A DEEP AUTO ENCODER WITH REINFORCEMENT LEARNING FOR ROTATING MACHINERY FAULT DIAGNOSIS

PROJECT REPORT

Submitted by

SAFANA F

REG NO : TKM21CSCE06

In partial fulfillment for the award of the degree of

MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Under the guidance of
Dr.Aneesh.G.Nath

Thangal Kunju Musaliar College of Engineering
Kerala

JULY 2023

DECLARATION

We undersigned hereby declare that the project report on **"A Deep Autoencoder With Reinforcement Learning For Rotating Machinery Fault Diaganosis"**,submitted for partial fulfillment of the requirements for the award of the degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under the supervision of **Dr. Aneesh G Nath**,Associate Professor of the Computer Science and Engineering Department,TKMCE.This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources.We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission.We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained.This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

KOLLAM SAFANA.F
7/07/2023

Thangal Kunju Musaliar College of Engineering
Department of Computer Science & Engineering

C E R T I F I C A T E

This is to certify that,this report titled ***A Deep Autoencoder With Reinforcement Learning For Rotating Machinery Fault Diagnosis*** is a bonafide record of the **Project** presented by **SAFANA F(TKM21CSCE06)** under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree, **M.Tech Computer Science & Engineering** in **APJ Abdul Kalam Technological University** .

Project Guide
Dr. Aneesh G Nath
Associate Professor
Computer Science & Engineering

Head Of The Department
Dr. Dimple A Shajahan
Professor
Computer Science & Engineering

Project Coordinator
Dr. Aneesh G Nath
Associate Professor
Computer Science & Engineering

ACKNOWLEDGEMENT

A successful project is a fruitful culmination of efforts by many people,some directly involved and some others indirectly, by providing support and encouragement.Firstly I would like to thank the almighty for giving me the wisdom and grace for making my project a memorable one. I thank him for steering me to the shore of fulfillment under his protective wings.

I express my sincere gratitude to **Dr. T A Shahul Hammed** , Principal of T.K.M College of Engineering for giving me an opportunity to present my poject.I would like to thank **Dr. Dimple A Shajahan**, Professor and Head of the Department,CSE,TKMCE,for her constant support and encouragement throughout the project work.

With a profound sense of gratitude, I would like to express my heartfelt thanks to my guide **Dr. Aneesh.G.Nath**, Associate Professor, CSE, TKMCE, for her expert guidance, cooperation and immense encouragement.I also extend my thanks to the entire faculty and staff of the Department of Computer Engineering,TKMCE, who has encouraged me throughout this work.

I also express my thanks to my loving parents, brother and friends, for their support and encouragement in the successful completion of this project work.

KOLLAM SAFANA.F
7/07/2023

ABSTRACT

Fault diagnosis in rotating machinery plays a crucial role in ensuring operational reliability and safety.Rotating machinery plays a critical role in various industrial applications,but ensuring its reliability and safety is of utmost importance.Fault diagnosis in rotating machinery is a vital task that involves identifying and addressing potential issues to prevent catastrophic accidents and enable effective maintenance.Traditional fault diagnosis methods have certain limitations, such as manual analysis and limited accuracy. In recent years,deep learning techniques have emerged as promising approaches for automating the fault diagnosis process.This study proposes a novel approach for fault diagnosis in rotating machinery by combining deep learning with reinforcement learning.The proposed method leverages a deep auto encoder augmented with reinforcement learning techniques to improve the accuracy and effectiveness of fault diagnosis.The deep auto encoder extracts relevant features by compressing input data into a lower-dimensional representation and reconstructing the original input.This process inherently performs feature extraction, capturing informative characteristics in the encoded layer.Furthermore,reinforcement learning, specifically a deep Q network, is employed to enhance the accuracy of failure mode diagnosis.By continuously interacting with the datasets and learning from the feedback received, the models can improve their diagnostic capabilities and handle compound failures more effectively.The performance of the proposed approach is evaluated using two real-world datasets, namely the CWRU and MAFAULDA datasets, which cover different fault diagnosis and time series analysis scenarios.Various models, including 1D CNN, LSTM, and GRU, are utilized to process the time series data and extract meaningful features.The evaluation metrics used to assess the effectiveness of the trained models include accuracy, precision, recall, and F1 score.Additionally, a confusion matrix and a classification report are generated to provide comprehensive insights into the performance of the models.The results demonstrate that the proposed approach,combining deep learning with reinforcement learning,holds significant potential for accurate fault diagnosis in rotating machinery.

Keywords:*Deep auto encoder, Fault diagnosis, Reinforcement learning, Deep Q network*

Contents

List of Figures

List of Tables

ABBREVATIONS

CNN	Convolutional Neural Network
LSTM	Long Short Term Memory
RL	Reinforcement Learning
GRU	Gated Recurrent Unit
CWRU	Case Western Reserve University
BPNN	Backpropagation neural network
SRF	Sensor Recorded Fault
KDE	Kernal Density Estimation
DFC	Distinctive Frequency Components

Chapter 1

INTRODUCTION

As the core component of the transmission system, rotating machinery is widely used in various mechanical systems.However,the fact that rotating machinery is one of the main sources of failure leads to the requirement of the fault diagnosis for rotating machinery which is crucial to ensure a stable and safe operation of mechanical systems.The most common fault diagnosis methods for bearing and gear, which are the most representative and widely used rotating machinery are based on vibration signal or acoustic signal .Vibration measurement is less affected by environmental noise, while acoustic measurement, as a non-contact measurement method, has a wider application range.However, it is usually necessary to establish different models for the fault diagnosis of different rotating machines, different working conditions, different signals,which is time-consuming and laborious, and the generality and portability of the models are poor.The research on general fault diagnosis framework (GFDF) can simplify the process of developing and deploying fault diagnosis algorithms, and realize one-time development and multi-terminal deployment,offering great engineering significance and practical value.GFDF's main feature lies in the evitable frequency domain transformation and noise reduction, which makes GFDF insensitive to signal type and has high diagnostic efficiency.The experimental results show that GFDF has high diagnostic accuracy and stability for acoustic signals and vibration signals of rolling bearing and planetary gear at different rotating speeds, which proves that GFDF has generality and portability and is potential for the application in other scenes.

1.1 Rotating Machinery Fault Diagnosis

More than 70% of the faults in rotating machinery and equipment are shaft and bearing faults.Bearing faults include inner ring faults, rotor faults, and outer ring faults.The main factors leading to the failure are unreasonable assembly,long-term overload operation, fatigue operation,and shortage of lubricant, which causes various faults in the shaft parts and the bearings of the rotating parts, such as friction, cracks, and eccentricity [2].Therefore, it is necessary to denoise and process the vibration signal collected by the sensor.To diagnose the fault of the rolling shaft of a rotating machine, it is necessary to first collect vibration signal, process signal noise reduction, extract fault characteristic signals,identify and classify fault states,and diagnose fault degrees[3]. Figure 1.1 shows the diagnostic flow of rotating machinery equipment.

Rotating machinery mainly refers to the machinery that can complete specific functions with the help of rotary action. The main vibration faults of rotating machinery include rotor unbalance, rotor misalignment, friction between moving and static parts and looseness of support parts. Rotating machinery constitutes approximately 40% of all machinery operated in the industry. Over a period of time, they are more prone to deterioration and failures in mechanical and electromechanical systems. However, the majority of the studies that appeared in the literature of artificial intelligence (AI)-based fault diagnosis of rotating machinery were dealing only with bearing or gear faults, leaving rotor faults diagnosis least

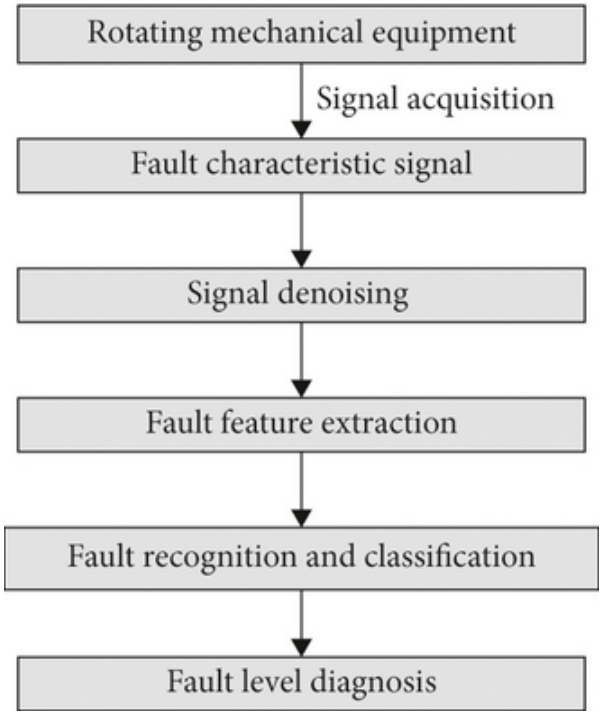

Figure 1.1: Flowchart of fault diagnosis of rotating machinery

addressed . Due to this, the rotor faults diagnosis literature looks fragmentary, lacking the opportunities for exploiting the fault-specific characteristics of the rotor faults and utilizing them in the feature engineering phase of AI to produce significant research improvements in the domain. Among the faults affecting the rotor, structural rotor fault (SRF) is a prevalent and straightforward fault, which includes unbalance (UB), misalignment, and looseness faults. Analyzing SRF is extremely important because it not only creates an unmediated baneful impact on the structural attributes and performance of the affected equipment, but it may cause secondary faults to the surrounding components, such as bearings and gears .

1.2 Bearing fault Diagnosis

The most important parts of rotating machinery are the rolling bearings, and with the development and demand of industry, the workload of most rotating machinery is huge. Periodic inspection and after-the-fact diagnosis are often used for traditional fault diagnosis of rotating machinery, but they are hard to carry out and time consuming for most rotating machinery operating environments.Modern rotary machinery and equipment must be more complicated and precise due to the swift advancement of science and technology. As a result, rolling bearings always function at high speeds, heavy loads,and significant impact forces . Once the error is made, it will unavoidably have negative effects. However,information regarding the machine's working conditions can often be found in the vibration signals recorded from the rolling bearing, making it crucial to correctly identify the data acquired from rolling bearing. Fig.1.2 shows, a rolling-element bearing has an outer-race,an inner race,

a cage, and a ball. This could be realized through various sensors and measured by employ-

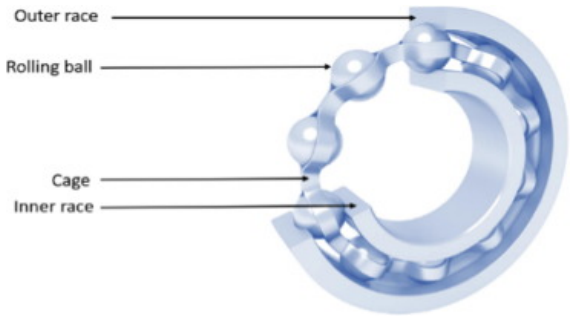

Figure 1.2: Rolling element

ing signal processing methods. Domestic and foreign scholars have carried out researches based on the rolling bearing operation characteristics and fault characteristics information in various forms and different methods. The researches have achieved positive results in the field of rolling bearing fault diagnosis. Three steps such as feature extraction, feature selection, and fault classification, are generally included in fault diagnosis.

