LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

With the technological advancement of machinery in various fields, failure of the machinery part is a serious issue. Failure of critical machinery part causes not only economic loss but also serious issues in the mechanical equipment used in hospitals, automobiles, agriculture, aircraft engines, marine machinery, and large-scale industries.

1.1 Rotating Machines

Rotating machines are vital devices applied in various sectors and these devices may cause damage due to tough working conditions and long-running duration. Roller Bearing – an important component in almost all rotatory modern industrial machines. Root cause of this failure of the rotating machine is often faults that occur in roller bearing elements as shown in fig 1.1. These faults must be monitored and detected in the early stages.

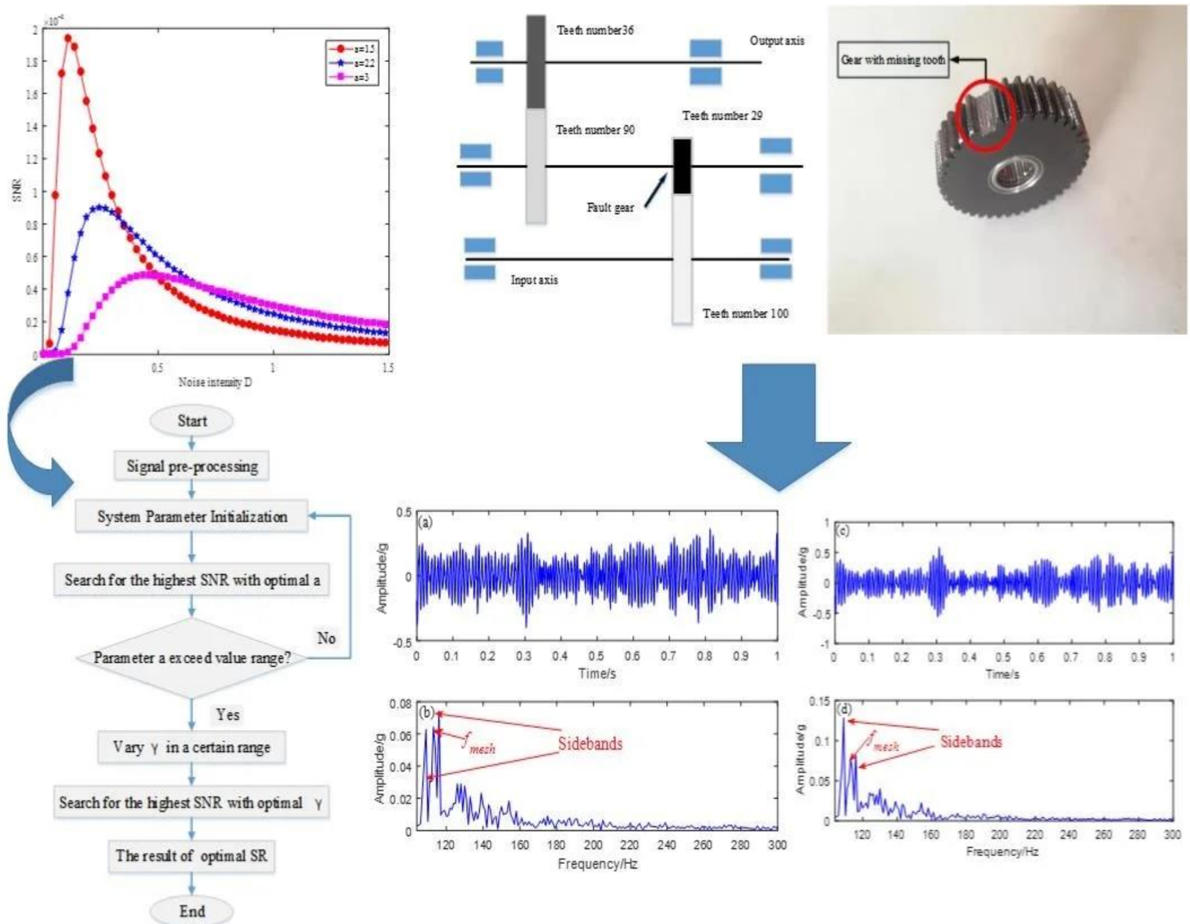

Fig 1.1: Roller Bearing

1.2 Components in Rotating Machine:

The major rotating mechanical components include steam, turbines, fans, pumps, aero engines, roller bearning, and shears. Anomaly Detection is a method to predict the main cause of machinery failure.

1.3 Significance of Anomaly Detection

Anomaly detection emerges as a pivotal method to forecast and forestall machinery failures, specifically targeting the primary cause underlying mechanical breakdowns. This method becomes indispensable in the early identification of potential faults within rotating machinery components, especially roller bearings, to preemptively avert machinery downtime and operational disruptions.

1.4 Traditional Methods

The traditional techniques for identifying bearing faults often involve manual analysis and expert judgment based on limited features or thresholds. The older techniques used to determine faults in roller contain time-domain analysis, frequency-domain analysis, envelope analysis, spectral analysis, and statistical analysis. These techniques involve manual analysis, simplified assumptions, and limited feature extraction capabilities, which can lead to limitations in accuracy, adaptability, and the ability to handle complex fault patterns.

1.5 Deep learning

Spectrograms, derived from the transformation of raw sensor data, offer a visual representation of the signal's frequency content across time intervals. The utilization of deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), on spectrogram data has demonstrated significant potential in automatically learning complex features and patterns associated with different fault conditions.

1.6 Role of Deep Learning in Fault Detection:

The advent of deep learning techniques has revolutionized fault detection methodologies, particularly in the context of rotating machinery. Spectrograms, derived from raw sensor data transformations, offer a comprehensive visual depiction of signal frequency distribution over time intervals. Employing deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) on spectrogram data unveils substantial potential in automatically deciphering complex fault-related features and patterns associated with different fault conditions within roller bearing as shown in fig.1.2.

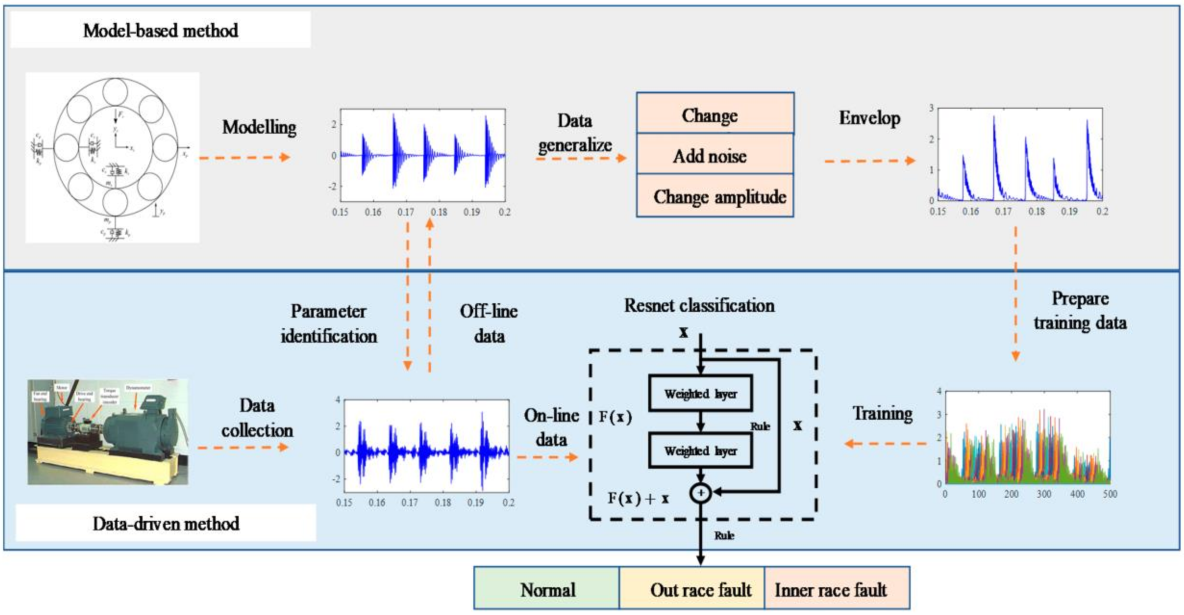

Fig 1.2 Faults Types

This project embarks on an exploration of fault diagnosis methodologies, specifically focusing on the utilization of sensor data derived from machinery components. The datasets employed in this endeavor originate from the Case Western Reserve University (CWRU) Bearing Data Center, encompassing recordings of machinery sensor signals acquired from various fault conditions in rotating machinery.

CHAPTER-2

LITERATURE SURVEY

In this chapter, we thoroughly explore the use of modern technologies like Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) for predicting faults in machinery. We look at the strengths and weaknesses of these technologies and the challenges they face. The focus is on how these smart algorithms can effectively identify different faults by analyzing vibrations in machinery. This exploration helps us understand how AI, ML, and DL play a crucial role in predicting and preventing issues in machinery systems.

Review on Fault Diagnosis

In the research conducted by Mohammed Hakim, Abdoulhdi A. Borhana Omran, Ali Najah Ahmed, Muhannad Al-Waily, Abdallah Abdellatif [1]. The paper provides a systematic review of rolling bearing fault diagnoses using deep learning and transfer learning. It explores the applications of various deep learning algorithms such as Convolutional Neural Networks, Recurrent Neural Networks, Autoencoder, and Generative Adversarial Network for bearing fault detection. The study emphasizes the significance of accurate and quick bearing fault diagnosis, addressing challenges in handling large monitoring data and diverse data types. Deep learning methods offer efficient and accurate bearing fault detection, addressing large data volumes effectively. Handling diverse data types and high data volumes can pose challenges for feature extraction in deep learning-based approaches.

In their study, Lijun Zhang, Yuejian Zhang, Guangfeng Li [3], addresses that the vulnerabilities and failure risks associated with essential components like rolling bearings and gears in rotating machinery. Emphasizing wind turbines as an example, the authors highlight the heavy stress and susceptibility of these components to damage, contributing to high maintenance costs.Issues with traditional methods like wavelet transforms and local mean decomposition (LMD) are noted due to their restrictive requirements and susceptibility to noise. The paper introduces variational-mode decomposition (VMD) and successive VMD (SVMD) as robust signal-processing methods with strong noise resilience, yet they face limitations in under-decomposition. Advantages are SVMD, similar to variational-mode decomposition (VMD), demonstrates strong noise resilience and performance in decomposing fault-vibration signals, ensuring reliable feature extraction. Limitations are Despite its robustness, SVMD, similar to VMD, faces limitations in under-decomposition, which may affect its ability to fully capture complex fault patterns, particularly when dealing with an unknown number of modes.

