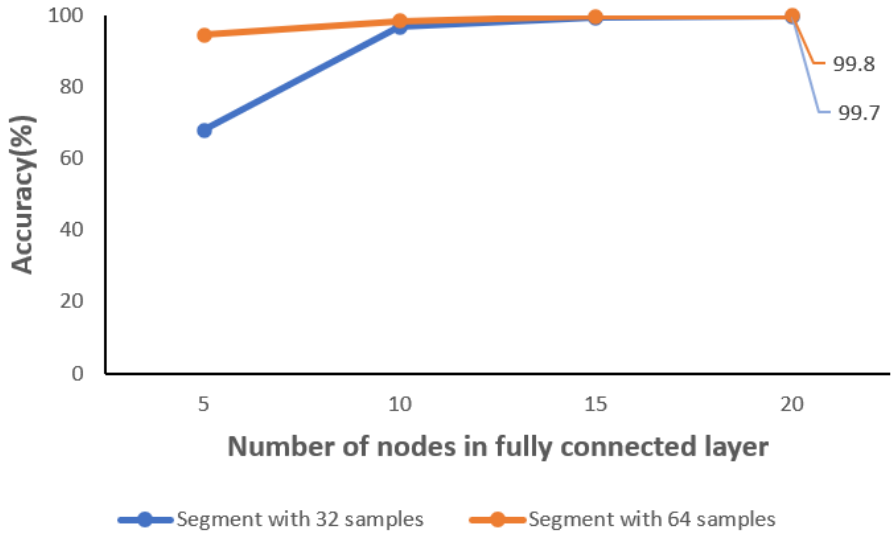


Figure 4.10: Accuracy under different numbers of nodes in the fully connected layer.

4.4.3 Effect of speed

In this part, the length of the layer was 20 and the epoch was 3000. The testing time was tested on different devices: a desktop computer and the Raspberry Pi. The results are shown in Table 4.5.

Table 4.5: Test results and time of raw data on Raspberry Pi and desktop computer.

Samples of Segment	Devices	Accuracy	Testing Time	Sampling Time
32	Desktop	99.7%	0.005 ms	1.25 ms
	Raspberry Pi	99.7%	0.028 ms	
64	Desktop	99.8%	0.009 ms	2.5 ms
	Raspberry Pi	99.8%	0.058 ms	

On the Raspberry Pi, for 64 samples of segments, the testing data were divided into 2400 segments, and each segment had 64 points.

The proposed system can efficiently process the sensor data and performs rapid analysis and prediction within 0.059 milliseconds per data segment. The experimental results demonstrate that the model achieves a 99.8% accuracy in detecting wind turbine bearing faults within milliseconds of their occurrence.

Regarding the 32 samples per segment, the testing dataset was divided into 4800 individual segments, with each segment consisting of 32 samples. In terms of system performance, the proposed approach exhibits efficient data processing capabilities, enabling swift analysis and prediction within a remarkable timeframe of 0.028 milliseconds per data segment. The experimental findings underscore the model's exceptional accuracy, with a reported 99.7% detection rate for wind turbine bearing faults within milliseconds of their manifestation.

On the desktop, for 64 samples of segments, the testing data were divided into 2400 segments, and each segment had 64 samples.

The proposed system can efficiently process the sensor data and performs rapid analysis and prediction within 0.009 milliseconds per data segment. The experimental results demonstrate that the model achieves a 99.8% accuracy in detecting wind turbine bearing faults within milliseconds of their occurrence.

In the case of segments consisting of samples, the testing dataset was partitioned into 4800 individual segments, with each segment containing 32 samples. The proposed system demonstrates notable efficiency in processing sensor data, facilitating swift analysis and prediction within an impressive timeframe of 0.005 milliseconds per data segment. The experimental results substantiate the model's remarkable performance, yielding a detection accuracy of 99.7% in identifying wind turbine bearing faults within milliseconds of

their manifestation.

Both the Raspberry Pi and the desktop achieved identical outcomes by utilizing the proposed method. However, the testing times varied. The Raspberry Pi was significantly slower than the PC when it came to test speeds. The proposed method was significantly better, with respect to accuracy and execution time, compared to the comparison networks. The proposed method demonstrated superior performance and aligns with the specific requirements. Multiple models were chosen based on our requirements, i.e., for rapid speed, we selected 32 samples per segment, and for high precision, we chose 64 samples per segment. Each model was approximately 5 kb in size.

4.5 Summary

This chapter developed a machine learning-based model on a Raspberry Pi to detect wind turbine bearing faults in real time. The model was designed and trained on a desktop computer due to its higher performance. Then, real-time implementation was achieved by running the model on a Raspberry Pi for real-time wind turbine bearing fault detection. The experimental results demonstrate that the model achieves a high accuracy and rapid detection of faults within milliseconds of their occurrence. The model achieved an accuracy rate of **99.8%** and the testing time was **0.059** ms, indicating its effectiveness and precision in detecting wind turbine bearing faults.

This chapter demonstrates the practical implications and potential applications of real-time fault detection in wind turbines using a neural network model. The model provides significant accuracy in quickly identifying bearing faults and providing immediate predictions during fault detection, resulting in reduced maintenance costs, increased turbine availability, and improved overall efficiency. By being able to identify faults during turbine operation in a timely manner, the development of this neural network model facilitates early warning and rapid response, minimizing downtime and associated maintenance costs. Ultimately, real-time fault detection ensures reliable turbine operation and increased availability.

The findings of this chapter are significant for promoting renewable energy generation and reducing reliance on fossil fuels. The model improves operational efficiency and reliability by enabling the real-time detection of wind turbine bearing faults, minimizing energy waste. This facilitates the wider adoption of renewable energy sources, reducing dependence on finite fossil fuel resources and promoting sustainable energy development and environmental protection.

In the future, many works will require enhancement, and network design optimization will be of paramount importance. This model can only be used on specific models of wind turbines and is restricted to the generator bearings of these turbines. In different environments, rather than fully retraining the model, only minor updates to classifier parameters or thresholds may be sufficient, since core signal patterns remain stable. Future work will investigate lightweight domain adaptation techniques (e.g., transfer learning, incremental learning) to support cross-environment generalization without needing large new datasets or retraining from scratch. Future research must establish a multi-channel, multi-type, and multi-scenario detection model that can be further optimized to improve the generalizability of models.

While the proposed system has demonstrated high accuracy under controlled conditions, its performance may degrade in different operational environments due to variations in noise characteristics, load conditions, and mechanical structures. In future work, model retraining or adaptation will be explored to address domain shifts. Techniques such as transfer learning, incremental learning, or domain adaptation will be considered to enable the system to remain effective across different machinery and environments without requiring complete re-training from scratch.
Continued innovation and improvement in this area will drive renewable energy generation, reducing dependence on fossil fuels and paving the way for a cleaner, more sustainable energy future.

Chapter 5

Fault Detection of Wheelset Bearings Through Vibration-Sound Fusion Data Based on Grey Wolf Optimizer and Support Vector Machine

While Chapter 4 presents a real-time monitoring system based on acoustic signals and embedded deployment, it still suffers from limitations in signal diversity and environmental robustness. Relying on a single signal source (sound) may lead to degraded performance in noisy or variable conditions, which are common in industrial environments.

This chapter addresses these limitations by proposing a multimodal fault detection framework that fuses both vibration and sound signals. A Grey Wolf Optimization (GWO) algorithm is employed to tune the hyperparameters of Support Vector Machine (SVM) classifiers. The goal is to enhance detection accuracy, repeatability, and adaptability across different fault types and operating conditions.

5.1 Introduction

Bearings play a crucial role in transportation, with their operational safety and reliability directly impacting logistics efficiency and economic benefits. The wheelset bearings are key components that bear significant loads and operate in complex environments. Over extended periods of use, these bearings are prone to various faults. Therefore, timely and effective detection of wheelset bearings faults is essential for ensuring the safety of transportation.

The primary objective of bearing fault diagnostics is to detect probable defects by examining diverse data. Vibration data is a widely utilized technique for diagnosing bearing failures, as these faults generally result in anomalous vibration characteristics. There are three predominant analysis methods: time domain analysis, frequency domain analysis, and time-frequency analysis. Time domain analysis involves the examination of the time waveform to identify impact signals and periodic components. The statistical properties, such as mean, variance, peak value, kurtosis, and skewness, can potentially reveal changes in the vibration signal [169]. The analysis typically involves computing these features from segmented windows of the vibration signal and monitoring their trends over time. Substantial departures from baseline values can suggest the existence and intensity of bearing faults [170–172]. Time domain analysis is a straightforward and efficient method for detecting bearing faults; however, it may not offer as much comprehensive diagnostic information as frequency domain techniques [170]. Frequency domain analysis involves applying a Fourier transform to convert signals in the time domain into signals in the frequency domain in order to identify and analyze certain frequency components [173]. Chen et al. [174] introduced power function-based Gini indices II and III (PFGI2 and PFGI3), and through mathematical derivation and experimental validation using envelope analysis in the frequency domain, demonstrated their superior sparsity quantification capabilities and fault feature characterization performance in bearing condition monitoring. Power Spectral Density (PSD) is a measure of the power distribution of a signal over different frequencies; it displays the amplitude of different frequency components and is commonly used to detect specific defects in bearings, such as defects in the outer ring, inner ring, or rolling elements [175]. Chen et al. [176] proposed two new blind deconvolution methods using the modified smoothness index (MSI) in the time and frequency domains for squared envelope applications, effectively enhancing sparse features for rolling bearing fault diagnosis and demonstrating excellent diagnostic performance and robustness in experiments. Time-frequency analysis involves the application of techniques that incorporate both time and frequency data, such as the Short-Time Fourier Transform (STFT) and Wavelet Transform [177, 178]. These methods are able to capture the transitory properties of a signal with more accuracy [179].

Additionally, sound signal analysis is becoming more crucial in diagnosing bearing faults, as variations in sound signals might indicate changes in the operating conditions of the bearing. The conventional techniques used are analogous to the analysis of vibration data, encompassing time domain analysis, frequency domain analysis, and time-frequency domain analysis. Two more methods exist: sound pressure level (SPL) analysis and sound signature recognition. SPL analysis is an effective technique for diagnosing bearing faults by analyzing sound emissions from bearings. The main sources of bearing noise are vibrations from the inner ring and rolling elements (balls or rollers) and as a bearing enters the failure stage, there is a rise in SPL of 12–16 dB over the baseline level, accompanied by a change in sound quality [180,181]. SPL analysis includes three techniques: time waveform analysis, frequency spectrum analysis and time-frequency domain analysis [170, 181]. Sound signature recognition in bearing fault detection involves utilizing sound signal analysis techniques to identify and diagnose bearing faults. This approach captures and analyzes the sound signals generated by bearings during operation, identifying abnormal patterns and features that indicate the bearing's health status. Advanced signal processing techniques such as FFT [125], Wavelet Transform [178], Empirical Mode Decomposition [182], and Hilbert Transform are commonly used to extract fault features from noisy sound data [183–187].

Nevertheless, there are limitations when it comes to evaluating individual vibration signals or sound signals. For example, the presence of machinery might influence vibrations, making it difficult to discern certain defect features. In a similar manner, background noise has the potential to disrupt signals by concealing important fault characteristics. In data fusion, vibration and sound signals provide complementary information about machine condition, and data fusion has the potential to greatly enhance the effectiveness and dependability of bearing fault diagnosis systems, offering robust assistance for equipment preventative maintenance and fault prediction [188–191].

To enhance the accuracy and reliability of bearing faults detection systems, research investigations highlight the significance of data fusion approaches, which involve merging information from multiple sensors. Wan et al. [192] proposed a Fusion Multiscale Convolutional Neural Network (F-MSCNN) method that processes raw sound and vibration signals to achieve high accuracy and stable fault diagnosis of rolling bearings under varying operating speeds. Shi et al. [193] proposed a two-stage sound-vibration signal fusion algorithm that combines and weights fault features from multiple sound measurement points, extracts features using empirical mode decomposition and kurtosis superposition, and then unifies sampling frequencies to fuse sound and vibration signals again, achieving weak fault detection in rolling bearings. This method significantly improves fault feature

detection accuracy and signal-to-noise ratio, aiding in the status monitoring of bearing systems. Duan et al. [194] provided a comprehensive review of multi-sensor information fusion for rolling bearings, highlighting the significance of combining data from diverse sensors for improved fault diagnosis capabilities. Wang et al. [195] conducted a study on bearing fault diagnosis using vibro-sound data fusion and a 1D-CNN network, demonstrating the benefits of integrating vibration and sound information for enhanced fault detection. Gu et al. [196] introduced an enhanced SE-ResNet sound-vibration fusion method for rolling bearing fault diagnosis, integrating various techniques to effectively process sound-vibration signals. By integrating vibration and sound data, a comprehensive method for detecting bearing faults is achieved. This approach combines the advantages of both signal types, resulting in a more precise and detailed depiction of the system's status. Researchers have successfully built sophisticated models that incorporate vibration and sound data using modern computational approaches such as deep learning, feature fusion, and adaptive signal processing. These models are used for precise problem identification in bearing systems. These studies highlight the significance of data fusion approaches in utilizing the combined benefits of vibration and sound inputs to improve the accuracy and effectiveness of bearing defect detection systems. However, these deep-learning-based methods need large datasets for training and they are not easily implemented for real-time detection.

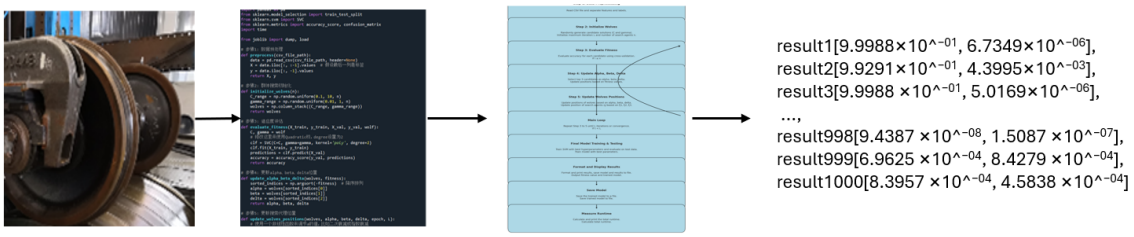
In this chapter, I introduce a novel approach by integrating the GWO [197] with a SVM [198] to optimize hyperparameters, specifically tailored for real-time analysis of vibration-sound fusion data. Yan et al. proposed GWO-SVM for smart emotion recognition, they used the Radial Basis Function (RBF) kernel of SVM and achieved high accuracy in their research [199]. I extend their method with various SVM kernels and provide rapid failure detection by preprocessing fusion data from vibrations and sounds. Data segmentation facilitated analysis, enabling the model to generate predictions at a remarkable speed of 0.0027 milliseconds per segment. In addition, the linear SVM model that was fine-tuned using GWO achieved a testing accuracy of 98.3%, outperforming the SVM and neural network models built in MATLAB. Furthermore, this model demonstrated significant efficiency in runtime assessments, making it extremely suitable for real-world settings. The proposed GWO-SVM model shows advantages in detecting defects in wheelset bearings. The model's capacity to generate real-time predictions and offer a comprehensive evaluation of the bearing's condition can greatly diminish maintenance expenses and enhance the accessibility and effectiveness of wheelset bearings. This chapter underscores the potential of integrating advanced optimization algorithms with machine learning techniques to enhance fault detection capabilities, ultimately contributing to more robust and efficient transportation systems.

5.2 Related works

5.2.1 Overview of the proposed method

An overview of the proposed method is shown in Figure 5.1. The vibration signal is collected by accelerometers, which are located on the axle box cover in the bearing area at the end of the wheelset; and the sound signal is collected by microphone located on both sides of the bearing. This data set offers vibration and sound data to efficiently identify various bearing faults.

Figure 5.1a shows the wheelset bearing. As shown in Figure 5.1b,c, FFT is utilized for analyzing vibration signals [125], whereas the MFCC is employed for analyzing sound signals [74]; split the data into different segments and then combine and integrate the characteristics of the two datasets by simply concatenating them; the model possesses the ability to rapidly examine every individual segment. The final stage involved the trained model making predictions for the score of each segment, as shown in Figure 5.1d.



(a) Wheelset (b) Python (c) GWO-SVM model (d) Predictions

Figure 5.1: Overview of the system: (**a**) wheelset; (**b**) modeling in python language; (**c**) GWO-SVM model; and (**d**) predictions.

5.2.2 The description of the GWO-SVM model

SVM has been widely applied for fault detection and diagnosis of bearings in rotating machinery. Pule et al. [200] proposed a method using PCA and SVM to achieve 97.4% accuracy in diagnosing bearing faults under varying speeds using vibration analysis. Yang et al. [201] introduced a triplet embedding-based method for classifying small sample rolling bearing fault datasets, achieving superior performance in fault diagnosis by using CNN for feature extraction and SVM for classification. Mo et al. [202] proposed a highly accurate (95.3%) and efficient (11.1608-s training time) method for diagnosing petrochemical rotating machinery bearing faults by combining ICEEMDAN-wavelet noise reduction, mutual dimensionless metrics, and MPGA-SVM, with further validation showing 97.1% accuracy on additional datasets.

In this chapter, I adopted a fast detection GWO-SVM model by analyzing fusion data from multiple models to assess bearing failure, as shown in Figure 5.2.

Step 1: Data preprocessing—Load the signals and extract their features and annotate their label.

Step 2: Initialize Wolves—Randomly generate the initial positions for the wolf pack, which are the candidate solutions for the SVM hyperparameters C_{SVM} and γ, initialize the maximum number of iterations L and the number of search agents n.

Step 3: Evaluate fitness—Evaluate accuracy for each candidate using cross-validation. If $i \leq n$:

a. Select the current candidate i: Choose the i-th wolf from the pack.

b. Train SVM model using current candidate: (1) Extract the SVM hyperparameters C_{SVM} and γ from the current candidate; (2) Initialize an SVM model with these parameters; (3) Train the SVM model on the training dataset.

c. Evaluate the model on the validation set: (1) Use the validation dataset to predict outcomes; (2) Calculate the accuracy of the predictions, which represents the fitness of the current candidate
The accuracy is defined as:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{5.1}$$

where TP and TN are the true positives and true negatives, while FP and FN are false positives and false negatives, respectively.

d. Record the fitness value of the current candidate: Save the fitness value for the current candidate.

e. Increment the index i: Move to the next candidate ($i = i+1$).

f. Repeat Step 3: Continue evaluating the next candidate until all candidates are evaluated.

Step 4: Update α_{wolf}, β_{wolf}, and δ_{wolf}.—Based on the fitness values, select the top three candidates as the α_{wolf}, β_{wolf}, and δ_{wolf}.

Step 5: Update Wolves positions—Update the positions of all wolves in the pack using the positions of the α_{wolf}, β_{wolf}, and δ_{wolf}.

Step 6: Main loop—Continue iterating through Steps 3 to 5 until the maximum number of iterations L is reached or the algorithm converges.

Step 7: Final model training and testing—Use the best hyperparameters found during the optimization to train the SVM model on the entire training dataset. Evaluate the trained model on the test dataset.

Step 8: Format and display results—Format the evaluation results and print them. Save the trained model and results to a file.

Step 9: Save model—Save the trained SVM model to a specified file path.

Step 10: Measure runtime—Calculate and print the total runtime and the testing time.

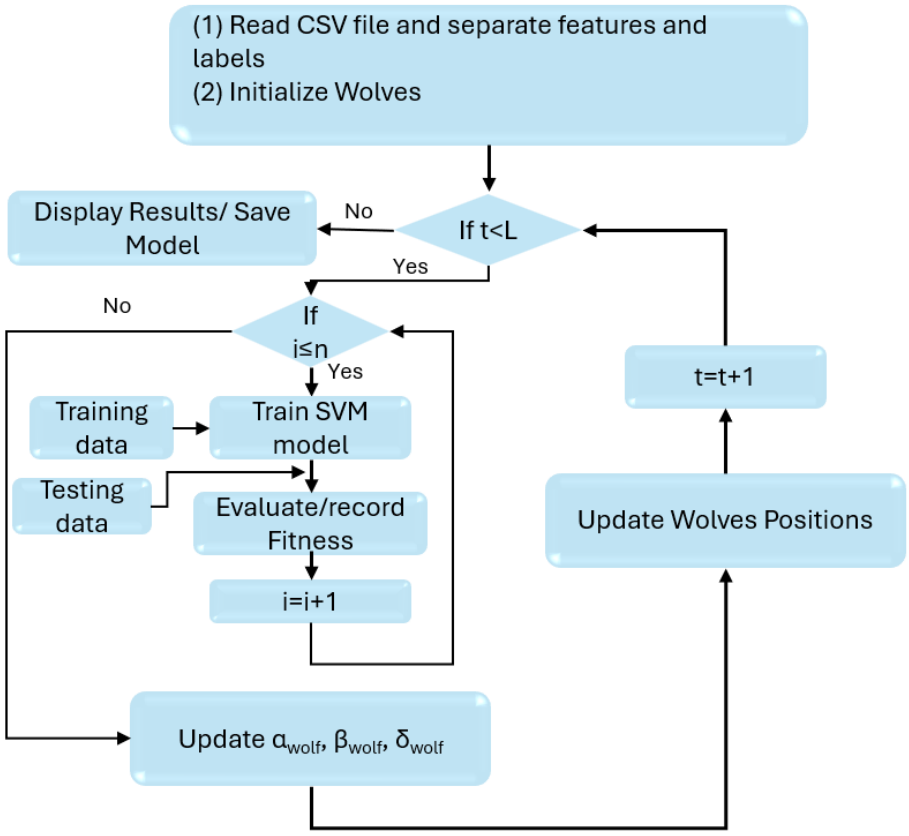


Figure 5.2: Proposed model, optimization is achieved by updating key parameters (e.g., Alpha, Beta, Delta) to enhance the model's performance.

Basic SVM formula

The decision function for SVM:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b \tag{5.2}$$

where α_i is the Lagrange multiplier, y_i is the label, x_i is the support vector, $K(x_i, x)$ is the kernel function, and b is the bias term.

GWO formulas

Updating parameter a:

$$a = 2 - 2\left(\frac{\text{epoch}}{L}\right)^2 \tag{5.3}$$

where epoch is the current iteration and L is the maximum number of iterations.
Calculating coefficients A and C_{wolf}:

$$\begin{aligned} A &= 2 \cdot a \cdot r_{wolf_{1-a}} \\ C_{wolf} &= 2 \cdot r_{wolf_2} \end{aligned} \tag{5.4}$$

where r_{wolf_1} and r_{wolf_2} are random numbers in the range $[0, 1]$.
Updating the positions of the wolves:

$$\begin{aligned} D_\alpha &= |C_{wolf} \cdot X_\alpha - X| \\ D_\beta &= |C_{wolf} \cdot X_\beta - X| \\ D_\delta &= |C_{wolf} \cdot X_\delta - X| \\ X_1 &= X_\alpha - A \cdot D_\alpha \\ X_2 &= X_\beta - A \cdot D_\beta \\ X_3 &= X_\delta - A \cdot D_\delta \\ X_{\text{new}} &= \frac{X_1 + X_2 + X_3}{3} \end{aligned} \tag{5.5}$$

Among them:

- D represents the distance between the current wolf and the α_{wolf}, β_{wolf}, and δ_{wolf}.
- $X_{\alpha,\beta,\delta}$ is the position of the α_{wolf}, β_{wolf}, and δ_{wolf}, representing the best solution found so far.

- X_{new} is the updated position of the wolf, the elements in X_{new} are essentially combinations of SVM hyperparameters, optimized through the GWO process to find the best parameter settings.

GWO algorithm steps

Linear Kernel function:

$$K(x_i, x_j) = x_i \cdot x_j \tag{5.6}$$

RBF Kernel function:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \tag{5.7}$$

Polynomial Kernel function:

$$K(x_i, x_j) = (x_i \cdot x_j + r)^d \tag{5.8}$$

Optimization objective:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C_{SVM} \sum_{i=1}^{n} \xi_i \tag{5.9}$$

Among them:

- w is the weight vector that determines the hyperplane for classification.
- b is the bias term.
- $\|w\|^2$ is the square norm of the weight vector used to control the complexity of the model.
- C_{SVM} is a regularization parameter used to balance the misclassification of training data and model complexity.
- ξ_i is a slack variable that allows certain samples to be misclassified.

The goal of this formula is to minimize the complexity and training error of the model, thereby improving its generalization ability.

Here, the GWO algorithm is used to optimize the hyperparameters of SVM (such as C_{SVM} and γ) to improve the accuracy and efficiency of fault detection.

Steps for combining GWO:

1. Initialize the positions of the wolves, for linear kernel, train the SVM with the linear kernel using the wolf's position parameter (C_{SVM}); train the SVM with the RBF kernel using the wolf's position parameters (C_{SVM}, γ), and train the SVM with the Polynomial kernel using the wolf's position parameters (C_{SVM}) and coefficient term (r), the degree of polynomial Kernel in this chapter is confirmed.
2. Evaluate the fitness of each wolf based on the classification accuracy of the SVM.
3. Update the positions of α_{wolf}, β_{wolf}, and δ_{wolf}.
4. Update the positions of the other wolves.
5. Repeat the above steps until the maximum number of iterations is reached.

Summary

The GWO algorithm is used to adjust the SVM hyperparameters C_{SVM}, γ and coefficient term r. These parameters significantly affect the model's performance. The adjustments are as follows:

- C_{SVM} determines the balance between minimizing the error on the training data and reducing the complexity of the model. A larger C_{SVM} value tries to classify every sample correctly, which may lead to overfitting, while a smaller C_{SVM} value allows some misclassifications, potentially improving generalization.
- γ controls the width of the Gaussian kernel. A larger γ value means higher sensitivity to individual data points, making the model focus more on local patterns, while a smaller γ value makes the model consider a broader range of data points.
- r adjusts the influence of higher-order terms in the polynomial kernel.

Using the GWO algorithm, I dynamically adjust the parameters C_{SVM}, γ and coefficient term r in the code to find the parameter combination that achieves the highest classification accuracy on the validation set. The adjusted model is then evaluated on the test set to assess its actual performance. Finally, the model is trained on the training set and validated on the test set, completing the process.

5.3 Experiments

5.3.1 Experiment dataset and baseline

The experiment used MATLAB 2022b and Python 3.9.

This fault detection has high accuracy and fast speed.
There are many sensors on the wheelset bearing. The data used in the experiment were the signal of one of the acceleration sensors and microphone sensors. The data were collected from the bearing in our laboratory. The data were collected at 25,600 samples/second.

Wheelset bearings are essential elements utilized in railway vehicles, including trains, subways, and light rail systems. The axles and wheels are supported by them, carrying the full weight of the vehicle. These bearings need to function consistently and dependably in diverse and intricate circumstances. It is shown in Figure 5.3a.

In order to monitor the health of bearings, sensors are installed on the bearings. It is shown in Figure 5.3b which is a cross-sectional view of Figure 5.3a, among them, 1 is tested bearing, 2 is accelerometer, 3 is auxiliary bearing, 4 is microphone sensor, 5 is friction wheel, 6 is motor, 7 is foundation. The height of the microphone sensor position is 300 mm, and the horizontal distance from the test bearing is 500 mm.

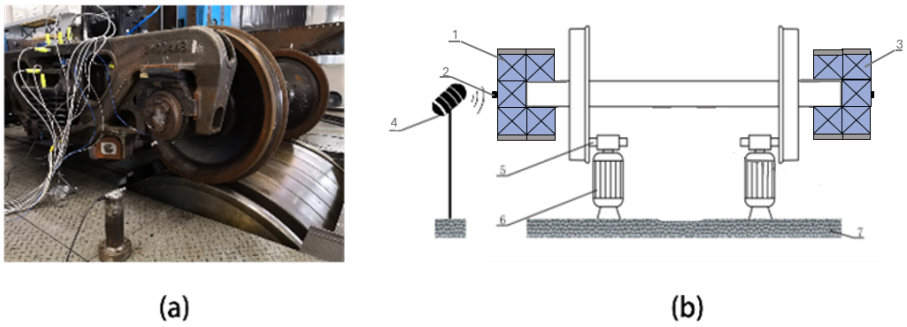


Figure 5.3: Data collection platform. (**a**) Wheelset bearings; (**b**) sensor installation location.

Accelerometers are predominantly employed to capture the vibration signals emitted by bearings. These signals can indicate the operating condition of the bearings, such as the existence of wear, imbalance, misalignment, or other mechanical problems. Microphone sensors are employed to capture the sound emissions generated by bearings during their functioning. Various sorts of flaws produce unique sound characteristics, and by analyzing these sounds, the state of the bearings can be initially evaluated. By integrating accelerometers and microphone sensors, it is possible to monitor and diagnose the operational condition of the bearings in a more comprehensive manner, enabling the quick identification and treatment of potential problems, ensuring the smooth functioning of the equipment.