Deep learning methods are more efficient and accurate than machine learning techniques because they address the issue from beginning to end .Deep learning-based fault diagnosis has become a popular area of study recently and has had great success in this field [8], [9]. In order to identify rolling bearing faults,Jiang et al. [10] used a deep recurrent neural network. For the purpose of diagnosing a rolling bearing issue,Shao et al. presented a deep belief network [11],[12]. Convolutional neural networks (CNN) are also commonly employed in the fault detection industry. An example of this is Guo et al.'s proposal [13] for a hierarchical adaptive deep convolutional neural network for bearing fault diagnosis.Experts have explored the development of more advanced network models, such as reinforcement learning [28–29], transfer learning [30–32], generative adversarial network [33–35], and ensemble learning [36–37], with the aim of creating more powerful functionalities. Among these approaches, reinforcement learning has gained significant attention from researchers, thanks to its robust ability for self-learning and independent decision-making. In reinforcement learning, a model learns by interacting with the environment and continuously adjusts its behavior to maximize the reward signal received from the environment during training (Sutton and Barto 2018).Unlike supervised machine learning algorithms, which focus on fitting the mapping function between input samples and labels, reinforcement learning aims to learn an optimal policy—a response mechanism that delivers the highest reward. Deep reinforcement learning (DRL), which combines the advantages of automatic feature extraction from deep learning and interactive learning from reinforcement learning, represents a revolutionary advancement in the field of artificial intelligence and shows promise for solving complex real-world problems (Arulkumaran et al., 2017).Reinforcement learning has been successfully employed by the Google DeepMind team to discover the optimal structure of neural networks .For instance, in the context of diagnosing rolling bearing faults, Wang et al. proposed a reinforcement neural architecture search approach [38]. This reinforcement

learning approach reduces the reliance on prior knowledge, eliminates the need to manually design intricate deep network models,and introduces a fresh perspective to research.

1.3 Reinforcement learning

Reinforcement learning is a learning framework where an agent interacts with an environment to achieve a desired goalThe main idea of reinforcement learning is that the agent can learn knowledge in a trial and error process. More specifically, the rewards obtained through interaction with the environment guide the behavior of the agent in an effort to maximize the reward for the agent . Reinforcement learning is different from the supervised learning in connectionism . The signal feedback from environment is an evaluation of the quality of the action, rather than telling the reinforcement learning system how to generate a right actionThe aim of reinforcement learning is to give a Markov decision process and find the optimal policy. The policy is a state-to-action mapping to tell the agent how to select the action.In this study, introduces a novel approach for bearing fault diagnosis that combines a deep autoencoder neural network with reinforcement learning.. Fig. 1.3 is the interaction process between environment and the agent: A fault identification mechanism utilizes a

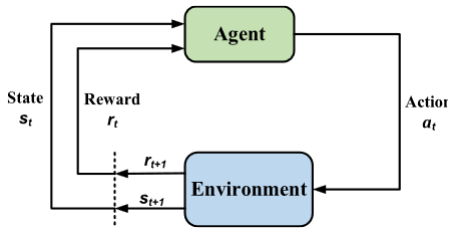

Figure 1.3: Reinforcement learning.

reinforcement learning model to generate predicted labels, which are compared with actual labels for fault status identification. The agent(a_t), representing the deep feature enhanced reinforcement learning model, selects actions based on different fault samples. Feedback in the form of rewards(if correct $r_t + 1$,if not correct $r_t - 1$) is received from the environment, indicating the correctness of the predicted labels. The state(s_t transitions from one fault sample to the next $(s_t + 1)$for continuous diagnosis.

1.4 Q-Learning

Q-learning algorithm uses Temporal difference (TD) method to solve reinforcement learning control problems. TD method can solve rein- forcement learning problems without using complete state sequences. The solution of this kind of reinforcement problem does not need the state transformation model of the environment. The solution of this kind of control problem is to update the strategy through value iteration, that is, through the update of value function, generate new state and immediate reward through the strategy, and then update the value function until the value function and strategy converge. The control

problem of TD can be divided into two categories. One is on-line control, that is, always using a strategy to update value function and select new action. The other is off-line control, that is, the Q-learning algorithm. $\gamma_t + 1+\gamma V(s_t + 1)$ is called TD target value, and $\gamma_t + 1+\gamma V(s_t + 1)-\mathrm{V}(\mathrm{s}_t)$ is called TD error.In this way,to solve the reinforcement learning problem by only two continuous states and corresponding rewards.The iterative formula of the value function of time series difference is:

$$V(s_t) = V(s_t) + \alpha(G_t - V(s_t)) \tag{1.1}$$

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(G_t - Q(s_t, a_t)) \tag{1.2}$$

In Q-learning algorithm, based on state s_t , using ϵ greedy method to select action , executesa_t , gets the reward r_{t+1} , and transfers to next state s_{t+1} . Based on state s_{t+1}, the greedy method is used to select actionn a_{t+1}, that is, select a_{t+1} that can maximize Q(s', a') as a, to update the value function. At this time, the selected action will only participate in the update of the value function and not be really executed. select real action by ϵ greedy method and update new value function by greedy method. It can be expressed by mathematical formula as follows :

$$Q(s, a) = Q(s, a) + \alpha \left(r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right) \tag{1.3}$$

In this research,an autoencoder model is used to learn a compressed representation of the input data. The attention feature maps capture the important features in the input data by highlighting regions that contribute significantly to the reconstruction process.The proposed model leverages the capabilities of deep autoencoders to learn a compressed representation of bearing vibration signals. By encoding the input data into a lower-dimensional space and then reconstructing it, the autoencoder captures essential features and patterns in the signals. This compact representation facilitates effective fault detection and diagnosis.To further enhance the fault diagnosis performance, reinforcement learning is incorporated into the model. Reinforcement learning agents are trained to dynamically adjust the reconstruction error threshold for fault detection based on expert feedback. This adaptive thresholding mechanism enables the model to adapt to different fault conditions, improving the accuracy of fault detection and reducing false positives.

The experimental data utilized in this study is obtained from two datasets: the CWRU bearing fault dataset and the MaFaulDa dataset. The MaFaulDa dataset [39] is a comprehensive database containing various types of failures, severity levels, and rotation frequencies. The second database, known as the CWRU dataset, is acquired from their test-stand [27]. It is employed to assess the model and evaluate its adaptability.The utilization of 1DCNN, LSTM, and GRU models in both CWRU and MAFAULDA datasets showcases their versatility and effectiveness in time series analysis, fault diagnosis, and other related applications.

These models address the limitations of individual algorithms and enable the extraction of spatial and temporal features, leading to improved accuracy, efficiency, and generalization ability.Overall, both the CWRU and MAFALDA datasets serve as valuable resources for evaluating the performance of machine learning algorithms in fault diagnosis and time series analysis.

Chapter 2

LITERATURE SURVEY

Most of machinery used in the modern world operates by means of motors and rotary parts which can develop faults. The monitoring of the operative conditions of a rotary machine provides a great economic improvement by reducing the operational and maintenance costs, as well as improving the safety level. Rolling bearings are critical components in rotating machinery, and their failure can have severe consequences on the system's operation. Traditional fault diagnosis methods, such as periodic inspection and after-the-fact diagnosis, are time-consuming and challenging to implement in real-time operating environments. Failure to address rolling bearing failures promptly can lead to machinery downtime, significant economic losses, and potential safety hazards for workers.

Researchers worldwide have conducted various studies using different approaches and methods based on the operational and fault characteristics of rolling bearings.Typically, fault diagnosis involves three main steps: feature extraction, feature selection, and fault classification.Due to the non stationary nature of vibration signals, methods like short-time Fourier transform, wavelet analysis, and empirical mode decomposition are commonly employed for feature extraction[1-3].The extracted features are then inputted into network models for classification, such as convolutional neural networks (CNNs), BP neural networks, long short-term memory networks (LSTMs), Bayesian classification, and others[4-7].Some studies have proposed novel methods for feature extraction and fault diagnosis. For instance, [14,15] wavelet transform was used to convert the original signal into a two-dimensional image containing fault information, which was then processed by CNN for robust feature learning and fault diagnosis. Another approach utilized variational mode decomposition (VMD), singular value decomposition (SVD), and CNN for feature extraction and fault diagnosis in planetary gears. Additionally, Yuan et al. [16] proposed an empirical modal decomposition (EMD) combined with CNN has been explored for fault feature extraction in rotating machinery, specifically for rolling bearings.While these methods showed promising results, some challenges remain. Certain approaches, although computationally efficient, suffer from high similarity among converted matrices, making it difficult to distinguish between them and leading to longer model training times and lower accuracy. Other methods that involve converting inputs into images increase the processing time and may result in information loss, compromising diagnostic accuracy.Despite the extensive research on bearing fault diagnosis and the application of intelligent diagnosis methods, the learning ability of diagnosis models still has limitations, impacting the effectiveness and accuracy of results. In recent years, deep learning models, known for their powerful data processing and feature learning capabilities, have gained popularity in bearing fault diagnosis. However, traditional models like the back-propagation neural network (BPNN) suffer from slow convergence, local minimization,and overfitting issues.Yin et al. [18] proposed a optimized models such as the cosine loss LSTM neural network (Cos-LSTM) and improved LSTM models to address these challenges.

While LSTM models improve upon some drawbacks of recurrent neural networks (RNNs) in handling sequence data, they can still have lengthy training processes. Variants like Bi LSTM[17] have been employed for fault diagnosis, but their effectiveness may not be optimal. To address these limitations, researchers have introduced improvements to LSTM models using weight amplification and macroscopic-microscopic attention mechanisms, proving their validity for bearing life monitoring through experiments. A novel Convolu-tional Long Short-Term Memory Recurrent Neural Network(CRNN), with higher accuracy and poorer generalizationability, was proposed by Amin Khorrama et al [19]. In order to handle large amounts of input information in deep learning models more efficiently, the attention mechanism has been introduced and proven effective. The attention mechanism allows the model to focus on important input information, improving the overall performance. This mechanism has been applied in deep learning methods with successful results.In the context of rolling bearing fault diagnosis, researchers have utilized the attention mechanism to enhance deep learning models. For example, Rathore proposed a bi-directional LSTM model combined with the attention mechanism for rolling bearing fault diagnosis [21]. Similarly, Li improved deep learning-based fault diagnosis of rolling bearings by incorporating the attention mechanism [20]. In addition to the attention mechanism, experts have also explored the design of more complex network models to achieve more powerful functionalities. These include reinforcement learning, transfer learning, generative adversarial networks, ensemble learning, and others. Among these approaches, reinforcement learning has garnered significant attention due to its strong self-learning and independent decision-making capabilities.

Most of the common methods used to detect bearing faults at present are supervised algorithms where labor label cost is pretty high. Because of the powerful function of hidden layers [40], the algorithm for training the classifier that detects bearing faults is NNs commonly. However, high-performance neural network classifiers also rely on numerous good-quality labeled data. But overfitting of NNs may occur when the samples used for training is noisy, or the amount is not enough, or the test distribution cannot be covered. Thus, the ability of NNs to generalize becomes poor particularly for complicated classification problems. Diagnosing bearing faults is challenging because the subject to be solved is a sophisticated mechanical instrument . Hence, lots of approaches have been put forward to figure this issue out, like senior techniques of processing the signals that are used for analyzing the vibration of signal to enable the extraction of helpful bearing fault features. But these advanced techniques require practitioners to have a deep knowledge reserve of vibration signal and the whole mechanical systems. Moreover, because it is difficult to obtain the professional knowledge related to this, these methods are not universal, meaning that they are not smart enough compared to machine learning. As an alternative and with the power of deep learning, deep neural networks (DNNs) are proposed to achieve unsupervised learning of characteristic. In recent years, considering the impressive achievements in the area of image recognition, deep learning has become the most popular and promising research methods. Among them, deep AE (autoencoder) network structures, as one of the representatives of unsupervised learning for detecting bearing faults, have become a common solution. In addition, as a well-known unsupervised feature learning method, denoising self-encoding is widely used for the realization of various tasks because it can be used to learn more robust feature expressions for input signals, that is, it has a strong generalization ability .