In the research conducted by Xiaoran Zhang, Kantilal Pitambar Rane, Ismail Kakaravada, and Mohammad Shabaz [4], The research paper discusses the challenges and complexities involved in fault diagnosis for rotating machinery components like rolling bearings and gears, emphasizing the significance of monitoring vibration signals. It introduces a new fault-diagnosis method employing successive variational-mode decomposition

(SVMD) and machine learning, aiming to improve fault detection accuracy. The paper provides insights into signal-decomposition algorithms, their limitations, and proposes a solution to address these challenges. Overall, it offers a structured approach to enhancing fault diagnosis in rotating machinery systems. Advantages are that the paper comprehensively discusses the challenges and complexities associated with fault diagnosis in rotating machinery, highlighting the significance of monitoring vibration signals. Limitations are Implementing IoT technologies, wireless sensor networks, machine learning algorithms, and cloud platforms for fault diagnosis can be complex and may require specialized expertise.

In their work, Jing Zhang, Deqing Zhang, Mingyue Yang, Xiaobin Xu, Weifeng Liu, Chenglin Wen [5] they discusses the challenges of detecting faults in rotating machinery due to complex fault types, noisy sensor signals, and limited fault samples. It proposes a novel method called Deep Fault Diagnosis (DFD) that utilizes a combination of shallow models (such as SVM, ELM) and deep learning (CNN) to address the scarcity of labeled samples. The main idea is to extract time-frequency domain features from vibration signals, train candidate SVM models, select the most discriminative features and best SVM models, and then use them to predict labels for unlabeled samples. These predicted labels, along with scarce labeled samples, form an augmented training set for training a deep CNN model.
Advantages are that DFD combines the strengths of both shallow models (SVM, ELM) and deep learning (CNN) to address the scarcity of labeled samples. This hybrid approach can leverage the advantages of each model type. Limitations are that Deep learning models, particularly CNNs, often demand substantial computational resources and training time, which might limit real-time application possibilities in some scenarios.

In the research conducted by Thanh Tran, Sebastian Bade, and Jan Lundgren [2]. The article explores the development of a machine failure detection system for drilling machines using sound analysis techniques. It discusses the challenges related to data imbalance and the need for an automated system due to the time-consuming nature of manual checks on drill bits. The article suggests data augmentation methods like SMOTE and VAE to address data scarcity and improve the performance of machine learning models. Deep learning approaches, including CNNs and VAEs, are highlighted as potential solutions for accurately analyzing sound signals to detect machine failures. Challenges related to feature extraction from short and complex sound waveforms are acknowledged, and the choice of using sound-based detection over vibration sensors due to practical constraints is explained. Utilizing deep learning models such as CNNs and VAEs offers potential for accurate analysis of sound signals, leading to more robust detection systems.
Implementing deep learning methodologies like VAE might require expertise and computational resources, potentially increasing the complexity of the solution.

In their work, N.Fathiah Waziralilah, Aminudin Abu, M. H Lim, Lee Kee Quen, and Ahmed Elfakharany [9] employed that the information highlights the evolution of CNNs in fault diagnosis, showcasing their potential for improved accuracy and direct feature extraction from raw data. However, challenges like data requirements, varied methods, and the risk of overfitting persist. Further research is crucial to optimize raw data representation, especially for real-time industrial application. Advantages are CNNs improve fault detection accuracy, Different CNN designs offer flexibility, CNNs spot faults directly in data, CNNs often outperform traditional methods, Research finds better data representations for CNNs.

Limitations are Makes comparing findings tricky, CNNs hunger for labelled data, especially for new machines, small datasets may narrow focus, more needed on raw data use, Industries lack data for CNN training.

In their research, David Verstraete, Andrés Ferrada, Enrique López Droguett, Viviana Meruane, and Mohammad Modarres [6] the advancement and integration of deep learning techniques, especially convolutional neural networks (CNNs), into fault diagnosis methodologies for rolling element bearings signify a paradigm shift in predictive maintenance. These methods leverage the power of sophisticated algorithms to analyze complex sensor data, providing a more automated and potentially more accurate means of fault detection and prognostics in rotating machinery. CNNs can learn intricate patterns within sensor data, potentially achieving higher accuracy in fault detection compared to traditional methods. CNNs are often considered as "black-box" models, making it challenging to understand how they arrive at specific decisions, reducing interpretability.

In the research conducted by Dhiraj Neupane, Yunsu Kim, Jongwon Seok, Jungpyo Hong [7] The 1-D CNN method proposed is excellent for detecting bearing faults with high accuracy, using fewer parameters and less computational time. However, its direct application might be limited to similar fault analysis tasks, and its performance heavily depends on the quality of the dataset used for training. Efficient with time-series data, achieves high accuracy with fewer parameters, computationally feasible. Limited applicability outside bearing fault analysis, reliance on dataset quality, challenges in interpretability.

In their work, Haisheng Wang,Jian Wei, Pengjin Li[8] investigated that Deep learning significantly enhances fault diagnosis accuracy, but it relies heavily on quality data and advanced technology, making its implementation complex and resource-demanding. Deep learning for fault diagnosis excels due to its accuracy boost and data-driven approach. It's adaptable across different systems and industries, efficiently extracting intricate data features. This method leads to cutting-edge results, outperforming traditional techniques. Deep learning hinges on quality data but faces challenges with errors and complexity. It demands robust hardware and relies heavily on advanced technology for optimal performance.

In the research conducted by Bo Zhang [10]. The provided information discusses the approach to detecting faults in rolling bearings, highlighting the advantages and disadvantages of utilizing diverse methods, including deep learning, for comprehensive fault analysis. It emphasizes the importance of preserving data relationships, classifying faults, and continuous improvement while acknowledging challenges like complexity, signal preservation, potential oversights, and resource demands. Advantages are Blending methods thoroughly, tailored for precision, focus on crucial correlations, offers clear insights, Committed to continuous improvement. Limitations are Integration complexity may rise, struggles in retaining full periodicity, Possibility of fault interaction overlook, Techniques might miss nuances, Deep learning needs specialized resources.

CHAPTER-3
RESEARCH GAPS OF EXISTING METHODS

3.1 Research Questions in Focus

this section describes six Research Questions (RQs) presented in Table 2, which are raised to provide a detailed review of various broader aspects of recommender systems. the following subsections attempt to present these aspects while answering the RQs.

RQ#	Research Question Statement
RQ1	What are the most effective methods to convert raw sensor data into spectrograms for fault diagnosis purposes?
RQ2	Which feature extraction techniques from spectrograms are most informative for distinguishing different fault conditions?
RQ3	How can transfer learning be effectively applied to adapt pre-trained models for fault diagnosis using spectrogram data?
RQ4	How robust are the developed models to variations in operating conditions, noise, or different types of faults?
RQ5	How does this approach contribute to minimizing downtime, reducing maintenance costs, and enhancing overall system reliability?
RQ6	What are the practical implications of integrating fault diagnosis using spectrograms and deep learning into industrial maintenance routines?

Table 2: Research Questions

Describes the approach taken in the study, including data collection, preprocessing techniques for generating spectrograms, and the selection of deep learning architectures for fault diagnosis. The section outlines the utilization of transfer learning principles and the rationale behind their application.
Details the steps involved in preprocessing raw sensor data to create spectrograms. It explains the transformation process and discusses parameters affecting spectrogram quality and informativeness for fault detection purposes.
Explores the feature extraction methods employed to identify fault signatures from spectrograms. Describes the design and rationale behind chosen deep learning architectures, including CNNs, RNNs, or hybrid models like CRNNs, emphasizing their suitability for processing spectrogram data.
Explores various transfer learning approaches utilized in adapting pre-trained models to the specific task of fault diagnosis. Discusses strategies such as fine-tuning, feature extraction, and model adaptation, highlighting their impact on model performance.

Details the experimental setup, including dataset description, model training procedures, evaluation metrics, and the presentation of experimental results. Provides quantitative and qualitative analyses of model performance in fault detection and condition monitoring tasks.
Analyzes the findings from the experiments, discussing the strengths and limitations of the proposed approach. It also compares the performance of spectrogram-based deep learning models with traditional methods and highlights the implications for industrial applications.

3.2 Strengths and weakness of Fault Diagnosis approaches:

In this subsection, a lot of scientific and technical literature has been reviewed to study and analyze the pros and cons of recommendation approaches. Each of these has its own strengths and weaknesses, which are summarized in Table 3 and queried in the forthcoming research question.

Traditional fault diagnosis methods have been employed for a long time and are well-understood within the industry. These methods often provide clear, interpretable results, allowing experts to understand the reasoning behind fault identifications. They can be effective for diagnosing straightforward and well-defined faults in machinery or systems.
Traditional methods struggle to handle complex fault patterns or anomalies that might have subtle variations. They often rely on predefined rules or heuristics, making them less adaptable to changing or diverse fault scenarios. Their effectiveness heavily relies on the expertise of the personnel defining rules and thresholds for fault detection.

3.3 Issues and Challenges in Fault Diagnosis

Here are some outlines the multifaceted challenges and issues faced in fault diagnosis within industrial systems, encompassing aspects of data, complexity, interpretability, scalability, generalization, integration, and ethical considerations. Addressing these challenges is crucial to the development and implementation of effective fault diagnosis methodologies in real-world industrial settings.

1. Data Quality and Availability:

Limited Labeled Data: Obtaining labeled data for training fault diagnosis models, especially for diverse fault conditions, can be challenging and expensive.
Data Imbalance: Unequal representation of different fault types in datasets can bias the model towards more prevalent faults, affecting its performance on rare fault instances.
Noisy and Incomplete Data: Real-world sensor data often contains noise or missing information, making it harder to discern fault signatures accurately.

2. Complex Fault Patterns:

Non-linear and Complex Fault Signatures: Some faults exhibit intricate patterns that might not be explicitly defined or easily recognized, requiring models capable of handling non-linear relationships.
Multimodal Data: Faults can manifest in multiple ways across various sensors or data sources, requiring techniques to effectively fuse information from different modalities.

3. Interpretability and Explainability:

Black-Box Nature of Models: Deep learning models, while effective, often lack interpretability, making it challenging to understand and trust the decision-making process behind fault predictions.
Need for Explainable AI (XAI): In critical industrial systems, explainability is crucial to understand how and why a model arrives at a particular diagnosis, ensuring trust and facilitating human decision-making.

4. Scalability and Real-Time Implementation:

Computational Resources: Deep learning models might require substantial computational power, hindering their real-time deployment in resource-constrained environments.
Scalability to Large Systems: Adapting fault diagnosis methods to large-scale industrial systems with numerous sensors and complex interactions presents scalability challenges.