Experiment 1

In this experiment, I tested the performance of each method with raw vibration data after FFT. There are four different states: Normal, Outer raceway scoring (referring to damage on the raceway surface where the rolling elements contact), Outer race scoring (a broader term encompassing damage on any part of the outer race), and Outer raceway pitting (referring to the pitting phenomenon on the raceway surface of the outer ring of a bearing). Each state has 100 segments, the length of each segment for vibration data is: 2048, 1024, 256, 128, 64, 32. The details are shown in Figure 5.4.

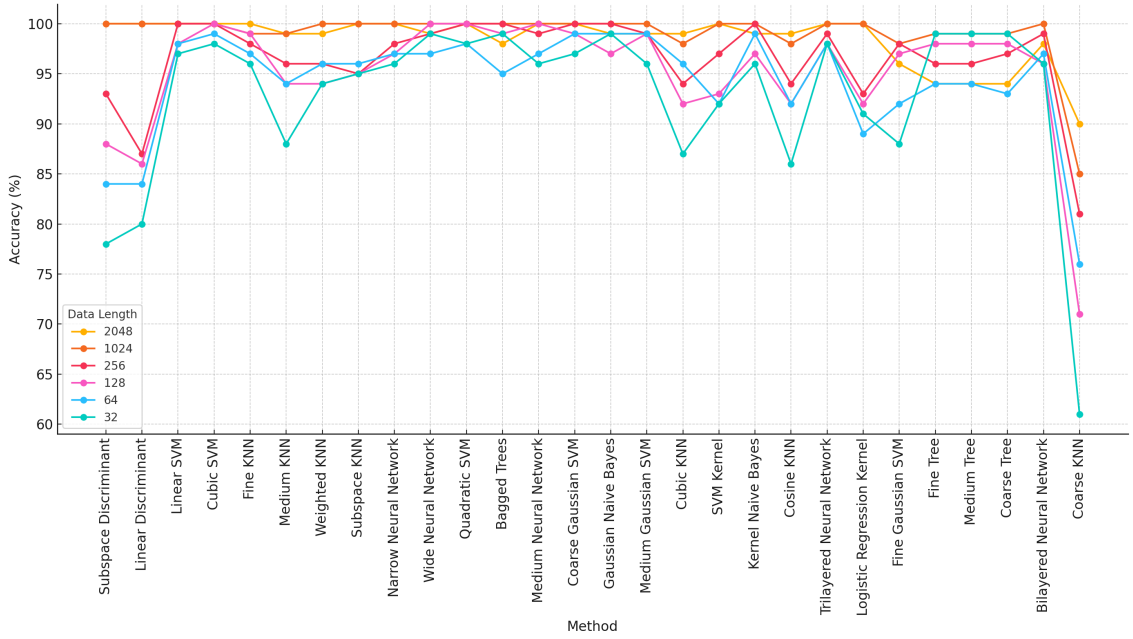


Figure 5.4: Testing accuracy of different lengths in MATLAB Classification Learner.

In Figure 5.4 and Table **??**, it can be observed that even with a length of 32, many classification methods (especially SVM and neural network models) still maintain high accuracy. For example, several SVM variants and neural networks achieve accuracies close to 96% or higher. This indicates that even with shorter data, the models are able to effectively capture key features and maintain strong accuracy levels.

Hence, choosing a length of 32 for the experiment allows for faster computation without sacrificing accuracy, making it ideal for efficient testing and resource-constrained environments. Additionally, the results of the figure show that a length of 32 already meets the high accuracy requirements for the experiment, making it an optimal choice for balancing speed and precision.

Table 5.1: Accuracy of different methods under varying segment lengths

Method	**Number of Data Points**					
	2048	**1024**	**256**	**128**	**64**	**32**
Subspace Discriminant	100%	100%	93%	88%	84%	72%
Linear Discriminant	100%	100%	87%	86%	84%	71%
Linear SVM	100%	100%	100%	98%	98%	77%
Cubic SVM	100%	100%	100%	100%	99%	75%
Fine KNN	100%	99%	98%	98%	97%	77%
Medium KNN	99%	99%	96%	94%	94%	69%
Weighted KNN	99%	99%	96%	94%	94%	71%
Subspace KNN	100%	100%	98%	97%	96%	76%
Narrow Neural Network	100%	100%	99%	97%	97%	72%
Wide Neural Network	99%	100%	100%	100%	97%	73%
Quadratic SVM	100%	100%	100%	100%	98%	80%
Bagged Trees	98%	100%	100%	99%	95%	80%
Medium Neural Network	100%	100%	99%	100%	97%	75%
Coarse Gaussian SVM	100%	100%	100%	99%	99%	83%
Gaussian Naïve Bayes	99%	100%	100%	97%	97%	82%
Medium Gaussian SVM	99%	100%	99%	99%	99%	76%
Cubic KNN	99%	98%	98%	92%	96%	68%
SVM Kernel	100%	100%	97%	97%	96%	75%
Kernel Naïve Bayes	99%	100%	100%	98%	94%	84%
Cosine KNN	99%	98%	94%	92%	91%	73%
Trilayered Neural Network	100%	100%	100%	98%	96%	75%
Logistic Regression Kernel	100%	100%	100%	97%	97%	73%
Fine Gaussian SVM	96%	98%	95%	97%	92%	66%
Fine Tree	94%	95%	98%	96%	90%	80%
Medium Tree	94%	94%	95%	94%	92%	82%
Coarse Tree	94%	94%	98%	94%	89%	78%
Bilayered Neural Network	98%	100%	95%	96%	97%	67%
Coarse KNN	90%	85%	81%	71%	76%	52%

Experiment 2

In this experiment, I tested the performance of each method with different noisy vibration data after FFT and sound data after MFCC, four different states: Normal, Outer raceway scoring, Outer race scoring, and Outer raceway pitting. Each state has 100 segments, and the length of each segment was 32 for vibration data.

In Figure 5.5 and 5.6, we observe that as noise levels increase, many methods experience a significant decrease in precision. However, SVM, especially Quadratic SVM and Medium Gaussian SVM, maintain relatively high accuracy even under higher noise levels (e.g., Noise 0.5 and Noise 1.0). The SVM structure is well suited for high-dimensional boundary separation, making it effective in noisy environments, particularly for non-linear classification problems.

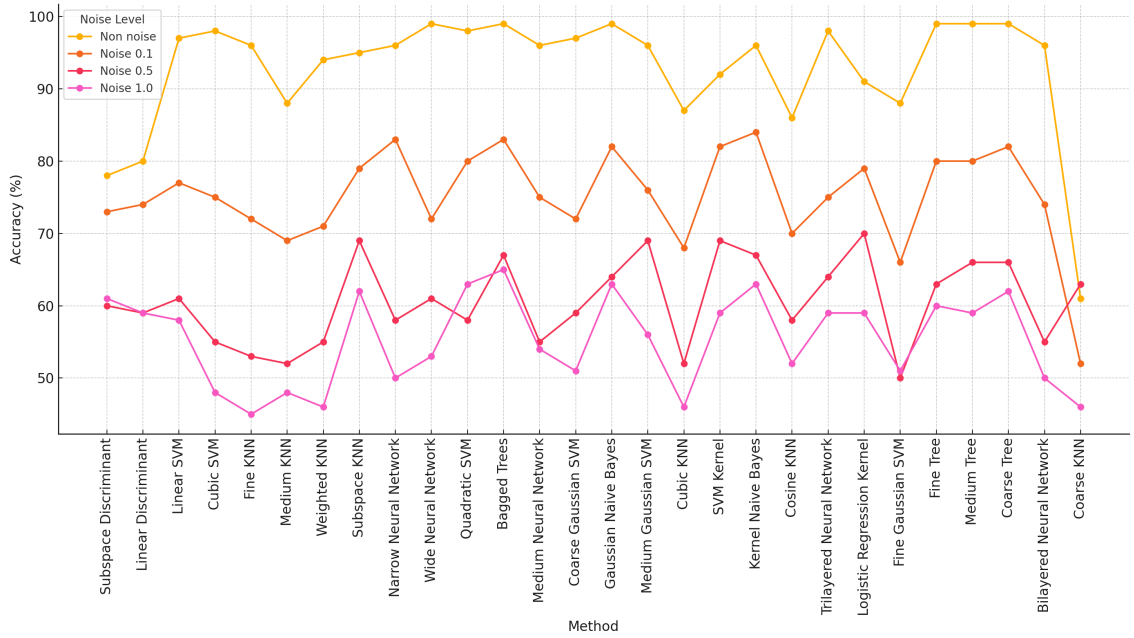


Figure 5.5: Testing accuracy of different noise levels for vibration data with 32 lengths in MATLAB Classification Learner.

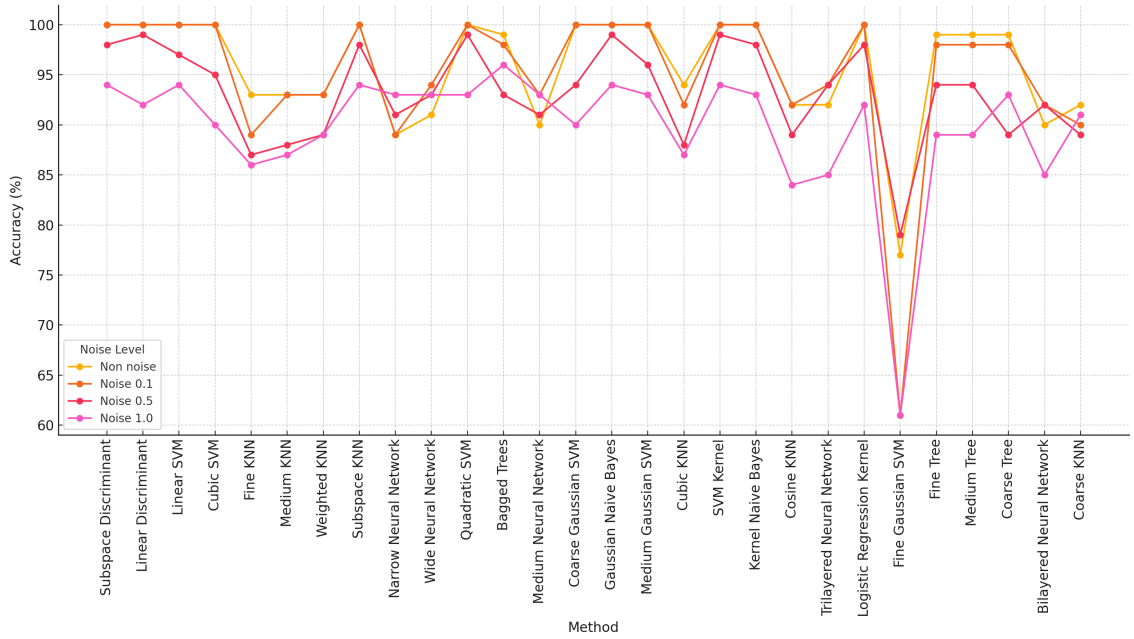


Figure 5.6: Testing accuracy of different noise levels for sound data in MATLAB Classification Learner.

Neural networks (such as narrow neural networks and wide neural networks) also show resilience, maintaining reasonable accuracy under various noise levels. The nonlinear architecture of neural networks allows them to capture complex patterns and extract deeper features from noisy data.

Both SVM and neural networks demonstrate strong performance on both vibration and sound data types. Sound data, in particular, often contains rich frequency-based features, which SVM can classify effectively by finding optimal decision boundaries in high-dimensional spaces, while neural networks can capture complex and hierarchical features through their layered structures. Therefore, choosing SVM and neural networks ensures that both vibration and sound data can be analyzed effectively, maximizing the versatility of the classification models.

On noise-free data, SVM and neural networks achieve very high accuracy (often above 90%). While there is some drop in accuracy as noise increases, SVM and neural networks generally experience a more controlled decrease compared to other methods, indicating that they provide relatively stable performance in noisy environments. neural networks (e.g., trilayered and wide neural networks) particularly benefit from their deeper network structures, which can learn features that are more robust to noise in higher-noise conditions.

Both SVM and neural networks are highly configurable algorithms, allowing for further parameter tuning to optimize performance for specific tasks. For instance, SVM kernel

parameters can be adjusted to enhance classification capabilities, while the layers, learning rates, and regularization parameters in neural networks can be fine-tuned to boost accuracy. Choosing SVM and neural networks also provides flexibility for more advanced modelling and tuning in the future, which can yield substantial performance gains, especially under varying noise levels.

SVM and neural networks are suitable choices due to their stable performance in noisy conditions, ability to handle diverse data types, adaptability to noise levels, and potential for advanced tuning. These characteristics make them well-suited for robust classification across different noise levels and data types in this context.

5.3.2 Experimental setup

Four different states are Normal, Outer raceway scoring (referring to damage on the raceway surface where the rolling elements contact), Outer race scoring (a broader term encompassing damage on any part of the outer race), and Outer raceway pitting (refers to the pitting phenomenon on the raceway surface of the outer ring of a bearing). Each state has 1000 segments, each vibration segment has 16 feature samples after FFT, and each sound segment has 14 feature samples after MFCC, concatenating these features to form a single data segment, hence each fusion segment has 30 feature samples.

The data in each state were evenly partitioned into 4 distinct groups with 250 segments each. The validation was performed using a 4-fold cross-validation approach. The details are shown in Table 5.2.

Table 5.2: Dataset was split into training and testing groups for the 4-fold cross-validation.

	Training Group	**Testing Group**
Validation 1	2, 3, 4	1
Validation 2	1, 3, 4	2
Validation 3	1, 2, 4	3
Validation 4	1, 2, 3	4

5.3.3 Data analysis

As shown in Figure 5.7a, the length of each sample of raw vibration data was 32, and then the Gaussian white noise was added to the vibration data. The reason for adding different levels of noise is that the data collected from our laboratory are too clear in comparison

with the data from real-world applications. After FFT, the length of each segment is 16. In Figure 5.7b, the length of each sample of raw sound data was 768 because the default window length in MATLAB depends on the specified sample rate: round (e.g. the number of frequency sampling $\times$ 0.03) and the frequency sampling is 25,600 samples/second, and then the Gaussian white noise is added to the sound data. After MFCC, the length of each segment is 14.

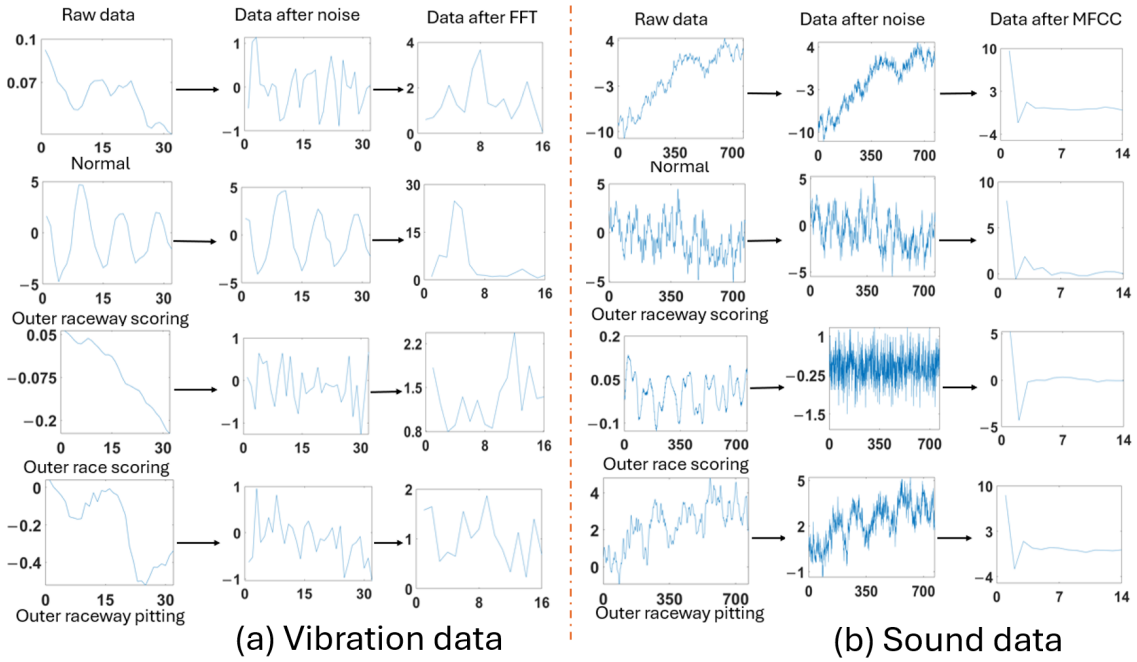


Figure 5.7: Data spectrum. (**a**) Vibration data; (**b**) sound data.

As shown in Figure 5.8, the length of each sample of fusion data was 30, and the difference states at the peak value were obvious.

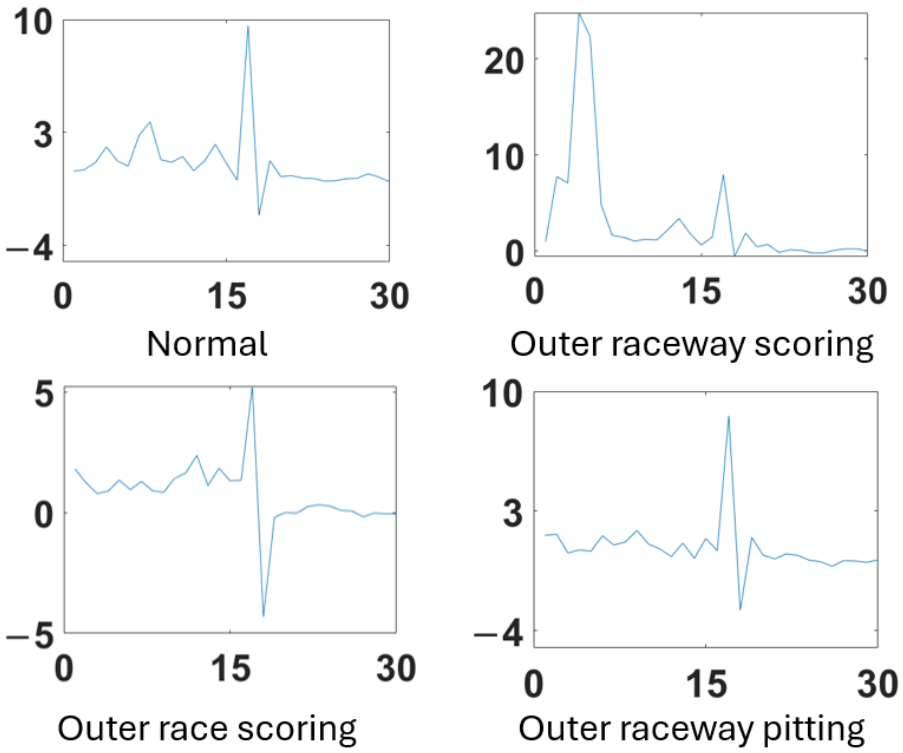


Figure 5.8: Fusion data, is a combination of vibration data and sound data.

5.3.4 Experimental results

Comparison performance

In this part, the fusion data were trained and tested with a few neural network methods and SVM methods [168]. In this section, the fusion data underwent training and testing using several neural networks and SVM algorithms [168]. For neural network methods, the best method was the trilayered neural network, whose accuracy was 97.8%. For SVM methods, the best method was the Quadratic SVM, whose accuracy was 97.9%. The accuracy data are shown in Figure 5.9. Among these 10 methods, the best result was demonstrated by Quadratic SVM. Additionally, the model of SVM is simpler than the neural network. This means SVM can achieve superior efficiency in data detection compared to neural networks. Hence, the suggested approach opts for SVM and enhances its performance by incorporating GWO.

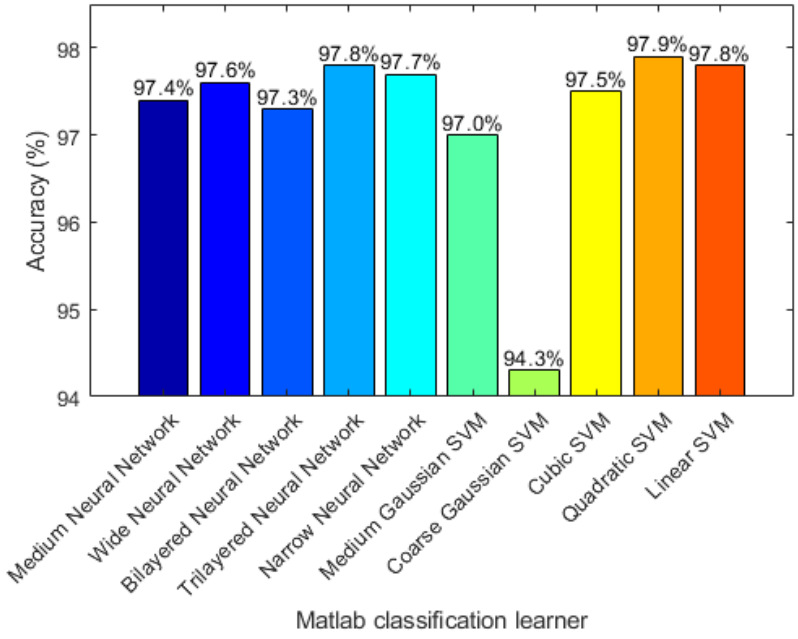


Figure 5.9: Comparison of accuracy of fusion data in Matlab classification learner.

5.4 Analysis

To create a functional model, it is necessary to achieve a high level of accuracy with real-time capability.

5.4.1 Evaluation of different noise level

In this part, five SVM methods in MATLAB are used for different noise level, including noise level 1 and noise level 0.5. The results are shown in Table 5.3.

Table 5.3: Testing accuracy of different noise level.

SVM Method	**Data Types**	**Noise Level**	
		1	**0.5**
Linear	Vibration	59.6%	67.3%
	Sound	94.1%	97.3%
Polynomial(Cubic)	Vibration	57.9%	64.7%
	Sound	91.9%	96.9%
Polynomial(Quadratic)	Vibration	59.5%	67.2%
	Sound	93.4%	97.2%
RBF	Vibration	59.3%	66.8%
	Sound	93.6%	97.6%

As shown in the table, most SVM methods achieve higher testing accuracy at a noise level of 0.5 for both Vibration and Sound data types. For example, Linear SVM reaches 97.325% on Sound data with a noise level of 0.5, compared to only 94.1% at a noise level of 1. A noise level of 1 introduces excessive noise, which reduces the model's accuracy. A noise level of 1 introduces a significant amount of noise, making it difficult for the model to identify meaningful features in the data, thus affecting classification performance. In contrast, a noise level of 0.5 is moderate, allowing the model to better distinguish useful information, particularly for Sound data. With a noise level of 0.5, the model can maintain high accuracy across various SVM methods and data types, indicating more stable performance during training and testing. Compared to a noise level of 1, a level of 0.5 achieves consistency without sacrificing classification quality, thereby enhancing model reliability.

While a noise level of 0.1 adds slight data perturbations to prevent overfitting, the low noise level may not be enough for the model to learn robust features. In contrast, a noise level of 0.5 provides sufficient perturbation to help the model handle real-world noise, leading to more stable performance on diverse data.

In real-world scenarios, data often contains a moderate level of noise. A noise level of 0.5 better approximates these typical noise levels, while a noise level of 0.1 might be too low to represent actual data conditions realistically. This makes a noise level of 0.5 more suitable for training models intended for real-world deployment, enhancing reliability in application.

A noise level of 0.5 offers a balanced level of perturbation that enhances data diversity and distinguishability, preventing overfitting without introducing excessive interference.

Compared to a noise level of 0.1, the level of 0.5 enriches the feature space, improving the model's generalization and adaptability.

In summary, a noise level of 0.5 strikes a balanced trade-off between accuracy, robustness, and data distinguishability, making it a more reasonable choice.

5.4.2 Evaluation of proposed method performance

In this part, the GWO-SVM method initializes the number of wolves as n = 10 and the number of iterations as L = 200; The software executes using a 4-fold cross-validation approach, with each group being repeated 5 times. Following each iteration, the wolves undergo a sorting process based on their fitness values, which allows for identifying the three wolves with the greatest fitness levels. These wolves are referred to as the alpha, beta, and delta wolves. The alpha wolf is prioritized due to its superior fitness. The ultimate trained model exclusively utilizes the parameters of the alpha wolf.

Training loss

For the proposed method, the model output was used to classify tasks (using 'SVM'). The training loss exhibits a quick decline in the initial iterations and thereafter reaches a stable state in the middle stages. This suggests that the process of training the model has achieved a state of convergence after 200 iterations, as shown in Figure 5.10.

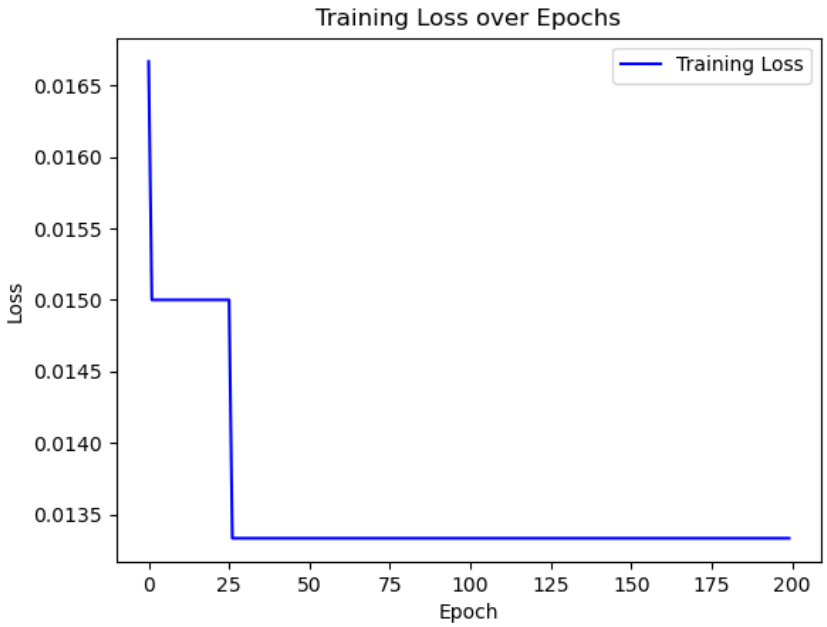


Figure 5.10: The loss curve of changes with training progress in GWO-SVM model.

For both vibration and sound data, as shown in Table 5.4, the best result without GWO was achieved by the RBF SVM, with accuracies of 68% and 97.6%, respectively. When optimized using GWO, the RBF SVM slightly improved to 68.1% for vibration and 97.7% for sound.

Table 5.4: Description of different kernels.

Kernel of SVM	**Training Model**	**Accuracy**		
		Vibration	**Sound**	**Fusion**
Linear	No GWO	67.6%	97.4%	97.8%
	GWO	68%	97.6%	98.3%
Polynomial(Quadratic)	No GWO	67.4%	97.2%	97.9%
	GWO	67.7%	97.3%	98%
Polynomial(Cubic)	No GWO	66.8%	95.5%	97.5%
	GWO	66.8%	97.2%	97.7%
RBF	No GWO	68%	97.6%	97%
	GWO	68.1%	97.7%	98.1%

Regarding the fusion data, a comparison was made between the baseline SVM and the GWO-optimized SVM. As shown in Table 5.4, the best result without GWO was obtained by the Quadratic SVM, achieving an accuracy of 97.9%. With GWO optimization, the Quadratic SVM slightly improved to 98%. Notably, the highest accuracy across all GWO-SVM configurations was achieved by the Linear kernel, reaching 98.3%.

Overall, Table 5.4 demonstrates that the proposed GWO-SVM method consistently outperformed traditional SVM models across vibration, sound, and fusion data.

Fusion of Multimodal Inputs

In this study, both sound and vibration signals are used as input modalities to enhance overall fault detection accuracy. The final prediction is calculated by applying a weighted average to the output probabilities of individual SVM classifiers trained on each modality, along with the output of a stacking ensemble model.

$$P_{\text{combined}} = \frac{w_s \cdot P_{\text{sound}} + w_v \cdot P_{\text{vibration}} + P_{\text{stacking}}}{2} \tag{5.10}$$

where:

- w_s and w_v are the weights assigned to the sound and vibration classifiers, respectively;
- P_{sound}, $P_{\text{vibration}}$, and P_{stacking} represent the predicted class probabilities from each model.

In this implementation, we set $w_s = 0.9$ and $w_v = 0.1$, based on empirical observations that sound-based models consistently outperform vibration-based models . The higher signal-to-noise ratio (SNR) in acoustic signals led to more distinctive fault-related features, particularly in early-stage bearing degradation.

The chosen weights were determined experimentally from validation performance [203], but they may be fine-tuned automatically using methods such as entropy-based weighting, attention mechanisms, or neural-network-based fusion in future work [192, 204].

Statistical Validation and Repeatability Analysis

To evaluate the stability and robustness of the proposed models, all experiments were conducted using 4-fold cross-validation, where each fold represents an independent train–test run with a different random split. The results for both baseline SVM classifiers (without optimization) and GWO-optimized SVMs are reported in Table 5.5.