Reinforcement learning has emerged as a powerful approach in various challenging projects, demonstrating remarkable success [22],[23],[24] [25]. Despite the existence of prior human expertise, the architectural design of neural networks still relies on extensive professional knowledge and consumes significant time. As a result, there is a growing interest in automatic methods for designing neural network architectures [27],[28]. Building on this trend, reinforcement learning is being utilized to explore more complex and deeper network structures. RL models learn by interacting with the environment and continuously adapt their behavior to maximize the feedback reward signal from the environment during training (Sutton and Barto, 2018). Unlike supervised machine learning algorithms, RL learns an optimal policy, which is a response mechanism that yields the highest reward, rather than fitting the mapping function between samples and labels. Deep Reinforcement Learning (DRL) combines the advantages of automatic feature extraction from deep learning and interactive learning from RL, making it a revolutionary approach in the field of artificial intelligence. DRL has shown promise in solving complex real-world problems, such as playing video game at a human player level (Mnih et al., 2013) and defeating world champions in games like Go (Silver et al., 2017).Recently, an increasing number of studies have applied DRL to fault diagnosis problems. For example, Ding et al. (2019) proposed a DRL method based on a sparse auto-encoder (SAE) for bearing and pump fault diagnosis, which showed comparable performance to the deep learning model SAE-softmax. Wang et al. (2022) developed a planetary gearbox fault diagnosis method using time-frequency representation (TFR) and a CNN-based DRL model, which performed well under various working conditions. Li et al. (2021) developed a DRL model based on capsule neural network (Cap-net) and an online feature dictionary method, which adapted to fault diagnosis tasks with variable working conditions. Zisheng Wang and Xuan (2021) proposed a 1D-CNN based DRL method for compound fault diagnosis in the presence of heavy background noise in bearings and tools.

While the above studies demonstrate the excellent fault diagnosis performance of DRL models in scenarios involving strong noise, varying working conditions, and sample imbalance, there is a relative lack of research on DRL-based fault diagnosis under small sample scenarios. Many existing studies focus on developing CNN-based DRL fault diagnosis models.In 2017, the Google DeepMind team proposed a neural network search method based on reinforcement learning [26]. They developed an RNN structure called NASCell, which was used to generate CNN structures. However, the complexity of the obtained network structures posed challenges in practical applications. Implementing these structures required substantial time, advanced equipment, and adequate financial resources.While the automatic design of neural network architecture through reinforcement learning holds great potential, practical implementation remains a hurdle due to the resource-intensive nature of the process. Nonetheless, researchers continue to explore and refine these methods in order to make them more accessible and applicable to real-world scenarios.

.

Chapter 3

METHODOLOGY

3.1 Data Preprocessing

The whole Framework can be split into four parts:- Preprocessing Module,Data feature enhanced with auto encoder, Deep Q neural network and Classification Module. In Mafaulda dataset,the data acquired is passed through subsampling and Time Domain and Frequency Domain Data are combined into a time-ordered sequential feature space. In the TD, summary data values are generated by identifying the maximum values within each bin. The raw data summarization interval is carefully chosen to ensure an adequate number of raw data points for accurate DFC (Diagnostic Feature Classification) extraction while maintaining an appropriate subsampling ratio. To mitigate overfitting, data augmentation is performed to synthetically create additional data. To extract the DFCs, the FFT (Fast Fourier Transform) spectrum is computed for each segment. Since the experiment considers fluctuating speeds and changing operating conditions in the industry, normalization is necessary to minimize variations in rates due to these conditions.This process is given by:

$$\mu_{\text{amp}} = \frac{1}{S_f} \sum_{i=1}^{S_f} f_i X_{\text{amp}} \tag{3.1}$$

$$N_{f_i x} = \frac{f_i x_{amp}}{\mu_{amp}} \tag{3.2}$$

Fig(3.1) shows the architecture of proposed method.The first step in the proposed approach is to generate summary data points that capture essential information from the continuous data stream. Each summary data point combines Time Domain (TD) signals and Frequency Domain Data Flow Characteristics (DFCs). These summary data points are assigned at regular intervals during the monitoring period of actual industrial scenarios. They are created in a subsampled feature space, providing the advantages of the temporal property of data points and sufficient discriminative information.

To determine the raw data summarization interval (T_s), two constraints are taken into account. Firstly, a minimum number of raw data points is required to ensure proper DFC extraction. This ensures that enough data points are available for accurate analysis. Secondly, the subsampling ratio is chosen to align with the unit observation period of industrial data-based solutions. This ensures that the summary data points are representative of the underlying patterns and characteristics of the industrial process.The number of sampling points per rotation (N_s) can be calculated as $N_s = 60 f_s / r_s$. Additionally, the minimum number of sampling points per segment (n_l) is determined such that it satisfies the condition: $s_i \leq \mathrm{n}_s \leq \mathrm{n}_l$,where s_i represents the number of interval points between the samples. This condition ensures that each segment contains one or more rotation data.Moreover, the maximum number of points is determined based on the data sampling rate of industry-based

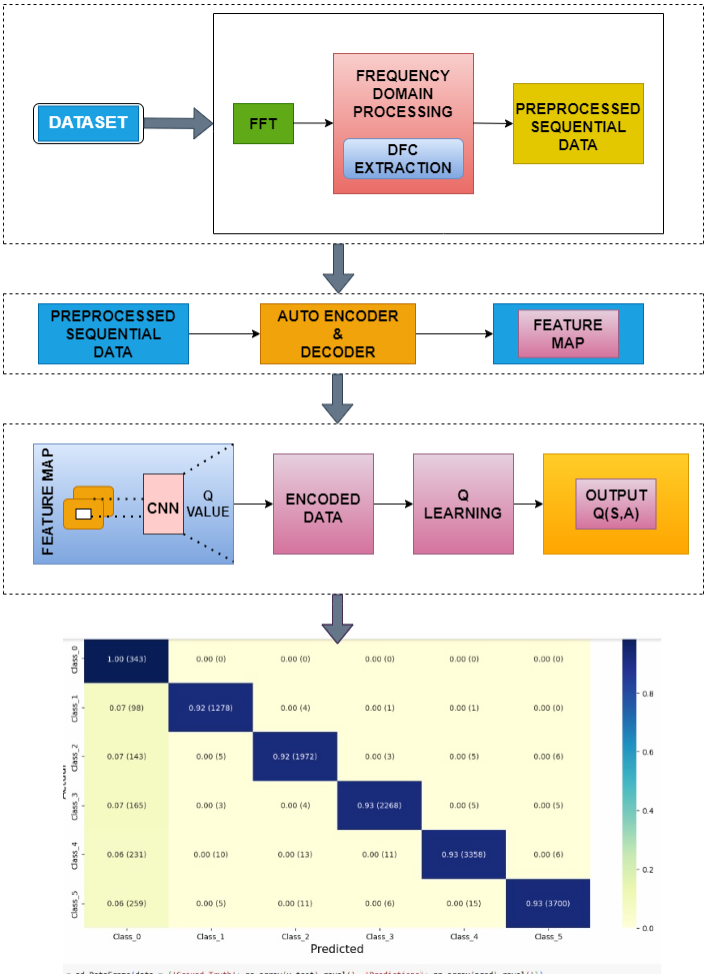

Figure 3.1: Architecture of proposed method

vibration monitoring solutions. For instance, a subsampling duration of one second may result in a segment length of F_S points in practical experiments.

3.2 DFC Extraction

To extract the DFCs, the proposed approach employs the Fast Fourier Transform (FFT) spectrum evaluation on each segment. Given the fluctuating speed and changing operating conditions in industrial scenarios,normalization is applied to reduce the rate difference under varying conditions.The normalization process is given by:

$$\mu_{\text{ampl}} = \frac{1}{N_f} \sum_{i=1}^{N_f} X_i \tag{3.3}$$

$$\bar{A}_i = \frac{A_i}{\mu_{amp}} \tag{3.4}$$

where N_f represents the number of significant frequency bands, A_i denotes the amplitude analogous to the $f_i X frequency$,$\mu_a mp$ is the mean of amplitudes corresponding to the significant frequency bands, and $\tilde{A}_i$ represents the normalized amplitude values.The DFCs are obtained by assigning $\tilde{A}i$ for i = 1, 2, ..., H_f, where h_f signifies the number of rotating frequency components. These DFCs are extracted using a multipass filter within the range $fiX - \Delta f$ to $fiX + \Delta f$,where Δ_f represents the fluctuating frequency range.he summary data values in the time domain (TD) are generated by identifying the maximum values within each bin of size $\tilde{N}_s$ data points and averaging fs/Ns values.

In this study, the CWRU dataset is utilized, which comprises time-series data representing diverse fault conditions within a system.The dataset consists of multiple fault classes, including '07_IR', '14_IR', '21_IR', '07_BA', '14_BA', '21_BA', '07_OR', '14_OR', '21_OR', and 'N'. To prepare the data for training a classification model, a sliding window approach is employed. A window of length 500 data points is moved along the time series with a stride of 300 data points. For each window, the features are extracted by excluding the last column, which represents the fault label. The features are reshaped to have a single channel, resulting in a three-dimensional input shape (number of windows, window length, and number of channels). The label for each window is derived from the last data point within the window.The dataset is preprocessed and divided into training and testing sets . The data is split with a test size of 30 percentage while ensuring stratified sampling. This means that the distribution of fault classes is maintained in both the training and testing sets, as the stratify parameter is set to the encoded labels.This data preprocessing and train-test split procedure enables the subsequent training and evaluation of classification models on the fault dataset.

3.3 Data feature enhanced with autoencoder

The autoencoder is a type of neural network that learns to compress and then reconstruct the input data.Autoencoder (AE) consists of an encoder and a decoder. Figure 3.1 shows are autoencoder neural network.Additionally, AE is one of the representative methods of unsupervised learning. The principle by which AE achieves unsupervised feature extraction is through the continuous minimization of the discrepancy between the input data and the reconstructed data.In this study, a feature enhancement module is designed. It can effectively enhance useful features from large amounts of input feature and improve pattern recognition ability.However, it defines an autoencoder model that can be considered as a form of feature enhancement.