5. Generalization and Adaptability:

Generalization to New Fault Conditions: Models trained on specific fault types might struggle to generalize to new, unseen fault scenarios, necessitating continuous adaptation and retraining.
Transferability to Different Systems: The transferability of fault diagnosis models across different types of machinery or systems with varying characteristics poses a significant challenge.

6. Integration with Maintenance Practices:

Operational Integration: Implementing fault diagnosis systems into existing maintenance workflows and integrating them with operational practices can be challenging.
Cost and ROI: Demonstrating the cost-effectiveness and return on investment (ROI) of implementing advanced fault diagnosis systems is essential for adoption.

7. Ethical and Safety Considerations:

Safety-Critical Environments: Fault diagnosis in industries involving safety-critical systems requires rigorous validation and safety measures to prevent catastrophic failures.

Features in Fault Diagnosis:

Feature Engineering from Sensor Data- Extraction of relevant features from sensor data (e.g., vibration, temperature, pressure) to characterize fault signatures.
Spectrogram Generation: Transforming raw sensor signals into spectrograms to capture time-frequency representations of fault-related patterns.
Multimodal Data Fusion: Integration of information from diverse sensors or data sources to enhance fault identification.
Model Interpretability: Incorporating features that allow interpretation of model decisions, contributing to explainability.

Challenges in Fault Diagnosis:

Data Scarcity and Quality- Limited labeled data availability and issues related to noise, incompleteness, and imbalance in datasets.
Complex Fault Patterns: Detection and recognition of non-linear, complex fault signatures that might vary across different operating conditions.

Interpretability and Explainability: Understanding and trust in the decision-making process of fault diagnosis models, especially deep learning-based approaches.
Computational Resources: Demands for significant computational power hindering real-time implementation in resource-constrained environments.
Generalization and Adaptability: Ability of models to generalize to new fault conditions and transfer across diverse industrial systems.

Overspecialization and Dependency:

Overfitting to Specific Faults - Models becoming overly specialized and performing well only on certain prevalent fault types, leading to poor performance on rare or new faults.
Dependency on Labeled Data- Overreliance on labeled data for training, limiting the adaptability of models to new or unseen fault scenarios.
Narrow Focus and Lack of Robustness: Models might be too narrowly trained on specific fault patterns, lacking robustness in handling variations or changes in operating conditions.

Quality and Other Aspects:

Quality Assurance in Data Collection- Ensuring high-quality labeled datasets with minimal noise and comprehensive coverage of various fault types.
Ethical Considerations: Safety-critical environments requiring stringent validation and adherence to ethical guidelines.
Integration with Maintenance Practices: Seamless integration of fault diagnosis systems into existing maintenance workflows and practices.
Cost-Benefit Analysis: Demonstrating the cost-effectiveness and tangible benefits of implementing fault diagnosis systems to justify investment and adoption.

Explainability and Interpretability:

Model Transparency- Ensuring that the decision-making process of fault diagnosis models is transparent and understandable, especially in safety-critical systems.
Explainable AI (XAI): Incorporating techniques and methodologies to make the inner workings of complex models interpretable for domain experts and stakeholders.

Continuous Learning and Adaptation:

Dynamic Model Updating - Establishing mechanisms for continuous learning and model adaptation to accommodate evolving fault patterns and system changes.
Incremental Learning: Implementing strategies to incrementally update models with new data while preserving previously learned knowledge.

Human-Machine Collaboration:

Integrating human expertise with automated fault diagnosis systems to validate and refine model predictions. Augmented Intelligence: Leveraging machine capabilities to assist human operators in identifying, interpreting, and resolving faults efficiently.

The key features involved in fault diagnosis, the challenges encountered, concerns related to overspecialization and dependency, considerations regarding quality and integration, and various other aspects crucial for developing effective fault diagnosis methodologies in industrial systems. Addressing these facets is essential for advancing fault diagnosis techniques to ensure reliability, efficiency, and safety in industrial operations.

3.4 Existing Solutions

In Fault diagnosis are being developed or implemented to address the challenges and aspects related to fault diagnosis in industrial systems:

1. Data Augmentation and Synthetic Data:

Generating synthetic data to supplement limited labeled datasets, addressing data scarcity and imbalance issues. Helps in improving model robustness by providing additional diverse examples for training.

2. Hybrid Models and Ensemble Methods:

Integrating traditional rule-based methods with deep learning approaches to combine interpretability with complex pattern recognition. Mitigates the black-box nature of deep learning models, offering insights into decision-making while leveraging the power of deep learning for intricate fault patterns.

3. Transfer Learning and Pre-trained Models:

Utilizing transfer learning with pre-trained models to leverage learned representations and adapt them to new fault conditions.
Addresses data scarcity by transferring knowledge from related tasks or domains, enhancing model performance with limited labeled data.

4. Explainable AI (XAI) Techniques:

Implementing XAI methods to provide transparency and interpretability in deep learning models' decision-making processes.
Improves trust and acceptance of model outputs, enabling domain experts to understand and validate model predictions.

5. Federated Learning and Edge Computing:

Employing federated learning techniques and edge computing for fault diagnosis at the edge devices to reduce latency and ensure real-time analysis.
Enables decentralized model training and inference, preserving data privacy and reducing dependence on centralized systems.

6. Continuous Learning and Adaptive Systems:

Implementing algorithms for continuous learning that adapt to evolving fault patterns and changes in operating conditions.
Ensures fault diagnosis models remain up-to-date and effective in detecting new or rare fault scenarios.

7. Collaborative Platforms and Standards:

Establishing collaborative platforms and industry standards to share datasets, methodologies, and best practices.
Facilitates knowledge sharing, fosters collaboration across industries, and promotes standardized fault diagnosis approaches.

8. Context-Aware Fault Diagnosis:

Incorporating contextual information (e.g., environmental factors, operational conditions) into fault diagnosis models.
Enhances model accuracy by considering the context in which faults occur, leading to more precise diagnosis and fewer false alarms.

The aspects of fault diagnosis challenges, aiming to improve the accuracy, interpretability, adaptability, and efficiency of fault diagnosis systems in industrial settings. Continued research and innovation in these areas are crucial to further advancing fault diagnosis methodologies and ensuring the reliability of industrial systems.

3.5 Case Western Reserve University (CWRU) Bearing Dataset

The dataset utilized in this project stems from the Case Western Reserve University (CWRU) Bearing Dataset—a comprehensive collection designed to advance research in fault diagnosis and condition monitoring of rotating machinery. This dataset specifically focuses on the drive end bearing of electric motors, capturing vibration signals through accelerometers. Key aspects of the dataset employed in this project are detailed below:

- **Objective and Significance:** The primary aim of the CWRU Bearing Dataset is to serve as a benchmark for studying fault diagnosis and predictive maintenance in bearings. Vibration signals, acquired at a high sampling frequency of 48 kHz, enable researchers to delve into the effects of various fault types and operating conditions on machinery health.
- **Drive End Bearing:** The dataset centers on the drive end bearing, a critical component in rotating machinery. Faults in this bearing can have profound effects on system performance.
- **Horsepower Variations:** The dataset includes data corresponding to different motor power levels—0hp, 1hp, 2hp, and 3hp. Variations in motor power offer insights into how changes in operating conditions impact vibration patterns and, consequently, fault detection.
- **Fault Types and Severity Levels:** Common fault types encompass outer race faults, inner race faults, and ball faults. Different severity levels of these faults are incorporated into the dataset, allowing for a nuanced exploration of fault detection capabilities.
- **Data Organization:** The dataset is structured into subdirectories, each reflecting specific operating conditions, fault types, and motor power levels. Individual data files, stored in MATLAB (.mat) format, represent distinct measurements or runs under the specified conditions.
- **Machine Learning Application:** Researchers and practitioners leverage the CWRU dataset for developing and evaluating machine learning models. Deep learning techniques, particularly convolutional neural networks (CNNs) and recurrent neural

networks (RNNs), are commonly applied to automate the detection and classification of faults based on vibration signals.

- **Research Impact:** The dataset has been extensively utilized in research studies and publications, establishing it as a foundational resource for assessing the efficacy of various fault diagnosis algorithms.

In summary, the 48k drive end bearing fault data from 0hp to 3hp within the CWRU dataset serves as a pivotal component of this project. It enables a focused exploration of fault detection capabilities under varying motor power conditions, contributing to the broader field of predictive maintenance and condition monitoring of rotating machinery.

3.6 Bearing Fault Types in Rotating Machinery:

1. Outer Race Faults:

Definition: Outer race faults involve damage to the outer ring of the bearing, the component encompassing and supporting the entire bearing structure.

Causes:

Misalignment: Incorrect bearing alignment induces stress on the outer race.
Overloading: Excessive loads or forces lead to fatigue and damage.
Contamination: Ingress of particles causes wear and pitting.
Manifestations:
Localized Pitting: Small, localized depressions on the outer race surface.
Spalling: Development of small, loose particles due to material fatigue.
Cracking: Structural cracks on the outer race.

2. Inner Race Faults:

Definition: Inner race faults involve damage to the inner ring of the bearing, encircling the rotating shaft.

Causes:

Misalignment: Incorrect bearing alignment induces stress on the inner race.
Overloading: Excessive loads or forces lead to fatigue and damage.
Contamination: Ingress of particles causes wear and pitting.
Manifestations:
Localized Pitting: Small, localized depressions on the inner race surface.
Spalling: Development of small, loose particles due to material fatigue.
Cracking: Structural cracks on the inner race.

3. Ball Faults:

Definition: Ball faults refer to defects in the rolling elements (balls) of the bearing.

Causes:

Insufficient Lubrication: Lack of proper lubrication leads to increased friction and wear on the balls.
Contamination: Foreign particles entering the bearing cause damage to the balls.
Overloading: Excessive loads or forces lead to fatigue and pitting.
Manifestations:

Pitting: Localized surface damage in the form of small depressions on the ball.
Cracking: Structural cracks on the surface of the ball.
Surface Damage: Abrasions or irregularities on the ball surface.

Understanding these distinct fault types is pivotal for effective fault diagnosis in rotating machinery. The comprehensive analysis of vibration data, especially from datasets like the CWRU Bearing Dataset, enables the development of machine learning models capable of automatically recognizing and classifying these faults. This, in turn, contributes to the implementation of proactive and targeted predictive maintenance strategies.