The results clearly demonstrate that the GWO-optimized classifiers consistently outperform their non-optimized counterparts, both in average accuracy and variance reduction:

Table 5.5: Comparison of Test Accuracy: SVM without vs with GWO (4-fold cross-validation)

	Cubic SVM		RBF SVM		Quadratic SVM		Linear SVM	
	No GWO	GWO	No GWO	GWO	No GWO	GWO	No GWO	GWO
Run1	97.5	98.28	97.8	99.0	97.9	98.6	98.4	98.84
Run2	97.2	97.4	96.9	98.6	97.82	97.82	97.6	98.2
Run3	97.8	98.0	97.1	98.2	97.7	97.7	97.9	98.04
Run4	97.6	97.2	96.3	98.2	97.7	97.7	97.2	98.12
Mean	97.525	97.72	97.025	98.5	97.9	97.955	97.775	98.3
Std. Dev	0.25	0.505	0.618	0.383	0	0.434	0.506	0.366

The RBF SVM improved from 97.025% to 98.5% with GWO optimization, while the Std. Dev dropped from 0.618 to 0.383;

Similarly, the Linear SVM saw accuracy increase from 97.775% to 98.3%, and Std. Dev reduced from 0.506 to 0.366.

These improvements confirm not only higher prediction accuracy but also better repeatability and robustness of the models across different data splits. This reinforces the validity and practical applicability of the proposed GWO-SVM framework for fault detection in real-world settings.

5.4.3 Evaluation of testing speed

This experiment measures the total testing time for all 1000 testing segments and then calculates its average for each segment.

In this part, as shown in Table 5.6, the testing time was tested with different SVM kernels in Python: linear kernel, gaussian kernel (rbf kernel) and quadratic-cubic kernel (polynomial kernel). The frequency sampling was 25,600 samples/second, the sampling time for each segment of vibration data was 1.25 ms and the sampling time for each segment of sound data was 30 ms; thus, the sampling time for each segment of fusion data was 31.25 ms.

Table 5.6: Test results and time of fusion data.

Different Kernel of GWO-SVM	Accuracy	Testing Time for Each Segment	Sampling Time
Linear SVM	98.3%	0.0027 ms	31.25 ms
Quadratic SVM	98%	0.0034 ms	
Cubic SVM	97.7%	0.0035 ms	
Gaussian SVM	98.1%	0.016 ms	

The proposed method demonstrates high efficiency in processing fusion data, enabling rapid analysis and prediction with a remarkable speed of 0.0027 milliseconds per data segment. The experimental results indicate that the model attains a 98.3% accuracy in promptly recognizing wheelset bearing defects. The proposed method exhibited superior performance in terms of both accuracy and execution time when compared to the comparative model. The proposed method exhibited exceptional performance and is in line with the specified requirements.

It should be mentioned here, this excellent experimental performance is based on this single case only, where there are only three fault and normal situations under our limited equipment conditions. However, it will be straightforward to extend our method to any other scenarios where more fault and sensors can be used in real situations. Of course, more evaluations are needed.

5.5 Summary

This work developed a GWO-SVM model for real-time identification of defects in wheelset bearings. The model was developed and trained using the Python programming language. Subsequently, the model was executed on a desktop computer to replicate fast wheelset bearing fault detection for possible real-time implementation. The results of the experiment show that the model achieves accuracy and quickly identifies faults within milliseconds of their occurrence. It achieved an accuracy rate of 98.3% with a testing duration of 0.0027 ms proving its effectiveness and precision in detecting wheelset bearing defects.

This work demonstrates the practical ramifications as well as potential uses of real-time identification of faults in wheelset bearings using a GWO-SVM model. The model demonstrates remarkable precision in rapidly detecting bearing defects and promptly making forecasts, leading to decreased maintenance expenses, enhanced bearing accessibility, and

improved overall effectiveness. By promptly detecting malfunctions during the operation of bearings, this model enables early detection and swift action, thereby reducing downtime and decreasing maintenance expenses.

While this chapter has made notable progress, there remains a substantial amount of effort to enhance the effectiveness and practicality of the suggested approach. Some guidelines for future endeavors include the following: 1. Enhanced data sources: Subsequent studies can explore the inclusion of other sensor data, such as temperature and pressure, to enhance the comprehensiveness and precision of defect identification. 2. Adaptive Algorithm Optimization: In real-world scenarios, researchers investigate adaptive optimization algorithms that allow the model to autonomously modify parameters according to the environment and operating conditions, thereby improving the model's resilience. 3. Long-term Performance Evaluation: Conduct long-term performance evaluations and maintenance cost analyses to verify the economic benefits and sustainability of the proposed method in practical operations. By exploring and improving these future work directions, it is expected that the performance and application of the GWO-SVM method in bearing fault detection can be further enhanced, providing a more reliable and efficient industrial maintenance solution.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

In this dissertation, I addressed the critical issue of bearing fault detection in three industrial contexts: high-speed rail, wind turbines, and railway wheelsets. I proposed and validated three innovative diagnostic methods that leverage frequency-domain signal processing, machine learning, and optimization algorithms to achieve high accuracy and real-time fault detection, contributing to the advancement of intelligent maintenance in industrial equipment.

This thesis presented a progressive investigation into intelligent bearing fault detection, structured in three stages. The first stage (Chapter 3) established a high-accuracy classification method using Bi-LSTM and frequency-domain features, but lacked real-time deployability. To address this, the second stage (Chapter 4) developed a compact monitoring system capable of running on embedded hardware, thus enabling real-time inference. However, this implementation used only acoustic signals, which limited its robustness under varying noise or operational conditions.

Consequently, the final stage (Chapter 5) introduced a multimodal framework that integrates both vibration and acoustic data. This was paired with SVM classifiers enhanced by Grey Wolf Optimization, significantly improving detection performance and adaptability. Together, these stages form an integrated pathway from algorithm development to practical, robust system deployment.

The main contributions of each method are summarized below:

I developed a model combining FFT and Bi-LSTM to transform time-domain signals into frequency-domain data and classify various fault types in high-speed rail bearings. The method achieved a fault classification accuracy of 99.7%, significantly outperforming traditional methods. This approach extends fault detection from the time domain to the frequency domain, proving effective in recognizing complex fault patterns in high-speed rail operations.

To meet the high-frequency diagnostic needs of wind turbine bearings, I designed and implemented an embedded real-time monitoring system on Raspberry Pi using a simplified neural network. This system can process each data segment within 0.06 milliseconds, achieving an accuracy of 99.8%. This efficient, edge-computing system substantially enhances the maintenance efficiency of wind turbines, demonstrating the potential of lightweight, real-time monitoring solutions in complex environments.

To enhance fault detection accuracy in the complex environments of railway wheelsets, I introduced a vibration and sound data fusion approach with GWO-optimized SVM parameters. The integration of multi-modal data improved the model's fault detection accuracy to 98.3%, with each segment processed in 0.0027 milliseconds, showcasing its significant advantages in both real-time capability and precision.

This thesis demonstrates the potential of combining frequency-domain signal processing and machine learning to overcome traditional limitations in data dimensionality, processing efficiency, and real-time response in fault detection. In the future, I plan to further explore multi-modal data fusion in additional industrial applications, as well as implement advanced deep learning and optimization techniques to improve diagnostic accuracy and adaptability. Furthermore, to meet the practical needs of industrial environments, future research will focus on lightweight model design and distributed deployment, enabling efficient, reliable monitoring and preventive maintenance in diverse industrial settings.

In conclusion, the three methods proposed in this dissertation offer novel solutions for intelligent diagnostics in industrial equipment, providing a strong technological foundation for future intelligent maintenance practices. These methods will support enhanced safety, reliability, and efficiency across a wide range of industrial systems.

6.2 Future works

Building on the contributions made in this thesis, several promising directions can be pursued to enhance the effectiveness, versatility, and applicability of intelligent bearing fault detection systems across industrial settings, the following are some future works.

While this thesis demonstrated the benefits of combining vibration and sound data, future work could explore integrating additional sensory data, such as temperature, load, or environmental factors, which could provide more comprehensive fault signatures. Incorporating multi-modal data from diverse sensor sources can potentially improve diagnostic robustness and accuracy, especially in complex and variable operating environments.

Leveraging more sophisticated models, such as transformer-based architectures or hybrid networks combining convolutional layers with LSTMs, could enhance the capture of temporal and spatial dependencies in fault patterns. Future work can also explore transfer learning to adapt pre-trained models to specific industrial settings, minimizing data requirements and improving model generalization across different types of equipment and fault types.

Developing lightweight, energy-efficient models suitable for deployment on edge devices is essential for real-time fault detection in remote or resource-constrained environments. Techniques such as model quantization, pruning, and knowledge distillation could be applied to reduce model complexity without compromising accuracy. This would enable practical, low-latency diagnostics directly on-site, reducing reliance on cloud-based processing and improving response times.

Future research could focus on adaptive monitoring systems that dynamically adjust diagnostic parameters based on real-time feedback from sensor data. By utilizing reinforcement learning or adaptive thresholding techniques, these systems could refine their sensitivity to emerging fault characteristics and optimize maintenance schedules, leading to more proactive and precise maintenance strategies.

Industrial equipment often shares similar fault characteristics across domains (e.g., bearings in rail, automotive, and aerospace applications). Future studies could aim to develop cross-domain fault detection models using domain adaptation techniques, enabling a single model to generalize across multiple applications. This could significantly reduce the time and resources required to deploy intelligent diagnostics in new industrial contexts.

To advance research and standardize comparisons, establishing a comprehensive fault data

repository that includes diverse types of faults and operational conditions would be valuable. This repository could facilitate benchmarking and model comparisons, accelerating the development and validation of new fault detection techniques.

In summary, future work efforts aimed at advancing multi-modal, adaptive, and edge-capable diagnostic systems will be instrumental in realizing the full potential of intelligent maintenance. By addressing these areas, intelligent fault detection systems can become even more responsive, efficient, and versatile, driving safer and more reliable operations across a wide range of industrial sectors.

Bibliography

[1] Jon S Wilson. *Sensor technology handbook*. Elsevier, 2004.

[2] Shaolong Xu, Chunyang Chen, Zhenjun Lin, Xiao Zhang, Jisheng Dai, and Liangjie Liu. Review and prospect of maintenance technology for traction system of high-speed train. *Transportation Safety and Environment*, 3(3):tdab017, 2021.

[3] Weigang Hu, Zhiming Liu, Dekun Liu, and Xue Hai. Fatigue failure analysis of high speed train gearbox housings. *Engineering Failure Analysis*, 73:57–71, 2017.

[4] Qiaoying Ma, Yongqiang Liu, Shaopu Yang, Yingying Liao, and Baosen Wang. A coupling model of high-speed train-axle box bearing and the vibration characteristics of bearing with defects under wheel rail excitation. *Machines*, 10(11):1024, 2022.

[5] Case Western Reserve University. Bearing data. Available online: https://engineering.case.edu/bearingdatacenter/download-data-file. (accessed on 1 October 2021).

[6] Mark Andrejevic and Mark Burdon. Defining the sensor society. *Television & New Media*, 16(1):19–36, 2015.

[7] Malik Tubaishat and Sanjay Madria. Sensor networks: an overview. *IEEE potentials*, 22(2):20–23, 2003.

[8] Wahyu Caesarendra and Tegoeh Tjahjowidodo. A review of feature extraction methods in vibration-based condition monitoring and its application for degradation trend estimation of low-speed slew bearing. *Machines*, 5(4):21, 2017.

[9] John Patrick Kuhn, Binh C Bui, and Gregory J Pieper. Acoustic sensor system for vehicle detection and multi-lane highway monitoring, August 25 1998. US Patent 5,798,983.

[10] John M Loeffler. Vibration sensing and diagnostic system for vehicle drive train components, April 13 1999. US Patent 5,893,892.

[11] Burcu Arman Kuzubasoglu and Senem Kursun Bahadir. Flexible temperature sensors: A review. *Sensors and Actuators A: Physical*, 315:112282, 2020.

[12] Stephen P Radzevich. *Dudley's Handbook of Practical Gear Design and Manufacture*. CRC Press, 2012.

[13] Subhasis Nandi, Hamid A Toliyat, and Xiaodong Li. Condition monitoring and fault diagnosis of electrical motors—a review. *IEEE transactions on energy conversion*, 20(4):719–729, 2005.

[14] Peng Chen, Xiaoqiang Zhao, and HongMei Jiang. A new method of fault feature extraction based on hierarchical dispersion entropy. *Shock and Vibration*, 2021(1):8824901, 2021.

[15] Juan Xu, Pengfei Xu, Zhenchun Wei, Xu Ding, and Lei Shi. Dc-nnmn: Across components fault diagnosis based on deep few-shot learning. *Shock and Vibration*, 2020(1):3152174, 2020.

[16] Lester E Alban. *Systematic analysis of gear failures*. ASM International, 1985.

[17] Heli Liu, Huaiju Liu, Caichao Zhu, and Jinyuan Tang. Study on gear contact fatigue failure competition mechanism considering tooth wear evolution. *Tribology International*, 147:106277, 2020.

[18] Muniyappa Amarnath, Sujatha Chandramohan, and Swarnamani Seetharaman. Experimental investigations of surface wear assessment of spur gear teeth. *Journal of Vibration and Control*, 18(7):1009–1024, 2012.

[19] John T Burwell Jr. Survey of possible wear mechanisms. *Wear*, 1(2):119–141, 1957.

[20] Ke Feng, Qing Ni, Michael Beer, Haiping Du, and Chuan Li. A novel similarity-based status characterization methodology for gear surface wear propagation monitoring. *Tribology International*, 174:107765, 2022.

[21] Ke Feng, Wade A Smith, Pietro Borghesani, Robert B Randall, and Zhongxiao Peng. Use of cyclostationary properties of vibration signals to identify gear wear mechanisms and track wear evolution. *Mechanical Systems and Signal Processing*, 150:107258, 2021.

[22] Ken Mao, W Li, Christopher J Hooke, and Douglas Walton. Friction and wear behaviour of acetal and nylon gears. *wear*, 267(1-4):639–645, 2009.

[23] American Gear Manufacturers Association and American National Standards Institute. *Appearance of Gear Teeth: Terminology of Wear and Failure*. American Gear Manufacturers Association, 1995.

[24] PJL Fernandes and C McDuling. Surface contact fatigue failures in gears. *Engineering Failure Analysis*, 4(2):99–107, 1997.

[25] Yufeng Long, Xianjun Shi, Qiangqiang Chen, Zhicai Xiao, Yufeng Qin, and Jiapeng Lv. Early fault diagnosis technology for bearings based on quantile multiscale permutation entropy. *Mathematical Problems in Engineering*, 2021(1):7718074, 2021.

[26] Fatih Karpat, Ahmet Emir Dirik, Onur Can Kalay, Celalettin Yüce, Oğuz Doğan, and Burak Korcuklu. Fault diagnosis with deep learning for standard and asymmetric involute spur gears. In *ASME International Mechanical Engineering Congress and Exposition*, volume 85697, page V013T14A029. American Society of Mechanical Engineers, 2021.

[27] Shuangwen Sheng. Monitoring of wind turbine gearbox condition through oil and wear debris analysis: A full-scale testing perspective. *Tribology Transactions*, 59(1):149–162, 2016.

[28] Bing Zhang, Andy CC Tan, and Jian-hui Lin. Gearbox fault diagnosis of high-speed railway train. *Engineering Failure Analysis*, 66:407–420, 2016.

[29] Qiaoying Ma, Shaopu Yang, and Yongqiang Liu. Contact analysis between rollers and raceway of axle box bearing under multi-source fault. In *International Conference on Mechanical System Dynamics*, pages 4049–4064, Singapore, 2023. Springer.

[30] Xianwen Zhou, Hao Zhang, Xu Hao, Xin Liao, and Qingkai Han. Investigation on thermal behavior and temperature distribution of bearing inner and outer rings. *Tribology International*, 130:289–298, 2019.

[31] Jason R Stack, Thomas G Habetler, and Ronald G Harley. Fault-signature modeling and detection of inner-race bearing faults. *IEEE transactions on Industry Applications*, 42(1):61–68, 2006.

[32] Dejie Yu, Junsheng Cheng, and Yu Yang. Application of emd method and hilbert spectrum to the fault diagnosis of roller bearings. *Mechanical systems and signal processing*, 19(2):259–270, 2005.

[33] XN Zhao, GX Chen, ZY Huang, and CG Xia. Study on the different effects of power and trailer wheelsets on wheel polygonal wear. *Shock and Vibration*, 2020(1):2587152, 2020.

[34] Yifan Li, Xihui Liang, Yuejian Chen, Zaigang Chen, and Jianhui Lin. Wheelset bearing fault detection using morphological signal and image analysis. *Structural Control and Health Monitoring*, 27(11):e2619, 2020.

[35] Ke Yan, Ning Wang, Qiang Zhai, Yongsheng Zhu, Jinhua Zhang, and Qingbo Niu. Theoretical and experimental investigation on the thermal characteristics of double-row tapered roller bearings of high speed locomotive. *International Journal of Heat and Mass Transfer*, 84:1119–1130, 2015.

[36] Andoni Zagouris, Arturo A Fuentes, Constantine M Tarawneh, Javier A Kypuros, and Andrea Arguelles. Experimentally validated finite element analysis of railroad bearing adapter operating temperatures. In *ASME International Mechanical Engineering Congress and Exposition*, volume Volume 7: Fluids and Heat Transfer, Parts A, B, C, and D, pages 565–576, Houston, Texas, USA, 2012. American Society of Mechanical Engineers.

[37] James F Manwell, Jon G McGowan, and Anthony L Rogers. *Wind energy explained: theory, design and application*. John Wiley & Sons, 2010.

[38] Tony Burton, Nick Jenkins, David Sharpe, and Ervin Bossanyi. *Wind energy handbook*. John Wiley & Sons, 2011.

[39] Erich Hau and Horst Renouard. *Wind turbines: fundamentals, technologies, application, economics*, volume 2. Springer, 2006.

[40] Thomas Ackermann and Lennart Söder. Wind energy technology and current status: a review. *Renewable and sustainable energy reviews*, 4(4):315–374, 2000.

[41] Begoña Guezuraga, Rudolf Zauner, and Werner Pölz. Life cycle assessment of two different 2 mw class wind turbines. *Renewable Energy*, 37(1):37–44, 2012.

[42] Rahman Saidur, Nasrudin A Rahim, Monirul Rafiqu Islam, and Khalid H Solangi. Environmental impact of wind energy. *Renewable and sustainable energy reviews*, 15(5):2423–2430, 2011.

[43] John K Kaldellis and Dimitris Zafirakis. The wind energy (r) evolution: A short review of a long history. *Renewable energy*, 36(7):1887–1901, 2011.

[44] European Wind Energy Association. *The economics of wind energy*. EWEA, 2009.

[45] Bent Sorensen. *Renewable energy*. Elsevier, 2004.

[46] European Wind Energy Association. *Wind energy-the facts: a guide to the technology, economics and future of wind power*. Routledge, 2012.

[47] Jason Jonkman and Paul Sclavounos. Development of fully coupled aeroelastic and hydrodynamic models for offshore wind turbines. In *44th AIAA aerospace sciences meeting and exhibit*, page 995, Reno, Nevada, 2006.

[48] Walt Musial, Sandy Butterfield, and Brian Mcniff. Improving wind turbine gearbox reliability. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2007.

[49] Henk Polinder, Frank FA Van der Pijl, G-J De Vilder, and Peter J Tavner. Comparison of direct-drive and geared generator concepts for wind turbines. *IEEE Transactions on energy conversion*, 21(3):725–733, 2006.

[50] Ruben Pena, JC Clare, and GM Asher. Doubly fed induction generator using back-to-back pwm converters and its application to variable-speed wind-energy generation. *IEE Proceedings-Electric power applications*, 143(3):231–241, 1996.

[51] Henk Polinder. Overview of and trends in wind turbine generator systems. In *2011 IEEE power and energy society general meeting*, pages 1–8, Detroit, MI, USA, 2011. IEEE.

[52] Fernando D Bianchi, Hernan De Battista, and Ricardo J Mantz. *Wind turbine control systems: principles, modelling and gain scheduling design*. Springer Science & Business Media, 2006.

[53] Paul Fleming, Jennifer Annoni, Jigar J Shah, Linpeng Wang, Shreyas Ananthan, Zhijun Zhang, Kyle Hutchings, Peng Wang, Weiguo Chen, and Lin Chen. Field test of wake steering at an offshore wind farm. *Wind Energy Science*, 2(1):229–239, 2017.

[54] Eduard Muljadi and Charles P Butterfield. Pitch-controlled variable-speed wind turbine generation. *IEEE transactions on Industry Applications*, 37(1):240–246, 2002.

[55] EA Bossanyi. Wind turbine control for load reduction. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 6(3):229–244, 2003.

[56] Stefan Oerlemans, Pieter Sijtsma, and B Méndez López. Location and quantification of noise sources on a wind turbine. *Journal of sound and vibration*, 299(4-5):869–883, 2007.

[57] Kenneth W Van Treuren. Wind turbine noise: Regulations, siting, perceptions and noise reduction technologies. In *Proceedings of Montreal 2018 Global Power and Propulsion Forum 7th–9th May*, Waco, Texas, USA, 2018.

[58] Jianzhong Zhang, Ming Cheng, Zhe Chen, and Xiaofan Fu. Pitch angle control for variable speed wind turbines. In *2008 Third international conference on electric utility deregulation and restructuring and power technologies*, pages 2691–2696, Nanjing, China, 2008. IEEE.

[59] Shuangwen Sheng. Wind turbine gearbox condition monitoring round robin study-vibration analysis. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2012.

[60] Hans Anton Buchholdt and Shodja Edin Moossavi Nejad. *Structural dynamics for engineers*, volume 715. Thomas Telford London, 1997.

[61] Meng Zhang, Bing Liu, Chongqi Gao, Md Nayim Hossain, and Guifeng Zhao. Wind-induced response analysis and fatigue reliability study of a steel–concrete composite wind turbine tower. *Buildings*, 14(6):1740, 2024.

[62] U Ritschel, I Warnke, J Kirchner, and B Meussen. Wind turbines and earthquakes. In *2nd World Wind Energy Conference*, pages 1–8, Norderstedt, Germany, 2003. Citeseer.

[63] Sondipon Adhikari and Subhamoy Bhattacharya. Dynamic analysis of wind turbine towers on flexible foundations. *Shock and vibration*, 19(1):37–56, 2012.

[64] Hao Bai, Younes Aoues, Jean-Marc Cherfils, and Didier Lemosse. Design of an active damping system for vibration control of wind turbine towers. *Infrastructures*, 6(11):162, 2021.

[65] Jutta Stauber, Mark Ott, Murray Snaith, and Matthias Puff. Wind turbine noise attenuation using modal structural damping. In *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, volume 253, pages 4562–4570, Hamburg, Germany, 2016. Institute of Noise Control Engineering.

[66] Peter E Hart, David G Stork, and Richard O Duda. *Pattern classification*. Wiley Hoboken, 2000.

[67] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti A Zadeh. *Feature extraction: foundations and applications*, volume 207. Springer, 2008.

[68] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[69] Robert Bell. *Introductory Fourier transform spectroscopy*. Elsevier, 2012.

[70] Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

[71] James W Cooley, Peter AW Lewis, and Peter D Welch. The fast fourier transform and its applications. *IEEE Transactions on Education*, 12(1):27–34, 1969.

[72] Steven W Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Pub, 1997.

[73] Leon Cohen. *Time-frequency analysis*, volume 778. Prentice Hall PTR New Jersey, 1995.

[74] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366, 1980.

[75] Zrar Kh Abdul and Abdulbasit K Al-Talabani. Mel frequency cepstral coefficient and its applications: A review. *IEEE Access*, 10:122136–122158, 2022.

[76] Zhang Guoliang Zheng, Fang and Zhanjiang Song. Comparison of different implementations of mfcc. *Journal of Computer science and Technology*, 16:582–589, 2001.

[77] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, and Dan Povey. The htk book. *Cambridge university engineering department*, 3(175):12, 2002.

[78] Shalbbya Ali, Safdar Tanweer, Syed Sibtain Khalid, and Naseem Rao. Mel frequency cepstral coefficient: a review. In *Proceedings of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development (ICIDSSD 2020)*, pages 92–101, Jamia Hamdard, New Delhi, India, 2020. ICIDSSD Organizing Committee.

[79] Zhenfa Bi, Le Kong, and Jun Zhang. Research on early fault detecting for wheelset of high-speed train based on metal magnetic memory method. In *Recent Developments in Mechatronics and Intelligent Robotics: Proceedings of International Conference*

on Mechatronics and Intelligent Robotics (ICMIR2018), pages 847–856. Springer, 2019.

[80] Xuesong Jin. Research progress of high-speed wheel–rail relationship. *Lubricants*, 10(10):248, 2022.

[81] Zhonghui Yin, Zhiwei Wang, and Weihua Zhang. Vibration characteristics of high-speed train axle box bearing with raceway defects considering track irregularity. In *International Conference on Electrical and Information Technologies for Rail Transportation*, pages 619–628, Singapore, 2023. Springer.

[82] Norden E Huang, Zheng Shen, SR Long, MC Wu, HH Shih, Quanan Zheng, C Tung, and H Liu. The empirical mode decomposition and hilbert spectrum for nonlinear and nonstationary time series analysis. *Proceedings of the Royal Society A*, 545(1971):903–995, 1998.

[83] Kais Khaldi, MONIA TURKI-HADJ ALOUANE, and Abdel-Ouahab Boudraa. Voiced speech enhancement based on adaptive filtering of selected intrinsic mode functions. *Advances in Adaptive Data Analysis*, 2(01):65–80, 2010.

[84] Jean-Claude Nunes and Eric Deléchelle. Empirical mode decomposition: Applications on signal and image processing. *Advances in Adaptive Data Analysis*, 1(01):125–175, 2009.

[85] Lotfi Saidi, Jaouher Ben Ali, and Farhat Fnaiech. Bi-spectrum based-emd applied to the non-stationary vibration signals for bearing faults diagnosis. *ISA transactions*, 53(5):1650–1660, 2014.

[86] Jaouher Ben Ali, Nader Fnaiech, Lotfi Saidi, Brigitte Chebel-Morello, and Farhat Fnaiech. Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals. *Applied Acoustics*, 89:16–27, 2015.

[87] Jacek Dybała and Radosław Zimroz. Rolling bearing diagnosing method based on empirical mode decomposition of machine vibration signal. *Applied Acoustics*, 77:195–203, 2014.

[88] Cong Wang and Meng Gan. Non-negative emd manifold for feature extraction in machinery fault diagnosis. *Measurement*, 70:188–202, 2015.

[89] Renping Shao, Wentao Hu, Yayun Wang, and Xiankun Qi. The fault feature extraction and classification of gear using principal component analysis and kernel principal component analysis based on the wavelet packet transform. *Measurement*, 54:118–132, 2014.

[90] Paolo Arena, Luca Patanè, and Angelo Giuseppe Spinosa. Data-based analysis of laplacian eigenmaps for manifold reduction in supervised liquid state classifiers. *Information Sciences*, 478:28–39, 2019.

[91] Weihua Li Shaohui Zhang. A local preserving projection algorithm based on feature space noise reduction and its application in bearing fault classification. *Journal of Mechanical Engineering*, 50(3):92–99, 2014.

[92] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[93] Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM journal on scientific computing*, 26(1):313–338, 2004.

[94] Qingbo He. Vibration signal classification by wavelet packet energy flow manifold learning. *Journal of Sound and Vibration*, 332(7):1881–1894, 2013.

[95] Benwei Li and Yun Zhang. Supervised locally linear embedding projection (sllep) for machinery fault diagnosis. *Mechanical Systems and Signal Processing*, 25(8):3125–3134, 2011.

[96] Jacek M Zurada. *Introduction to artificial neural systems*, volume 8. West St. Paul, 1992.

[97] Jun Ueda and Yuichi Kurita. *Human modeling for bio-inspired robotics: mechanical engineering in assistive technologies*. Academic Press, 2016.

[98] Chao Ma, Stephan Wojtowytsch, and Lei Wu. Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't. *arXiv preprint arXiv:2009.10713*, 2020.

[99] Shenhao Wang, Qingyi Wang, Nate Bailey, and Jinhua Zhao. Deep neural networks for choice analysis: A statistical learning theory perspective. *Transportation Research Part B: Methodological*, 148:60–81, 2021.

[100] Yuchen Wu and Junwen Feng. Development and application of artificial neural network. *Wireless Personal Communications*, 102:1645–1656, 2018.

[101] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval*, pages 21–49. Springer, 2008.

[102] Kyunghyun Cho. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[103] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman. 1d convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, 151:107398, 2021.

[104] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[105] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

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

[107] Corinna Cortes. Support-vector networks. *Machine Learning*, 1995.

[108] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[109] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[110] B Schölkopf. Learning with kernels: support vector machines, regularization, optimization, and beyond, 2002.

[111] Lewis Rosado, Nelson H Forster, Kevin L Thompson, and Jason W Cooke. Rolling contact fatigue life and spall propagation of aisi m50, m50nil, and aisi 52100, part i: experimental results. *Tribology Transactions*, 53(1):29–41, 2009.