In an autoencoder, the encoder part learns to extract relevant features from the input data and compress it into a lower-dimensional representation. The decoder part then reconstructs the original input from this compressed representation. This process inherently performs feature extraction by capturing the most informative features in the encoded layer.By training the autoencoder , the model learns to enhance relevant features by optimizing the reconstruction of the input data. The layers with the 'relu' activation function contribute to this

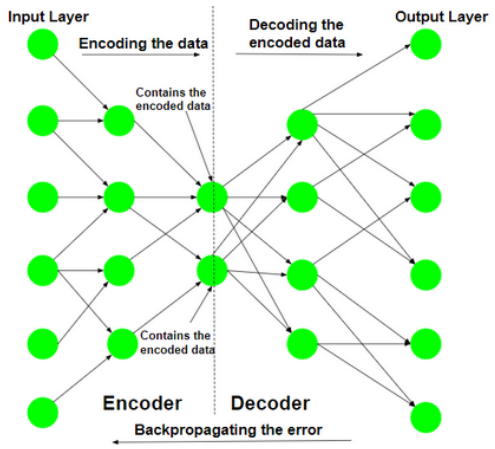

Figure 3.2: Autoencoder

process by introducing non-linearity and capturing complex patterns in the data.After this, performs data normalization, builds and trains autoencoder models, and generates attention feature maps for both the training and test data. The attention feature maps represent the enhanced and compressed representations of the original data, obtained through the autoencoder's encoding process. These attention feature maps can then be utilized for classification in fault diagnosis tasks. They capture the most salient and informative features from the input data, enabling improved performance in fault diagnosis and related tasks.

3.4 Fault diagnosis with deep Q neural network

In this study, a deep Q network (DQN) module is designed to achieve higher fault classification accuracy by maximizing long-term cumulative rewards. The DQN module is implemented to transform the fault diagnosis task into a reinforcement learning task. The reinforcement learning environment is equivalent to the fault identification mechanism established, where the agent represents the DQN module. The state (s_t) represents the fault sample passed into the DQN module, and the action (a_t) refers to the predicted label selected by the DQN module based on the current state (s_t), matching the current fault sample.

When starting fault diagnosis, the fault samples are input to the DQN module, and the agent selects the corresponding action (a_t) based on this fault sample. After executing the action (a_t), the environment provides the agent with feedback ($r_t + 1$), representing the reward. The reward reflects the correctness or incorrectness of the predicted label ($r_t + 1 =$ +1 if correct and $r_t + 1 =$ -1 if incorrect). Simultaneously, the state transitions from (s_t) to ($s_t + 1$), and the training experience s_t, a_t, $r_t + 1$, $s_t + 1$ is saved.Following this, the next training sample is input to the DQN module, and the agent continues to diagnose the next fault sample. The agent employs the -greedy method in Formula 3.1 to choose the action (a) that maximizes the value function Q(s, a). The value function Q(s, a) can be represented by the Bellman optimization equation:

$$Q(s,a) = Q(s,a) + \alpha\left(r + \gamma \max_a Q(s',a') - Q(s,a)\right) \tag{3.5}$$

in which α is set to be 0.5. ϵ greed method adopted linear decreasing function,

$$\epsilon = max\{\epsilon_{min}, 1\frac{1-\epsilon_{min} * step}{total}\} \tag{3.6}$$

in which $\epsilon_{min} = 0.01$.The DQN module fits the Q-value with a convolutional neuralnetwork. i.e., the value function is expressed in terms of the CNN weights, bias parameters, denoted as θ. When the value function is updated, θ is updated accordingly. The loss function L(θ) is trained to update the Q(s, a; θ).

$$L(\theta) = \mathrm{E}\left[\left(r + \gamma \max_a Q(s',a',\theta) - Q(s,a)\right)^2\right] \tag{3.7}$$

where E[(•) 2] is the mean-squared error. Q(s, a;θ) is the value function.The main technique used in DQN training is experience replay in order to break the correlation between samples while satisfying the independent and identically distributed requirements of CNN. The training experience s_t , a_t , $r_t + 1$,$s_t + 1$ obtained at each step is deposited into the memory D. After that, the uniform random sampling method is used to randomly select samples s_t , a_t , $r_t + 1$,$s_t + 1$ from D for training DQN module.

During fault diagnosis, the input consists of fault samples containing a massive number of data points. Storing and updating a Q table with such a large amount of data using the traditional Q-learning method is not feasible. To overcome this limitation, we utilize Convolutional Neural Networks (CNNs) with their powerful feature extraction capabilities to extract relevant features from the fault samples.The CNN is employed to extract features from the input fault sample, and subsequently, a fully connected layer is connected to encode the features and obtain the Q-value of the current state (s_t). Additionally, the action selected by the DQN module based on the current state (s_t), represented as the one-hot code of the predicted label, is taken as another input.

Algorithm 1:Deep Q-learning with Experience Replay

Initialization:
Initialize Replay memory D
Initialize $Q(s,a,\theta)$ with random weight θ
for iteration $i = 1$ to N **do**
    Randomly input fault samples to initialize the state s_1
    for $t = 1$ to T **do**
        With probability ϵ select a random action a_t
        Otherwise select $a_t = \arg\max_a Q(s_t,a,\theta)$
        Execute action a_t and get reward r_{t+1}
        Transfer to next state s_{t+1}
        Store transition $(s_t, a_t, r_{t+1}, s_{t+1})$

Sample random minibatch of transitions $(s_j, a_j, r_{j+1}, s_{j+1})$ from D

$$y_i = \begin{cases} \gamma_j & \text{for terminal state } s_{j+1} \\ \gamma_j + \gamma_{\max_a} \cdot Q(s_{j+1}, a, \theta) & \text{for non-terminal state } s_{j+1} \end{cases}$$

Perform a gradient descent step on $(y_j) - Q(s_j, a_j, \theta)$

EndFor

To incorporate the action information into the Q-value calculation, we multiply the action a_t) with the Q-value encoded by the CNN from the previous step. This enables us to obtain the Q(s, a; θ) value, which contains both state and action information. We then use gradient descent to update the Q(s, a; θ) value, facilitating the learning process of the DQN module.

The specific procedures are summarized as follows:

1.Preprocessing step: For each input feature x_i at index i, the input is transformed using a reference value r_i as follows: $z_i = x_i$ r.

2.Randomly select fault samples as the current state $s_t$and input s_t to DQN module.

3.Autoencoder architectures takes the input shape as an argument and returns the

constructed model. The model consists of a series of dense layers with decreasing sizes for the encoder part and increasing sizes for the decoder part.

4.The activation function used is ReLU for the encoder layers and sigmoid for the final decoder layer.Then preprocesses the input data by normalizing it.

5.The normalized values are computed by subtracting the mean of the training and test data, respectively, and dividing by the standard deviation.

6.Encode the input attention feature map with convolution layer and full connected layer to obtain Q-value.

7.The ϵ greedy method is used to choose action a_t , execute action at and get the reward r_t+1 (if correct, the reward + 1, and if wrong, the reward 1), and enter the next state s_t+1. Store the transition s_t, a_t ,$r_t + 1$, $s_t + 1$ in memory D.

8.Updates the Q-value of a state-action pair using the Q-learning formula. It calculates the new Q-value based on the current Q-value, the immediate reward, and the maximum Q-value of the next state.
Q-learning formula: Q(s, a) = (1 - α) $\cdot Q(s, a) + \alpha \cdot (r + \gamma \cdot \max(Q(s', a')))$

9.Repeat the step (2) to step (8) and sample random minibatch of transitions s_t, a_t ,r_t+1, $s_t + 1$ from D to update θ of Q(s, a; θ).

10. Predicts the labels for the given features using the trained Q-learning agent. It iterates over each state and performs the following steps:(a)Compare the predicted label with

the actual label and calculate the classification accuracy.

11.It calculates the classification accuracy by dividing the discordance count by the total number of labels.Verify the efficacy of proposed method.

3.5 Classification Module

LSTM, GRU, and convolutional layers are powerful components in building deep learning models that excel at capturing temporal dependencies in sequential data while extracting relevant features from the input. A 1DCNN is a type of convolutional neural network that operates on one-dimensional data, such as time series. By applying convolutional filters over the input sequence, 1DCNNs can effectively capture local patterns and features within the data.When used in conjunction with LSTM or GRU architectures, a 1DCNN layer can serve as a preprocessing step to extract relevant local features before passing the processed data to the recurrent layers. This combination allows the model to benefit from both the feature extraction capabilities of the 1DCNN and the long-term dependency modeling of LSTM .By incorporating a 1DCNN layer, the model can capture short-term patterns and features in the input sequence while the LSTM or GRU layers can focus on capturing long-term dependencies. This hybrid architecture can be particularly effective in time series analysis and prediction tasks where both local patterns and long-term dependencies are crucial for accurate classification.

Convolutional Neural Network :A CNN is a powerful feed-forward deep neural network known for its strong feature extraction capabilities, inspired by the primate perception mechanism . It is specifically designed to process data with a grid-like topology, such as images. The network operates by progressively extracting representative features from the input data using multiple filters . By employing sparse connectivity and parameter weight sharing mechanisms, CNNs are able to downsample and refine the data's dimensionality in both time and space. This approach effectively reduces the number of trainable parameters and helps prevent overfitting.

By analyzing Figure 3.3, the output of the convolution operation is expressed as:

$$y_t = w_t * x_l + b_l \tag{3.8}$$

where l is the number of layers of the convolutional layer, $w_l$represents the weight matrix of convolution kernels, x_l represents the input of the layer l,b_l is the bias of layer l, and y_l is the output of layer l.A typical CNN includes a convolutional layer for feature extraction, pooling layers for dimensionality reduction, activation layers for introducing non-linearity, and fully connected layers for classification or regression tasks.. Convolutional layers are the core component of CNN . The input of one-dimensional convolution is one-dimensional data. In an **LSTM**, each cell has three main components: the input gate, the forget gate, and the output gate. These gates are responsible for regulating the flow of information.The equations for an LSTM cell at time step t are as follows:

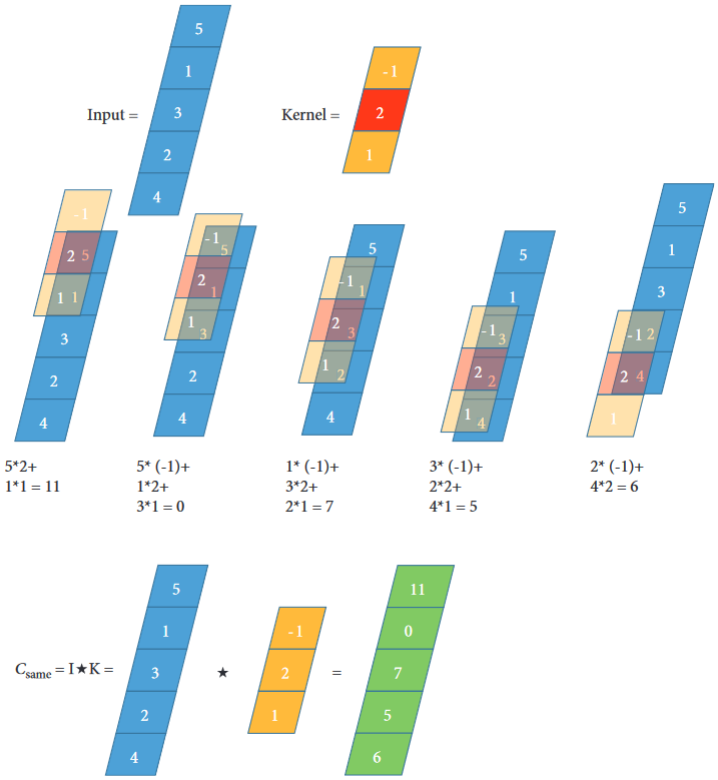

Figure 3.3: The operation of the one-dimensional convolution.

$$Inputgate(i_t) : i_t = (W_{xi}x_t + W_hih_t - 1 + W_cic_t - 1 + b_i) \tag{3.9}$$

$$Forgetgate(f_t) : f_t = (W_{xf}x_t + W_hfh_t - 1 + W_cfc_t - 1 + b_f) \tag{3.10}$$

$$Cellstate(c_t) : c_t = (f_t * c_t - 1 + i_t * tanh(W_{xc}x_t + W_{hc}h_t - 1 + b_c)) \tag{3.11}$$

$$Outputgate(o_t) : o_t = (W_{x0}x_t + W_h0h_t - 1 + W_coc_t + b_o) \tag{3.12}$$

$$Hiddengate(h_t) : h_t = (o_t * tanh(c_t)) \tag{3.13}$$

In these equations, x_t represents the input at time step t, h_t represents the hidden state at time step t, $c_t$represents the cell state at time step t. The sigmoid function is denoted as , * represents element-wise multiplication, and tanh represents the hyperbolic tangent function.