CHAPTER-4

PROPOSED MOTHODOLOGY

The reliability of industrial machinery is the cornerstone of efficient and uninterrupted operations. Early detection of faults, particularly in critical components like roller bearings, is instrumental in preventing costly unplanned downtime. Identifying faults in roller bearings before they escalate is a complex task that demands a systematic approach. This presentation aims to introduce a comprehensive methodology for fault diagnosis in roller bearings, offering a proactive solution to enhance industrial machinery reliability.

4.1 Dataset Exploration and Organization:

Objective: Organize and explore the CWRU dataset to understand its structure and contents.
Steps:

- Organize the dataset into a structured directory.
- Use os.path.join or f-strings for robust path creation.
- Iterate through the directory using os.walk.
- Load sample .mat files to inspect their structure using mat.keys().

4.2 Data Loading and Preprocessing:

Objective: Load vibration data from .mat files and create a structured DataFrame for analysis.
Steps:

- Create an empty DataFrame with columns 'DE_data' and 'fault'.
- Iterate through files, load data, and create fault labels.
- Concatenate individual DataFrames into a single DataFrame.
- Visualize individual fault samples for a better understanding of the data.

4.3 Data Exploration and Visualization:

Objective: Explore the characteristics of the vibration data.
Steps:

- Plot individual fault samples to visualize vibration patterns.
- Utilize scatter plots or line plots to understand variations across different faults.
- Explore statistical measures for key insights.

4.4 Data Preprocessing and Saving:

Objective: Prepare the data for model training and save it for future use.
Steps:

- Save the preprocessed DataFrame to a CSV file.
- Ensure data is normalized if needed.

4.5 Convolutional Neural Network (CNN) Model Design:

Objective: Design a CNN model for fault diagnosis.
Steps:

- Choose appropriate input dimensions based on the data shape.
- Design a CNN architecture suitable for feature extraction from vibration data.
- Specify activation functions and layer configurations.

4.6 Model Training:

Objective: Train the CNN model on the preprocessed data.
Steps:

- Split the data into training and testing sets.
- Compile the CNN model with appropriate loss function and optimizer.
- Train the model for a specified number of epochs.

4.7 Model Evaluation and Visualization:

Objective: Assess the model's performance and visualize results.
Steps:

- Generate predictions on the test set.
- Create a confusion matrix and visualize it using a heatmap.
- Display a classification report and additional performance metrics.
- Analyze and interpret results.

4.8 Documentation and Refinement:

Objective: Document findings, insights, and refine the code for future reference.
Steps:

- Document key parameters, architecture, and performance metrics.
- Refine code for clarity and potential improvements.
- Consider adding comments for better understanding.

CHAPTER-5
OBJECTIVES

The primary objective of this project is to develop an advanced fault diagnosis system for roller bearings based on vibration data. The system aims to leverage state-of-the-art techniques, including data exploration, preprocessing, visualization, and the implementation of a Convolutional Neural Network (CNN), to automate the identification of different fault modes in roller bearings. The specific objectives include:

Data Exploration and Understanding:

- Explore the dataset containing vibration data from roller bearings to gain insights into its structure and characteristics.
- Identify the types and distributions of faults present in the roller bearing dataset.

Data Preprocessing:

- Develop an efficient and systematic approach for loading and preprocessing vibration data from roller bearings.
- Extract relevant features from the vibration data to represent distinctive patterns associated with different fault modes.

Visualization of Fault Patterns:

- Visualize the vibration patterns corresponding to various fault types in roller bearings.
- Utilize scatter plots and other visualization techniques to highlight unique features indicative of specific faults.

CNN Model Implementation:

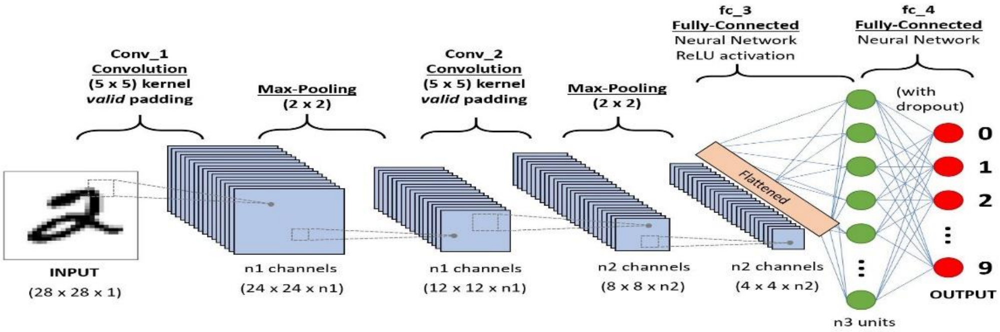

Figure 5.1 CNN Architecture

- Design a specialized CNN architecture tailored for learning and capturing fault-related features from vibration data.
- Implement the CNN model with appropriate activation functions, layers, and optimization techniques.

Model Training and Evaluation:

- Split the roller bearing dataset into training and testing sets.
- Train the CNN model on the training set, monitoring its performance over epochs.
- Evaluate the model on the test set using key metrics such as accuracy, precision, recall, and F1 score.

Documentation and Insights:

- Document the key parameters, hyperparameters, and architectural choices made during the CNN model implementation.
- Provide insights into the system's effectiveness in accurately diagnosing various fault modes in roller bearings based on vibration data.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

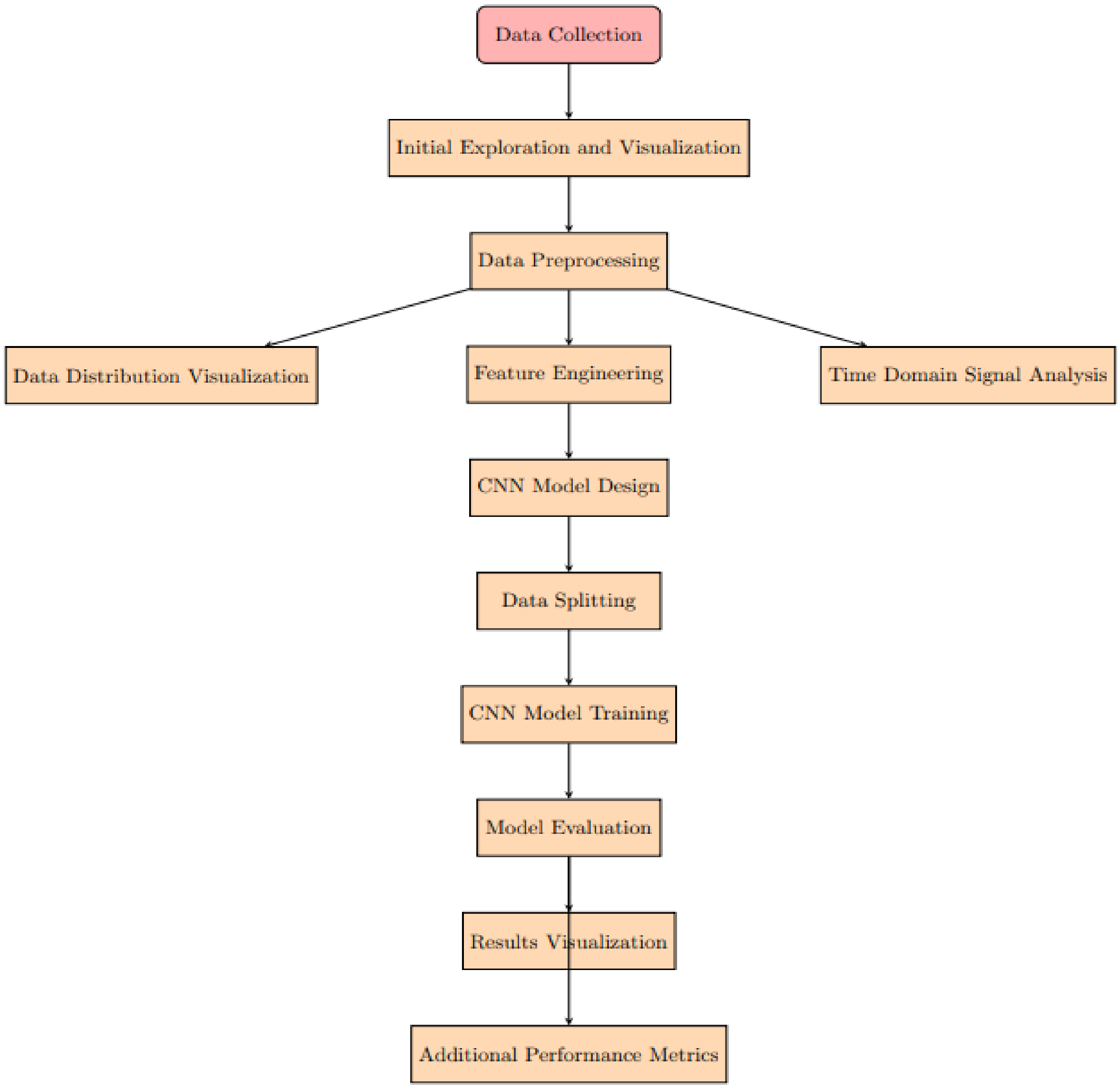

Figure 6.1 Flowchart - Fault Diagnosis in Drive End Bearings using CNN-based Machine Learning

The flowchart illustrates the step-by-step process involved in the Fault Diagnosis in Drive End Bearings using CNN-based Machine Learning. The key steps include data collection, initial exploration, data preprocessing, feature engineering, CNN model design, data splitting, model training, model evaluation, results visualization, data distribution visualization, time domain signal analysis, and additional performance metrics.

6.1 Code 1: Data Exploration, Processing, and Visualization System Design:

Data Exploration:

- Traverse through the dataset directory using os.walk.
- Print the paths of all files within the directory.

Loading and Plotting a Single File:

- Load a specific '.mat' file (OR007_0.mat) using scipy.io.loadmat.
- Extract relevant data from the loaded file.
- Create a DataFrame (df_temp) with the extracted data.
- Plot the 'DE_data' column from df_temp.

Batch Processing Files and Data Concatenation:

- Iterate through all '.mat' files in the dataset directory.
- Load each file, extract vibration data (DE_data), and create a simplified fault label.
- Concatenate individual DataFrames into a single DataFrame (df).
- Display unique fault labels.

Saving Data to CSV:

- Save the resulting DataFrame (df) to a CSV file ('0hp_all_faults.csv') for future use.
- Individual Fault Plots:
- Iterate through unique fault labels and create individual plots for each fault.

Scatter Plot:

- Create a scatter plot for a subset of the DataFrame, visualizing variations in 'DE_data' with different fault labels.

Implementation:

- Utilizes libraries such as scipy, seaborn, numpy, pandas, and matplotlib.
- Employs os.walk for directory traversal and scipy.io.loadmat for loading '.mat' files.
- Creates and manipulates DataFrames for efficient data storage and processing.
- Saves the processed data to a CSV file for future use.
- Generates plots to visualize individual faults and variations in the dataset. .