[112] Chunyan Wu, Jian Liu, Fuqiang Peng, Dejie Yu, and Rong Li. Gearbox fault diagnosis using adaptive zero phase time-varying filter based on multi-scale chirplet sparse signal decomposition. *Chinese Journal of Mechanical Engineering*, 26(4):831–838, 2013.

[113] Xinglong Li, Shengguo Liu, Shuo Cheng, Jindi Lin, Rongchun Liu, Leyu Wang, and Zhilin Zhou. Research on ultrasonic quantitative evaluation technology of complex defects based on neural network. In *Journal of Physics: Conference Series*, volume 2196, page 012022. IOP Publishing, 2022.

[114] Yuan Cao, Yuntong An, Shuai Su, Guo Xie, and Yongkui Sun. A statistical study of railway safety in china and japan 1990–2020. *Accident Analysis & Prevention*, 175:106764, 2022.

[115] Xiaofei He and Partha Niyogi. Locality preserving projections. *Advances in neural information processing systems*, 16, 2003.

[116] Tianhao Zhang, Jie Yang, Deli Zhao, and Xinliang Ge. Linear local tangent space alignment and application to face recognition. *Neurocomputing*, 70(7-9):1547–1553, 2007.

[117] Shao Haidong, Jiang Hongkai, Li Xingqiu, and Wu Shuaipeng. Intelligent fault diagnosis of rolling bearing using deep wavelet auto-encoder with extreme learning machine. *Knowledge-Based Systems*, 140:1–14, 2018.

[118] Haidong Shao, Hongkai Jiang, Ying Lin, and Xingqiu Li. A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders. *Mechanical Systems and Signal Processing*, 102:278–297, 2018.

[119] Jie Tao, Yilun Liu, Dalian Yang, Fang Tang, and Chi Liu. Fault diagnosis of rolling bearing using deep belief networks. In *2015 International Symposium on Material, Energy and Environment Engineering*, pages 566–569. Atlantis Press, 2015.

[120] Meng Gan and Cong Wang. Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings. *Mechanical Systems and Signal Processing*, 72:92–104, 2016.

[121] Zhuang Zilong and Qin Wei. Intelligent fault diagnosis of rolling bearing using one-dimensional multi-scale deep convolutional neural network based health state classification. In *2018 IEEE 15th International conference on networking, sensing and control (ICNSC)*, pages 1–6, Zhuhai, China, 2018. IEEE.

[122] Xiaojie Guo, Liang Chen, and Changqing Shen. Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. *Measurement*, 93:490–502, 2016.

[123] Ruixin Wang, Hongkai Jiang, Xingqiu Li, and Shaowei Liu. A reinforcement neural architecture search method for rolling bearing fault diagnosis. *Measurement*, 154:107417, 2020.

[124] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

[125] Paul Heckbert. Fourier transforms and the fast fourier transform (fft) algorithm. *Computer Graphics*, 2(1995):15–463, 1995.

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

[127] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[128] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. In *International conference on artificial neural networks*, pages 799–804, Berlin, Heidelberg, 2005. Springer.

[129] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

[130] Davide Chicco. Siamese neural networks: An overview. *Artificial neural networks*, pages 73–94, 2021.

[131] International energy agency. World energy outlook 2021. Available online: https://www.iea.org/reports/world-energy-outlook-2021. (accessed on 1 March 2024).

[132] Junyan Ma, Yiping Yuan, and Pan Chen. Spatiotemporal attention-based long short-term memory auto-encoder network for fault detection of wind turbine generators. Available online: https://www.researchsquare.com/article/rs-2206291/v1. (accessed on 3 April 2024).

[133] Yi-Hsiang Liao, Lijuan Wang, and Yong Yan. Instantaneous rotational speed measurement of wind turbine blades using a marker-tracking method. In *2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–5, Ottawa, ON, Canada, 2022. IEEE.

[134] Meng-Hui Wang, Cheng-Che Hsieh, and Shiue-Der Lu. Fault diagnosis of wind turbine blades based on chaotic system and extension neural network. *Sensors & Materials*, 33, 2021.

[135] Zhiwei Gao and Xiaoxu Liu. An overview on fault diagnosis, prognosis and resilient control for wind turbine systems. *Processes*, 9(2):300, 2021.

[136] Wai-Xi Liu, Rui-Peng Yin, and Ping-Yu Zhu. Deep learning approach for sensor data prediction and sensor fault diagnosis in wind turbine blade. *IEEE Access*, 10:117225–117234, 2022.

[137] Francesc Pozo, Yolanda Vidal, and Josep M Serrahima. On real-time fault detection in wind turbines: Sensor selection algorithm and detection time reduction analysis. *Energies*, 9(7):520, 2016.

[138] Fan Zhang, Wenlei Sun, Hongwei Wang, and Tiantian Xu. Fault diagnosis of a wind turbine gearbox based on improved variational mode algorithm and information entropy. *Entropy*, 23(7):794, 2021.

[139] Magda Ruiz, Luis E Mujica, Santiago Alférez, Leonardo Acho, Christian Tutivén, Yolanda Vidal, José Rodellar, and Francesc Pozo. Wind turbine fault detection and classification by means of image texture analysis. *Mechanical Systems and Signal Processing*, 107:149–167, 2018.

[140] Mingzhu Tang, Yutao Chen, Huawei Wu, Qi Zhao, Wen Long, Victor S Sheng, and Jiabiao Yi. Cost-sensitive extremely randomized trees algorithm for online fault detection of wind turbine generators. *Frontiers in Energy Research*, 9:686616, 2021.

[141] Han Peng, Hai Zhang, Yisa Fan, Linjian Shangguan, and Yang Yang. A review of research on wind turbine bearings' failure analysis and fault diagnosis. *Lubricants*, 11(1):14, 2022.

[142] He Tian, Huaicong Fan, Mingwen Feng, Ranran Cao, and Dong Li. Fault diagnosis of rolling bearing based on hpso algorithm optimized cnn-lstm neural network. *Sensors*, 23(14):6508, 2023.

[143] Amin Khorram, Mohammad Khalooei, and Mansoor Rezghi. End-to-end cnn+ lstm deep learning approach for bearing fault diagnosis. *Applied Intelligence*, 51(2):736–751, 2021.

[144] Fanghong Zhang, Yuze Zhu, Chuanjiang Zhang, Peng Yu, and Qingan Li. Abnormality detection method for wind turbine bearings based on cnn-lstm. *Energies*, 16(7):3291, 2023.

[145] Zifei Xu, Chun Li, and Yang Yang. Fault diagnosis of rolling bearing of wind turbines based on the variational mode decomposition and deep convolutional neural networks. *Applied Soft Computing*, 95:106515, 2020.

[146] Zifei Xu, Xuan Mei, Xinyu Wang, Minnan Yue, Jiangtao Jin, Yang Yang, and Chun Li. Fault diagnosis of wind turbine bearing using a multi-scale convolutional neural network with bidirectional long short term memory and weighted majority voting for multi-sensors. *Renewable Energy*, 182:615–626, 2022.

[147] Lixiao Cao, Zheng Qian, Hamidreza Zareipour, Zhenkai Huang, and Fanghong Zhang. Fault diagnosis of wind turbine gearbox based on deep bi-directional long short-term memory under time-varying non-stationary operating conditions. *IEEE Access*, 7:155219–155228, 2019.

[148] Prince Waqas Khan, Chan Yeob Yeun, and Yung Cheol Byun. Fault detection of wind turbines using scada data and genetic algorithm-based ensemble learning. *Engineering Failure Analysis*, 148:107209, 2023.

[149] Dezun Zhao, Tianyang Wang, and Fulei Chu. Deep convolutional neural network based planet bearing fault classification. *Computers in Industry*, 107:59–66, 2019.

[150] Zixu Wang, Jianhong Zhu, Juping Gu, Junjie Hu, Bojun Zhou, and Jiahao Zhao. Fault diagnosis of wind turbine bearing on satlbo-mlp. In *2022 China Automation Congress (CAC)*, pages 6622–6627, Xiamen, China, 2022. IEEE.

[151] Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo, and Junshan Zhang. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8):1738–1762, 2019.

[152] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[153] Ian Goodfellow. Deep learning, 2016.

[154] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[155] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[156] Jian Fu, Jingchun Chu, Peng Guo, and Zhenyu Chen. Condition monitoring of wind turbine gearbox bearing based on deep learning model. *IEEE Access*, 7:57078–57087, 2019.

[157] Henrik Asmuth and Henry Korb. Wakenet 0.1-a simple three-dimensional wake model based on convolutional neural networks. In *Journal of Physics: Conference Series*, volume 2265, page 022066. IOP Publishing, 2022.

[158] Yuan Xie, Jisheng Zhao, Baohua Qiang, Luzhong Mi, Chenghua Tang, and Longge Li. Attention mechanism-based cnn-lstm model for wind turbine fault prediction using ssn ontology annotation. *Wireless Communications and Mobile Computing*, 2021(1):6627588, 2021.

[159] Rudolf Kruse, Sanaz Mostaghim, Christian Borgelt, Christian Braune, and Matthias Steinbrecher. Multi-layer perceptrons. In *Computational intelligence: a methodological introduction*, pages 53–124. Springer, 2022.

[160] Juncai He, Lin Li, Jinchao Xu, and Chunyue Zheng. Relu deep neural networks and linear finite elements. *arXiv preprint arXiv:1807.03973*, 2018.

[161] Min-Hang Bao. *Micro mechanical transducers: pressure sensors, accelerometers and gyroscopes*. Elsevier, 2000.

[162] Krassimir Hristov Denishev and Mihaela Rangelova Petrova. Accelerometer design. *Proceedings of ELECTRONICS*, 2007:159–164, 2007.

[163] John Vetelino and Aravind Reghu. *Introduction to sensors*. CRC press, Piscataway, NJ, USA, 2017.

[164] John Park and Steve Mackay. *Practical data acquisition for instrumentation and control systems*. Newnes, Oxford, UK, 2003.

[165] Maurizio Di Paolo Emilio. *Data Acquisition Systems*. Springer, Cham, Switzerland, 2013.

[166] Charles Severance. Eben upton: Raspberry pi. *Computer*, 46(10):14–16, 2013.

[167] Ashwin Pajankar. *Raspberry Pi Image Processing Programming: With NumPy, SciPy, Matplotlib, and OpenCV*. Apress, New York, NY, USA, 2022.

[168] M Senthamil Selvi and S Jansi Rani. Classification of admission data using classification learner toolbox. In *Journal of Physics: Conference Series*, volume 1979, page 012043. IOP Publishing, 2021.

[169] Om Prakash Yadav and GL Pahuja. Bearing health assessment using time domain analysis of vibration signal. *International Journal of Image, Graphics and Signal Processing*, 10(3):27, 2020.

[170] NCD.IO. Bearing fault detection vibration analysis-how to measure vibration frequency domain predictive analysis. Available online: https://ncd.io/blog/bearing-fault-detection-vibration-analysis/. (accessed on 23 June 2023).

[171] Muhammad Altaf, Tallha Akram, Muhammad Attique Khan, Muhammad Iqbal, M Munawwar Iqbal Ch, and Ching-Hsien Hsu. A new statistical features based approach for bearing fault diagnosis using vibration signals. *Sensors*, 22(5):2012, 2022.

[172] Jay Prakash Kumar, Premanand S Chauhan, and Prem Prakash Pandit. Time domain vibration analysis techniques for condition monitoring of rolling element bearing: A review. *Materials Today: Proceedings*, 62:6336–6340, 2022.

[173] Tianhao Wang, Rui Qin, Hongying Meng, MaoZhen Li, Meng Cheng, and Yi Liu. Frequency domain feature extraction and long short-term memory for rolling bearing fault diagnosis. In *2022 International Conference on Machine Learning, Control, and Robotics (MLCR)*, pages 72–77, Suzhou, China, 2022. IEEE.

[174] Bingyan Chen, Fengshou Gu, Weihua Zhang, Dongli Song, Yao Cheng, and Zewen Zhou. Power function-based gini indices: New sparsity measures using power function-based quasi-arithmetic means for bearing condition monitoring. *Structural Health Monitoring*, 22(6):3677–3706, 2023.

[175] Ping Huang, Ziwei Pan, Xiaoli Qi, and Jiapeng Lei. Bearing fault diagnosis based on emd and psd. In *2010 8th World Congress on Intelligent Control and Automation*, pages 1300–1304, Jinan, China, 2010. IEEE.

[176] Bingyan Chen, Weihua Zhang, Dongli Song, Yao Cheng, Fengshou Gu, and Andrew D Ball. Squared envelope sparsification via blind deconvolution and its application to railway axle bearing diagnostics. *Structural Health Monitoring*, 22(6):3637–3658, 2023.

[177] Tatsuro Baba. Time-frequency analysis using short time fourier transform. *The Open Acoustics Journal*, 5:32–38, 2012.

[178] Dengsheng Zhang. Wavelet transform. *Fundamentals of image data mining: Analysis, Features, Classification and Retrieval*, pages 35–44, 2019.

[179] Gang Yu. A concentrated time–frequency analysis tool for bearing fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 69(2):371–381, 2019.

[180] Qitao Zhang, Jiapeng Yang, and Qi An. Noise calculation method for deep groove ball bearing with considering raceway surface waviness and roller size error. *Frontiers in Mechanical Engineering*, 4:13, 2018.

[181] BearingNews. Diagnosing bearing failures using ultrasound spectrum analysis. Available online: https://www.bearing-news.com/diagnosing-bearing-failures-using-ultrasound-spectrum-analysis/. (accessed on 26 April 2022).

[182] Norden E Huang, Zheng Shen, Steven R Long, Manli C Wu, Hsing H Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H Liu. 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, 1998.

[183] Niranjan Hiremath and D Mallikarjuna Reddy. Bearing fault detection using acoustic emission signals analyzed by empirical mode decomposition. *International Journal of Research in Engineering and Technology*, 3:426–431, 2014.

[184] Jarek Grebenik, Yu Zhang, Chris Bingham, and Saket Srivastava. Roller element bearing acoustic fault detection using smartphone and consumer microphones comparing with vibration techniques. In *2016 17th International Conference on Mechatronics-Mechatronika (ME)*, pages 1–7, Prague, Czech Republic, 2016. IEEE.

[185] Chi Li, Changzheng Chen, and Xiaojiao Gu. Acoustic-based rolling bearing fault diagnosis using a co-prime circular microphone array. *Sensors*, 23(6):3050, 2023.

[186] Maciej Orman, Pawel Rzeszucinski, Agnieszka Tkaczyk, Karthik Krishnamoorthi, Cajetan T Pinto, and Maciej Sulowicz. Bearing fault detection with the use of acoustic signals recorded by a hand-held mobile phone. In *2015 International Conference on Condition Assessment Techniques in Electrical Systems (CATCON)*, pages 252–256, Bangalore, India, 2015. IEEE.

[187] Emmanuel Attal, Elga Ossonemane Asseko, Edgard Sekko, Nadia Sbai, and Philippe Ravier. Study of bearing fault detectability on a rotating machine by vibro-acoustic characterisation as a function of a noisy surrounding machine. In *Surveillance, Vibrations, Shock and Noise*, Toulouse, France, 2023.

[188] Oliver Mey, André Schneider, Olaf Enge-Rosenblatt, Dirk Mayer, Christian Schmidt, Samuel Klein, and Hans-Georg Herrmann. Condition monitoring of drive trains by data fusion of acoustic emission and vibration sensors. *Processes*, 9(7):1108, 2021.

[189] Yin Kaicheng and Zhong Chaosheng. Vibration data fusion algorithm of auxiliaries in power plants based on wireless sensor networks. In *2011 International Conference on Computer Science and Service System (CSSS)*, pages 935–938, Nanjing, China, 2011. IEEE.

[190] Li Xuejun, Bin Guangfu, and Balbir S Dhillon. A new method of multi-sensor vibration signals data fusion based on correlation function. In *2009 WRI World Congress on Computer Science and Information Engineering*, volume 6, pages 170–174, Los Angeles, CA, USA, 2009. IEEE.

[191] Luttfi A Al-Haddad, Alaa Abdulhady Jaber, Mohsin N Hamzah, and Mohammed A Fayad. Vibration-current data fusion and gradient boosting classifier for enhanced stator fault diagnosis in three-phase permanent magnet synchronous motors. *Electrical Engineering*, 106(3):3253–3268, 2024.

[192] Haibo Wan, Xiwen Gu, Shixi Yang, and Yanni Fu. A sound and vibration fusion method for fault diagnosis of rolling bearings under speed-varying conditions. *Sensors*, 23(6):3130, 2023.

[193] Huaitao Shi, Yangyang Li, Xiaotian Bai, Ke Zhang, and Xianming Sun. A two-stage sound-vibration signal fusion method for weak fault detection in rolling bearing systems. *Mechanical Systems and Signal Processing*, 172:109012, 2022.

[194] Zhihe Duan, Tonghai Wu, Shuaiwei Guo, Tao Shao, Reza Malekian, and Zhixiong Li. Development and trend of condition monitoring and fault diagnosis of multi-sensors information fusion for rolling bearings: a review. *The International Journal of Advanced Manufacturing Technology*, 96:803–819, 2018.

[195] Xin Wang, Dongxing Mao, and Xiaodong Li. Bearing fault diagnosis based on vibro-acoustic data fusion and 1d-cnn network. *Measurement*, 173:108518, 2021.

[196] Xiaojiao Gu, Yang Tian, Chi Li, Yonghe Wei, and Dashuai Li. Improved se-resnet acoustic–vibration fusion for rolling bearing composite fault diagnosis. *Applied Sciences*, 14(5):2182, 2024.

[197] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in engineering software*, 69:46–61, 2014.

[198] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

[199] Xucun Yan, Zihuai Lin, Zhiyun Lin, and Branka Vucetic. A novel exploitative and explorative gwo-svm algorithm for smart emotion recognition. *IEEE Internet of Things Journal*, 10(11):9999–10011, 2023.

[200] Mushabi Pule, Oduetse Matsebe, and Ravi Samikannu. Application of pca and svm in fault detection and diagnosis of bearings with varying speed. *Mathematical Problems in Engineering*, 2022(1):5266054, 2022.

[201] Kaisi Yang, Lianyu Zhao, and Chenglin Wang. A new intelligent bearing fault diagnosis model based on triplet network and svm. *Scientific Reports*, 12(1):5234, 2022.

[202] Changchun Mo, Huizi Han, Mei Liu, Qinghua Zhang, Tao Yang, and Fei Zhang. Research on svm-based bearing fault diagnosis modeling and multiple swarm genetic algorithm parameter identification method. *Mathematics*, 11(13):2864, 2023.

[203] Ahmed Chennana, Ahmed Chaouki Megherbi, Noureddine Bessous, Salim Sbaa, Ali Teta, El Ouanas Belabbaci, Abdelaziz Rabehi, Mawloud Guermoui, and Takele Ferede Agajie. Vibration signal analysis for rolling bearings faults diagnosis based on deep-shallow features fusion. *Scientific Reports*, 15(1):9270, 2025.

[204] You Keshun, Lian Zengwei, and Gu Yingkui. A performance-interpretable intelligent fusion of sound and vibration signals for bearing fault diagnosis via dynamic came. *Nonlinear Dynamics*, 112(23):20903–20940, 2024.

Appendix

A.1 Published work 1: Real-Time Monitoring of Wind Turbine Bearing Using Simple Neural Network on Raspberry Pi

Frequency Domain Feature Extraction and Long Short-Term Memory for Rolling Bearing Fault Diagnosis

Tianhao Wang
Department of Electronic and Electrical Engineering
Brunel University London
London, UK
1815203@brunel.ac.uk

Rui Qin
Department of Computing and Mathematics
Manchester Metropolitan University
Manchester, UK
r.qin@mmu.ac.uk

Hongying Meng
Department of Electronic and Electrical Engineering
Brunel University London
London, UK
hongying.meng@brunel.ac.uk

MaoZhen Li
Department of Electronic and Electrical Engineering
Brunel University London
London, UK
Maozhen.Li@brunel.ac.uk

Meng Cheng
Department of Electronic and Electrical Engineering
Brunel University London
London, UK
2104413@brunel.ac.uk

Yi Liu
Department of New Media
Beijing Institute of Graphic Communication
Beijing, China
yiliu61easy@163.com

2022 International Conference on Machine Learning, Control, and Robotics (MLCR) | 978-1-6654-5459-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/MLCR57210.2022.00022

***Abstract*—With the rapid development of the high-speed railways, the speed of trains is getting faster and faster, and the dynamic load between the wheels and rails of the vehicle increases accordingly. The rolling bearing is a key part of the high-speed train transmission system. The train is subjected to high-frequency vibration for a long time during operation, and the bearing is prone to fatigue damage, which affects the safe operation of the train. Nowadays, many methods have been applied in fault diagnosis like reinforcement learning, convolutional neural networks and autoencoders. One of the typical methods is the reinforcement neural architecture research method. It makes neural network design automatic and eliminates the bottleneck associated with choosing network architectural parameters. However, this method focuses on the time domain signal, and a time domain signal cannot capture the particular properties of a frequency domain signal. In order to solve these problems, we propose a new method containing two Steps: Use FFT to convert the time domain signal to the frequency domain and use Bi-LSTM neural network model to recognize different faults. For each fault, the time series signal has some correlation with some specific frequencies. The frequency domain is more intuitive than the time domain and describes different states of faulty types. For recognition, LSTM is better at classifying sequence data than other methods, and Bi-LSTM can predict the sequence from both directions, achieving higher accuracy. Experiments on public data sets demonstrate the efficiency of the proposed method.**

***Index Terms*—Bi-LSTM, FFT, Bearing fault**

I. Introduction

A few essential components ensure that the train operates safely and efficiently, and the development of high-speed rail has considerably improved traffic conditions in many countries. The transmission system is one of them and is crucial to the efficient running of high-speed trains. In the majority of high-speed train transmission systems, the traction motor serves as the power source and is a critical component. Therefore, it is important to recognise traction motor breakdowns. The most frequent type of motor failure is motor bearing failure [1]. For instance, according to the research, 40% of motor failures are caused by motor bearing failures. Fatigue spalling, which includes partial damage or fall-off on the inner ring, outer ring, rolling element, and other bearing surfaces, is one of the most common defects in motor bearings. Fatigue stress from alternating loads is the main factor in bearing fatigue spalling [2]. When a bearing experiences a fatigue spalling failure, a particular frequency of shock pulse will manifest. As a result, rolling bearing maintenance is quite expensive and very important for every country. For instance, the US spends hundreds of billions of dollars every year on maintaining machinery and routinely replacing vital components [3].

However, if the crucial components of the equipment are not updated in a timely manner, catastrophic tragedies could happen. For instance, on June 3, 1998, a high-speed train in Germany's elastic wheel burst due to prolonged use, resulting in 101 fatalities and 194 serious injuries [4]. 72 people lost their lives and 416 were hurt when 9 carriages of the NO.T195 train from Beijing to Qingdao derailed and crashed with the

978-1-6654-5459-9/22/$31.00 ©2022 IEEE
DOI 10.1109/MLCR57210.2022.00022

NO.5034 train from Yantai to Xuzhou on April 28, 2008, in China. [5].

Consequently, rolling bearing detection and recognition for health monitoring has emerged as one of the key research fields in order to lower the cost of rolling bearing maintenance and maintain the safety of operation for high-speed trains. In recent years, some laboratories used accelerometers located at the driving end of the motor housing to adopt a significant amount of vibration data. The sampling frequency is usually 12000 samples per second, 25600 samples per second and 48000 samples per second. These vibration data lengths are generally more than 120000, and many subcategories of vibration signals exist.

Hence, signal processing and status recognition have taken centre stage in rolling bearings research. For signal processing, Huang et al. [6] proposed a method of adaptively decomposing non-stationary signals into a series of zero-mean intrinsic modal functions (IMF), which was called empirical mode decomposition (EMD). Saidi et al. [7] used EMD to dissect the non-stationary signal into several IMFs according to the local characteristic time scale of the signal. Ali et al. [8] used the Intrinsic Mode Function (IMF) energy bribe generated by empirical mode decomposition to describe seven different bearing states. Dybała and Zimroz [9] proposed an early damage detection method for rolling bearings based on EMD. Wang et al. [10] proposed a new non-negative EMD manifold (NEM) bearing failure feature extraction method. Popular learning has been a more popular dimensionality reduction method in recent years. It has been used in a wide variety of fields of fault diagnosis. Among them, Arena et al. [11] proposed Laplacian Eigenmaps (LE), He and Partha [12] proposed Locality Preservation Projection (LPP), Roweis and Saul [13] proposed Locally Linear (LLE), Zhang et al. [14] proposed linear local tangent space alignment (LLTSA). He [15] used LLE to extract the popular features of wavelet packet energy and effectively distinguished bearing and gear failures with different failure degrees. Li and Zhang [16] used the supervised LLE algorithm to map the features from the high-dimensional space to the embedding space and performed bearing fault classification in the embedding space.

More importantly, status recognition achieves much success as well. For instance, Shao et al. [17] proposed a Deep wavelet auto-encoder (DWAE) with an extreme learning machine (ELM). They used the wavelet function to design a wavelet autoencoder, to get data features and improve the ability to study unsupervised features. ELM is a classifier. The result is 95.2%. Shao et al. [18] proposed ensemble deep auto-encoders (EDAEs). Use the Unsupervised feature learning from the raw vibration data and design a strategy to ensure accuracy and stability. The result is 97.18%. Tao et al. [19] proposed deep belief networks (DBN). DBN can reduce energy loss between the output and input vibration signals. The result is 96.67%. Gan and Wang [20] proposed hierarchical diagnosis network (HDN) can achieve 99.03%. Zhuang and Qin [21] proposed a multi-scale deep CNN (MS-DCNN) model that can reach 99.27%. Guo et al. [22] constructed a hierarchical adaptive deep convolutional neural network (ADCNN), the accuracy is 97.7%.

In these recognition methods, the ability to reinforce learning is the most similar to manually detection. Among them, Wang et al. [23] proposed a reinforcement neural architecture search method to achieve success. The article suggested and validated the neural network architecture automatic search method. The framework of the article includes two parts: the controller model and the child model.

The controller model has 2 Nascell layers, and the output of this model are convolutional kernel size and kernel number and a pooling kernel size of each layer. They formed the CNN [24]. The child models are CNNs. The model consists of an input layer. The two groups of the same convolutional layer, the pooling layer, take turns to each other. The complete connection layer.

However, the time domain is the main emphasis of these methods. The time domain analysis is unable to observe the frequency-dependent signal properties for the vibration signal. The frequency domain analysis is more succinct than the time domain. Following the signal in the frequency domain provides a deeper and more practical analysis of the issue.

This paper proposed feature extraction and recognition for rolling bearing fault diagnosis based on frequency domain and Long Short-Term Memory (LSTM) to overcome the shortcomings mentioned before. In this method, uses Fast Fourier Transform (FFT) to alter the bearing's time-domain signal before it is transmitted to the network. We only need to fine-tune the maxepochs and hidden units in the process.

II. Methodology

A. Overview of system

The structure of the proposed method is shown in Fig. 1, the vibration signal is collected by accelerometers which are located drive end of the motor housing, and the signal is measured at 1750 RPM in each working state. Then the vibration signal is divided into several overlapping samples, and each sample is window processed and transformed with FFT. And then input these data into networks, the neural network train these data, gets outputs and calculates the accuracy. Adjust the number of the hidden units and MaxEpochs until the result reaches the best.

The layers in the neural network system include the input layer, Bi-LSTM layer, fully connection layer, softmax Layer and classification Layer.

B. Frequency domain analysis

The existence of FFT makes Discrete Fourier Transform play a central role in algorithms in digital signal processing. The calculation formula of Discrete Fourier Transform is [25]:

$X(k) = \sum_{n=0}^{N-1} x(n)W^{nk}$,(0≤ $k \leq N-1$)

$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)W^{-nk}$,(0≤ $k \leq N-1$)

Where x stands a limited long sequence, X stands data after Discrete Fourier transformation, N is sampled n points in a sinusoidal cycle, and $W = e^{\frac{-j2p}{N}}$ is the Fourier factor. For a

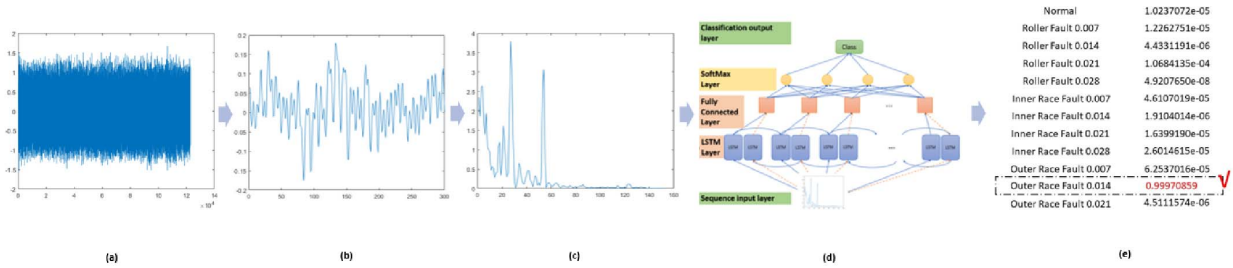


Fig. 1: Overview of the system, (a) is raw data example from CWRU dataset [29], (b) is egmented data with 300 points, (c) is segmented data using FFT, (d) is training data sent to the network and (e) is score of 12 classes.

discrete signal x, FFT will transform the signal in frequency domain. Fig. 2 shows that the signal changes before and after FFT.