The proposed fault diagnosis model in this study combines 1DCNN and LSTM to create an end-to-end architecture. It comprises stacked convolutional layers, pooling layers, an

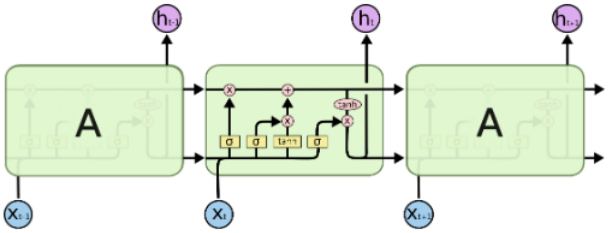

Figure 3.4: LSTM Architecture

LSTM layer, and a fully connected layer. The first two pooling layers utilize max pooling to extract prominent features, while the last layer employs average pooling to retain important information. The LSTM layer captures temporal dependencies in the features extracted by the convolutional operation, effectively modeling time series data. The model concludes with a dense layer containing 10 cells and a softmax activation function, which outputs a set of values between 0 and 1. These values are used to determine the fault type.

The **GRU** architecture simplifies the LSTM architecture by combining the cell state and the hidden state into a single state called the "update gate" (z_t) and a "reset gate" (r_t) that control the flow of information within the cell. The GRU equations for a cell at time step t are as follows:

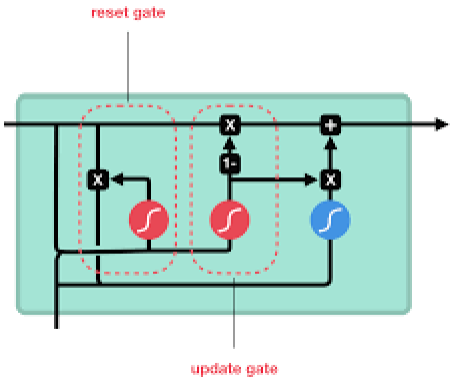

Figure 3.5: GRU Architecture

$$Resetgate(r_t) : r_t = (W_{xr}x_t + W_hrh_t - 1 + +b_r) \tag{3.14}$$

$$Updategate(z_t) : z_t = (W_{xz}x_t + W_hzh_t - 1 + b_z) \tag{3.15}$$

$$CandidateActivation(h_t) : h_t = tanh(W_{xh}x_t + r_t * (W_{hh}h_t - 1 + b_h)) \tag{3.16}$$

$$Hiddengate(h_t) : h_t = (1 - z_t) * h_t - 1 + z_t * h_t \tag{3.17}$$

In these equations, x_t represents the input at time step t, h_t represents the hidden state at time step t,r_t represents the reset gate at time step t, z_t represents the update gate at time step t, is the sigmoid function,* represents element-wise multiplication, and tanh is the hyperbolic tangent function. The LSTM and GRU architectures have proven to be valuable tools for time series analysis and forecasting. Due to their gating mechanisms, both architectures excel in capturing long-term dependencies in time series data. They have been widely used in various tasks such as stock market prediction, natural language processing, speech recognition, and machine translation. These architectures are effective in modeling and predicting complex sequential patterns, making them crucial for time series applications.

Chapter 4

EXPERIMENT AND RESULTS

4.1 Datasets and Experimental Settings

For experimentation, two datasets were used: the publicly available MaFaulDa (Machinery Fault Dataset) and the CWRU (Case Western Reserve University (CWRU)) dataset, as shown in Figure 4.1. comparison of setup conditions for these data-sets is detailed in Table 4.1

CWRU (Case Western Reserve University) Dataset

The bearing fault dataset collected at the test-stand (shown in Figure 4.1) of Case Western Re- serve University (CWRU) [27] is used as the experimental data.The subject bearing used in the experiments is SKF6205. The sampling frequency of the data collection platform is 12 kHz, which means that 12,000 samples per second were collected. These samples represent vibration acceleration velocity values.

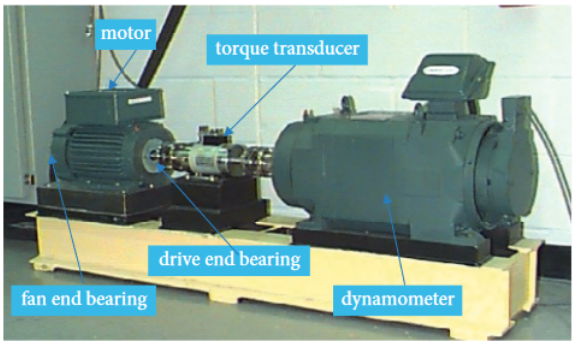

Figure 4.1: Motor driving mechanical system used by CWRU

The dataset from CWRU(DS-1) includes three bearing components: the inner raceway, the outer raceway, and the rolling element.Four different load conditions were considered: 0 hp, 1 hp, 2 hp, and 3 hp. The dataset covers a total of nine fault types, which were created by implanting damage with diameters of 0.007 inches, 0.014 inches, and 0.021 inches in each component of the bearing.The damage was introduced using the electrodischarge machining (EDM) technique.The dataset under 0 hp, 1 hp, 2 hp, and 3 hp load condition is selected for the model validation experiment and model migration generalization. Each dataset contains 10 statuses, including 9 fault statuses and one normal status. In each kind of load condition, there are 1000 samples for each status, resulting in a total of 10000 samples.Each sample consists of 2048 data points.The time-domain waveform diagrams of 10 bearing operating states are shown in Figure 4.2, respectively, where the x-axis represents the sampling indices, and the y-axis represents the amplitude.For each bearing state, 70%f the samples are randomly selected for model training, 20% are used for model validation, and 10% are left for model testing.

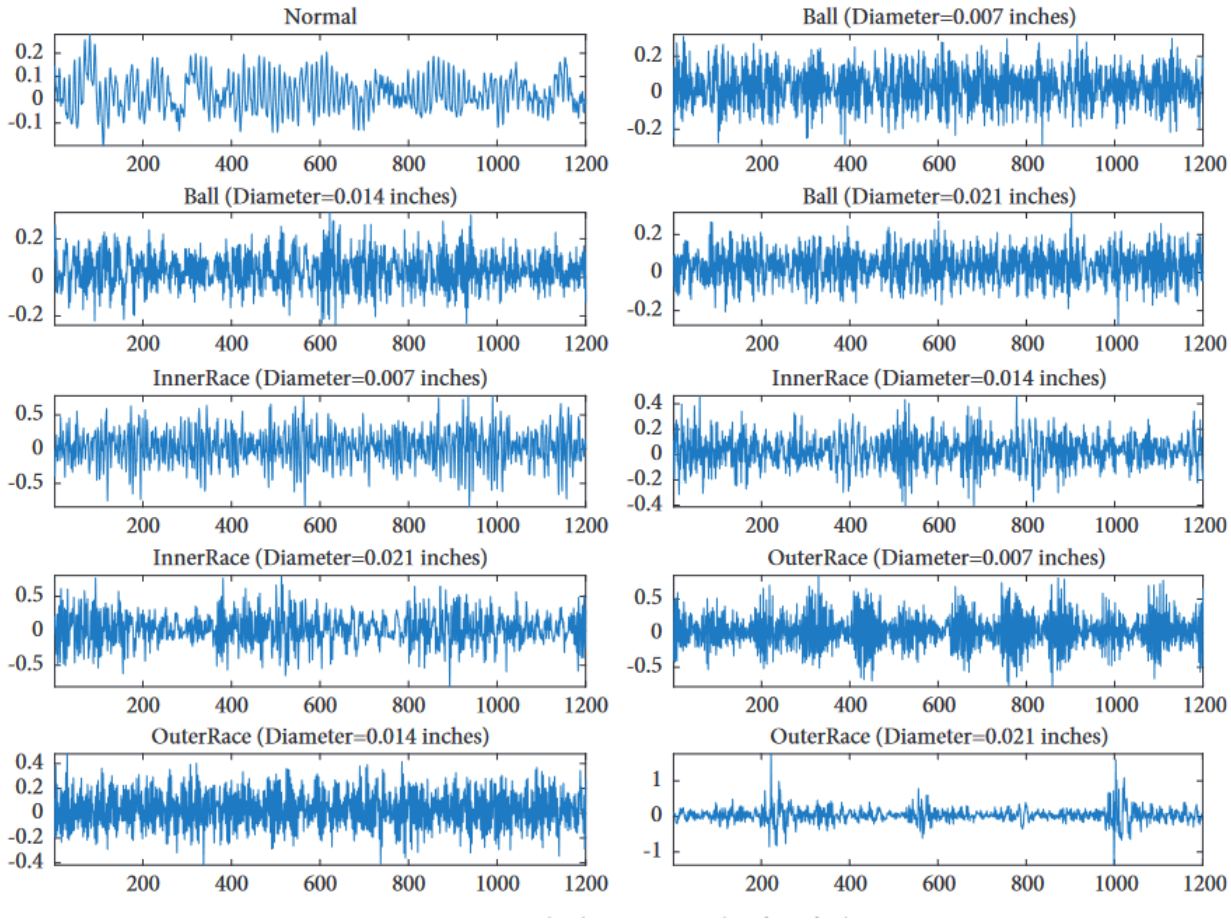

Figure 4.2: Typical vibration signals of 10 faults

MaFaulDa (Machinery Fault Dataset)

The dataset(DS-2) contains multivariate sequences of SRF (Sensor Recorded Fault) data acquired using sensors on SpectraQuest's Machinery Fault Simulator (MFS). The MFS is capable of simulating six different states: normal, imbalance, horizontal misalignment, vertical misalignment, underhang, and overhang. The MFS generates eight signals, which include a tachometer signal, radial, axial, and tangential amplitudes from two sensors, and a microphone signal.Fig:4.3 shows the visualization of mafaulda dataset. The dataset consists of a total of 1951 sequences, and each sequence corresponds to a simulated state. The data within each sequence is sampled at a frequency of 50 kHz for a duration of 5 seconds. The filenames of the sequences correspond to the rotational frequency of the motor used in that specific sequence. Each data point within a sequence contains amplitude values for each of the eight signals mentioned.