6.2 Code 2: Convolutional Neural Network (CNN) for Fault Diagnosis System Design:

Data Loading and Preprocessing:

- Load preprocessed data from a CSV file into a DataFrame (df).
- Extract segments of data (X) and corresponding labels (Y) using a sliding window approach.

Data Reshaping and Encoding:

- Reshape the data (X) to fit the input shape of the CNN.

- One-hot encode the labels (Y) using to_categorical.

Model Architecture:

- Define a sequential CNN model with convolutional and pooling layers, followed by fully connected layers.
- Configure the model with 'tanh' activation functions and softmax activation in the output layer.

Model Compilation and Training:

- Compile the model using categorical cross-entropy loss and stochastic gradient descent (SGD) optimizer.
- Split the data into training and testing sets using train_test_split.
- Train the model for a specified number of epochs.

Prediction and Evaluation:

- Make predictions on the test set.
- Transform predictions and true labels back to their original form.
- Generate a confusion matrix, classification report, and additional performance metrics.

Visualization:

- Plot a normalized confusion matrix using seaborn and matplotlib.
- Display a classification report and additional performance metrics.

Implementation:

- Utilizes libraries such as scipy, seaborn, numpy, pandas, tensorflow, sklearn, and matplotlib.
- Implements a CNN model using the Keras API within TensorFlow.
- Applies data preprocessing steps such as reshaping and one-hot encoding.
- Compiles and trains the CNN model on the preprocessed data.
- Evaluates the model performance using metrics like accuracy, precision, recall, and F1 score.
- Visualizes the results through a confusion matrix and classification report.
- Considerations and Suggestions:
- Verify correct file paths and names in both codes.
- Experiment with different hyperparameters for optimal model performance.
- Visualize training history to monitor model convergence and potential overfitting.
- Tailor the system design based on specific project requirements and dataset characteristics.

CHAPTER-7
TIMELINE FOR EXECUTION OF PROJECT
(GANTT CHART)

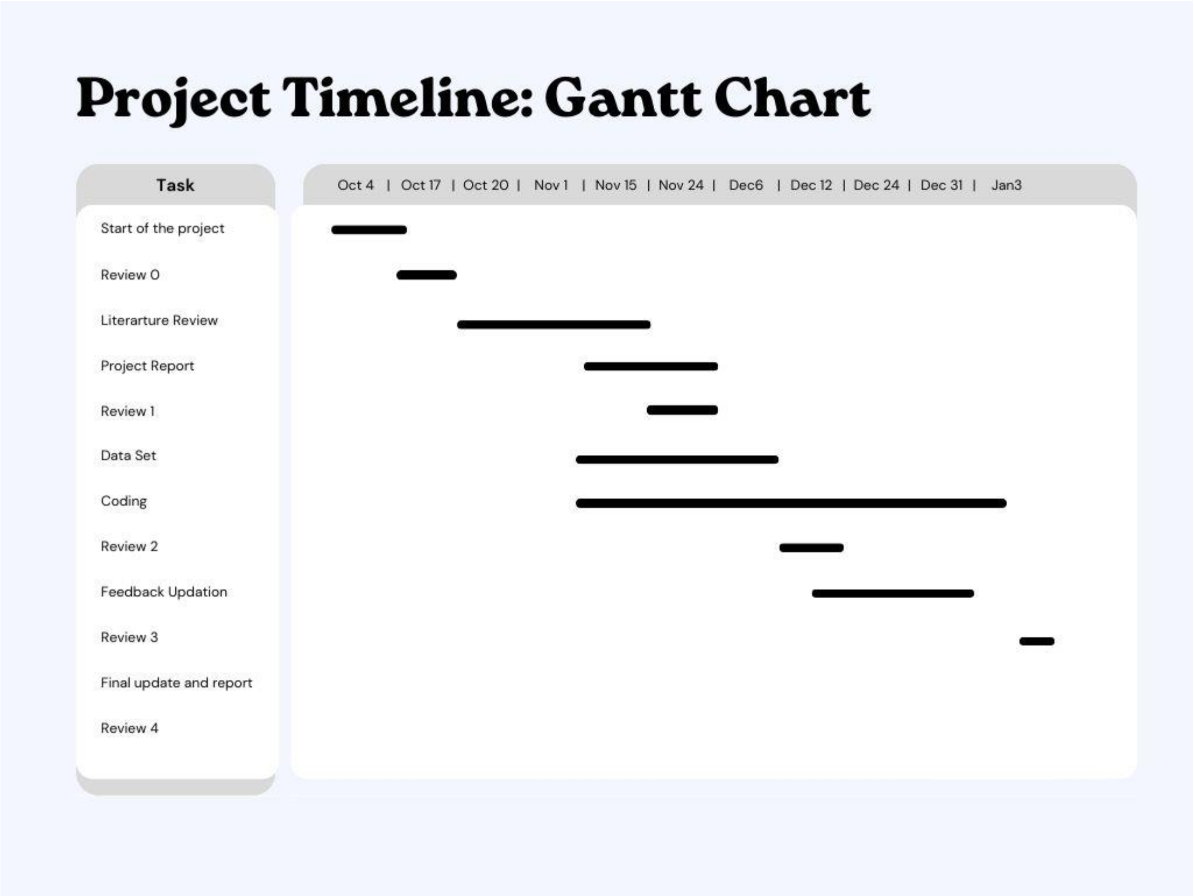

Fig .7.1 Gantt chart

CHAPTER-8

OUTCOMES

In the realm of machinery fault diagnosis, the implementation of the CNN-based fault diagnosis system for the CWRU 48k Drive End Bearing Fault Dataset has resulted in transformative outcomes. The project serves as a central hub, providing real-time insights into the health of industrial machinery and creating a dynamic database that revolutionizes how maintenance teams respond to emerging issues.

The primary and most pivotal outcome of the project is its substantial contribution to minimizing machinery faults. Through early detection capabilities, the system acts as the foundation for a rapid and proactive response to potential faults, effectively reducing the overall incidence of machinery-related issues. This positive impact aligns seamlessly with the broader objective of enhancing machinery health and preventing the escalation of faults.
Beyond its direct impact on fault prevention, the system has ushered in a paradigm shift in industrial machinery health assessment. Serving as a reliable source of fault data, the application forms the basis for understanding fault patterns, thereby enhancing the overall effectiveness of machinery maintenance programs. The analysis of existing data reveals subtle and evolving fault patterns, bolstering resilience and aiding in the prevention of machinery breakdowns.

Real-time execution of fault diagnosis has become an indispensable element in supporting maintenance processes. Stakeholders, including maintenance departments and relevant industrial organizations, benefit from the immediate, accurate information about the location and severity of potential faults. This new approach empowers these organizations with the tools needed to develop efficient strategies and response plans, ultimately reducing the impact of machinery faults.

A secondary but significant benefit of the fault diagnosis system is its role in reducing maintenance costs associated with machinery faults. By acting as a preventive measure against the spread of faults, the system contributes to the broader societal goal of optimizing maintenance resources and lowering economic costs associated with machinery health.
Moreover, the application promises various societal benefits, extending beyond its technical outcomes. It serves as a powerful educational tool, raising awareness among industrial communities about potential machinery faults, risks, and preventive measures. This increased awareness plays a pivotal role in fostering a culture of proactive machinery maintenance and creating a healthier and more efficient industrial environment.

The fault diagnosis system transcends its role as a passive informant and evolves into a source of active community support. Providing local, real-time information about machinery health, the application empowers industrial communities to take crucial steps in protecting their machinery assets. This approach not only promotes community participation in machinery health but also enhances understanding of organizational processes and the role of community members.

An essential facet of the system's potential impact lies in its contribution to improving machinery health and reliability. By providing real-time information on potential faults, the system aids maintenance departments in identifying and addressing issues promptly, promising a safer and more reliable machinery infrastructure.

The outcomes of this fault diagnosis project extend far beyond immediate fault control. The system has emerged as a versatile tool that not only reduces fault transmission but also improves machinery health surveillance, enables more effective response to emerging issues, and strengthens industrial communities. A positive force in industrial machinery health, the application has the potential to bring about transformative change in how society responds to machinery faults, reducing the risk of unplanned machinery downtime and enhancing overall industrial productivity.

CHAPTER-9
RESULTS AND DISCUSSIONS

In this section, we delve into the results obtained from the implemented fault diagnosis system for drive end bearings using a Convolutional Neural Network (CNN). The analysis begins with an exploration of the time domain signals, providing an understanding of the intricate patterns associated with different faults. Subsequently, we showcase the distribution of data points in the feature space, offering insights into how the model could potentially differentiate between fault categories.

The CNN-based machine learning model is then trained and evaluated, with a detailed examination of training and validation metrics, confusion matrix, and classification report metrics. Finally, the results are visualized, providing a qualitative assessment of the model's predictions on randomly selected samples.

9.1 Data Preprocessing Results:

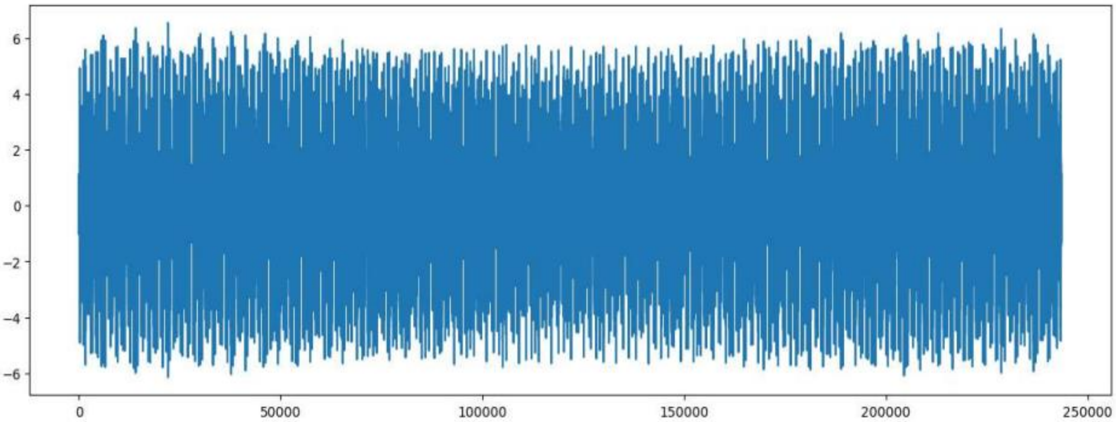

Figure 9.1 Sample Vibration Data

Figure 9.1 provides a visual representation of a sample of vibration data extracted from the CWRU dataset. The time domain signals showcase the intricate patterns associated with different fault types in drive end bearings. Notably, variations in amplitude, frequency, and transient behavior can be observed, forming the basis for subsequent machine learning analysis.