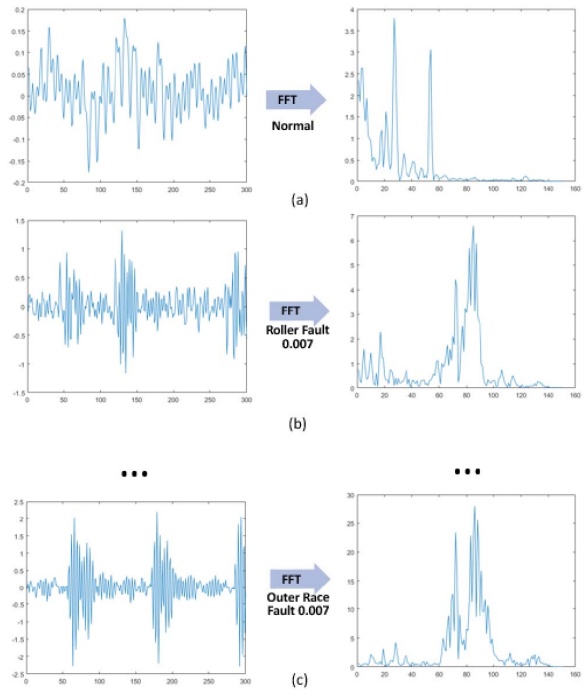


Fig. 2: Signal transformed by FFT. Left part is raw data, each of them have 300 points, right part is data with FFT, each of them have 150 points. And (a) is an example of Normal signal, (b) is an example of Roller Fault size 0.007 inches signal, (c) is an example of Outer Fault size 0.007 inches signal.

C. Bi-LSTM

Long Short Term Memory (LSTM) is a special Recurrent Neural Network (RNN) that can learn long-term dependencies [26]. Vanishing and exploding gradient problems are hard to avoid in traditional RNNs. LSTM learned the long-term dependence on the network with passed these problems. The hidden layer of traditional RNN is usually a $tanh$ function or ReLU. A typical LSTM unit will conclude 3 $sigmoid$ layer and 1 $tanh$ layer.

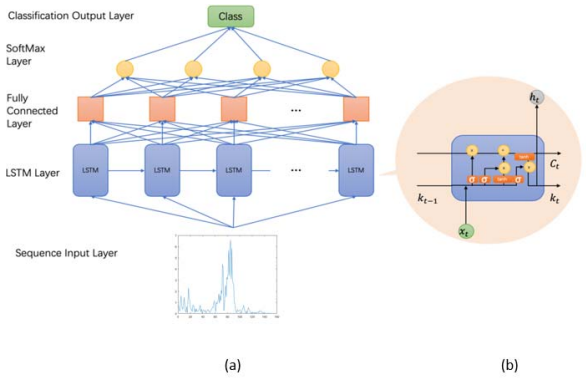


Fig. 3: (a) is notwork structure of LSTM, (b) is LSTM unit. The repetitive module in LSTM has four interaction layers, three sigmoid and one $tanh$, and they interact in a unique way.

LSTM consists of three gate variables: Input gate, Forgetting Gate and Output gate.

A cell state C is applied in LSTM with only a few linear operating on it, which could retain information easily. The first gate in LSTM is forget gate, which decides what information should be discarded. x_t will be send to a sigmoid function with h_{t-1} and get a value between 0 and 1 which multiplied with the cell state C_{t-1}. The output of the sigmoid function will decide how much information remains. Part of the information in the last layer $t-1$ has been forgotten in the cell state C_{t-1}, and the new information in the current layer will be added by a $tanh$ function and a sigmoid function. This sigmoid function is called input gate and the output of it will multiply by a $tanh$ function. When the value of it is 0, the cell state doesn't need to update.

Then the last cell state C_{t-1} multiply with forget gate f_t to discard part of information and update the information from $i_t \times C_t$. The output gate concludes the information in updated cell state C_t and the output after a $tanh$ function and a sigmoid function. A brief figure of LSTM is shown in Fig. 3.

TABLE I: Description of 12 states, one normal state and eleven failure states

Data no.	Fault type	Fault size/inches	Motor speed (r/min)	Sampling frequency (kHz)	Size of training / testing samples
1	Normal	0	1750	12	350/50
2	Roller Fault	0.007	1750	12	350/50
3	Roller Fault	0.014	1750	12	350/50
4	Roller Fault	0.021	1750	12	350/50
5	Roller Fault	0.028	1750	12	350/50
6	Inner Race Fault	0.007	1750	12	350/50
7	Inner Race Fault	0.014	1750	12	350/50
8	Inner Race Fault	0.021	1750	12	350/50
9	Inner Race Fault	0.028	1750	12	350/50
10	Outer Race Fault-Center@6:00	0.007	1750	12	350/50
11	Outer Race Fault-Center@6:00	0.014	1750	12	350/50
12	Outer Race Fault-Center@6:00	0.021	1750	12	350/50

LSTM has the advantages of long-term trajectory memory and short memory unification, simulation of selective brain forgetting, and more accurate trajectory modelling. Therefore, the multi-layer structure can be mixed to solve the efficiency and stability problems of massive data training.

Bi-LSTM is an RNN with LSTM unit and will predict the sequence from both directions [27]. It could perform better in a sequence without directionality [28]. Actually the sequences are sent to two LSTM unit indenpently with different directions. The structure of Bi-LSTM is shown in (a) of Fig. 4. In this paper, Bi-LSTM model was selected instead of LSTM model because the input data is the FFT of the vibration signals. In frequency domain, there is no strong dependent relationship between current and previous components. Bi-LSTM can model both relationship of the one from low frequency to high frequency as well as the one from high frequency to low frequency.

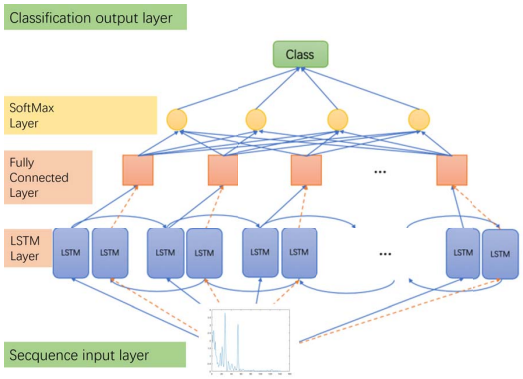


Fig. 4: Bi-LSTM networks. The LSTM framework is used to merge the input sequence's front and backward directions. The two LSTM layers' vectors can be added to, averaged out, or connected.

III. Experiment

A. Dataset

The data are collected from the Electrical Engineering Laboratory at Case Western Reserve University [29]. The data was collected at 12,000 samples/second and has four different fault inches: 0.007, 0.014, 0.021 and 0.028 inches.

B. Parameters

The parameter information is shown in Table 1. Twelve data in different states include one normal state and eleven faults states. Each of these has 400 samples. Among them, the first 350 samples are training samples, and the left 50 samples are test samples, a totally of 4800 samples, and the length of each sample is 300. The last vision of network Parameters in this paper is in the following Table II.

TABLE II: Parameters of network

Name of Parameters	Size of Parameters
Mini Batch-Size	**20**
Input Size	**1**
Number Hidden Units	**100**
Number Classes	**12**
Max Epochs	**100**

C. Experimental results

In this experiment, the experimental setting follows the same one in paper [23] for a fair comparison. In this setting, the subset data related to 12 faulty types are used, as shown in Table 1. In [23], the result of the recognition accuracy is 98.47%, and the standard deviation obtained by reinforcement neural architecture after 10 experiments are 0.61.

The average classification accuracy rate of proposed method on raw data can reach 94.88%, and the accuracy standard deviation after 10 experiments is 1.90 as shown in Table 3. This paper's average classification accuracy rate for data with FFT can reach 99.70±0.23%, improved by 4.82%. Compared with neural reinforcement architecture, enhanced by 1.23%, the classification accuracy of test samples is summarized in Table 3, which is better than reinforcement neural architecture. Therefore, it can be considered that FFT data in Bi-LSTM has better performance.

TABLE III: Results of comparison paper, Bi-LSTM for raw data and FFT data

Method	Average accuracy (%) ± standard deviation (%)
Reinforcement neural architecture	98.47 ± 0.61 [23]
Raw data/Bi-LSTM	94.88±1.90
FFT/Bi-LSTM	**99.70±0.23**

And the 10 times of results show in table 4, the best result for raw data is 97.83% and the worst is 92%. the best result for data with FFT approached 100% and the worst is 99.33%, which improved 2.17% and 7.33% respectively.

TABLE IV: Test results of Raw data and Data with FFT

NO. Times	Raw data/Bi-LSTM %	FFT/Bi-LSTM %
1	97.83	100
2	96.67	100
3	96	99.83
4	96.5	99.83
5	95.67	99.83
6	94	99.67
7	93.67	99.5
8	93.67	99.5
9	92.83	99.5
10	92	99.33

It is significantly better than the comparison network structures, it can be seen that the proposed method has better performance. the diagnostic results are summarized in Fig. 5.

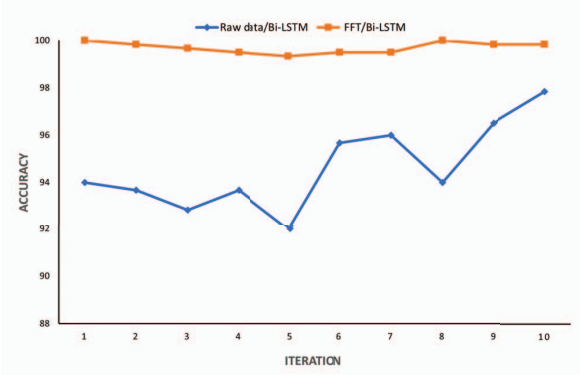


Fig. 5: Accuracy of the 10 times.The horizontal axis is the number of iterations and Vertical axis is accuracy. The blue folding line is the result of the raw signal combined with the Bi-LSTM network, the orange folding line is the result of the signal after FFT combined with the BI-LSTM network.

Conclusion

Even if the data from the time sequence is transferred to the frequency domain after being processed by FFT, it remains a continuous signal. It is equally applicable to engineering practice and produces positive outcomes. In the paper [30], [31], [32], combination method of FFT and LSTM has achieved good results.

As a result of comparisons between paper [23], the experimental procedures are completely consistent. Paper [23] only uses training and testing sets, and there is no verification set. Therefore, there is no verification set in the experiments described in this paper.

Finally, deep learning methods, in particular, have become increasingly popular in recent years. As a result, deep learning-based automated recognition is now utilised in numerous new sectors. The goal of increasingly more automatic recognition systems is to simplify our lives. In this paper, the signal is transformed from the time domain to the frequency domain by fully using of the FFT. The best result, when combined with BI-LSTM, is 100%, while the mean value over ten times is 99.70±0.23 %. Future studies on the existing research issue can include a variety of methods, including EMD. Practical engineering challenges will be solved using some methodologies that will be enhanced and employed in this research.

Acknowledgment

The authors would like to thank the Natural Science Foundation of China (NSFC) on the International Collaboration and Exchange Project under Grant 6196020601 for the support.

References

[1] S. Nandi, H. A. Toliyat and X. Li, “Condition monitoring and fault diagnosis of electrical motors—A review“, IEEE transactions on energy conversion, 2005, 20(4): 719-729.

[2] L. Rosado, N. H. Forster, K. L. Thompson and J. W. Cooke, et al. “Rolling contact fatigue life and spall propagation of AISI M50, M50NiL, and AISI 52100, Part I: experimental results“, Tribology Transactions, 2009, 53(1): 29-41.

[3] C. Wu, J. Liu, F. Peng and R. Li. “Gearbox fault diagnosis using adaptive zero phase time-varying filter based on multi-scale chirplet sparse signal decomposition“, Chinese Journal of Mechanical Engineering, 2013, 26(4): 831-838.

[4] X. Li, S. Liu, S. Cheng, J. Lin, R. Liu, L. Wang and Z. Zhou, “Research on Ultrasonic Quantitative Evaluation Technology of Complex Defects Based on Neural Network”, Journal of Physics: Conference Series. IOP Publishing, 2022, 2196(1): 012022.

[5] Y. Cao, Y. An, S. S, G. X and Y. S, “A statistical study of railway safety in China and Japan 1990–2020.” Accident Analysis Prevention 175 (2022): 106764.

[6] N. E Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, Q. Zheng, NC. Yen, C. C. Tung and H. H. Liu, “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, vol. 454, no. 1971, pp. 903-995, 1998. Available: 10.1098/rspa.1998.0193.

[7] L. Saidi, J. Ali and F. Fnaiech, “Bi-spectrum based-EMD applied to the non-stationary vibration signals for bearing faults diagnosis”, ISA Transactions, vol. 53, no. 5, pp. 1650-1660, 2014. Available: 10.1016/j.isatra.2014.06.002.

[8] J. B. Ali, N. Fnaiech, L. Saidi, B. Chebel-Morello and F. Fnaiech, “Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals”, Applied Acoustics, vol. 89, pp. 16-27, 2015. Available: 10.1016/j.apacoust.2014.08.016.

[9] J. Dybała and R. Zimroz, “Rolling bearing diagnosing method based on Empirical Mode Decomposition of machine vibration signal”, Applied Acoustics, vol. 77, pp. 195-203, 2014. Available: 10.1016/j.apacoust.2013.09.001.

[10] C. Wang, M. Gan and C. Zhu, “Non-negative EMD manifold for feature extraction in machinery fault diagnosis”, Measurement, vol. 70, pp. 188-202, 2015. Available: 10.1016/j.measurement.2015.04.006.

[11] P. Arena, L. Patanè and A. Spinosa, “Data-based analysis of Laplacian Eigenmaps for manifold reduction in supervised Liquid State classifiers”, Information Sciences, vol. 478, pp. 28-39, 2019. Available: 10.1016/j.ins.2018.11.017.

[12] X. He and N. Partha, “Locality preserving projections.” Advances in neural information processing systems 16 (2003).

[13] S. Roweis and L. Saul, “Nonlinear Dimensionality Reduction by Locally Linear Embedding”, Science, vol. 290, no. 5500, pp. 2323-2326, 2000. Available: 10.1126/science.290.5500.2323.

[14] T. Zhang, J. Yang, D Zhao and X. Ge. “Linear local tangent space alignment and application to face recognition.” Neurocomputing 70.7-9 (2007): 1547-1553.

[15] Q. He, "Vibration signal classification by wavelet packet energy flow manifold learning", Journal of Sound and Vibration, vol. 332, no. 7, pp. 1881-1894, 2013. Available: 10.1016/j.jsv.2012.11.006.
[16] B. Li and Y. Zhang, "Supervised locally linear embedding projection (SLLEP) for machinery fault diagnosis", Mechanical Systems and Signal Processing, vol. 25, no. 8, pp. 3125-3134, 2011. Available: 10.1016/j.ymssp.2011.05.001.
[17] H. Shao, H. Jiang, X. Li and S. Wu, "Intelligent fault diagnosis of rolling bearing using deep wavelet auto-encoder with extreme learning machine", Knowledge-Based Systems, 2018, 140: 1-14.
[18] H. Shao, H. Jiang, Y. Lin and X. Li, "A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders", Mechanical Systems and Signal Processing, 2018, 102: 278-297.
[19] J. Tao, Y. Liu, D. Yang, F. Tang and C. Liu, "Fault diagnosis of rolling bearing using deep belief networks", 2015 International Symposium on Material, Energy and Environment Engineering. Atlantis Press, 2015: 566-569.
[20] M. Gan and C. Wang, "Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings", Mechanical Systems and Signal Processing, 2016, 72: 92-104.
[21] Z. Zhuang and W. Qin, "Intelligent fault diagnosis of rolling bearing using one-dimensional multi-scale deep convolutional neural network based health state classification", 2018 IEEE 15th International conference on networking, sensing and control (ICNSC). IEEE, 2018: 1-6.
[22] X. Guo, L. Chen, C. Shen, "Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis", Measurement, 2016, 93: 490-502.
[23] R. Wang, H. Jiang, X. Li and S. Liu, "A reinforcement neural architecture search method for rolling bearing fault diagnosis", Measurement, vol. 154, p. 107417, 2020. Available: 10.1016/j.measurement.2019.107417.
[24] B. Zoph and Q.V. Le, "Neural architecture search with reinforcement learning", arXiv preprint arXiv:1611.01578, 2016.
[25] P. Heckbert, "Fourier transforms and the fast Fourier transform (FFT) algorithm", Computer Graphics, 1995, 2: 15-463.
[26] J. Schmidhuber, and S. Hochreiter. "Long short-term memory", Neural Comput 9.8 (1997): 1735-1780.
[27] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks", in IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, Nov. 1997, doi: 10.1109/78.650093
[28] A. Graves, S. Fernández and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition", International conference on artificial neural networks. Springer, Berlin, Heidelberg, 2005: 799-804.
[29] CWRU, Case Western Reserve University, "Bearing data", [Online]. Available:https://engineering.case.edu/bearingdatacenter/download-data-file
[30] J. Wan, L. Shih, Y. Hong and H. Che, "Evaluation of deep learning neural networks for surface roughness prediction using vibration signal analysis." Applied Sciences 9.7 (2019): 1462.
[31] M. Hachemi, A. Ghomari, Y. Hadjadj-Aoul and G. Rubino, "Mobile traffic forecasting using a combined FFT/LSTM strategy in SDN networks." 2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR). IEEE, 2021.
[32] H. Park, E. Kwon, S. Byon, W. Shin, E. Jung and Y. Lee, "Punch Analysis with FFT and LSTM of Accelerometer and Gyroscope Data." 2020 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2020.

A.2 Published work 2: Frequency Domain Feature Extraction and Long Short-Term Memory for Rolling Bearing Fault Diagnosis

applied sciences

MDPI

Article

Real-Time Monitoring of Wind Turbine Bearing Using Simple Neural Network on Raspberry Pi

Tianhao Wang [1,*], Hongying Meng [1], Rui Qin [2], Fan Zhang [3] and Asoke Kumar Nandi [1]

[1] Department of Electronic and Electrical Engineering, Brunel University London, London UB8 3PH, UK; hongying.meng@brunel.ac.uk (H.M.); asoke.nandi@brunel.ac.uk (A.K.N.)
[2] School of Computing and Mathematical Sciences, University of Leicester, Leicester LE1 7RH, UK; rq20@leicester.ac.uk
[3] School of Design, Southwest Jiaotong University, Chengdu 610031, China; fan.zhang@swjtu.edu.cn
* Correspondence: 1815203@brunel.ac.uk

Abstract: Wind turbines are a crucial part of renewable energy generation, and their reliable and efficient operation is paramount in ensuring clean energy availability. However, the bearings in wind turbines are subjected to high stress and loads, resulting in faults that can lead to costly downtime and repairs. Fault detection in real time is critical to minimize downtime and reduce maintenance costs. In this work, a simple neural network model was designed and implemented on a Raspberry Pi for the real-time detection of wind turbine bearing faults. The model was trained to accurately identify complex patterns in raw sensor data of healthy and faulty bearings. By splitting the data into smaller segments, the model can quickly analyze each segment and generate predictions at high speed. Additionally, simplified algorithms were developed to analyze the segments with minimum latency. The proposed system can efficiently process the sensor data and performs rapid analysis and prediction within 0.06 milliseconds per data segment. The experimental results demonstrate that the model achieves a 99.8% accuracy in detecting wind turbine bearing faults within milliseconds of their occurrence. The model's ability to generate real-time predictions and to provide an overall assessment of the bearing's health can significantly reduce maintenance costs and increase the availability and efficiency of wind turbines.

Keywords: wind turbines; neural network; real-time implementation; bearing fault detection

Citation: Wang, T.; Meng, H.; Qin, R.; Zhang, F.; Nandi, A.K. Real-Time Monitoring of Wind Turbine Bearing Using Simple Neural Network on Raspberry Pi. *Appl. Sci.* **2024**, *14*, 3129. https://doi.org/10.3390/app14073129

Academic Editors: Javier Martinez-Roman, Manuel Pineda-Sanchez, Martin Riera-Guasp, Angel Sapena-Bano and Jordi Burriel-Valencia

Received: 1 March 2024
Revised: 3 April 2024
Accepted: 4 April 2024
Published: 8 April 2024

1. Introduction

Wind energy has now become a component of the energy mix, significantly bolstering its role in renewable energy sources. According to the International Energy Agency (IEA), in 2020, the global installed capacity for wind power amounted to 721 GW [1]. Wind power plays a role in minimizing impact and fostering the adoption of low carbon alternatives over traditional fossil fuels. The growing demand for energy has driven the acceptance of wind turbines as a pivotal technology for generating renewable energy, which is essential for our worldwide shift towards cleaner and more sustainable energy sources. This transition is essential in reducing our reliance on fossil fuels, curbing carbon emissions and promoting progress. The reliability of wind turbines largely determines the consistency of the electricity supply. When wind turbines are operating, they experience loads and stress in key components like bearings. Bearings are essential in converting wind energy into power. However, these components are susceptible to wear and damage, especially when exposed to conditions such as high winds, temperature fluctuations, or salty offshore atmospheres. Any defects in the bearings possibly affect the performance of wind turbines and may lead to unforeseen unexpected downtime and substantial maintenance costs. This ultimately impacts the viability and overall energy output of wind farms.

One of the most common reasons for wind turbine downtime is generator failure, which accounts for 37% of all failure downtimes [2]. Real-time generator defect detection can aid in preventing system shutdowns and mitigating their effects.

There are a number of defect detection systems for wind turbine generators. These methods include spatiotemporal attention-based long short-term memory auto-encoder networks [2], marker-tracking for immediate rotational speed measurement [3], chaotic system and extension neural network fault diagnostics [4], time-varying models with augmented observers [5], deep learning approaches for sensor data prediction and fault diagnosis [6], sensor selection algorithms for real-time fault detection [7], enhanced variational mode algorithm fault diagnosis [8], image texture analysis for fault detection and classification [9], and cost-sensitive algorithms for online fault detection [10].

Wind turbine bearings are one of the key components, and the performance and lifespan of the device are significantly influenced by their normal operation. However, wind turbine bearings frequently sustain damage and malfunction in long-term operation due to high loads and harsh weather conditions. These flaws, which pose major risks to the wind turbine's ability to operate safely, may include rolling bead fatigue, insufficient lubrication, and an unbalanced load [11].

For prompt maintenance actions, shorter stop times, and lower maintenance costs, the real-time detection of wind turbine bearing failures is crucial. Preventive maintenance can be accomplished through real-time monitoring, quickly and precisely detecting bearings, preventing further degradation of faults, and correcting faults to maintain wind turbines' continuous functioning and dependability.

Currently, the detection of bearing faults mainly relies on traditional vibration analysis and oil analysis [11]. These techniques often involve processing amounts of raw data, which makes real-time detection and prediction more complicated. Recently, deep learning techniques such as convolutional neural networks (CNNs) and long short-term memory networks (LSTMs) have also been used in bearing fault detection [12–14]. However, their complex model structures and numerous parameters make it challenging to deploy them in real time on edge devices. With variational mode decomposition with deep convolutional neural networks (VMD-DCNNs) [15], it is possible to diagnose rolling bearing faults that do not require manual labor or human experience. The performance variations of existing approaches in various contexts and situations are obtained by extracting characteristics from each inherent mode function (IMF). The multi-scale convolutional neural network with bidirectional long short-term memory (MSCNN-BiLSTM) model [16], using a weighted majority voting rule, enhance the intelligent fault diagnosis of bearings in wind turbines under complex working and testing environments, achieving improved diagnostic performance compared to existing methods. The deep bi-directional long short-term memory (DB-LSTM) [17] method circumvents feature selection challenges, enhances computational efficiency, and generates simulated data that closely resemble real-world settings, hence improving the approach's applicability. Supervisory Control And Data Acquisition (SCADA) is ensemble method that consists of XGBoost [18], a framework based on ensemble learning and genetic algorithms, utilizing SCADA data to detect faults in wind turbines. The deep convolutional neural network (DCNN) and Synchro Squeezing Transform (SST) [19], compared to traditional spectral analysis methods, can automatically identify fault features, avoiding misdiagnosis and missed diagnosis that may be caused by manual identification, and its excellent classification effect has been verified through experiments. The Self-Adaptive Teaching-Learning-Based Optimization Multi-Layer Perceptron (SATLBO-MLP) [20] is a data-based wind turbine bearing fault diagnosis method that is applied to the SATLBO algorithm optimization MLP model. The related experimental results demonstrate the effectiveness of the method.

Some studies have attempted to address this issue by training models in the cloud [21] and then transferring them to edge devices. However, this method relies on a network connection and has challenges in meeting real-time performance demands in practical applications. Currently, research mostly concentrates on the impact of models, while giving

less consideration to the practical viability of models, such as optimizing model size and predictive latency. This aspect requires further enhancement.

Overall, real-time defect detection research on wind turbine bearing failure has not been conducted, and most methods need to process raw data, which will cost time in practical engineering applications, resulting in the inability to make strategic adjustments and take appropriate measures in a timely manner, which may lead to irreparable losses. In general, the existing approaches for detecting bearing faults still have limitations in terms of real-time capability, user-friendliness, and other factors, making it challenging to fulfill the requirements of real-time monitoring and prediction in industrial settings.

In spite of that, deep learning exhibits significant promise for utilization in wind turbine health monitoring, particularly in the detection of bearing faults. Compared to traditional machine learning techniques, the main benefits of deep learning include the following: Unsupervised learning has the capacity to extract intricate and non-linear patterns from data without the need for manual configuration, making it particularly well suited for complicated datasets [22]. Deep learning acquires intricate functional connections by utilizing multi-layer network architectures with strong fitting capabilities [23]. Finally, it implements end-to-end learning, autonomously capturing the geographical and temporal relationships within the data [24]. Due to the growing processing power and data volume, performance is consistently enhancing and has surpassed manually built algorithms in numerous tasks [25]. Furthermore, the operational data of bearings contain abundant information in both the time and frequency domains. The neural network model has the capability to automatically acquire these intricate characteristics by utilizing its multi-layer network architecture. Furthermore, it can proficiently address the difficulties posed by substantial data quantities and the fusion of multiple sensors. Thanks to advancements in edge computing and mobile internet technology, the neural network model can now be deployed in real time on edge devices. For example, the neural network model can be deployed on the control box of a wind turbine [6], resulting in a significant improvement in the response speed of bearing fault monitoring. In summary, neural networks have shown huge application potential in the monitoring of wind turbine bearing failure monitoring.

In order to achieve the real-time detection of wind turbine bearing failure, this study aimed to apply machine learning methods in a Raspberry Pi to achieve high-precision and low-delay real-time monitoring with this portable device, which is of great significance for improving the reliability and utilization rate of wind power equipment. Our model can accurately identify patterns by training the physical characteristics of health and fault bearings. The data were divided into smaller segments so that the model can quickly analyze each segment and generate high-speed predictions. In addition, in order to achieve a minimal delay treatment, high-efficiency algorithms were developed. The network was trained and, subsequently, the NN algorithm was embedded into the Raspberry Pi. The network has two fully connected layers, and the tested time was 0.06 milliseconds on the Raspberry Pi. The results of this study indicate that the model can accurately detect wind turbine bearing faults and provide real-time predictions within milliseconds of the fault occurrence. This model possesses the capability to generate real-time predictions and assess the holistic health condition of bearings, thereby substantially diminishing maintenance expenses and enhancing the accessibility and efficacy of wind turbines. In essence, this study demonstrates the potential of AI-driven solutions in optimizing the generation of renewable energy and mitigating reliance on fossil fuels.

2. Methodology

2.1. Overview of the Proposed System

The overview of the proposed method is shown in Figure 1. The vibration signal is collected by accelerometers, which are located at the non-drive end in the wind turbine. Wind turbines are usually constructed in areas where wind energy is abundant, such as coastlines, mountains, plains, and deserts.

Figure 1a shows wind turbines constructed on grasslands. The data were trained in the model, which was implemented in Python language on PC. The data were divided into small segments; thus, the model can quickly analyze each segment without other processing methods, as shown in Figure 1b,c. The trained neural model was embedded in the Raspberry Pi, as is shown in Figure 1d. The last step was for the trained model to predict each segment's score, as is shown in Figure 1e.