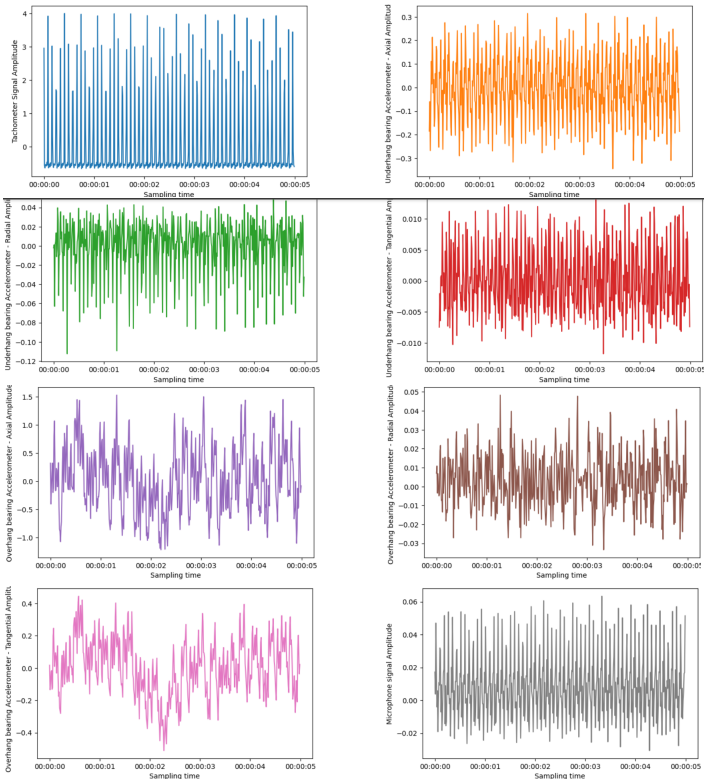

Figure 4.3: visualization of mafaulda dataset

Table 4.1: Database Comparison

Features	**CWRU**	**MAFAULDA**
Accelerometr	3 unidirectional	1 unidirectional
Frequency / RPM	Sampling Fz: 51.2 kHz	Sampling Fz: 51.2 kHz RPM Range: 700–3600 rpm
Direction	Axial, radial, tangential	Axial
Faults	Inner race, outer race, ball, normal	Imbalance, normal, horizontal, vertical
Diameter	0.007, 0.014, 0.021	Rotor: 152.4mm and shaft:16.0mm

4.2 Classification

The pre-processed data-sets are then split into train and test sets with a 80-20 split ratio. Two sequential models:1DCNN- LSTM and GRU are then used to classify the faults on the data. The performance metric used for calculating the effectiveness of the model is :- **Accuracy**: The percentage of samples where predicted class and actual class are same, over each sample in the test set, formally defined as (eq:4.1):

$$\text{Accuracy} = \frac{\sum_{i=1}^{N}(\hat{y}_i = y_i)}{N} \times 100 \tag{4.1}$$

Table 4.2: Test Accuracy

Model	DS-1		DS-2	
	Train Accu (%)	Test Accu (%)	Train Accu (%)	Test Accu (%)
(C_1)1D-CNN, LSTM	0.989	0.986	0.993	0.987
(C_2)GRU	0.992	0.989	1.00	0.997

The comparison of accuracy between the 1DCNN-LSTM(C_1) and GRU (C_2)models over a training period of 20 epochs is described in Table . The table 4.2 shows that both models have similar performance C_2 with having a slight advantage in DS-2. It is also observed that as the complexity increases, meaning as the number of epochs increases, there is a significant improvement in accuracy until around epoch X. However, after that point, there is little to no change in accuracy with each additional epoch. As can be seen from Figure 4.3, the

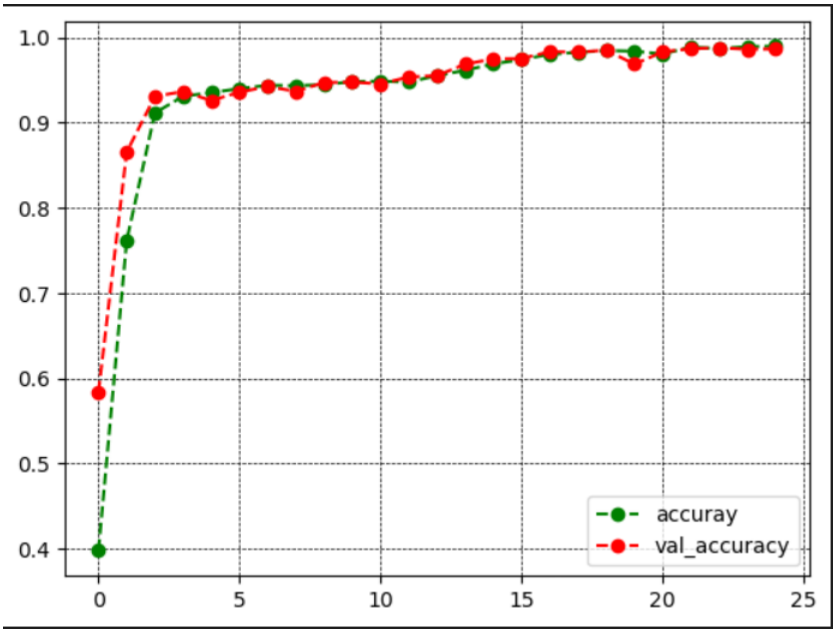

Figure 4.4: Train and validation accuracy

accuracy of the model is 0.98 in the fifteenth training epoch and remains stable in subsequent training epochs. During testing, although the accuracy fluctuates initially, it consistently increases after the fourth training epoch and stabilizes after the fifth training epoch. This demonstrates the powerful feature learning ability of the proposed model. These results indicate that the proposed model can be effectively applied to bearing diagnosis tasks when provided with vibration datasets.

4.3 Verification of proposed method

The main objective of this study is to train a Q-learning agent capable of accurately predicting fault conditions based on encoded features extracted from the CWRU and Mafaulda datasets. The study utilizes autoencoder models, which consist of dense layers for both en-

Table 4.3: Performance Evaluation

Method	**DS-1 (%)**	**DS-2 (%)**
Precision	94.56	87.01
Recall	93.08	93.8
F1-score	93.27	86.86
Accuracy	**93.01**	**90.09**

coding and decoding, to learn representations of the input data. The models are optimized using an appropriate optimizer and loss function during the training phase, which involves normalizing the input data to enhance convergence and performance. The trained models are then employed to generate attention feature maps, highlighting salient areas in the input data. These maps are derived by passing the normalized training data through the trained models and extracting relevant representations. Additionally, a convolutional neural network (CNN) model is developed to encode the input data. The encoded data is further utilized to train a Q-learning agent, which predicts labels for the testing data. Q-learning agent is trained using the encoded training data, updating the Q-table based on rewards obtained from the labels.The Q-learning agent makes predictions on the encoded testing data, and the performance of the classification task is evaluated using metrics such as precision, recall, F1-score, and accuracy.

In case of CWRU: The Q-learning agent achieved an accuracy of 93.00% on the validation set, indicating that it correctly classified 93.00% of the samples. The precision score of 95.00% suggests that when the agent predicted a particular class, it was accurate 95.00% of the time. The recall score of 93.50% indicates that the agent successfully identified 93.50% of the positive samples in the dataset. The F1-score, which considers both precision and recall, is also calculated and found to be 93.50%.

In case of Mafaulda: It achieved an accuracy of 90.09% on the validation set, indicating that it correctly classified 90.09% of the samples. The precision score of 87.01% suggests that when the model predicted a particular class, it was accurate 87.01% of the time. The recall score of 93.8% indicates that the model successfully identified 93.8% of the positive samples in the dataset. The F1-score, which considers both precision and recall, is also calculated and found to be 86.86%.Comparing these results Mafaulda model exhibits slightly lower accuracy, precision, and F1-score values. However, the recall score of the Mafaulda model is higher, indicating that it has a better ability to correctly identify positive samples. The macro avg and weighted avg values provide average and weighted averages of precision, recall, and F1-score across all classes.In CWRU dataset generally exhibits better performance in terms of macro average and weighted average scores compared to the Mafaulda dataset. For the macro average, which calculates the average performance across all classes without considering class imbalance, CWRU dataset has higher values compared to Mafaulda. The precision for CWRU is 0.95, while for Mafaulda it is 0.87. The recall for CWRU is 0.94, while for Mafaulda it is 0.94. The F1-score for CWRU is 0.93, while for Mafaulda it is 0.87.These results indicate that, on average, the CWRU dataset performs better across all classes compared to Mafaulda.

The confusion matrix was visualized to gain further insights into the classification performance. The matrix shows the number of true positives, true negatives, false positives, and false negatives for each class. It provides a comprehensive overview of the agent's performance across different classes.The overall performance indicated by the confusion matrix highlights the model's ability to correctly identify and classify samples from different classes.Visualisation of confusion matrix for both dataset shown in fig (4.5 and 4.6).

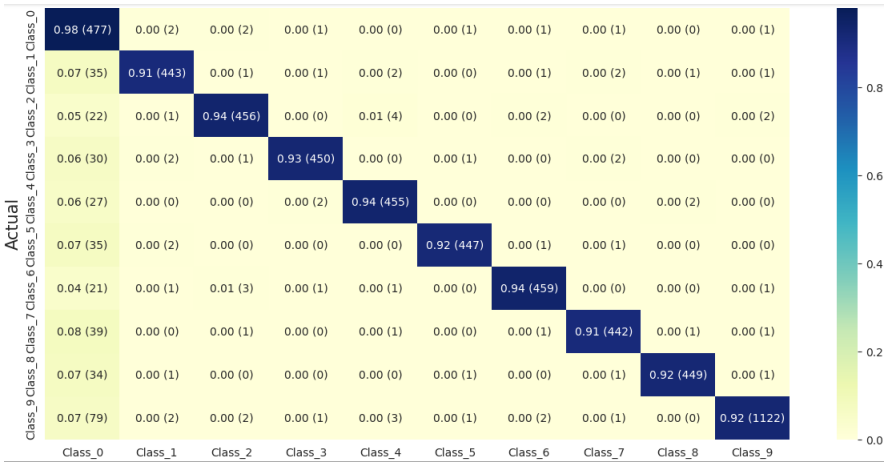

Figure 4.5: (A) Confusion matrix: CWRU

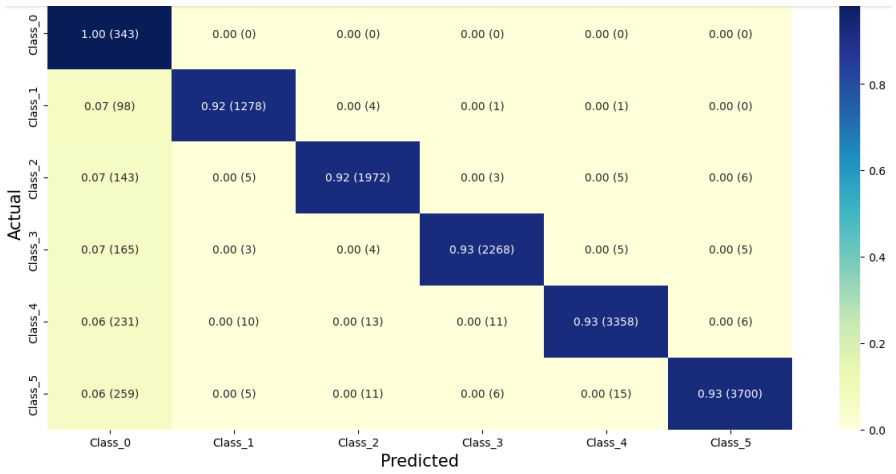

Figure 4.6: (B) Confusion matrix:Mafaulda

It is important to note that while the Mafaulda model achieves high recall scores for most classes, there are variations in performance when compared to the CWRU model. The Mafaulda model seems to have a higher recall for Class 0 and Class 1, indicating its ability to better identify samples from these classes. However, the CWRU model shows slightly higher recalls for Class 4 and Class 6.These comparisons highlight the variations in

performance between the two datasets and the models trained on them. The differences could be attributed to variations in data characteristics, feature representations, or model architectures.