9.1.1 Time Domain Signals for Different Faults:

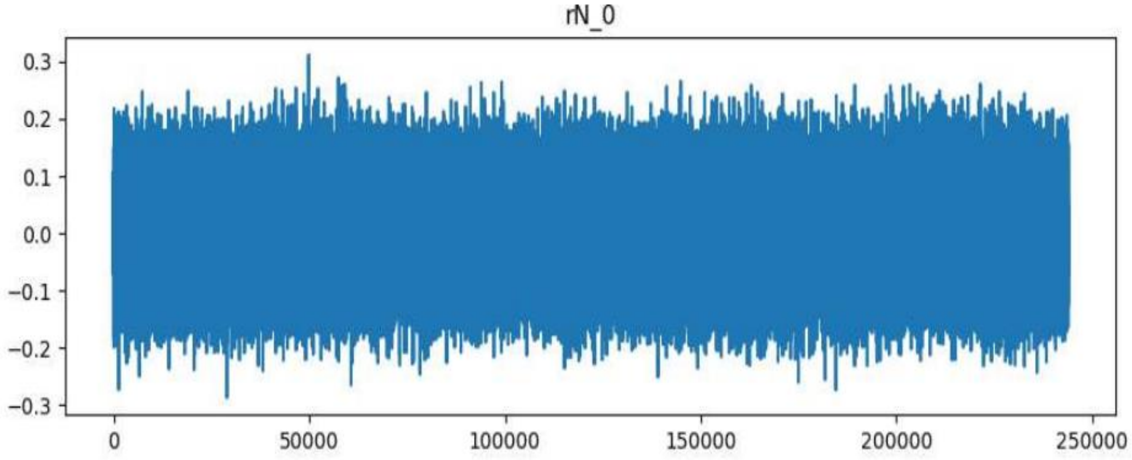

Figure 9.2 (a): Time-domain feature representation for Normal Bearing

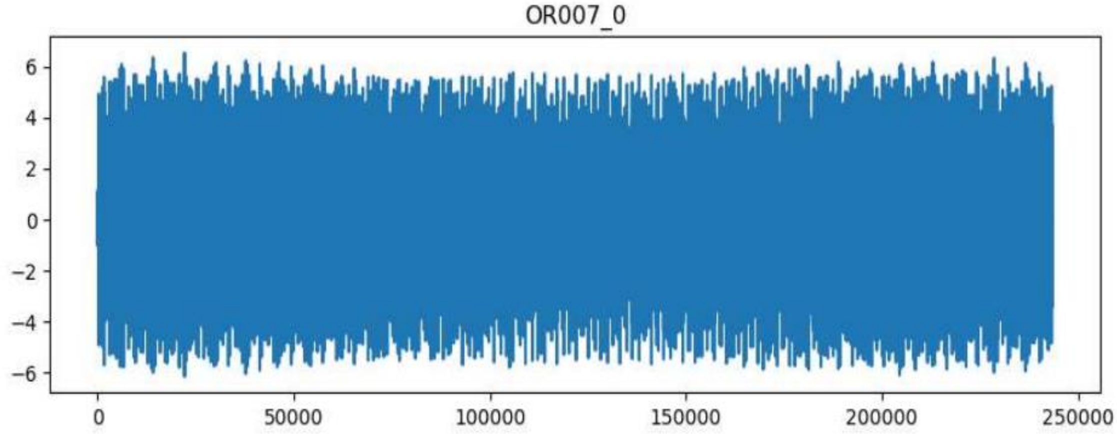

Figure 9.2 (b): Time-domain feature representation for Outer Race Fault

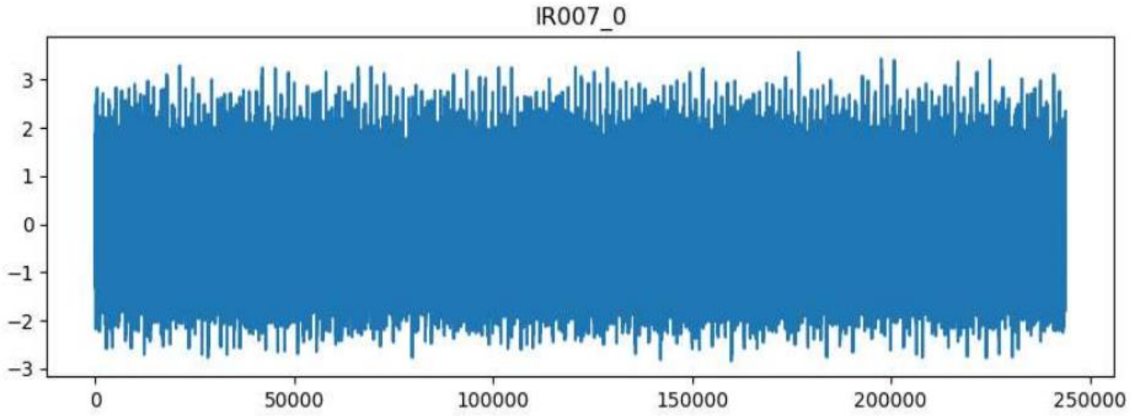

Figure 9.2 (c): Time-domain feature representation for Inner Race Fault

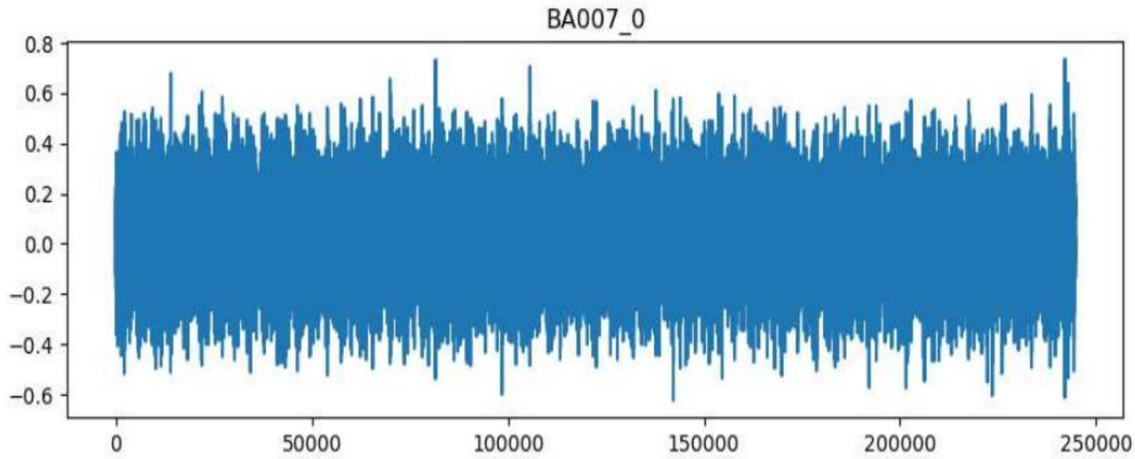

Figure 9.2 (d): Time-domain feature representation for Ball Fault

Figure 9.2 further delves into the time domain signals, specifically highlighting instances of inner race, outer race, and ball faults. Each subplot illustrates the distinctive characteristics of these fault types, aiding in the understanding of the underlying vibration patterns. Inner race faults typically manifest as repetitive impacts, outer race faults as irregular patterns, and ball faults as periodic impulses.

9.1.2 Visualizing Data Distribution:

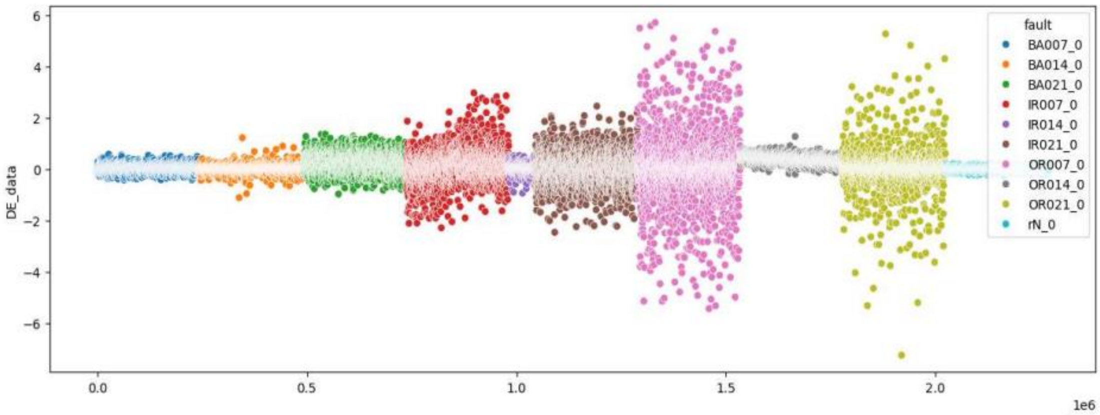

Figure 9.3 Data Distribution Visualization

Figure 9.3 showcases the distribution of data points in the feature space. The scatter plot visualizes the data distribution by displaying every 100th data point, with the x-axis representing the index of data points and the y-axis representing the corresponding 'DE_data' values. Different fault types are distinguished by color, providing insights into how the model could potentially learn and differentiate between these fault categories.