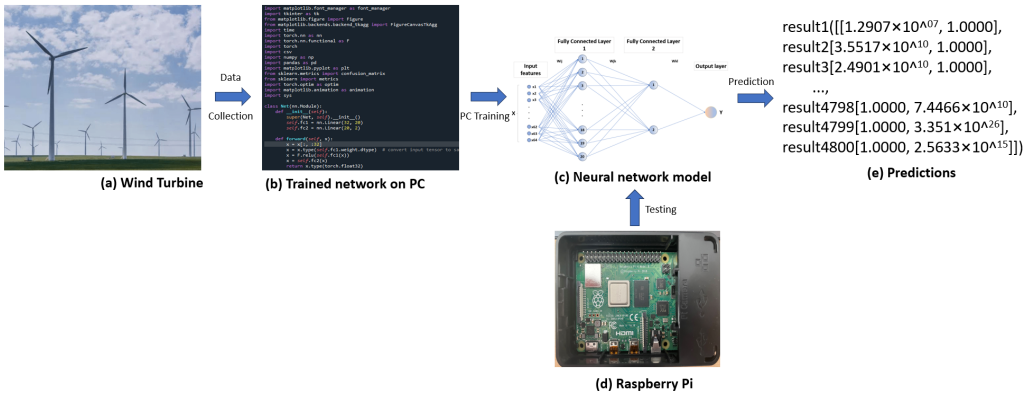


Figure 1. Overview of the system: (**a**) wind turbine; (**b**) modeling in python language; (**c**) neural network model; (**d**) Raspberry Pi implementation; and (**e**) predictions.

2.2. The Description of the Neural Network Model

Many research works use neural networks for fault detection or the monitoring of bearings. Fu et al. [26] monitored wind turbines with deep learning. In their paper, CNN and LSTM are used to analyze variables and monitor wind turbine gears. Gearbox bearing temperature data are processed for AI monitoring and troubleshooting. Asmuth and Korb [27] provide a basic three-dimensional CNN-based wake model to predict wind turbine wake flow fields. The model accurately predicts wake flow characteristics, showing its potential for wind turbine wake forecasts. Xie et al. [28] proposed an attention mechanism-based CNN-LSTM wind turbine fault prediction model. The CNN extracts features, LSTM captures time sequence correlations, and an attention algorithm gathers fault-related target information from the ontology-annotated Semantic Sensor Network. The model was shown to be precise and generalized.

In this paper, we adopted a real-time detection neural network model to assess wind turbine bearing failure, as shown in Figure 2. The model adopts the architecture of a Multi-Layer Perceptron (MLP) [29].

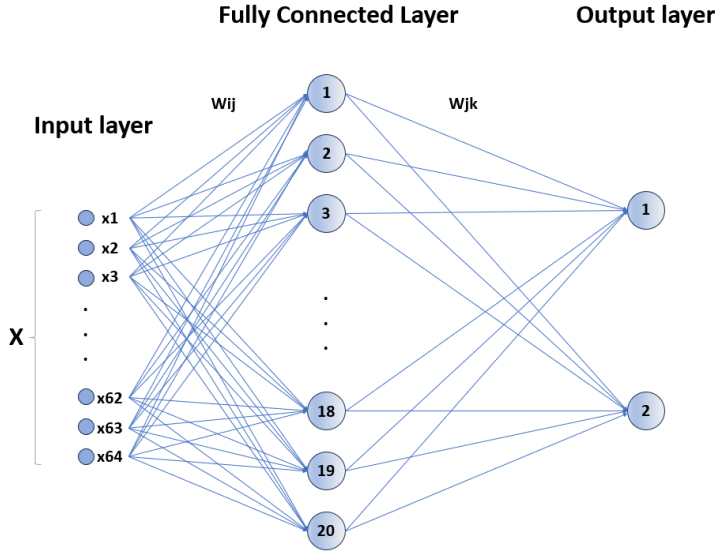


Figure 2. Proposed network.

The neural network model contains two fully connected layers (FC1 and FC2), and the second fully connected layer (FC2) acts as the output layer. The first full connected layer (FC1) has 64 input features and 20 hidden units, which are used to learn complex features in the input data. The output layer (FC2) is a layer that takes the 20 output features from FC1 and reduces them to two output features, which are used to map the learned features to faults and normal categories.

2.3. From Input Layer to Hidden Layer

The input vector $x \in \mathbb{R}^{64}$. The first fully connected layer has a weight matrix $W_1 \in \mathbb{R}^{20\times64}$ and a bias vector $b_1 \in \mathbb{R}^{20}$. The operation performed by this layer can be expressed as:

$$h = W_1 \times x + b_1 \tag{1}$$

Then, the ReLU (Rectified Linear Unit) activation function is applied to h. The ReLU function is defined as $f(z) = \max(0, z)$, which introduces non-linearity, allowing the model to learn more complex functions:

$$h' = \max(0, h) \tag{2}$$

In the hidden layer of the model, the ReLU (Rectified Linear Unit) [30] is used as the activation function. The ReLU function can enhance the non-linear modeling capabilities of the model and help capture the complex features of the input data.

2.4. From Hidden Layer to Output Layer

The vector h' is then passed through the output layer ('FC2'), which transforms the 20-dimensional input into a 2-dimensional output suitable for binary classification. Let the weight matrix of the output layer be $W_2 \in \mathbb{R}^{2\times20}$ and the bias vector be $b_2 \in \mathbb{R}^2$. The final output y can be expressed as:

$$y = W_2 \times h' + b_2 \tag{3}$$

During the forward-direction of the model, the input data that first passed the first full connection layer (FC1) and activated the function ReLU are applied to the output of the hidden layer. Then, the final prediction result is output through the output layer (FC2).

In order to enable the model to accurately identify the mode of health and bearing faults, the method of monitoring learning was adopted for the training of model parameters. The labeling dataset was used for training and optimizing the weight and bias of the model by minimizing the loss function, so that the prediction results of the model were as close to the real label as possible.

The structure of neural network models is simple and effective, with less parameters and computing complexity, and it can operate efficiently on edge equipment such as a Raspberry Pi. Through this model, we can realize the real-time detection of wind turbine bearing faults and provide fast and accurate predictive results to provide feasible solutions for reducing maintenance costs and improving the availability and efficiency of wind turbines.

2.5. Acceleration Sensor and Data Logger

As shown in Figure 3a, the primary purpose of the acceleration sensor located on the bearing is to monitor and assess the condition and functionality of the bearings. These sensors offer an efficient means for detecting and preventing possible malfunctions by monitoring the acceleration of the vibration in the bearing while it is in operation [31–33].

As shown in Figure 3b, the data logger obtains an analog signal from the acceleration sensor at the bearing, and uses the A/D converter to convert it to a digital signal. Then, the conversion signal is stored [34,35].

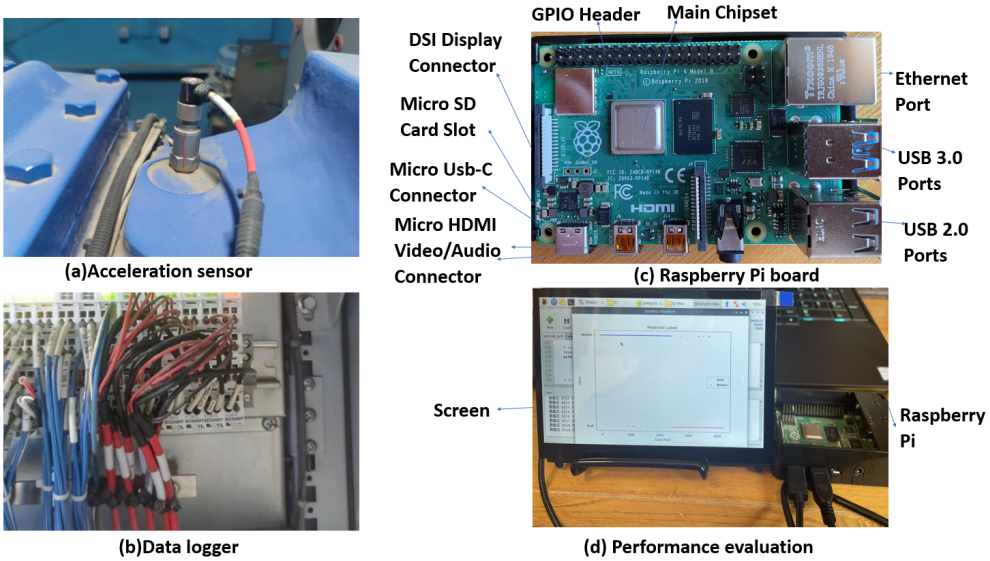


Figure 3. (**a**) Acceleration sensor, (**b**) data logger, (**c**) Raspberry Pi board, and (**d**) performance evaluation.

2.6. Raspberry Pi

With Eben Upton serving as the project manager, the "Raspberry Pi Charity Foundation" was established in the UK. The world's smallest desktop computer, sometimes referred to as a card-type computer, was officially introduced in March 2012 by Emben Apoton [36], a Cambridge University research lab.

In this paper, the network model was trained on the desktop and it was embedded into a Raspberry Pi 4 [37]. It can be seen as a small computer and is easy to carry. In Figure 3c, some parts are labeled and the touch screen is shown in Figure 3d.

3. Experiment

3.1. Dataset

There are many acceleration sensors on the generator bearing of a wind turbine. The data used in the experiment were the signal of one of the acceleration sensors. The data were collected from the turbine generator of a company. The vibration data were collected at 25,600 samples/second and a collection time of 3 s.

The generator is a device that converts mechanical energy into electrical energy. It usually consists of a rotor and a stator, where the rotor is supported by bearings rotating on a shaft. The generator bearing is an important component that supports the rotor, and its proper functioning is critical to the performance and reliability of the generator. It is shown in Figure 4a.

In order to monitor the health of generator bearings, acceleration sensors are installed on the bearings. An acceleration sensor is a device that measures the acceleration of an object. When a generator is in operation, the bearings are subjected to a variety of forces and vibrations that cause the bearings to change in acceleration. The acceleration sensor senses and measures these changes and converts them into a digital signal. It is shown in Figure 4b.

Important information about generator bearing vibration can be obtained by analyzing the data collected by the acceleration sensor. Vibration data can include parameters such as the vibration amplitude, the frequency, and the time variation. These data are critical for monitoring the health of the bearing. Abnormal vibration data may indicate bearing failure or wear, such as loose bearings, overheating, or damage. By detecting these problems in time, the safety monitoring system can take appropriate repair and maintenance measures to prevent further damage or failure from occurring.

Figure 4. Data collection platform. (**a**) Wind turbine generator; (**b**) acceleration sensor.

Therefore, acceleration sensor data acquisition is crucial for generator bearing monitoring and maintenance. It provides valuable information to better manage and optimize the generator system, ensuring its proper operation and extended service life.

Two different states are Normal and Fault. Each state has 9600 segments in testing, 2400 segments in training, and each segment has 64 samples; details are shown in Table 1.

Table 1. Description of data.

Fault Type	Training Segments	Testing Segments	Samples in Each Segment
Normal	9600	2400	64
Fault	9600	2400	64

3.2. Experimental Setup

3.2.1. Data Analysis

As shown in Figure 5, the length of each sample of raw data was 64, and the raw data produced irregular waveform diagrams. The difference in the amplitude value was readily apparent. After FFT, the difference between health and faults at the peak value was also obvious.

3.2.2. Data Groups

The parameter information is shown in Table 2. The data were evenly divided into 10 groups of data with different states, including five normal states (Group 1–5) and five fault states (Group 6–10). In the experiment, in order to validate the performance of the methodology, comparison models were introduced, i.e., a Medium Neural Network, Wide Neural Network, Bi-layer Neural Network, Tri-layer Neural Network, and Narrow Neural Network. The validation of the method accuracy was determined under 5-fold cross-validation.

Table 2. Dataset was split into training and testing subsets.

Training Group		Testing Group	
Subset 1	2, 7, 3, 8, 4, 9, 5, 10	Subset 1	1, 6
Subset 2	1, 6, 3, 8, 4, 9, 5, 10	Subset 2	2, 7
Subset 3	1, 6, 2, 7, 4, 9, 5, 10	Subset 3	3, 8
Subset 4	1, 6, 2, 7, 3, 8, 5, 10	Subset 4	4, 9
Subset 5	1, 6, 2, 7, 3, 8, 4, 9	Subset 5	5, 10

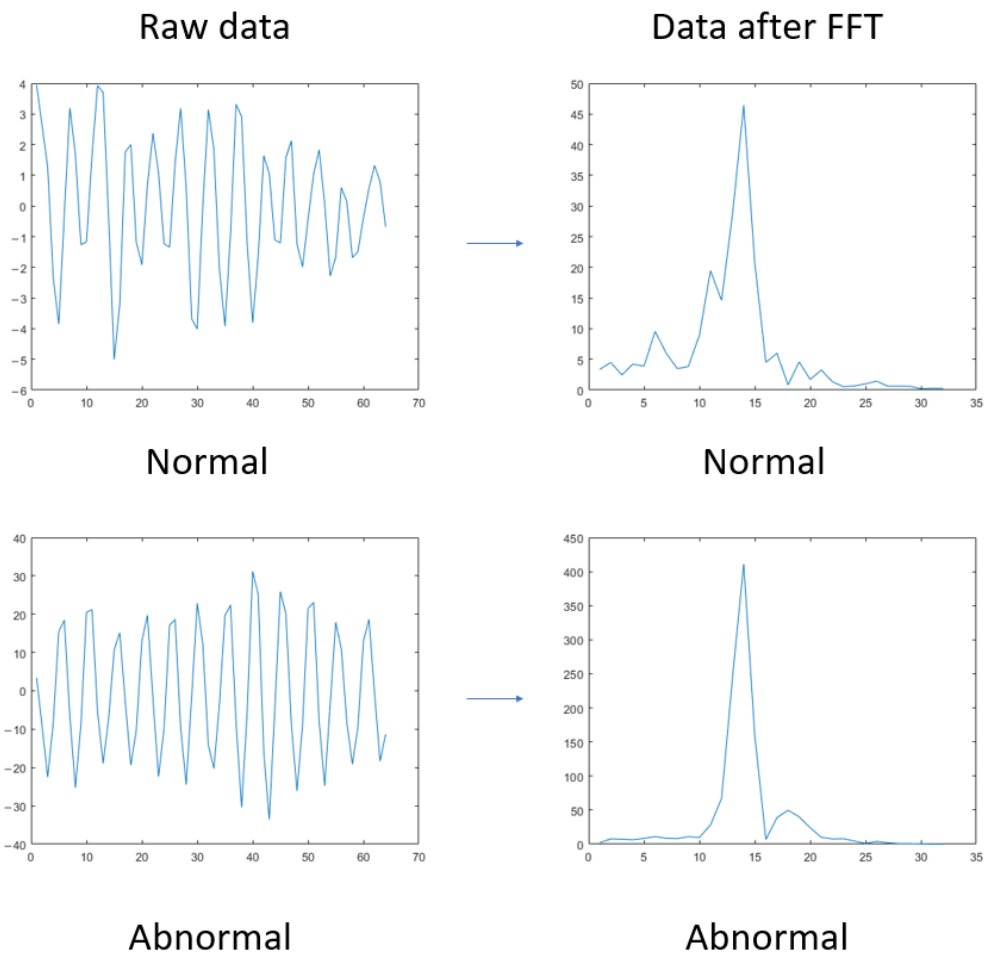


Figure 5. Data spectrum.

3.3. Experimental Results

In this experiment, to verify the performance, five comparison networks were introduced. The first was used to confirm the accuracy of each method and the second was used to confirm the testing time of each method.

3.3.1. Comparison Performance

In this part, the raw data were trained and tested with a few neural network methods [38]. For testing the comparison methods, the best method was the Medium Neural Network, whose accuracy was 99.0%. The poorest one was the Narrow Neural Network, whose accuracy was 98.6%. For the proposed method, the accuracy achieved was 99.8%. The accuracy data are shown in Figure 6.

For proposed method, the network output was used to classify tasks (using 'CrossEntropLoss'). The loss curve of changes with training progress shows that the training converged, as is shown in Figure 7.

3.3.2. Speed

In this part, the tested times are shown in Figure 8. From the five comparison methods, the fastest was the Narrow Neural Network, whose testing time was 0.72 ms. The slowest was the Wide Neural Network, whose testing time was 3.67 ms. For the proposed method, the testing time achieved was 0.06 ms.

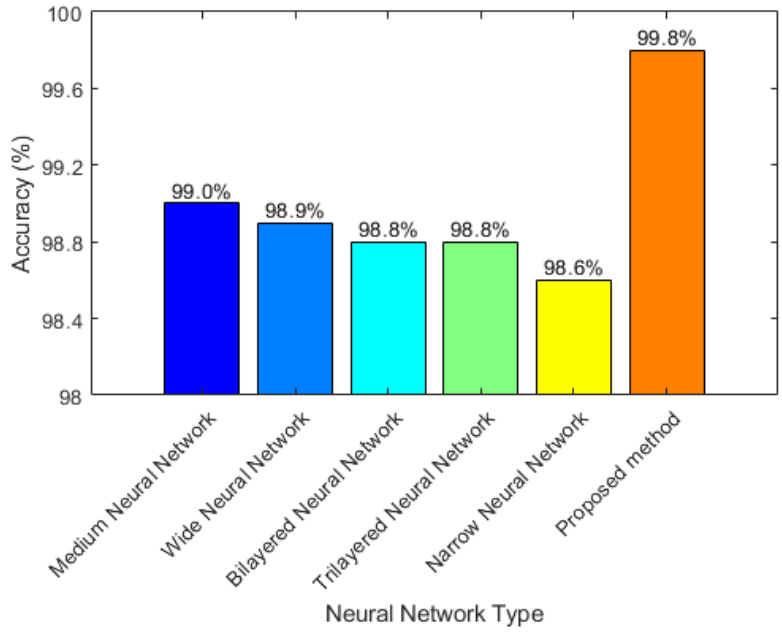


Figure 6. Comparison of accuracy of raw data.

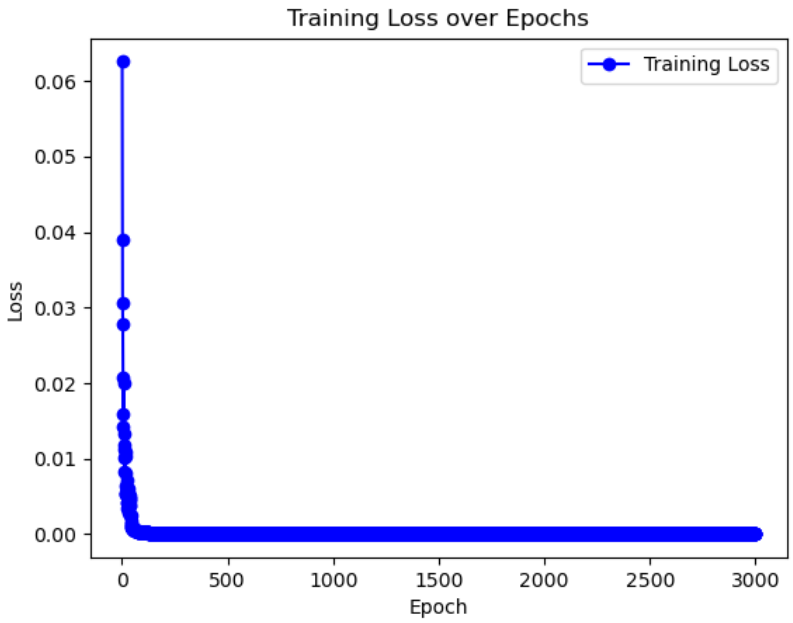


Figure 7. The loss curve of changes with training progress.

For the compared methods, they had a similar accuracy at around 99.0%. However, the testing time of the Wide Neural Network was greater than that of the other methods at 3.67 ms. The main reason for this is that this network has a wide layer. The fastest comparison method was the Narrow Neural Network because of its narrow layer. The average accuracy of the proposed method was 99.8% and the testing time was within 0.06 ms under 5-fold cross-validation. Compared with the other methods, these values represent improvements of 0.8% and 0.66 ms, respectively. It is obvious from the Figure 8 that the proposed method had a better performance than the other methods.

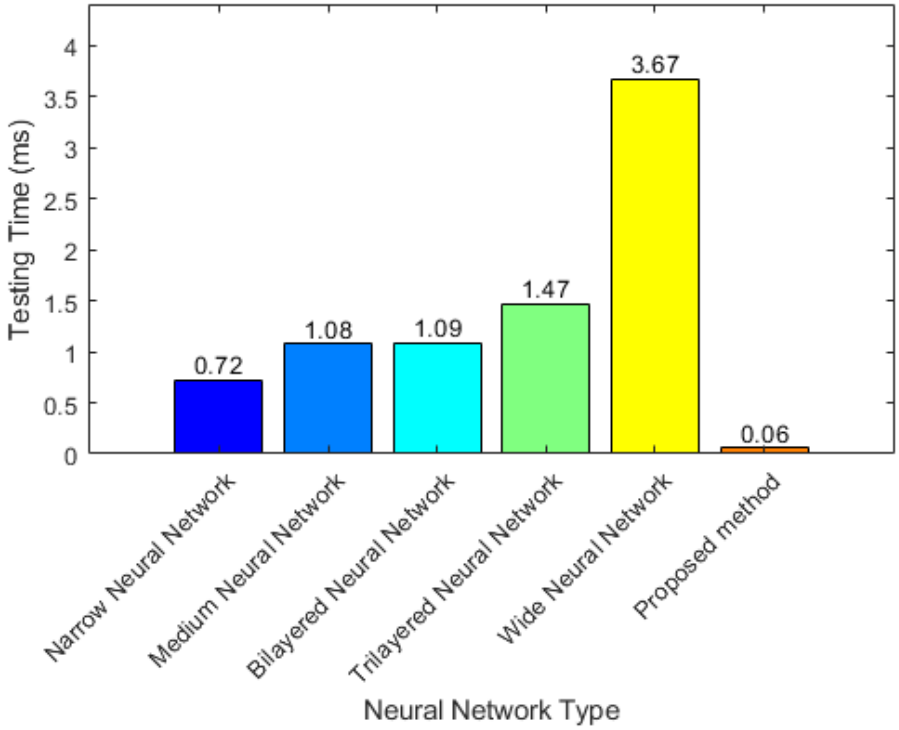


Figure 8. Comparison of testing time of raw data.

4. Discussion

Achieving a practical model requires a small model, with a high accuracy that can function in real time.

4.1. Effect of Different Segments and Epochs

There are four types of segments: the first kind of segment has 8 samples, the second segment has 16 samples, the third segment has 32 samples, and the last segment has 64 samples. In this experiment, different epochs were tested: 20 epochs, 200 epochs, 1000 epochs, 2000 epochs, and 3000 epochs. In addition, the layer size in this experiment was 10. The results are shown in Figure 9.

From Figure 9, the best performance was observed for 32 sample segments and 64 sample segments. Among them, the best results were seen at 3000 epochs. For 32 sample segments, the result was 96.8% and, for 64 sample segments, it was 98.5%. Hence, the segments of 32 and 64 samples at 3000 epochs were tested in the next experiment.

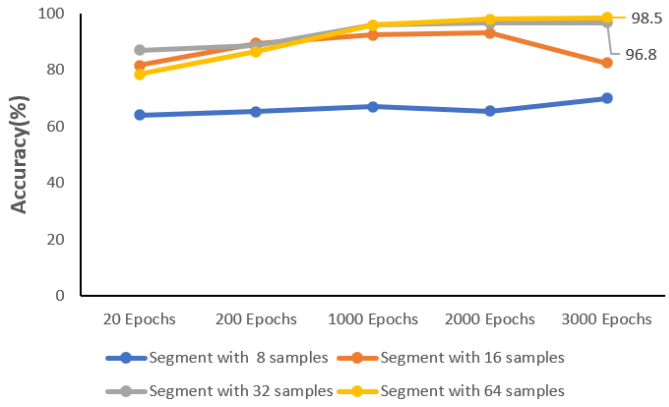


Figure 9. Accuracy of different samples in each segment.

4.2. Effect of Different Number of Nodes

For testing the effect of a different number of nodes in the network, two ablation experiments were introduced. Details are shown in Table 3.

Table 3. Description of different segments.

Samples in Each Segment	Fault Type	Training Segments	Testing Segments
32	Normal	19,200	4800
	Fault	19,200	4800
64	Normal	9600	2400
	Fault	9600	2400

In the first experiment, each of these had 2400 segments, and each segment had 32 samples. Among them, 5-fold cross-validation was applied. Moreover, 19,200 segments were training segments, and the remaining 4800 segments were test segments.

For the second experiment, each of these had 1200 segments, and each segment had 64 samples. Among them, 5-fold cross-validation was applied. Moreover, 9600 segments were training segments, and the remaining 2400 segments were test segments.

In this part, testing was performed with different numbers of nodes, i.e., 5, 10, 15, and 20, in the fully connected layer and at 3000 epochs to validate the model. There were five groups. The final model was obtained through training data and was tested with the remaining data from this model.

As shown in Figure 10, the best performance was obtained for a 20-layer ANN, which achieved 99.8% for 32 samples and 99.7% for 64 samples with a model size of around 5 kb. Therefore, the next step was to confirm the testing time so that the experiment could achieve real-time detection.

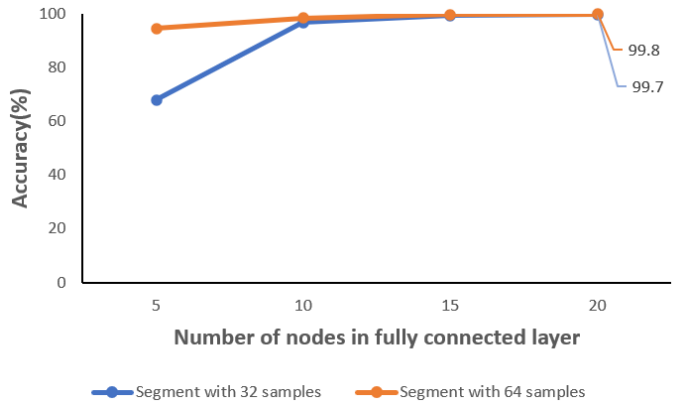


Figure 10. Accuracy under different numbers of nodes in the fully connected layer.

4.3. Experiment of Speed

In this part, the length of the layer was 20 and the epoch was 3000. The testing time was tested on different devices: a desktop computer and the Raspberry Pi. The results are shown in Table 4.

Table 4. Test results and time of raw data on Raspberry Pi and desktop computer.

Samples of Segment	Devices	Accuracy	Testing Time	Sampling Time
32	Desktop	99.8%	0.009 ms	2.5 ms
	Raspberry Pi	99.8%	0.059 ms	
64	Desktop	99.7%	0.005 ms	1.25 ms
	Raspberry Pi	99.7%	0.028 ms	

On the Raspberry Pi, for 64 samples of segments, the testing data were divided into 2400 segments, and each segment had 64 points.

The proposed system can efficiently process the sensor data and performs rapid analysis and prediction within 0.059 milliseconds per data segment. The experimental results demonstrate that the model achieves a 99.8% accuracy in detecting wind turbine bearing faults within milliseconds of their occurrence.

Regarding the 32 samples per segment, the testing dataset was divided into 4800 individual segments, with each segment consisting of 32 samples. In terms of system performance, the proposed approach exhibits efficient data processing capabilities, enabling swift analysis and prediction within a remarkable timeframe of 0.028 milliseconds per data segment. The experimental findings underscore the model's exceptional accuracy, with a reported 99.7% detection rate for wind turbine bearing faults within milliseconds of their manifestation.

On the desktop, for 64 samples of segments, the testing data were divided into 2400 segments, and each segment had 64 samples.

The proposed system can efficiently process the sensor data and performs rapid analysis and prediction within 0.009 milliseconds per data segment. The experimental results demonstrate that the model achieves a 99.8% accuracy in detecting wind turbine bearing faults within milliseconds of their occurrence.

In the case of segments consisting of samples, the testing dataset was partitioned into 4800 individual segments, with each segment containing 32 samples. The proposed system demonstrates notable efficiency in processing sensor data, facilitating swift analysis and prediction within an impressive timeframe of 0.005 milliseconds per data segment. The experimental results substantiate the model's remarkable performance, yielding a detection accuracy of 99.7% in identifying wind turbine bearing faults within milliseconds of their manifestation.

Both the Raspberry Pi and the desktop achieved identical outcomes by utilizing the proposed method. However, the testing times varied. The Raspberry Pi was significantly slower than the PC when it came to test speeds. The proposed method was significantly better, with respect to accuracy and execution time, compared to the comparison networks. The proposed method demonstrated superior performance and aligns with the specific requirements. Multiple models were chosen based on our requirements, i.e., for rapid speed, we selected 32 samples per segment, and for high precision, we chose 64 samples per segment. Each model was approximately 5 kb in size.

5. Conclusions

This study developed a machine learning-based model on a Raspberry Pi to detect wind turbine bearing faults in real time. The model was designed and trained on a desktop computer due to its higher performance. Then, real-time implementation was achieved by running the model on a Raspberry Pi for real-time wind turbine bearing fault detection. The experimental results demonstrate that the model achieves a high accuracy and rapid detection of faults within milliseconds of their occurrence. The model achieved an accuracy rate of 99.8% and the testing time was 0.059 ms, indicating its effectiveness and precision in detecting wind turbine bearing faults.