4.4 Classification Report

The classification report shows the precision, recall, and F1-score for each class, as well as the overall accuracy and other aggregated measures.In case of CWRU:For each individual class, observe varying levels of precision, recall, and F1-score. Precision measures the proportion of correctly predicted instances out of the total instances predicted as a particular class. Recall represents the proportion of correctly predicted instances out of the total instances of that class in the dataset. The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance.

Table 4.4: Classification Report: CWRU

	precision	recall	f1-score	support
Class_0	0.60	0.98	0.74	486
Class_1	0.98	0.91	0.94	487
Class_2	0.98	0.94	0.96	487
Class_3	0.99	0.93	0.96	486
Class_4	0.98	0.94	0.96	486
Class_5	0.99	0.92	0.95	486
Class_6	0.98	0.94	0.96	487
Class_7	0.98	0.91	0.94	486
Class_8	0.99	0.92	0.96	487
Class_9	0.99	0.92	0.96	1213
accuracy			0.93	5591
macro avg	0.95	0.93	0.93	5591
weighted avg	0.95	0.93	0.94	5591

In case of CWRU dataset, the model performs well on most classes, with precision, recall, and F1-scores above 0.90 for $Class_1$ to $Class_9$. $Class_0$ has relatively lower precision (0.60) but high recall (0.98), indicating that the model frequently identifies instances correctly but may also generate false positives for this class. Precision, recall, and F1-score values for the macro average are 0.95, 0.93, and 0.93, respectively. The support value indicates the number of instances considered for calculating these metrics, which is 5591 in this case.

In case of Mafaulda: For each class, the model exhibits high precision, recall, and F1-scores, ranging from 0.92 to 0.99,except for $Class_0$, where the precision is low. The model demonstrates effectiveness in identifying most of the classes with high accuracy. However, there is room for improvement in correctly classifying samples in $Class_0$. The macro avg and weighted avg values again provide average and weighted averages of precision, recall, and F1-score across all classes. In this case, the macro avg values show a high level of performance,with the weighted avg values taking into account the support (number of samples) for each class.The precision, recall, and F1-score values for the macro average are 0.87,

0.94, and 0.87, respectively. The support value indicates the number of instances considered for calculating these metrics, which is 13959 in this case.

Table 4.5: Classification Report: Mafaulda

	precision	recall	f1-score	support
Class_0	0.27	1.00	0.43	339
Class_1	0.98	0.92	0.95	1404
Class_2	0.99	0.93	0.96	2193
Class_3	0.99	0.93	0.96	2434
Class_4	0.99	0.93	0.96	3629
Class_5	0.99	0.93	0.96	3960
accuracy			0.90	13959
macro avg	0.87	0.94	0.87	13959
weighted avg	0.97	0.93	0.94	13959

4.5 Comparison of Learning Rates

Learning rate is a crucial hyperparameter in machine learning algorithms, including Q-learning. In this comparison, we explore the impact of different learning rates on the performance of the Q-learning agent using two distinct datasets: MaFaulda and CWRU.The learning rate determines the balance between exploration and exploitation during the learning process and can significantly affect the agent's ability to learn optimal policies.The evaluation metric employed to measure the agent's effectiveness will be the accuracy achieved on a validation set. By analyzing the accuracy obtained for each learning rate, we can determine which learning rate yields the highest classification accuracy, thereby identifying the most optimal learning rate for fault classification using the Q-learning agent.The performance of the Q-learning agents with different learning rates on the MaFaulda and CWRU datasets was compared.

The model's accuracy is relatively stable across different learning rates. The accuracies range from 90% to 93.15%, with slight variations.Based on the provided accuracies, it seems that the learning rates in the range of 0.1 to 0.5 yield relatively high accuracy, with small variations between them. For the CWRU dataset, the accuracies range from 92.5% to 93.31% as the learning rate increases from 0.0 to 1.0. The highest accuracy is achieved at a learning rate of 1.0. On the other hand, for the Mafaulda dataset, the accuracies range from 92.5% to 93.33% as the learning rate increases. Similar to the CWRU dataset, the highest accuracy is obtained at a learning rate of 1.0.Both datasets show a general increase in accuracy as the learning rate increases, indicating that a higher learning rate allows the models to converge faster and achieve better performance. However, it is important to note that the improvements in accuracy are relatively small between different learning rates.Based on the comparison, the accuracies achieved by both datasets are similar, with the Mafaulda dataset having slightly higher accuracies in most cases. However, the differences in accuracy between the learning rates are not significant for either dataset.

4.6 Ground Truth vs. Predicted Magnitudes

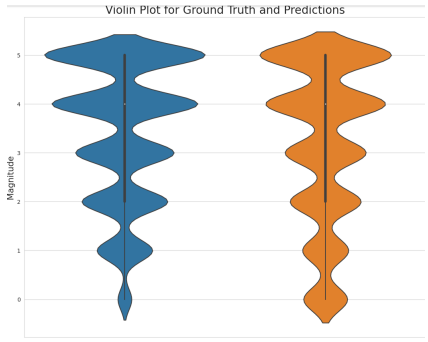

Figure 4.7: (A) Violin plot for Mafaulda

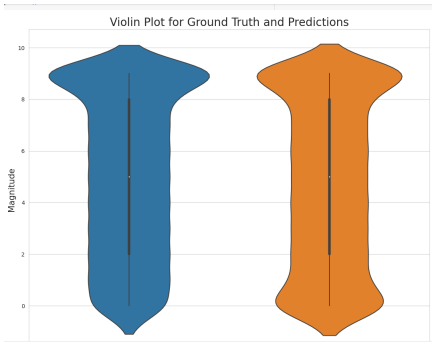

Figure 4.8: (B) Violin plot for CWRU

The Violin Plot is a data visualization technique that combines aspects of a box plot and a kernel density plot. It provides a concise summary of the distribution of a continuous variable, allowing for the comparison of multiple groups or categories.In both datasets,the Violin Plot displays(fig:4.7 and 4.8) the distribution of the ground truth and predicted magnitudes. The x-axis represents the two categories, 'Ground Truth' and 'Predictions', while the y-axis represents the magnitude values.The width of the violins represents the density or frequency of data points at different magnitudes. A wider region indicates a higher density of data points, while a narrower region indicates a lower density.Consequently, if a violin appears wider in specific regions, it indicates a higher concentration of data points at those corresponding magnitudes. In both cases,violin is wider in certain regions,it indicates a higher concentration of data points at those magnitudes.

4.7 Accuracy Comparison: CWRU vs Mafaulda Datasets in Fault Diagnosis

A bar plot that compares the accuracy of two datasets,namely CWRU and Mafaulda.The x-axis will display the dataset names, while y-axis will indicate the accuracy values.The

plot aims to visually represent and compare the performance of these datasets in terms of accuracy.By visualizing the accuracy comparison through a bar plot(fig:4.9),it becomes easier to observe any differences or similarities between the performance of the CWRU and Mafaulda datasets.This plot can be a valuable tool for presenting and analyzing the accuracy metrics of these datasets, aiding in the evaluation and comparison of their respective performance in fault diagnosis .

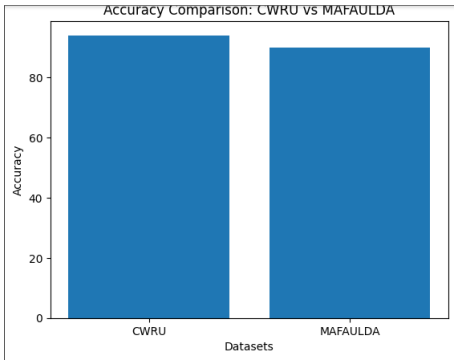

Figure 4.9: Bar Diagram

4.8 Comparison of Distribution Patterns: KDE Plots for Ground Truth and Predicted Magnitudes

A Kernal Density Estimation plot is a non-parametric way to estimate the probability density function of a random variable. By comparing the KDE plots, to gain insights into the distribution and similarity of the ground truth and predicted values.The KDE plot visualizes the density of data points along the magnitude axis. The x-axis represents the magnitude, while the y-axis represents the density.It visualize the distribution of the ground truth labels and the corresponding predictions.Fig(4.10 and 4.11) shows the comparison of KDE plot.The resulting KDE plot provides an overall view of the alignment between the predicted values and the actual values. By analyzing the plot, one can determine how closely the predicted values match the ground truth labels. This information is crucial in evaluating the performance of the model and gaining insights into the accuracy of its predictions.