9.2 CNN Implementation Results:

```
42/42 [==============================] - 3s 46ms/step - loss: 2.1245 - accuracy: 0.1985 - val_loss: 1.9160
- val_accuracy: 0.2978
Epoch 2/50
42/42 [==============================] - 2s 37ms/step - loss: 1.7203 - accuracy: 0.4630 - val_loss: 1.5256
- val_accuracy: 0.6293
Epoch 3/50
42/42 [==============================] - 2s 38ms/step - loss: 1.3393 - accuracy: 0.6325 - val_loss: 1.1880
- val_accuracy: 0.6611
Epoch 4/50
42/42 [==============================] - 1s 35ms/step - loss: 1.0731 - accuracy: 0.6880 - val_loss: 0.9981
- val_accuracy: 0.6836
Epoch 5/50
42/42 [==============================] - 2s 39ms/step - loss: 0.9221 - accuracy: 0.7255 - val_loss: 0.8812
- val_accuracy: 0.7919
Epoch 6/50
42/42 [==============================] - 1s 35ms/step - loss: 0.8225 - accuracy: 0.7945 - val_loss: 0.7945
- val_accuracy: 0.7477
Epoch 7/50
42/42 [==============================] - 2s 38ms/step - loss: 0.7459 - accuracy: 0.8163 - val_loss: 0.7271
- val_accuracy: 0.7901
Epoch 8/50
42/42 [==============================] - 2s 36ms/step - loss: 0.6838 - accuracy: 0.8316 - val_loss: 0.6706
- val_accuracy: 0.8489
Epoch 9/50
42/42 [==============================] - 2s 36ms/step - loss: 0.6312 - accuracy: 0.8634 - val_loss: 0.6247
- val_accuracy: 0.8246
Epoch 10/50
42/42 [==============================] - 1s 35ms/step - loss: 0.5850 - accuracy: 0.8754 - val_loss: 0.5787
- val_accuracy: 0.8524
Epoch 11/50
42/42 [==============================] - 1s 33ms/step - loss: 0.5436 - accuracy: 0.8812 - val_loss: 0.5396
- val_accuracy: 0.8767
Epoch 12/50
42/42 [==============================] - 1s 35ms/step - loss: 0.5054 - accuracy: 0.8951 - val_loss: 0.5057
- val_accuracy: 0.8719
Epoch 13/50
42/42 [==============================] - 1s 36ms/step - loss: 0.4710 - accuracy: 0.8998 - val_loss: 0.4733
- val_accuracy: 0.8922
Epoch 14/50
42/42 [==============================] - 2s 36ms/step - loss: 0.4385 - accuracy: 0.9081 - val_loss: 0.4523
- val_accuracy: 0.8683
Epoch 15/50
```

Figure 9.4 Training, Validation Loss and Validation Accuracy

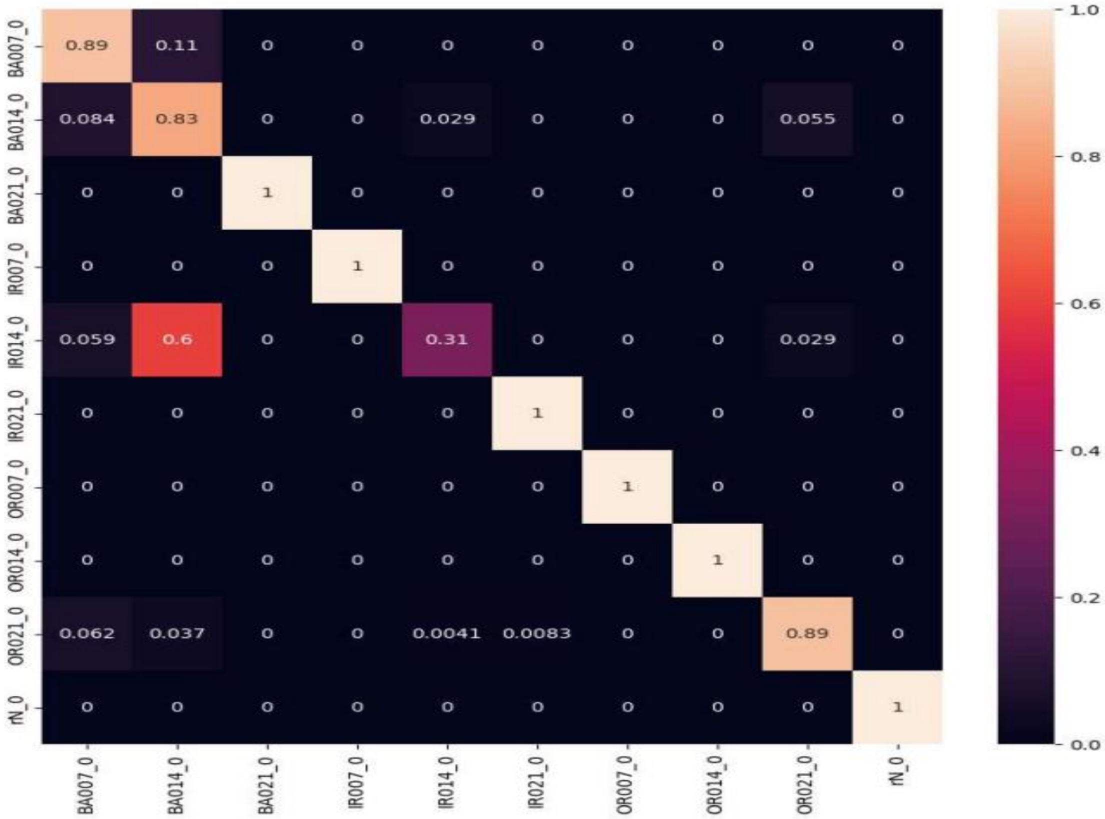

Figure 9.5 Confusion Matrix

Classification Report:

	precision	recall	f1-score	support
BA007_0	0.85	0.89	0.87	254
BA014_0	0.72	0.83	0.77	238
BA021_0	1.00	1.00	1.00	248
IR007_0	1.00	1.00	1.00	249
IR014_0	0.72	0.31	0.43	68
IR021_0	0.99	1.00	1.00	243
OR007_0	1.00	1.00	1.00	248
OR014_0	1.00	1.00	1.00	241
OR021_0	0.93	0.89	0.91	242
rN_0	1.00	1.00	1.00	232
accuracy			0.94	2263
macro avg	0.92	0.89	0.90	2263
weighted avg	0.94	0.94	0.93	2263

Figure 9.6 Classification Report Metrics

The results from the CNN-based fault diagnosis model are presented in *Figures 9.4-9.6*. *Figures 4-6* illustrate the training and validation loss, accuracy, confusion matrix, and classification report metrics. These visualizations provide comprehensive insights into the model's performance, demonstrating convergence during training and its ability to accurately classify different fault types.

9.2.1 Visualizing Results:

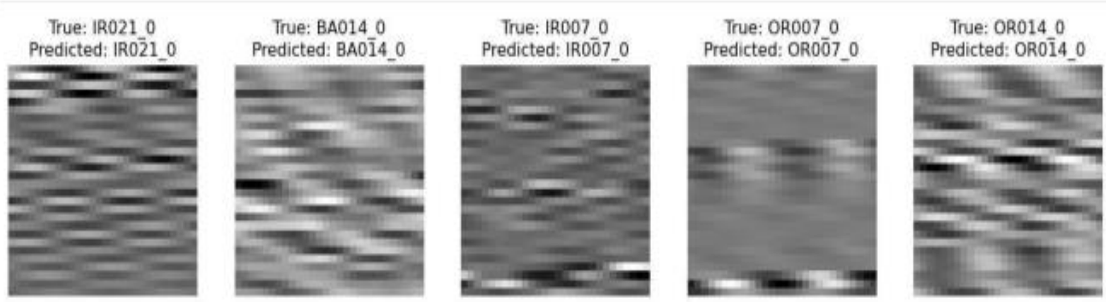

Figure 9.7 Randomly Selected Sample Predictions

Figure 9.7 showcases randomly selected samples from the test set alongside their true and predicted fault labels. Each subplot provides a visual representation of the vibration signal, allowing for an intuitive understanding of the model's performance on individual instances. The true fault labels and corresponding predictions are presented, aiding in the interpretation of the model's decision-making process.

9.3 Discussion:

The comprehensive analysis of time domain signals in *Figure 9.2* and the distribution visualization in *Figure 9.3* offer valuable insights into the nature of different faults in drive end bearings. This understanding becomes crucial for the subsequent training and evaluation of the CNN model.

The training and validation metrics in *Figures 9.4-9.6* collectively demonstrate the effectiveness of the CNN-based fault diagnosis model. The convergence of loss and the increase in accuracy signify the model's capacity to learn and generalize. The confusion matrix and classification report metrics provide a detailed breakdown of the model's performance across various fault types, offering insights into its strengths and areas for improvement.

Figure 9.7 provides a qualitative assessment of the model's predictions on randomly selected samples. The visual representation of vibration signals, coupled with true and predicted fault labels, enhances the interpretability of the model's decisions.

In conclusion, the integration of time domain signal analysis, CNN-based machine learning, and comprehensive visualizations has resulted in a robust fault diagnosis system for drive end bearings. The model exhibits promising capabilities in accurately classifying faults, laying the foundation for practical implementation in industrial settings. Future work may involve further refinement of the model, exploration of alternative architectures, and integration with real-time monitoring systems to enhance predictive maintenance practices.

CHAPTER-10
CONCLUSION

In conclusion, this project aimed to develop a fault diagnosis system for drive end bearings using machine learning techniques, with a focus on the CWRU dataset. The project involved two main phases: data preprocessing and the implementation of a Convolutional Neural Network (CNN).

Data Preprocessing: The initial stage involved loading and processing vibration data from the CWRU dataset. The dataset comprised various fault types in drive end bearings, and the data were organized into a structured format using Pandas DataFrames. The data were visualized to gain insights into the distribution of faults, and a CSV file was generated for further use.

CNN Implementation: The CNN-based machine learning model was designed to classify faults in drive end bearings using the preprocessed data. The CNN architecture consisted of convolutional layers for feature extraction, pooling layers for down-sampling, and fully connected layers for classification. The model was trained on a subset of the data, and its performance was evaluated on a separate test set.

Results and Performance Metrics: The trained CNN demonstrated promising results in fault classification. The confusion matrix, classification report, and additional performance metrics, including accuracy, precision, recall, and F1 score, provided a comprehensive evaluation of the model's performance across different fault categories. The model exhibited the ability to effectively differentiate between various fault types in drive end bearings.

Visualization of Results: To enhance the interpretability of the model, a visualization section showcased randomly selected samples from the test set. These samples were presented alongside their true and predicted fault labels, providing a qualitative understanding of the model's decision-making process.

Project Impact and Future Work: This project lays the foundation for automated fault diagnosis in drive end bearings, contributing to the field of predictive maintenance. Future work could involve further refinement of the model, exploration of alternative architectures, and integration with real-time monitoring systems. Additionally, collaboration with domain experts could enhance the interpretability and domain-specific applicability of the model.

In summary, the project successfully developed a fault diagnosis system using a CNN-based approach, demonstrating its potential for practical implementation in industrial scenarios. The insights gained from this project contribute to the broader goal of enhancing the reliability and efficiency of machinery through intelligent fault detection and diagnosis.