This study demonstrates the practical implications and potential applications of real-time fault detection in wind turbines using a neural network model. The model provides significant accuracy in quickly identifying bearing faults and providing immediate predictions during fault detection, resulting in reduced maintenance costs, increased turbine availability, and improved overall efficiency. By being able to identify faults during turbine operation in a timely manner, the development of this neural network model facilitates early warning and rapid response, minimizing downtime and associated maintenance costs. Ultimately, real-time fault detection ensures reliable turbine operation and increased availability.

The findings of this study are significant for promoting renewable energy generation and reducing reliance on fossil fuels. The model improves operational efficiency and

reliability by enabling the real-time detection of wind turbine bearing faults, minimizing energy waste. This facilitates the wider adoption of renewable energy sources, reducing dependence on finite fossil fuel resources and promoting sustainable energy development and environmental protection.

In the future, many works will require enhancement, and network design optimization will be of paramount importance. This model can only be used on specific models of wind turbines and is restricted to the generator bearings of these turbines. Future research must establish a multi-channel, multi-type, and multi-scenario detection model that can be further optimized to improve the generalizability of models.

Continued innovation and improvement in this area will drive renewable energy generation, reducing dependence on fossil fuels and paving the way for a cleaner, more sustainable energy future.

Author Contributions: Conceptualization, H.M., T.W., R.Q. and F.Z.; methodology, H.M. and T.W.; validation, T.W.; formal analysis, T.W. and R.Q.; investigation, T.W., R.Q. and F.Z.; resources, T.W.; data curation, T.W.; writing—original draft preparation, T.W., R.Q. and F.Z.; writing—review and editing, T.W., H.M. and A.K.N.; supervision, H.M. and T.W.; project administration, H.M. and T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Royal Society award (number IEC\NSFC\223294) to Professor Asoke K. Nandi.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to commercial privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. IEA. World Energy Outlook 2021. International Energy Agency. 2021. Available online: https://www.iea.org/reports/world-energy-outlook-2021 (accessed on 1 March 2024).
2. Ma, J.; Yuan, Y.; Chen, P.; Sitahong, A. Spatiotemporal Attention-Based Long Short-Term Memory Auto-Encoder Network for Early Fault Detection of Wind Turbine Generators. 14 November 2022. Available online: https://www.researchsquare.com/article/rs-2206291/v1 (accessed on 3 April 2024).
3. Liao, Y.-H.; Wang, L.; Yan, Y. Instantaneous rotational speed measurement of wind turbine blades using a marker-tracking method. In Proceedings of the 2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Ottawa, ON, Canada, 16–19 May 2022; pp. 1–5.
4. Wang, M.-H.; Hsieh, C.-C.; Lu, S.-D. Fault diagnosis of wind turbine blades based on chaotic system and extension neural network. *Sens. Mater.* **2021**, *33*, 2879–2895. [CrossRef]
5. Gao, Z.; Liu, X. An overview on fault diagnosis, prognosis and resilient control for wind turbine systems. *Processes* **2021**, *9*, 300. [CrossRef]
6. Liu, W.; Yin, R.; Zhu, P. Deep learning approach for sensor data prediction and sensor fault diagnosis in wind turbine blade. *IEEE Access* **2022**, *10*, 117225–117234. [CrossRef]
7. Pozo, F.; Vidal, Y.; Serrahima, J.M. On real-time fault detection in wind turbines: Sensor selection algorithm and detection time reduction analysis. *Energies* **2016**, *9*, 520. [CrossRef]
8. Zhang, F.; Sun, W.; Wang, H.; Xu, T. Fault diagnosis of a wind turbine gearbox based on improved variational mode algorithm and information entropy. *Entropy* **2021**, *23*, 794. [CrossRef] [PubMed]
9. Ruiz, M.; Mujica, L.E.; Alferez, S.; Acho, L.; Tutiven, C.; Vidal, Y.; Rodellar, J.; Pozo, F. Wind turbine fault detection and classification by means of image texture analysis. *Mech. Syst. Signal. Process* **2018**, *107*, 149–167. [CrossRef]
10. Tang, M.; Chen, Y.; Wu, H.; Zhao, Q.; Long, W.; Sheng, V.S.; Yi, J. Cost-sensitive extremely randomized trees algorithm for online fault detection of wind turbine generators. *Front. Energy Res.* **2021**, *9*, 686616. [CrossRef]
11. Peng, H.; Zhang, H.; Fan, Y.; Shangguan, L.; Yang, Y. A review of research on wind turbine bearings' failure analysis and fault diagnosis. *Lubricants* **2022**, *11*, 14. [CrossRef]
12. Tian, H.; Fan, H.; Feng, M.; Cao, R.; Li, D. Fault diagnosis of rolling bearing based on hpso algorithm optimized cnn-lstm neural network. *Sensors* **2023**, *23*, 6508. [CrossRef] [PubMed]
13. Khorram, A.; Khalooei, M.; Rezghi, M. End-to-end CNN+ LSTM deep learning approach for bearing fault diagnosis. *Appl. Intell.* **2021**, *51*, 736–751. [CrossRef]

14. Zhang, F.; Zhu, Y.; Zhang, C.; Yu, P.; Li, Q. Abnormality detection method for wind turbine bearings based on CNN-LSTM. *Energies* **2023**, *16*, 3291. [CrossRef]
15. Xu, Z.; Li, C.; Yang, Y. Fault diagnosis of rolling bearing of wind turbines based on the variational mode decomposition and deep convolutional neural networks. *Appl. Soft. Comput.* **2020**, *95*, 106515. [CrossRef]
16. Xu, Z.; Mei, X.; Wang, X.; Yue, M.; Jin, J.; Yang, Y.; Li, C. Fault diagnosis of wind turbine bearing using a multi-scale convolutional neural network with bidirectional long short term memory and weighted majority voting for multi-sensors. *Renew. Energy* **2022**, *182*, 615–626. [CrossRef]
17. Cao, L.; Qian, Z.; Zareipour, H.; Huang, Z.; Zhang, F. Fault diagnosis of wind turbine gearbox based on deep bi-directional long short-term memory under time-varying non-stationary operating conditions. *IEEE Access* **2019**, *7*, 155219–155228. [CrossRef]
18. Khan, P.W.; Yeun, C.Y.; Byun, Y.C. Fault detection of wind turbines using SCADA data and genetic algorithm-based ensemble learning. *Eng. Fail. Anal.* **2023**, *148*, 107209. [CrossRef]
19. Zhao, D.; Wang, T.; Chu, F. Deep convolutional neural network based planet bearing fault classification. *Comput. Ind.* **2019**, *107*, 59–66. [CrossRef]
20. Wang, Z.; Zhu, J.; Gu, J.; Hu, J.; Zhou, B.; Zhao, J. Fault Diagnosis of Wind Turbine Bearing on SATLBO-MLP. In Proceedings of the 2022 China Automation Congress (CAC), Xiamen, China, 25–27 November 2022.
21. Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Luo, K.; Zhang, J. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. In *Proceedings of the IEEE*; IEEE: Piscataway, NJ, USA, 2019; Volume 107, pp. 1738–1762. [CrossRef]
22. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
23. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
24. Lipton, Z.C.; Berkowitz, J.; Elkan, C. A critical review of recurrent neural networks for sequence learning. *arXiv* **2015**, arXiv:1506.00019.
25. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef]
26. Fu, J.; Chu, J.; Guo, P.; Chen, Z. Condition monitoring of wind turbine gearbox bearing based on deep learning model. *IEEE Access* **2019**, *7*, 57078–57087. [CrossRef]
27. Asmuth, H.; Korb, H. WakeNet 0.1-A simple three-dimensional wake model based on convolutional neural networks. *J. Phys. Conf. Ser.* **2022**, *2265*, 22066. [CrossRef]
28. Xie, Y.; Zhao, J.; Qiang, B.; Mi, L.; Tang, C.; Li, L. Attention mechanism-based CNN-LSTM model for wind turbine fault prediction using SSN ontology annotation. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 6627588. [CrossRef]
29. Kruse, R.; Mostaghim, S.; Borgelt, C.; Braune, C.; Steinbrecher, M. Multi-layer perceptrons. In *Computational Intelligence: A Methodological Introduction*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 53–124.
30. He, J.; Li, L.; Xu, J.; Zheng, C. ReLU deep neural networks and linear finite elements. *arXiv* **2018**, arXiv:1807.03973.
31. Bao, M.H. *Micro Mechanical Transducers: Pressure Sensors, Accelerometers and Gyroscopes*; Elsevier: Amsterdam, The Netherlands, 2000.
32. Denishev, K.H.; Petrova, M.R. Accelerometer design. In Proceedings of the Electronics 2007, Sozopol, Bulgaria, 19–21 September 2007; pp. 159–164.
33. Vetelino, J.; Reghu, A. *Introduction to Sensors*; CRC Press: Piscataway, NJ, USA, 2017.
34. Park, J.; Mackay, S. *Practical Data Acquisition for Instrumentation and Control Systems*; Newnes: Oxford, UK, 2003.
35. Emilio, M.D.P. *Data Acquisition Systems*; Springer: Cham, Switzerland, 2013.
36. Severance, C. Eben upton: Raspberry pi. *Computer* **2013**, *46*, 14–16. [CrossRef]
37. Pajankar, A. *Raspberry Pi Image Processing Programming: With NumPy, SciPy, Matplotlib, and OpenCV*; Apress: New York, NY, USA, 2022.
38. Selvi, M.S.; Rani, S.J. Classification of admission data using classification learner toolbox. *J. Phys. Conf. Ser.* **2021**, *1979*, 12043. [CrossRef]

A.3 Published work 3: Fault Detection of Wheelset Bearings through Vibration-Sound Fusion Data Based on Grey Wolf Optimizer and Support Vector Machine

technologies

MDPI

Article

Fault Detection of Wheelset Bearings through Vibration-Sound Fusion Data Based on Grey Wolf Optimizer and Support Vector Machine

Tianhao Wang [1,*], Hongying Meng [1,*], Fan Zhang [2] and Rui Qin [3]

[1] Department of Electronic and Electrical Engineering, Brunel University London, London UB8 3PH, UK
[2] School of Design, Southwest Jiaotong University, Chengdu 610031, China; fan.zhang@swjtu.edu.cn
[3] School of Computing and Mathematical Sciences, University of Leicester, Leicester LE1 7RH, UK; rq20@leicester.ac.uk
* Correspondence: 1815203@brunel.ac.uk (T.W.); hongying.meng@brunel.ac.uk (H.M.)

Abstract: This study aims to detect faults in wheelset bearings by analyzing vibration-sound fusion data, proposing a novel method based on Grey Wolf Optimizer (GWO) and Support Vector Machine (SVM). Wheelset bearings play a vital role in transportation. However, malfunctions in the bearing might result in extensive periods of inactivity and maintenance, disrupting supply chains, increasing operational costs, and causing delays that affect both businesses and consumers. Fast fault identification is crucial for minimizing maintenance expenses. In this paper, we proposed a new integration of GWO for optimizing SVM hyperparameters, specifically tailored for handling sound-vibration signals in fault detection. We have developed a new fault detection method that efficiently processes fusion data and performs rapid analysis and prediction within 0.0027 milliseconds per data segment, achieving a test accuracy of 98.3%. Compared to the SVM and neural network models built in MATLAB, the proposed method demonstrates superior detection performance. Overall, the GWO-SVM-based method proposed in this study shows significant advantages in fault detection of wheelset bearing vibrations, providing an efficient and reliable solution that is expected to reduce maintenance costs and improve the operational efficiency and reliability of equipment.

Keywords: support vector machine; grey wolf optimizer; bearing fault detection; fusion data

Citation: Wang, T.; Meng, H.; Zhang, F.; Qin, R. Fault Detection of Wheelset Bearings through Vibration-Sound Fusion Data Based on Grey Wolf Optimizer and Support Vector Machine. *Technologies* **2024**, *12*, 144. https://doi.org/10.3390/technologies12090144

Academic Editor: Pietro Zanuttigh

Received: 17 July 2024
Revised: 19 August 2024
Accepted: 27 August 2024
Published: 28 August 2024

1. Introduction

Bearings play a crucial role in transportation, with their operational safety and reliability directly impacting logistics efficiency and economic benefits. The wheelset bearings are key components that bear significant loads and operate in complex environments. Over extended periods of use, these bearings are prone to various faults. Therefore, timely and effective detection of wheelset bearings faults is essential for ensuring the safety of transportation.

The primary objective of bearing fault diagnostics is to detect probable defects by examining diverse data. Vibration data is a widely utilized technique for diagnosing bearing failures, as these faults generally result in anomalous vibration characteristics. There are three predominant analysis methods: time domain analysis, frequency domain analysis, and time-frequency analysis. Time domain analysis involves the examination of the time waveform to identify impact signals and periodic components. The statistical properties, such as mean, variance, peak value, kurtosis, and skewness, can potentially reveal changes in the vibration signal [1]. The analysis typically involves computing these features from segmented windows of the vibration signal and monitoring their trends over time. Substantial departures from baseline values can suggest the existence and intensity of bearing faults [2–4]. Time domain analysis is a straightforward and efficient method

for detecting bearing faults; however, it may not offer as much comprehensive diagnostic information as frequency domain techniques [2]. Frequency domain analysis involves applying a Fourier transform to convert signals in the time domain into signals in the frequency domain in order to identify and analyze certain frequency components [5]. Chen et al. [6] introduced power function-based Gini indices II and III (PFGI2 and PFGI3), and through mathematical derivation and experimental validation using envelope analysis in the frequency domain, demonstrated their superior sparsity quantification capabilities and fault feature characterization performance in bearing condition monitoring. Power Spectral Density (PSD) is a measure of the power distribution of a signal over different frequencies; it displays the amplitude of different frequency components and is commonly used to detect specific defects in bearings, such as defects in the outer ring, inner ring, or rolling elements [7]. Chen et al. [8] proposed two new blind deconvolution methods using the modified smoothness index (MSI) in the time and frequency domains for squared envelope applications, effectively enhancing sparse features for rolling bearing fault diagnosis and demonstrating excellent diagnostic performance and robustness in experiments. Time-frequency analysis involves the application of techniques that incorporate both time and frequency data, such as the Short-Time Fourier Transform (STFT) and Wavelet Transform [9,10]. These methods are able to capture the transitory properties of a signal with more accuracy [11].

Additionally, sound signal analysis is becoming more crucial in diagnosing bearing faults, as variations in sound signals might indicate changes in the operating conditions of the bearing. The conventional techniques used are analogous to the analysis of vibration data, encompassing time domain analysis, frequency domain analysis, and time-frequency domain analysis. Two more methods exist: sound pressure level (SPL) analysis and sound signature recognition. SPL analysis is an effective technique for diagnosing bearing faults by analyzing sound emissions from bearings. The main sources of bearing noise are vibrations from the inner ring and rolling elements (balls or rollers) and as a bearing enters the failure stage, there is a rise in SPL of 12–16 dB over the baseline level, accompanied by a change in sound quality [12,13]. SPL analysis includes three techniques: time waveform analysis, frequency spectrum analysis and time-frequency domain analysis [2,13]. Sound signature recognition in bearing fault detection involves utilizing sound signal analysis techniques to identify and diagnose bearing faults. This approach captures and analyzes the sound signals generated by bearings during operation, identifying abnormal patterns and features that indicate the bearing's health status. Advanced signal processing techniques such as Fast Fourier Transform (FFT) [14], Wavelet Transform [10], Empirical Mode Decomposition [15], and Hilbert Transform are commonly used to extract fault features from noisy sound data [16–20].

Nevertheless, there are limitations when it comes to evaluating individual vibration signals or sound signals. For example, the presence of machinery might influence vibrations, making it difficult to discern certain defect features. In a similar manner, background noise has the potential to disrupt signals by concealing important fault characteristics. In data fusion, vibration and sound signals provide complementary information about machine condition, and data fusion has the potential to greatly enhance the effectiveness and dependability of bearing fault diagnosis systems, offering robust assistance for equipment preventative maintenance and fault prediction [21–24].

To enhance the accuracy and reliability of bearing faults detection systems, research investigations highlight the significance of data fusion approaches, which involve merging information from multiple sensors. Wan et al. [25] proposed a fusion multiscale convolutional neural network (F-MSCNN) method that processes raw sound and vibration signals to achieve high accuracy and stable fault diagnosis of rolling bearings under varying operating speeds. Shi et al. [26] proposed a two-stage sound-vibration signal fusion algorithm that combines and weights fault features from multiple sound measurement points, extracts features using empirical mode decomposition and kurtosis superposition, and then unifies sampling frequencies to fuse sound and vibration signals again, achieving weak fault detec-

tion in rolling bearings. This method significantly improves fault feature detection accuracy and signal-to-noise ratio, aiding in the status monitoring of bearing systems. Duan et al. [27] provided a comprehensive review of multi-sensor information fusion for rolling bearings, highlighting the significance of combining data from diverse sensors for improved fault diagnosis capabilities. Wang et al. [28] conducted a study on bearing fault diagnosis using vibro-sound data fusion and a 1D-CNN network, demonstrating the benefits of integrating vibration and sound information for enhanced fault detection. Gu et al. [29] introduced an enhanced SE-ResNet sound-vibration fusion method for rolling bearing fault diagnosis, integrating various techniques to effectively process sound-vibration signals. By integrating vibration and sound data, a comprehensive method for detecting bearing faults is achieved. This approach combines the advantages of both signal types, resulting in a more precise and detailed depiction of the system's status. Researchers have successfully built sophisticated models that incorporate vibration and sound data using modern computational approaches such as deep learning, feature fusion, and adaptive signal processing. These models are used for precise problem identification in bearing systems. These studies highlight the significance of data fusion approaches in utilizing the combined benefits of vibration and sound inputs to improve the accuracy and effectiveness of bearing defect detection systems. However, these deep-learning-based methods need large datasets for training and they are not easily implemented for real-time detection.

In this study, we introduce a novel approach by integrating the Grey Wolf Optimizer (GWO) [30] with a Support Vector Machine (SVM) [31] to optimize hyperparameters, specifically tailored for real-time analysis of vibration-sound fusion data. Yan et al. proposed GWO-SVM for smart emotion recognition, they used the Radial Basis Function (RBF) kernel of SVM and achieved high accuracy in their research [32]. We extend their method with various SVM kernels and provide rapid failure detection by preprocessing fusion data from vibrations and sounds. Data segmentation facilitated analysis, enabling the model to generate predictions at a remarkable speed of 0.0027 milliseconds per segment. In addition, the linear SVM model that was fine-tuned using GWO achieved a testing accuracy of 98.3%, outperforming the SVM and neural network models built in MATLAB. Furthermore, this model demonstrated significant efficiency in runtime assessments, making it extremely suitable for real-world settings. The proposed GWO-SVM model shows advantages in detecting defects in wheelset bearings. The model's capacity to generate real-time predictions and offer a comprehensive evaluation of the bearing's condition can greatly diminish maintenance expenses and enhance the accessibility and effectiveness of wheelset bearings. This study underscores the potential of integrating advanced optimization algorithms with machine learning techniques to enhance fault detection capabilities, ultimately contributing to more robust and efficient transportation systems.

2. Methodology

The list of abbreviations are shown in Table 1:

Table 1. Description of abbreviations.

Abbreviations	Description
CNN	Convolutional Neural Network
FFT	Fast Fourier Transform
GWO	Grey Wolf Optimizer
MFCC	Mel-Frequency Cepstral Coefficients
PCA	Principal Component Analysis
PSD	Power Spectral Density
RBF	Radial Basis Function
SPL	Sound Pressure Level
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine

2.1. Overview of the Proposed Method

An overview of the proposed method is shown in Figure 1. The vibration signal is collected by accelerometers, which are located on the axle box cover in the bearing area at the end of the wheelset; and the sound signal is collected by microphone located on both sides of the bearing. This data set offers the vibration and sound data to efficiently identify bearing various faults.

Figure 1a shows the wheelset bearing. As shown in Figure 1b,c, FFT is utilized for analyzing vibration signals [14], whereas the Mel-Frequency Cepstral Coefficients (MFCC) are employed for analyzing sound signals [33]; split the data into different segments and then combine and integrate the characteristics of the two datasets by simply concatenating them; the model possesses the ability to rapidly examine every individual segment. The final stage involved the trained model making predictions for the score of each segment, as shown in Figure 1d.

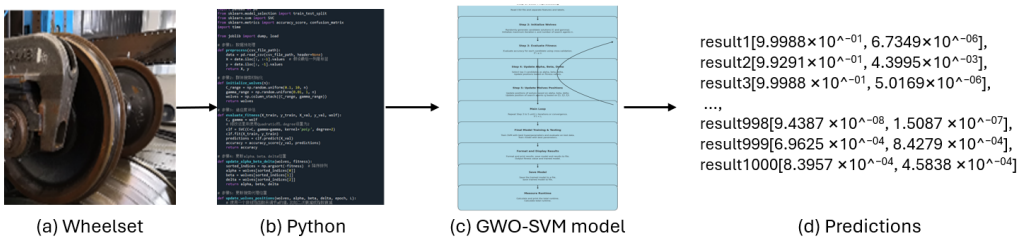


Figure 1. Overview of the system: (**a**) wheelset; (**b**) modeling in python language; (**c**) GWO-SVM model; and (**d**) predictions.

2.2. The Description of the GWO-SVM Model

SVM have been widely applied for fault detection and diagnosis of bearings in rotating machinery. Pule et al. [34] proposed a method using principal component analysis (PCA) and SVM to achieve 97.4% accuracy in diagnosing bearing faults under varying speeds using vibration analysis. Yang et al. [35] introduced a triplet embedding-based method for classifying small sample rolling bearing fault datasets, achieving superior performance in fault diagnosis by using CNN for feature extraction and SVM for classification. Mo et al. [36] proposed a highly accurate (95.3%) and efficient (11.1608-s training time) method for diagnosing petrochemical rotating machinery bearing faults by combining ICEEMDAN-wavelet noise reduction, mutual dimensionless metrics, and MPGA-SVM, with further validation showing 97.1% accuracy on additional datasets.

In this paper, we adopted a fast detection GWO-SVM model by analyzing fusion data from multiple models to assess bearing failure, as shown in Figure 2

Step 1: Data preprocessing—Load the signals and extract their features and annotate their label.

Step 2: Initialize Wolves—Randomly generate the initial positions for the wolf pack, which are the candidate solutions for the SVM hyperparameters C_{SVM} and γ, initialize the maximum number of iterations L and the number of search agents n.

Step 3: Evaluate fitness—Evaluate accuracy for each candidate using cross-validation. If $i \leq n$:

a. Select the current candidate i: Choose the i-th wolf from the pack.
b. Train SVM model using current candidate: (1) Extract the SVM hyperparameters C_{SVM} and γ from the current candidate; (2) Initialize an SVM model with these parameters; (3) Train the SVM model on the training dataset.
c. Evaluate the model on the validation set: (1) Use the validation dataset to predict outcomes; (2) Calculate the accuracy of the predictions, which represents the fitness of the current candidate
d. Record the fitness value of the current candidate: Save the fitness value for the current candidate.

e. Increment the index i: Move to the next candidate ($i = i + 1$).
f. Repeat Step 3: Continue evaluating the next candidate until all candidates are evaluated.

Step 4: Update α_{wolf}, β_{wolf}, and δ_{wolf}.—Based on the fitness values, select the top three candidates as the α_{wolf}, β_{wolf}, and δ_{wolf}.

Step 5: Update Wolves positions—Update the positions of all wolves in the pack using the positions of the α_{wolf}, β_{wolf}, and δ_{wolf}.

Step 6: Main loop—Continue iterating through Steps 3 to 5 until the maximum number of iterations L is reached or the algorithm converges.

Step 7: Final model training and testing—Use the best hyperparameters found during the optimization to train the SVM model on the entire training dataset. Evaluate the trained model on the test dataset.

Step 8: Format and display results—Format the evaluation results and print them. Save the trained model and results to a file.

Step 9: Save model—Save the trained SVM model to a specified file path.

Step 10: Measure runtime—Calculate and print the total runtime and the testing time.

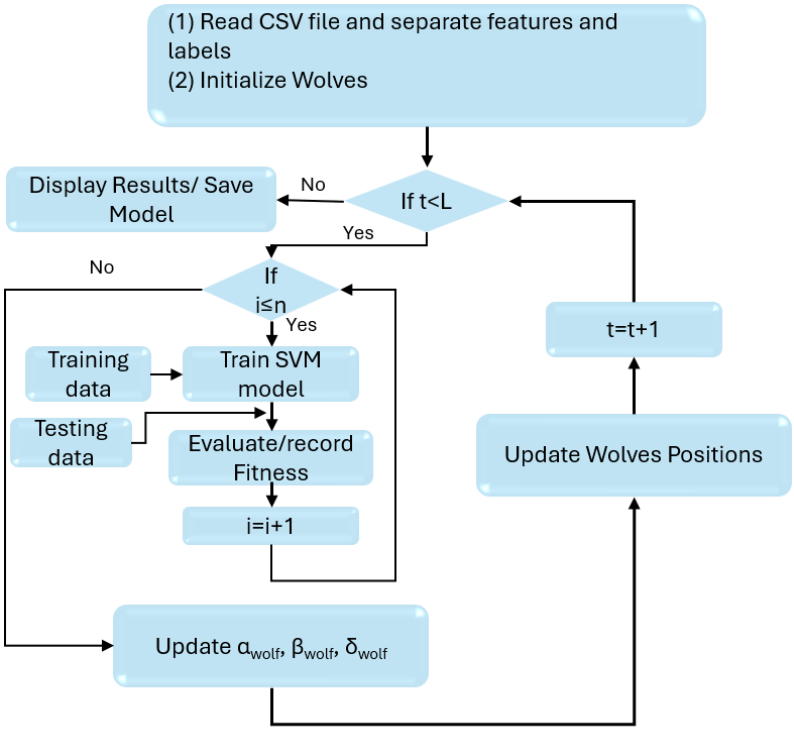


Figure 2. Proposed model.

2.2.1. Basic SVM Formula

The decision function for SVM:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b \tag{1}$$

where α_i is the Lagrange multiplier, y_i is the label, x_i is the support vector, $K(x_i, x)$ is the kernel function, and b is the bias term.

2.2.2. GWO Formulas

Updating parameter a:

$$a = 2 - 2\left(\frac{\text{epoch}}{L}\right)^2 \tag{2}$$

where epoch is the current iteration and L is the maximum number of iterations.

Calculating coefficients A and C_{wolf}:

$$\begin{aligned} A &= 2 \cdot a \cdot r_{wolf_{1-a}} \\ C_{wolf} &= 2 \cdot r_{wolf_2} \end{aligned} \tag{3}$$

where r_{wolf_1} and r_{wolf_2} are random numbers in the range $[0, 1]$.

Updating the positions of the wolves:

$$\begin{aligned} D_\alpha &= |C_{wolf} \cdot X_\alpha - X| \\ D_\beta &= |C_{wolf} \cdot X_\beta - X| \\ D_\delta &= |C_{wolf} \cdot X_\delta - X| \\ X_1 &= X_\alpha - A \cdot D_\alpha \\ X_2 &= X_\beta - A \cdot D_\beta \\ X_3 &= X_\delta - A \cdot D_\delta \\ X_{\text{new}} &= \frac{X_1 + X_2 + X_3}{3} \end{aligned} \tag{4}$$

Among them:

- D represents the distance between the current wolf and the α_{wolf}, β_{wolf}, and δ_{wolf}.
- $X_{\alpha,\beta,\delta}$ is the position of the α_{wolf}, β_{wolf}, and δ_{wolf}, representing the best solution found so far.
- X_{new} is the updated position of the wolf, the elements in X_{new} are essentially combinations of SVM hyperparameters, optimized through the GWO process to find the best parameter settings.

2.2.3. GWO Algorithm Steps

Linear Kernel function:

$$K(x_i, x_j) = x_i \cdot x_j \tag{5}$$

RBF Kernel function:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \tag{6}$$

Polynomial Kernel function:

$$K(x_i, x_j) = (x_i \cdot x_j + r)^d \tag{7}$$

Optimization objective:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C_{SVM} \sum_{i=1}^{n} \xi_i \tag{8}$$

Among them:

- w is the weight vector that determines the hyperplane for classification.
- b is the bias term.
- $\|w\|^2$ is the square norm of the weight vector used to control the complexity of the model.
- C_{SVM} is a regularization parameter used to balance the misclassification of training data and model complexity.
- ξ_i is a slack variable that allows certain samples to be misclassified.

The goal of this formula is to minimize the complexity and training error of the model, thereby improving its generalization ability.

Here, the GWO algorithm is used to optimize the hyperparameters of SVM (such as C_{SVM} and γ) to improve the accuracy and efficiency of fault detection.

Steps for combining GWO:

1. Initialize the positions of the wolves, for linear kernel, train the SVM with the linear kernel using the wolf's position parameter (C_{SVM}); train the SVM with the RBF kernel using the wolf's position parameters (C_{SVM}, γ), and train the SVM with the Polynomial kernel using the wolf's position parameters (C_{SVM}) and coefficient term (r), the degree of polynomial Kernel in this study is confirmed.
2. Evaluate the fitness of each wolf based on the classification accuracy of the SVM.
3. Update the positions of α_{wolf}, β_{wolf}, and δ_{wolf}.
4. Update the positions of the other wolves.
5. Repeat the above steps until the maximum number of iterations is reached.