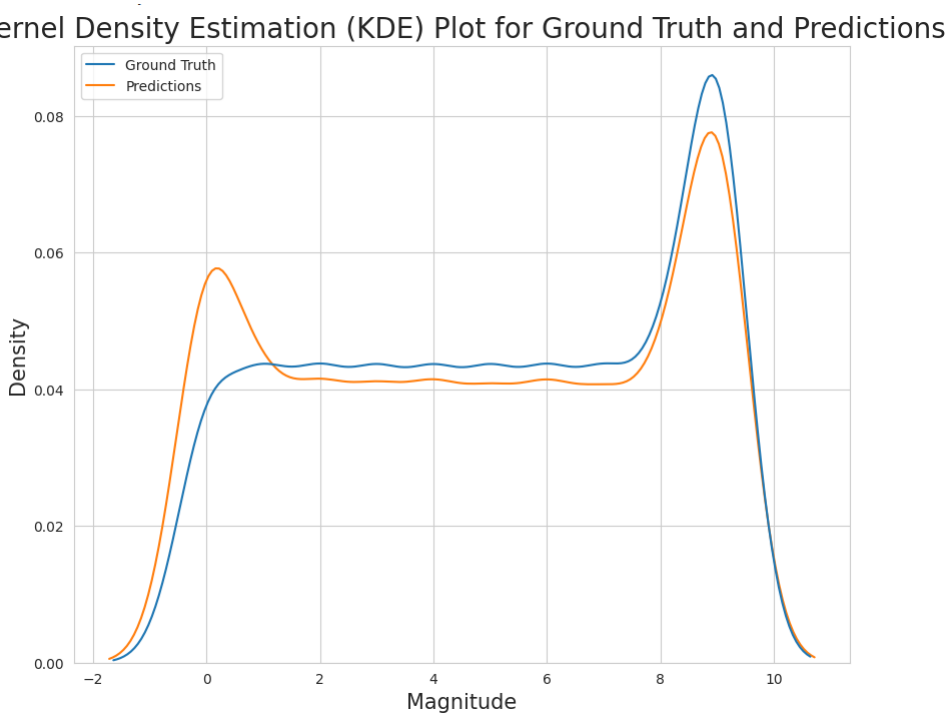

Figure 4.10: (A) KDE plot for CWRU

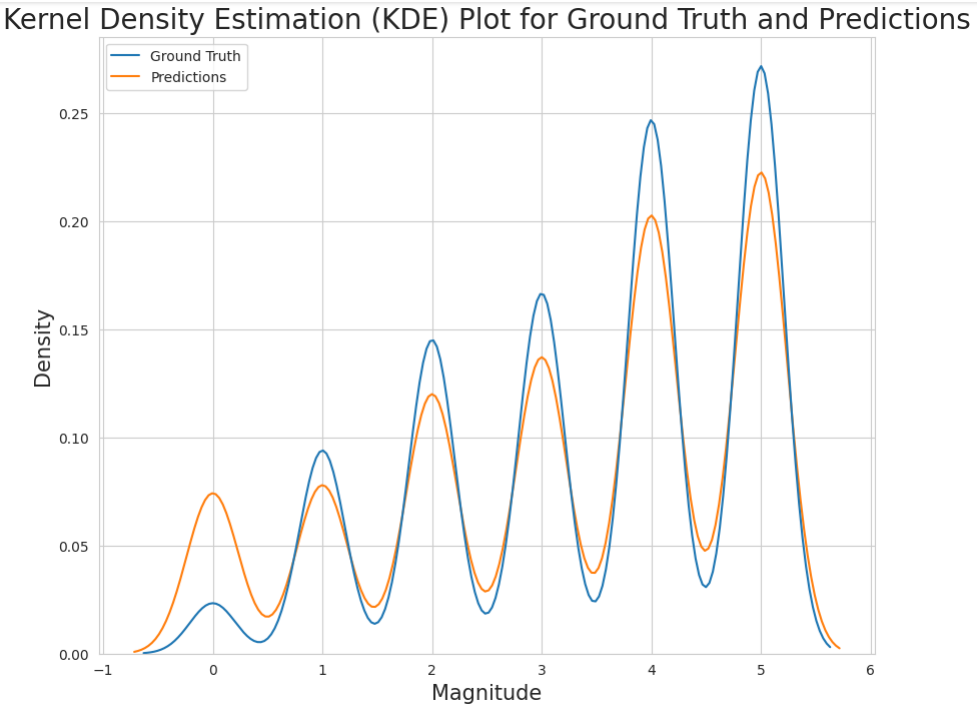

Figure 4.11: (A) KDE plot for Mafaulda

Chapter 5

CONCLUSION

In this study, a deep feature-enhanced reinforcement learning method is proposed for fault diagnosis of rotating machinery. The method integrates various deep learning techniques, including 1D Convolutional Neural Networks (1DCNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU), to effectively extract fault features from one-dimensional vibration signals and capture the correlation among these features. To reduce bias and improve robustness, the RELU activation function is utilized to modify the neural network architecture.

The proposed method also incorporates an autoencoder for feature enhancement. The autoencoder aims to capture essential information and improve the model accuracy by learning a compressed representation of the input data through an encoder network and reconstructing the original input from this representation using a decoder network. By doing so, the autoencoder effectively enhances important features of different datasets, leading to improved fault identification accuracy.Furthermore, a Deep Q-Network module is constructed for fault label prediction. The DQN module optimizes long-term cumulative rewards through reinforcement learning, breaking the dependence on the model structure in previous fault diagnosis methods and achieving higher recognition accuracy.

The superiority and generalization ability of the proposed method are verified through two experimental cases. The results demonstrate that the autoencoder module effectively enhances important features across different datasets, thereby improving fault identification accuracy. Moreover,the reinforcement learning mechanism enables autonomous fault feature learning, capturing temporal dependencies,and accurately detecting and classifying faults.

REFERENCES

[1] H. Jiang, C. Li, H. Li, An improved eemd with multiwavelet packet for rotating machinery multi-fault diagnosis, Mech. Syst. Sig. Process. 36 (2) (2013) 225–239.

[2] M. A. Kester, S. D. Cook, A. F. Harding, R. P. Rodriguez, and C. S. Pipkin, "An evaluation of the mechanical failure modalities of a rotating hinge knee prosthesis," Clinical Orthopaedics and Related Research, vol. 37, no. 228, pp. 156–163, 1988.

[3] M. Z. A. Bhuiyan, J. Wu, G. Wang, Z. Chen, J. Chen, and T. Wang, "Quality-guaranteed event-sensitive data collection and monitoring in vibration sensor networks," IEEE Transactions on Industrial Informatics, vol. 13, no. 2, pp. 572–583, 2017.

[4] L. Jing, M. Zhao, P. Li, and X. Xu, "A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox," Measurement, vol. 111,pp. 1–10, 2017.

[5] L. Zhu, G. Tang, and X. Hu, "Fault identification of gearbox based on BP neural network improved by artificial fish swarm algorithm," International Core Journal of Engineering, vol. 5, no. 8, 2019.

[6] A. Yin, Y. Yan, Z. Zhang, C. Li, and R.-V. S´anchez, "Fault diagnosis of wind turbine gearbox based on the optimized LSTM neural network with cosine loss," Sensors, vol. 20, no. 8, Article ID 2339, 2020.

[7] J. Yang, Y. Guo, and W. Zhao, "Long short-term memory neural network based fault detection and isolation for electro- mechanical actuators," Neurocomputing, vol. 360, pp. 85–96, 2019

[8] Yi C, Yong L, Zhang D, et al. Quaternion singular spectrum analysis using convex optimization and its application to fault diagnosis of rolling bearing[J].Measurement, 2017, 103:321-332.

[9] Cong W, Meng G, Zhu C. Non-negative EMD manifold for feature extraction in machinery fault diagnosis[J]. Measurement, 2015, 70:188-202.

[10] Shao H, Jiang H, Ying L, et al. A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders[J]. Mechanical Systems Signal Processing, 2018, 102:278-297.

[11] Shao H, Jiang H, Zhang H, et al. Rolling bearing fault feature learning using improved convolutional deep belief network with compressed sensing[J]. Mechanical Systems Signal Processing, 2018, 100:743-765.

[12] Shao H, Jiang H, Zhang H, et al. Electric Locomotive Bearing Fault Diagnosis Using a Novel Convolutional Deep Belief Network[J]. IEEE Transactions on Industrial Electronics, 2018, 65(3):2727-2736.

[13] Guo X, Chen L, Shen C. Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis[J]. Measurement, 2016, 93:490-502..

[14] Y. Cheng, M. Lin, J. Wu, H. Zhu, and X. Shao, "Intelligent fault diagnosis of rotating machinery based on continuous wavelet transform-local binary convolutional neural net- work," Knowledge-Based Systems, vol. 216, Article ID 106796,2021.

[15] Y.-M. Hsueh, V. R. Ittangihal, W.-B. Wu, H.-C. Chang, and C.-C. Kuo, "fault diagnosis system for induction motors by CNN using empirical wavelet transform," Symmetry, vol. 11,no. 10, Article ID 1212, 2019.

[16] X. Yuan, T. Zhang, and G. Boscato, "Fault diagnosis for rotating machinery based on convolutional neural network and empirical mode decomposition," Shock and Vibration, vol. 2017, Article ID 3084197, 12 pages, 2017

[17] T. Zhang, Q. Fei, N. Li, and D. Ma, "Fault diagnosis based on modified BiLSTM neural network," in Proceedings of the 2020 5th International Conference on Intelligent Information Tech- nology (ICIIT 2020), pp. 21–26, Hanoi, Viet Nam, February 2020

[18] A. Yin, Y. Yan, Z. Zhang, C. Li, and R.-V. S´anchez, "Fault diagnosis of wind turbine gearbox based on the optimized LSTM neural network with cosine loss," Sensors, vol. 20, no. 8, Article ID 2339, 2020.

[19] K. Amin, M. Khalooei, and M. Rezghi, "End-to-end CNN+LSTMdeep learning approach for bearing fault diag- nosis [J]," Applied Intelligence, vol. 51, pp. 736–751, 2020.

[20] X. Li, W. Zhang, Q. Ding, Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism, Signal Process. 161 (AUG.) (2019) 136–154.

[21] M.S. Rathore, S.P. Harsha, Prognostics analysis of rolling bearing based on bi- directional lstm and attention mechanism, J. Fail. Anal. Prev. 22 (2) (2022) 704–723..

[22] Albarrán-Arriagada F, Retamal J C, Solano E, et al. Measurement-based adaptation protocol with quantum reinforcement learning[J]. 2018, 98(4):042315.

[23] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[J]. Computer Science, 2013.

[24] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[J]. Computer Science, 2013.

[25] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540):529.

[26] Zoph B , Le Q V . Neural Architecture Search with Reinforcement Learning[J]. 2016.

[27] The Case Western Reserve University Bearing Data Center,Bearing Data center Seeded Fault Test Data [EB/OL], 2012.

[28] X. Li, H. Jiang, M. Xie, A reinforcement ensemble deep transfer learning network for rolling bearing fault diagnosis with Multi-source domains, Adv. Eng. Inform. 51 (2022) 101480. ISSN 1474-0346.

[29] W. Dai, Z. Mo, C. Luo, et al., Fault Diagnosis of Rotating Machinery Based on Deep Reinforcement Learning and Reciprocal of Smoothness Index[J], IEEE Sens. J. 20 (15) (2020) 8307–8315

[30] H.e. Zhiyi, S. Haidong, D. Ziyang, J. Hongkai, C. Junsheng, Modified deep auto- encoder driven by multi-source parameters for fault transfer prognosis of aero- engine[J], IEEE Trans. Ind. Electron. 69 (1) (2022) 845–855.

[31] Zhenghong Wu, Hongkai Jiang, Shaowei Liu, Ruixin Wang, A deep reinforcement transfer convolutional neural network for rolling bearing fault diagnosis,ISA Transactions, 2022, ISSN 0019-0578.

[32] Zhao Ke, Jiang Hongkai*, Liu Chaoqiang, et al. A new data generation approach with modified Wasserstein auto-encoder for rotating machinery fault diagnosis with limited fault data[J], Knowledge-Based Systems, 2022, 238: 107892.

[33] S.W. Liu, H.K. Jiang, Z.H. Wu, X.Q. Li, Data synthesis using deep feature enhanced generative adversarial networks for rolling bearing imbalanced fault diagnosis, Mech. Syst. Signal Process. 16 (2022) 108139.

[34] Wei Li, Xiang Zhong, Haidong Shao, Baoping Cai, Xingkai Yang, Multi-mode data augmentation and fault diagnosis of rotating machinery using modified ACGAN designed with new framework[J], Adv. Eng. Inform, 52 (2022) 101552, ISSN 1474-0346.

[35] S.W. Liu, H.K. Jiang, Y.F. Wang, et al., A deep feature alignment adaptation network for rolling bearing intelligent fault diagnosis, Adv. Eng. Inform. 52 (2022) 101598.

[36] De-Cheng Feng, Zhen-Tao Liu, Xiao-Dan Wang, Zhong-Ming Jiang, Shi-Xue Liang, Failure mode classification and bearing capacity prediction for reinforced concrete columns based on ensemble machine learning algorithm[J], Adv. Eng. Inform. 45 (2020) 101126, ISSN 1474-0346.

[37] J. Abualdenien, A. Borrmann, Ensemble-learning approach for the classification of Levels Of Geometry (LOG) of building elements[J], Adv. Eng. Inform. 51 (2022)

[38] R. Wang, H. Jiang, X. Li, S. Liu, A reinforcement neural architecture search method for rolling bearing fault diagnosis, Measurement, 154.

[39] Ribeiro, F. M. L. (2018). Machinery Fault Database(MaFaulDa) - multivariate time-series acquired by sensors on a SpectraQuest's Machinery Fault Simulator (MFS) Alignment-Balance-Vibration

[40] Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in International conference on machine learning, pp. 478–487, New York, USA, 2016.