REFERENCES

[1] Mohammed Hakim, Abdoulhdi A.Borhana Omran, Ali Najah Ahmed, Muhannad Al-Waily, Abdallah Abdellatif(2023) .A systematic review of rolling bearing fault diagnoses based on deep learning and transfer learning: Taxonomy, overview, application, open challenges weaknesses and recommendations. Ain Shams Engineering Journal, volume 14, Issue 4.
https://doi.org/10.1016/j.asej.2022.101945

[2] Thanh Tran, Sebastian Bader, and Jan Lundgren(2022). An artificial neural network-based system for detecting machine failures using a tiny sound dataset: A case study. IEEE International Symposium on Multimedia (ISM)
https://doi.org/10.1109/ISM55400.2022.00036

[3] Lijun Zhang, Yuejian Zhang, Guangfeng Li (2023). Fault-Diagnosis Method for Rotating Machinery Based on SVMD Entropy and Machine Learning. MDIP, 16(6), 304.
https://www.mdpi.com/1999-4893/16/6/304

[4] Xiaoran Zhang, Kantilal Pitambar Rane, Ismail Kakaravada, and Mohammad Shabaz(2021). Research on vibration monitoring and fault diagnosis of rotating machinery based on internet of things technology. Nonlinear Engineering ,10(1):245-254.
https://www.degruyter.com/document/doi/10.1515/nleng-2021-0019/html

[5] Jing Zhang, Deqing Zhang, Mingyue Yang, Xiaobin Xu, Weifeng Liu, Chenglin Wen(2018). Fault Diagnosis for Rotating Machinery with Scarce Labelled Samples: A Deep CNN Method Based on Knowledge-Transferring from Shallow Models. IEEE International Conference on Control, Automation and Information Sciences (ICCAIS).
https://ieeexplore.ieee.org/document/8570515

[6] David Verstraete, Andrés Ferrada, Enrique López Droguett, Viviana Meruane, and Mohammad Modarres (2017). Deep Learning Enabled Fault Diagnosis Using Time-

Frequency Image Analysis of Rolling Element Bearings. Shock and Vibration, volume 2017. https://doi.org/10.1155/2017/5067651

[7] Dhiraj Neupane, Yunsu Kim, Jongwon Seok, Jungpyo Hong(2021). CNN-Based Fault Detection for Smart Manufacturing. Applied science, 11732. https://in.search.yahoo.com/search?fr=mcafee&type=E211IN826G0&p=.https%3A%2F%2F+doi.org%2F10.3390%2Fapp112411732#:~:text=no.%2024%3A%2011732.,https%3A//doi.org/10.3390/app112411732,-Note%20that%20from

[8] Haisheng Wang, Jian Wei, Pengjin Li (2022). Research on Fault Diagnosis Technology Based on Deep Learning. Journal of Physics: Conference Series, volume 2187. https://iopscience.iop.org/article/10.1088/1742-6596/2187/1/012041

[9] N.Fathiah Waziralilah, Aminudin Abu, M. H Lim, Lee Kee Quen, and Ahmed Elfakharany (2019). A Review on Convolutional Neural Network in Bearing Fault Diagnosis. MATEC Web of Conferences 255, 06002 (2019). https://doi.org/10.1051/matecconf/201925506002

[10] Bo Zhang (2022). Rolling Bearing Fault Detection System and Experiment Based on Deep Learning. Comput Intell Neurosci.2022: 8913859. https://doi.org/10.1155/2022/8913859

APPENDIX-A

PSUEDOCODE

Importing necessary libraries

```
import scipy.io
import seaborn as sns
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
```

DATA PREPROCESSING SECTION

In the data processing and analysis phase, we applied the following steps to prepare and visualize the dataset:

Creating the path for data files

```
data_dir = "CWRU_dataset/48k_drive_end/0hp"
for root, dirs, files in os.walk(data_dir, topdown=False):
    for file_name in files:
        path = os.path.join(root, file_name)
        print(path)
```

Loading data from a sample file

```
path = f'CWRU_dataset/48k_drive_end/0hp/OR007_0.mat'
mat = scipy.io.loadmat(path)
```

```
# Extracting the key name for data
key_name = list(mat.keys())[3]

# Creating a DataFrame with a sample data file
fault = np.full((len(mat[key_name]), 1), file_name[:-4])
df_temp = pd.DataFrame({'DE_data': np.ravel(mat[key_name]), 'fault': np.ravel(fault)})

# Plotting a sample data file
plt.figure(figsize=(15, 5))
plt.plot(df_temp.iloc[:, 0])
plt.show()

# Initializing a DataFrame for all data files
df = pd.DataFrame(columns=['DE_data', 'fault'])

# Iterating through all data files
data_dir = "CWRU_dataset/48k_drive_end/0hp"
for root, dirs, files in os.walk(data_dir, topdown=False):
    for file_name in files:
        # Check if the file has a .mat extension
        if file_name.endswith('.mat'):
            path = os.path.join(root, file_name)
            print(path)

            try:
                mat = scipy.io.loadmat(path)
                key_name = list(mat.keys())[3]
                DE_data = mat.get(key_name)
```

```
            # Simplifying fault creation
            fault = np.full((len(DE_data), 1), file_name[:-4])

            # Concatenating directly without creating df_temp
            df = pd.concat([df, pd.DataFrame({'DE_data': np.ravel(DE_data), 'fault':
np.ravel(fault)})], axis=0)
            print(df['fault'].unique())

        except Exception as e:
            print(f"Error processing file {file_name}: {e}")
```

 # Save the resulting DataFrame to a CSV file

```
df.to_csv('CWRU_dataset/48k_drive_end/0hp/0hp_all_faults.csv', index=False)
```

 # Display the resulting DataFrame

```
df
```

 # Plotting samples for each fault type

```
for f in df['fault'].unique():
    plt.figure(figsize=(10, 3))
    plt.plot(df[df['fault'] == f].iloc[:, 0])
    plt.title(f)
    plt.show()
```

 # Visualizing data distribution

```
plt.figure(figsize=(15, 5))
sns.scatterplot(data=df.iloc[::100, :], y='DE_data', x=np.arange(0, len(df), 100), hue='fault')
plt.show()
```

CNN IMPLEMENTATION SECTION

In this phase, we employed a Convolutional Neural Network (CNN) for fault classification based on vibration data. Key steps and outcomes include:

```
# Loading the preprocessed data
df = pd.read_csv('CWRU_dataset/48k_drive_end/0hp/0hp_all_faults.csv')

# Data preprocessing parameters
win_len = 784
stride = 300
X = []
Y = []

# Creating windows for CNN input
for k in df['fault'].unique():
    df_temp_2 = df[df['fault'] == k]

    for i in np.arange(0, len(df_temp_2) - (win_len), stride):
        temp = df_temp_2.iloc[i:i + win_len, :-1].values
        temp = temp.reshape((1, -1))
        X.append(temp)
        Y.append(df_temp_2.iloc[i + win_len, -1])

# Preparing data for CNN input
X = np.array(X)
X = X.reshape((X.shape[0], 28, 28, 1))
Y = np.array(Y)

# One-hot encoding the target variable
encoder = LabelEncoder()
```

```
encoder.fit(Y)
encoded_Y = encoder.transform(Y)
OHE_Y = to_categorical(encoded_Y)
```

Splitting data into training and testing sets

```
X_train, X_test, y_train, y_test = train_test_split(X, OHE_Y, test_size=0.3, shuffle=True)
```

Creating the CNN model

```
cnn_model = Sequential()
cnn_model.add(Conv2D(32, kernel_size=(3, 3), activation='tanh', input_shape=(X.shape[1],
X.shape[2], 1), padding='same'))
cnn_model.add(MaxPooling2D((2, 2), strides=(2, 2), padding='same'))
cnn_model.add(Conv2D(64, (3, 3), activation='tanh', padding='same'))
cnn_model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2), padding='same'))
cnn_model.add(Flatten())
cnn_model.add(Dense(128, activation='tanh'))
cnn_model.add(Dense(len(df['fault'].unique()), activation='softmax'))
```

Compiling the CNN model

```
cnn_model.compile(loss='categorical_crossentropy', optimizer='sgd', metrics=['accuracy'])
```

Training the CNN model

```
epochs = 50
history = cnn_model.fit(X_train, y_train, batch_size=128, epochs=epochs, verbose=1,
validation_data=(X_test, y_test), shuffle=True)
```

Function to inverse transform predictions

```
def inv_Transform_result(y_pred):
    y_pred = y_pred.argmax(axis=1)
    y_pred = encoder.inverse_transform(y_pred)
```

```
    return y_pred
```

Making predictions on the test set

```
y_pred = cnn_model.predict(X_test)
Y_pred = inv_Transform_result(y_pred)
Y_test = inv_Transform_result(y_test)
```

Confusion Matrix

```
plt.figure(figsize=(10, 10))
cm = confusion_matrix(Y_test, Y_pred, normalize='true')
f = sns.heatmap(cm, annot=True, xticklabels=encoder.classes_,
yticklabels=encoder.classes_)
plt.show()
```

Classification Report

```
print("Classification Report:")
print(classification_report(Y_test, Y_pred, target_names=encoder.classes_))
```

Additional Performance Metrics

```
accuracy = np.sum(np.diag(cm)) / np.sum(cm)
precision = np.diag(cm) / np.sum(cm, axis=0)
recall = np.diag(cm) / np.sum(cm, axis=1)
f1_score = 2 * (precision * recall) / (precision + recall)

print("\nAdditional Performance Metrics:")
print(f"Accuracy: {accuracy:.4f}")
print("Precision per class:")
for fault, prec in zip(encoder.classes_, precision):
    print(f"{fault}: {prec:.4f}")
print("Recall per class:")
```

```
for fault, rec in zip(encoder.classes_, recall):
    print(f"{fault}: {rec:.4f}")
print("F1 Score per class:")
for fault, f1 in zip(encoder.classes_, f1_score):
    print(f"{fault}: {f1:.4f}")

# Visualizing Results
num_samples_to_visualize = 5

# Randomly selecting some samples from the test set
random_indices = np.random.choice(len(X_test), num_samples_to_visualize, replace=False)
sample_images = X_test[random_indices]
true_labels = Y_test[random_indices]

# Predicting the labels for the selected samples
predicted_labels = inv_Transform_result(cnn_model.predict(sample_images))

# Plotting the selected samples along with true and predicted labels
plt.figure(figsize=(15, 8))
for i in range(num_samples_to_visualize):
    plt.subplot(1, num_samples_to_visualize, i + 1)
    plt.imshow(sample_images[i, :, :, 0], cmap='gray')
    plt.title(f'True: {true_labels[i]}\nPredicted: {predicted_labels[i]}')
    plt.axis('off')

plt.show()
```

APPENDIX-B

Fault Diagnosis in Drive End Bearings using CNN-based Machine Learning

ORIGINALITY REPORT

19%	16%	14%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

6%

★ www.ijraset.com

Internet Source

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

APPENDIX-C