2.2.4. Summary

The GWO algorithm is used to adjust the SVM hyperparameters C_{SVM}, γ and coefficient term r. These parameters significantly affect the model's performance. The adjustments are as follows:

- C_{SVM} determines the balance between minimizing the error on the training data and reducing the complexity of the model. A larger C_{SVM} value tries to classify every sample correctly, which may lead to overfitting, while a smaller C_{SVM} value allows some misclassifications, potentially improving generalization.
- γ controls the width of the Gaussian kernel. A larger γ value means higher sensitivity to individual data points, making the model focus more on local patterns, while a smaller γ value makes the model consider a broader range of data points.
- r adjusts the influence of higher-order terms in the polynomial kernel.

Using the GWO algorithm, we dynamically adjust the parameters C_{SVM}, γ and coefficient term r in the code to find the parameter combination that achieves the highest classification accuracy on the validation set. The adjusted model is then evaluated on the test set to assess its actual performance. Finally, the model is trained on the training set and validated on the test set, completing the process.

3. Experimental Section

3.1. Experimental Setup and Dataset

The experiment used MATLAB 2022b and Python 3.9.

This fault detection has high accuracy and fast speed.

There are many sensors on the wheelset bearing. The data used in the experiment were the signal of one of the acceleration sensors and microphone sensors. The data were collected from the bearing in our laboratory. The data were collected at 25,600 samples/second.

Wheelset bearings are essential elements utilized in railway vehicles, including trains, subways, and light rail systems. The axles and wheels are supported by them, carrying the full weight of the vehicle. These bearings need to function consistently and dependably in diverse and intricate circumstances. It is shown in Figure 3a.

In order to monitor the health of bearings, sensors are installed on the bearings. It is shown in Figure 3b which is a cross-sectional view of Figure 3a, among them, 1 is tested bearing, 2 is accelerometer, 3 is auxiliary bearing, 4 is microphone sensor, 5 is friction wheel, 6 is motor, 7 is foundation. The height of the microphone sensor position is 300 mm, and the horizontal distance from the test bearing is 500 mm.

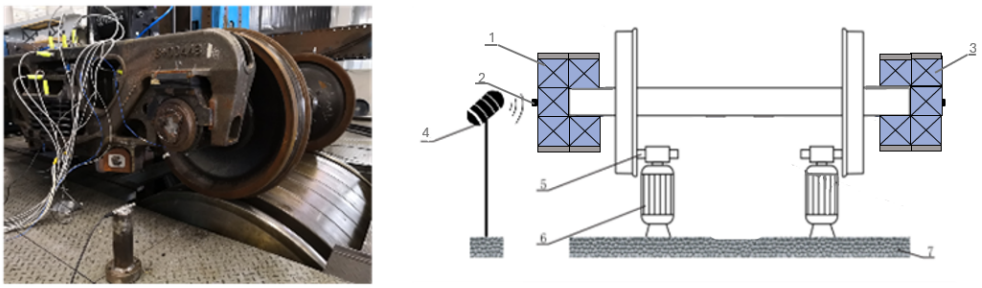


Figure 3. Data collection platform. (**a**) Wheelset bearings; (**b**) sensor installation location.

Accelerometers are predominantly employed to capture the vibration signals emitted by bearings. These signals can indicate the operating condition of the bearings, such as the existence of wear, imbalance, misalignment, or other mechanical problems. Microphone sensors are employed to capture the sound emissions generated by bearings during their functioning. Various sorts of flaws produce unique sound characteristics, and by analyzing these sounds, the state of the bearings can be initially evaluated. By integrating accelerometers and microphone sensors, it is possible to monitor and diagnose the operational condition of the bearings in a more comprehensive manner, enabling the quick identification and treatment of potential problems, ensuring the smooth functioning of the equipment.

Four different states are Normal, Outer raceway scoring (referring to damage on the raceway surface where the rolling elements contact), Outer race scoring (a broader term encompassing damage on any part of the outer race), and Outer raceway pitting (refers to the pitting phenomenon on the raceway surface of the outer ring of a bearing). Each state has 1000 segments, each vibration segment has 16 feature samples after FFT, and each sound segment has 14 feature samples after MFCC, concatenating these features to form a single data segment, hence each fusion segment has 30 feature samples.

The data in each state were evenly partitioned into 4 distinct groups with 250 segments each. The validation was performed using a 4-fold cross-validation approach. The details are shown in Table 2.

Table 2. Dataset was split into training and testing groups for the 4-fold cross-validation.

	Training Group	Testing Group
Validation 1	2, 3, 4	1
Validation 2	1, 3, 4	2
Validation 3	1, 2, 4	3
Validation 4	1, 2, 3	4

3.2. Data Analysis

As shown in Figure 4a, the length of each sample of raw vibration data was 32, and then the Gaussian white noise was added to the vibration data. The reason for adding different levels of noise is that the data collected from our laboratory are too clear in comparison with the data from real-world applications. After FFT, the length of each segment is 16. In Figure 4b, the length of each sample of raw sound data was 768 because the default window length in MATLAB depends on the specified sample rate: round (e.g., the number of frequency sampling $\times$ 0.03) and the frequency sampling is 25,600 samples/second, and then the Gaussian white noise is added to the sound data. After MFCC, the length of each segment is 14.

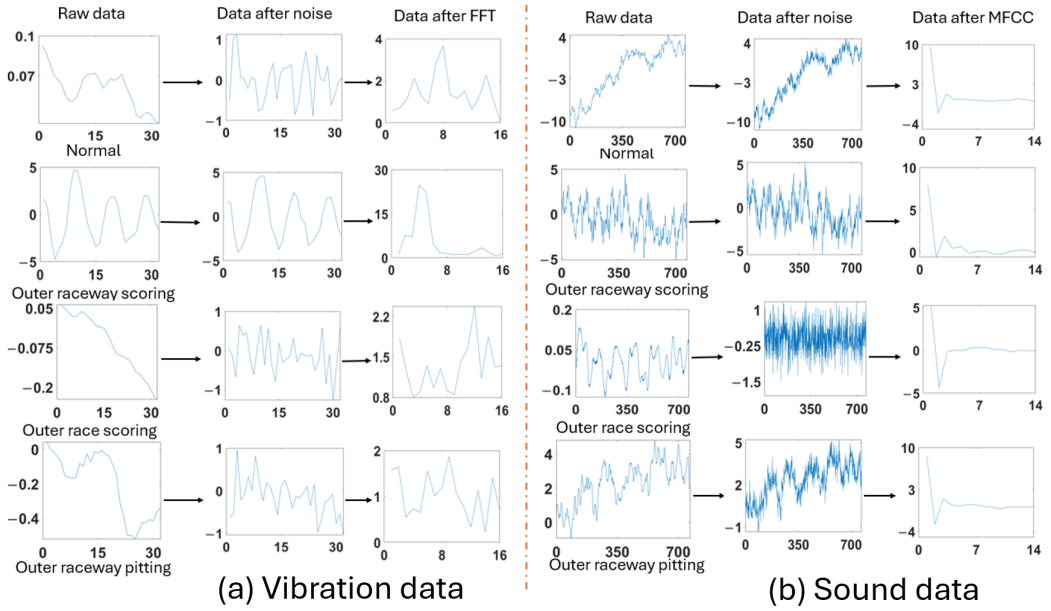


Figure 4. Data spectrum. (**a**) vibration data; (**b**) sound data.

As shown in Figure 5, the length of each sample of fusion data was 30, the difference states at the peak value were obvious.

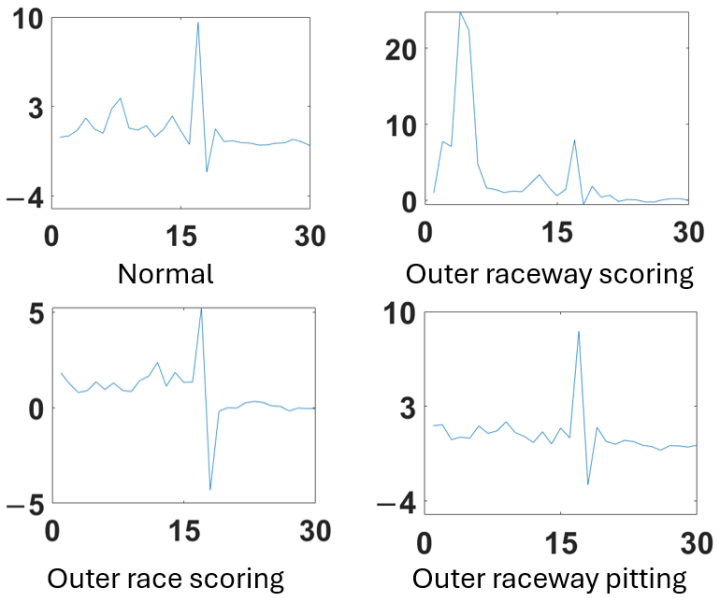


Figure 5. Fusion data.

3.3. Experimental Results

In this experiment, ten comparative networks were introduced to validate the performance and confirm the accuracy of each method. To validate the performance of the methodology, the Neural Network and SVM comparison models in MATLAB were introduced.

3.3.1. Comparison Performance

In this part, the fusion data were trained and tested with a few neural network methods and SVM methods [37]. In this section, the fusion data underwent training and testing using several neural network and SVM algorithms [37]. For neural network methods, the best method was the Trilayered Neural Network, whose accuracy was 97.8%. For SVM methods, the best method was the Quadratic SVM, whose accuracy was 97.9%. The accuracy data are

shown in Figure 6. Among these 10 methods, the best result was demonstrated by Quadratic SVM. Additionally, the model of SVM is simpler than neural network. This means SVM can achieve superior efficiency in data detection compared to neural networks. Hence, the suggested approach opts for SVM and enhances its performance by incorporating GWO.

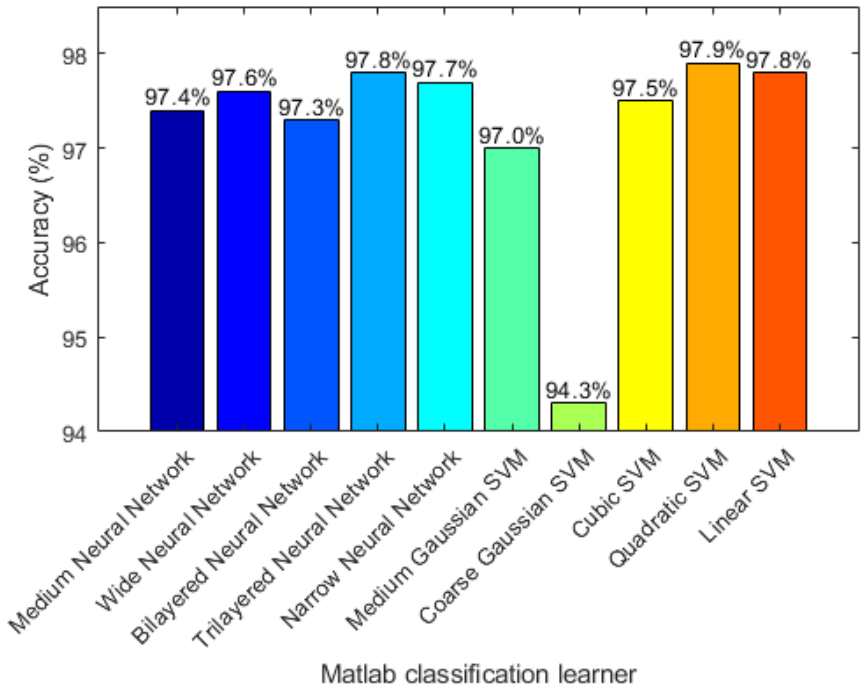


Figure 6. Comparison of accuracy of fusion data in Matlab classification learner.

3.3.2. Training Loss

For the proposed method, the model output was used to classify tasks (using 'SVM'). The training loss exhibits a quick decline in the initial iterations and thereafter reaches a stable state in the middle stages. This suggests that the process of training the model has achieved a state of convergence after 200 iterations, as is shown in Figure 7.

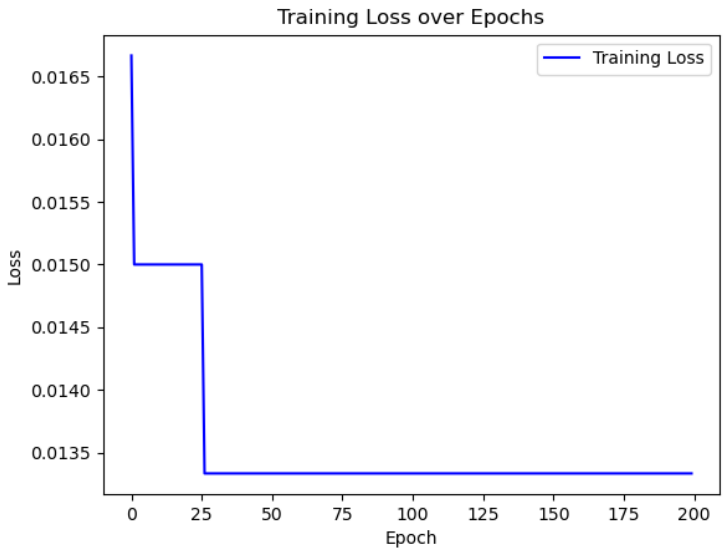


Figure 7. The loss curve of changes with training progress.

4. Discussion

To create a functional model, it is necessary to achieve a high level of accuracy with real-time capability.

4.1. Evaluation of Different Lengths in Each Segment

In this part, five SVM methods in MATLAB are used for different lengths of testing in each segment, including lengths of 256, 128, 64, and 32. The results are shown in Table 3, for the length of 32, the best accuracy achieved is 98% which shows that the length of 32 in each segment contains enough characteristic information.

Table 3. Testing accuracy of different lengths in each segment.

SVM Method	Data Types	Length			
		256	128	64	32
Linear SVM	Vibration	100%	98%	98%	97%
	Sound	99%	100%	99%	97%
Cubic SVM	Vibration	100%	100%	99%	98%
	Sound	99%	100%	98%	97%
Quadratic SVM	Vibration	100%	100%	98%	98%
	Sound	99%	100%	98%	97%
Coarse Guassian SVM	Vibration	100%	99%	99%	97%
	Sound	99%	100%	97%	93%
Medium Gaussian SVM	Vibration	99%	99%	99%	96%
	Sound	99%	100%	100%	98%

4.2. Evaluation of Proposed Method Performance

In this part, the GWO-SVM method initializes the number of wolves as n = 10 and the number of iterations as L = 200; The software executes using a 4-fold cross-validation approach, with each group being repeated 5 times. Following each iteration, the wolves undergo a sorting process based on their fitness values, which allows for the identification of the three wolves with the greatest fitness levels. These wolves are referred to as the alpha, beta, and delta wolves. The alpha wolf is prioritized due to its superior fitness. The ultimate trained model exclusively utilizes the parameters of the alpha wolf.

For vibration data and sound data, as shown in Table 4, the best result in Matlab was Medium Gaussian SVM, whose accuracy was 68% in vibration data and 97.6% in sound data, respectively. The result obtained by the GWO-SVM model using Medium Gaussian SVM was 68.1% and 97.7%, respectively.

Table 4. Description of different segments.

Kernel of SVM	Training Model	Vibration Accuracy	Sound Accuracy	Fusion Accuracy
Linear SVM	Matlab	67.6%	97.4%	97.8%
	GWO-SVM	68%	97.6%	98.3%
Quadratic SVM	Matlab	67.4%	97.2%	97.9%
	GWO-SVM	67.7%	97.3%	98%
Cubic SVM	Matlab	66.8%	95.5%	97.5%
	GWO-SVM	66.8%	97.2%	97.7%
Gaussian SVM	Matlab-Medium Gaussian SVM	68%	97.6%	97%
	Matlab-Coarse Gaussian SVM	67.9%	95.2%	94.3%
	GWO-SVM	68.1%	97.7%	98.1%

For fusion data, a comparison was made between the GWO-SVM in Python and the SVM method in MATLAB. As shown in Table 4, the best result in MATLAB was Quadratic SVM, whose accuracy was 97.9%; The result obtained by the GWO-SVM model using Quadratic SVM was 98%. Additionally, the best result in GWO-SVM was Linear GWO-SVM, whose accuracy was 98.3%.

It is obvious from Table 4 that the proposed method had a better performance than the traditional SVM methods in both vibration, sound and fusion data.

4.3. Evaluation of Testing Speed

This experiment measures the total testing time for all 1000 testing segments and then calculates its average for each segment.

In this part, as shown in Table 5, the testing time was tested with different SVM kernels in Python: linear kernel, gaussian kernel (rbf kernel) and quadratic-cubic kernel (polynomial kernel). The frequency sampling was 25,600 samples/second, the sampling time for each segment of vibration data was 1.25 ms and the sampling time for each segment of sound data was 30 ms; thus, the sampling time for each segment of fusion data was 31.25 ms.

Table 5. Test results and time of fusion data.

Different Kernel of GWO-SVM	Accuracy	Testing Time for Each Segment	Sampling Time
Linear SVM	98.3%	0.0027 ms	31.25 ms
Quadratic SVM	98%	0.0034 ms	
Cubic SVM	97.7%	0.0035 ms	
Gaussian SVM	98.1%	0.016 ms	

The proposed method demonstrates high efficiency in processing fusion data, enabling rapid analysis and prediction with a remarkable speed of 0.0027 milliseconds per data segment. The experimental results indicate that the model attains a 98.3% accuracy in promptly recognizing wheelset bearing defects. The proposed method exhibited superior performance in terms of both accuracy and execution time when compared to the comparative model. The proposed method exhibited exceptional performance and is in line with the specified requirements.

It should be mentioned here, this excellent experimental performance is based on this single case only, where there are only three fault and normal situations under our limited equipment conditions. However, it will be straightforward to extend our method to any other scenarios where more fault and sensors can be used in real situations. Of course, more evaluations are needed.

5. Conclusions

This work developed a GWO-SVM model for real-time identification of defects in wheelset bearings. The model was developed and trained using the Python programming language. Subsequently, the model was executed on a desktop computer to replicate fast wheelset bearing fault detection for possible real-time implementation. The results of the experiment show that the model achieves accuracy and quickly identifies faults within milliseconds of their occurrence. It achieved an accuracy rate of 98.3% with a testing duration of 0.0027 ms proving its effectiveness and precision in detecting wheelset bearing defects.

This work demonstrates the practical ramifications as well as potential uses of real-time identification of faults in wheelset bearings using a GWO-SVM model. The model demonstrates remarkable precision in rapidly detecting bearing defects and promptly making forecasts, leading to decreased maintenance expenses, enhanced bearing accessibility, and improved overall effectiveness. By promptly detecting malfunctions during the

operation of bearings, this model enables early detection and swift action, thereby reducing downtime and decreasing maintenance expenses.

While this study has made notable progress, there remains a substantial amount of effort to enhance the effectiveness and practicality of the suggested approach. Some guidelines for future endeavors include the following: 1. Enhanced data sources: Subsequent studies can explore the inclusion of other sensor data, such as temperature and pressure, to enhance the comprehensiveness and precision of defect identification. 2. Adaptive Algorithm Optimization: In real-world scenarios, researchers investigate adaptive optimization algorithms that allow the model to autonomously modify parameters according to the environment and operating conditions, thereby improving the model's resilience. 3. Long-term Performance Evaluation: Conduct long-term performance evaluations and maintenance cost analyses to verify the economic benefits and sustainability of the proposed method in practical operations. By exploring and improving these future work directions, it is expected that the performance and application of the GWO-SVM method in bearing fault detection can be further enhanced, providing a more reliable and efficient industrial maintenance solution.

Author Contributions: Conceptualization, H.M., T.W., R.Q. and F.Z.; methodology, H.M. and T.W.; validation, T.W.; formal analysis, T.W. and F.Z.; investigation, T.W., R.Q. and F.Z.; resources, T.W.; data curation, T.W.; writing—original draft preparation, T.W., R.Q. and F.Z.; writing—review and editing, T.W. and H.M.; supervision, H.M. and T.W.; project administration, H.M. and T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the Royal Society award (IEC\NSFC\223294).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to commercial privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yadav, O.P.; Pahuja, G.L. Bearing health assessment using time domain analysis of vibration signal. *Int. J. Image Graph. Signal Process.* **2020**, *10*, 27. [CrossRef]
2. NCD.IO. Bearing Fault Detection Vibration Analysis-How to Measure Vibration Frequency Domain Predictive Analysis. Available online: https://ncd.io/blog/bearing-fault-detection-vibration-analysis/ (accessed on 23 June 2023).
3. Altaf, M.; Akram, T.; Khan, M.A.; Iqbal, M.; Ch, M.M.I.; Hsu, C.H. A new statistical features based approach for bearing fault diagnosis using vibration signals. *Sensors* **2022**, *22*, 2012 [CrossRef]
4. Kumar, J.P.; Chauhan, P.S.; Pandit, P.P. Time domain vibration analysis techniques for condition monitoring of rolling element bearing: A review. *Mater. Today Proc.* **2022**, *62*, 6336–6340. [CrossRef]
5. Wang, T.; Qin, R.; Meng, H.; Li, M.; Cheng, M.; Liu, Y. Frequency Domain Feature Extraction and Long Short-Term Memory for Rolling Bearing Fault Diagnosis. In Proceedings of the 2022 International Conference on Machine Learning, Control, and Robotics (MLCR), Suzhou, China, 29–31 October 2022.
6. Chen, B.; Gu, F.; Zhang, W.; Song, D.; Cheng, Y.; Zhou, Z. Power function-based Gini indices: New sparsity measures using power function-based quasi-arithmetic means for bearing condition monitoring. *Struct. Health Monit.* **2022**, *22*, 3677–3706. [CrossRef]
7. Huang, P.; Pan, Z.; Qi, X.; Lei, J. Bearing fault diagnosis based on EMD and PSD. In Proceedings of the 2010 8th World Congress on Intelligent Control and Automation, Jinan, China, 7–9 July 2010; pp. 1300–1304.
8. Chen, B.; Zhang, W.; Song, D.; Cheng, Y.; Gu, F.; Ball, A.D. Squared envelope sparsification via blind deconvolution and its application to railway axle bearing diagnostics. *Struct. Health Monit.* **2023**, *22*, 3637–3658. [CrossRef]
9. Baba, T. Time-frequency analysis using short time Fourier transform. *Open Acoust. J.* **2012**, *5*, 32–38. [CrossRef]
10. Zhang, D. Wavelet transform. In *Fundamentals of Image Data Mining: Analysis, Features, Classification and Retrieval*; Springer: Cham, Switerland, 2019; pp. 35–44.
11. Yu, G. A concentrated time–frequency analysis tool for bearing fault diagnosis. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 371–381. [CrossRef]
12. Zhang, Q.; Yang, J.; An, Q. Noise calculation method for deep groove ball bearing with considering raceway surface waviness and roller size error. *Front. Mech. Eng.* **2018**, *4*, 13. [CrossRef]

13. BearingNews. Diagnosing Bearing Failures Using Ultrasound Spectrum Analysis. 2022. Available online: https://www.bearing-news.com/diagnosing-bearing-failures-using-ultrasound-spectrum-analysis/ (accessed on 26 April 2022).
14. Heckbert, P. Fourier transforms and the fast Fourier transform (FFT) algorithm. *Comput. Graph.* **1995**, *2*, 15–463.
15. 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. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. London. Ser. A Math. Phys. Eng. Sci.* **1998**, *454*, 903–995. [CrossRef]
16. Hiremath, N.; Reddy, D.M. Bearing fault detection using acoustic emission signals analyzed by empirical mode decomposition. *Int. J. Res. Eng. Technol.* **2014**, *3*, 426–431.
17. Grebenik, J.; Zhang, Y.; Bingham, C.; Srivastava, S. Roller element bearing acoustic fault detection using smartphone and consumer microphones comparing with vibration techniques. In Proceedings of the 2016 17th International Conference on Mechatronics-Mechatronika (ME), Prague, Czech Republic, 7–9 December 2016; pp. 1–7.
18. Li, C.; Chen, C.; Gu, X. Acoustic-Based Rolling Bearing Fault Diagnosis Using a Co-Prime Circular Microphone Array. *Sensors* **2023**, *23*, 3050. [CrossRef]
19. Orman, M.; Rzeszucinski, P.; Tkaczyk, A.; Krishnamoorthi, K.; Pinto, C.T.; Sulowicz, M. Bearing fault detection with the use of acoustic signals recorded by a hand-held mobile phone. In Proceedings of the 2015 International Conference on Condition Assessment Techniques in Electrical Systems (CATCON), Bangalore, India, 10–12 December 2015; pp. 252–256.
20. Attal, E.; Asseko, E.O.; Sekko, E.; Sbai, N.; Ravier, P. Study of bearing fault detectability on a rotating machine by vibro-acoustic characterisation as a function of a noisy surrounding machine. In *Surveillance, Vibrations, Shock and Noise*; Institut Supérieur de l'Aéronautique et de l'Espace: Toulouse, France, 2023.
21. Mey, O.; Schneider, A.; Enge-Rosenblatt, O.; Mayer, D.; Schmidt, C.; Klein, S.; Herrmann, H.-G. Condition Monitoring of Drive Trains by Data Fusion of Acoustic Emission and Vibration Sensors. *Processes* **2021**, *9*, 1108. [CrossRef]
22. Kaicheng, Y.; Chaosheng, Z. Vibration data fusion algorithm of auxiliaries in power plants based on wireless sensor networks. In Proceedings of the 2011 International Conference on Computer Science and Service System (CSSS), Nanjing, China, 27–29 June 2011; pp. 935–938.
23. Xuejun, L.; Guangfu, B.; Dhillon, B.S. A new method of multi-sensor vibration signals data fusion based on correlation function. In Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering, Los Angeles, CA, USA, 31 March–2 April 2009; Voulme 6, pp. 170–174.
24. Al-Haddad, L.A.; Jaber, A.A.; Hamzah, M.N.; Fayad, M.A. Vibration-current data fusion and gradient boosting classifier for enhanced stator fault diagnosis in three-phase permanent magnet synchronous motors. *Electr. Eng.* **2024**, *106*, 3253–3268. [CrossRef]
25. Wan, H.; Gu, X.; Yang, S.; Fu, Y. A Sound and Vibration Fusion Method for Fault Diagnosis of Rolling Bearings under Speed-Varying Conditions. *Sensors* **2023**, *23*, 3130. [CrossRef] [PubMed]
26. Shi, H.; Li, Y.; Bai, X.; Zhang, K.; Sun, X. A two-stage sound-vibration signal fusion method for weak fault detection in rolling bearing systems. *Mech. Syst. Signal Process.* **2022**, *172*, 109012. [CrossRef]
27. Duan, Z.; Wu, T.; Guo, S.; Shao, T.; Malekian, R.; Li, Z. Development and trend of condition monitoring and fault diagnosis of multi-sensors information fusion for rolling bearings: A review. *Int. J. Adv. Manuf. Technol.* **2018**, *96*, 803–819. [CrossRef]
28. Wang, X.; Mao, D.; Li, X. Bearing fault diagnosis based on vibro-acoustic data fusion and 1D-CNN network. *Measurement* **2021**, *173*, 108518. [CrossRef]
29. Gu, X.; Tian, Y.; Li, C.; Wei, Y.; Li, D. Improved SE-ResNet Acoustic–Vibration Fusion for Rolling Bearing Composite Fault Diagnosis. *Appl. Sci.* **2024**, *14*, 2182. [CrossRef]
30. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
31. Hearst, M.A.; Dumais, S.T.; Osuna, E.; Scholkopf, B. Support vector machines. *IEEE Intell. Syst. Their Appl.* **1998**, *13*, 18–28. [CrossRef]
32. Yan, X.; Lin, Z.; Lin, Z.; Vucetic, B. A Novel Exploitative and Explorative GWO-SVM Algorithm for Smart Emotion Recognition. *IEEE Internet Things J.* **2023**, *10*, 9999–10011. [CrossRef]
33. Davis, S.; Mermelstein, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoust. Speech Signal Process.* **1980**, *28*, 357–366. [CrossRef]
34. Pule, M.; Matsebe, O.; Samikannu, R. Application of PCA and SVM in fault detection and diagnosis of bearings with varying speed. *Math. Probl. Eng.* **2022**, *2022*, 5266054. [CrossRef]
35. Yang, K.; Zhao, L.; Wang, C. A new intelligent bearing fault diagnosis model based on triplet network and SVM. *Sci. Rep.* **2022**, *12*, 5234. [CrossRef] [PubMed]
36. Mo, C.; Han, H.; Liu, M.; Zhang, Q.; Yang, T.; Zhang, F. Research on SVM-Based Bearing Fault Diagnosis Modeling and Multiple Swarm Genetic Algorithm Parameter Identification Method. *Mathematics* **2023**, *11*, 2864. [CrossRef]
37. Selvi, M.S.; Rani, S.J. Classification of admission data using classification learner toolbox. *J. Phys. Conf. Ser.* **2021**, *1979*, 12043. [CrossRef]