People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Dr. Yahia Farès University of Medea

Faculty of Technology



Thesis

presented to obtain the degree of doctor in:
Electrical Engineering

By: LACHENANI SID AHMED

Theme

Monitoring and fault diagnosis of industrial machines using a Wireless Sensor Network

Defended in front of a jury composed of :

Mr. BOUHEDDA Mounir	Professor	at University of MEDEA	Chair.
Mr. Ould Zmirli Mohamed	Professor	at University of MEDEA	Thesis Supervisor.
Mr. HOUASSINE Hamza	Professor	at University of BOUIRA	Examiner.
Mr. SEBAA Karim	Professor	at ENP	Examiner.

Academic year : 2025/2026

N° d'ordre : …….

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Yahia Farès de Médéa
Faculté de Technologie
Département de Génie Électrique

THÈSE

En vue de l'obtention du diplôme de
Doctorat en Sciences
Spécialité: Génie électrique

Présenté par:

LACHENANI Sid Ahmed

THÈME:

Surveillance et diagnostic des défauts des machines industrielles par un réseau de capteurs sans fil.

Soutenue publiquement le : 11/12/2025

Devant le jury composé de :

M. BOUHEDDA Mounir	Prof. Université de Médéa	Président
M. OULD ZMIRLI Mohamed	Prof. Université de Médéa	Directeur de Thèse
M. HOUASSINE Hamza	Prof. Université de Bouira	Examinateur
M. SEBAA Karim	Prof. ENP	Examinateur

Thèse préparée au sein du Laboratoire des Systèmes Électroniques Avancés (LSEA)
de l'Université de Médéa, ALGÉRIE

Dedication

To my mother
To my father
To my wife, daughter, and son
To all my friends and family members

— LACHENANI Sid Ahmed

الملخّص

تقدم هذه الأطروحة **BearNet**، وهو إطار للتعلّم العميق بالنقل مخصّص لتشخيص أعطال المحامل في بيئات WSN محدودة الموارد. يعتمد BearNet على بنية YAMNet المدرَّبة مسبقًا والمصمَّمة أساسًا للتعرّف على الصوت، مع تكييفها لتصنيف أعطال المحامل انطلاقًا من إشارات الاهتزاز. تُحوَّل القياسات الخام الصادرة عن العقد منخفضة الاستهلاك إلى مخططات طيفية ميل، تُستعمل كمدخلات مدمجة وغنية بالمعلومات. يُعاد ضبط طبقات مختارة فقط من YAMNet، بينما تضيف طبقات مخصّصة حساسية أكبر لخصائص المحامل، مما يقلّص الحمل الحسابي واستهلاك الذاكرة مع الحفاظ على الدقة.

تمّ تقييم الإطار على مجموعتي بيانات CWRU وMFPT، حيث أظهر BearNet تفوّقًا بلغ 3.1% مقارنةً بالضبط القياسي لـYAMNet، وتجاوز أحدث النماذج الأخرى بنسبة تراوحت بين 0.02% و5.26%. يُثبت ذلك أنّ BearNet حلّ فعّال وقابل للنشر لمراقبة حالة المحامل في التطبيقات الصناعية الواقعية، ويمهّد الطريق لأبحاث مستقبلية حول التشخيص الفوري المتكيّف داخل البنى التحتية اللاسلكية.

الكلمات المفتاحية: شبكات الاستشعار اللاسلكية، التعلّم العميق، التعلّم بالنقل، أعطال المحامل، تشخيص الأعطال، الصناعة 5.0، التصنيف، بيانات CWRU وMFPT.

Abstract

This thesis proposes **BearNet**, a deep transfer learning framework for bearing fault diagnosis in resource-constrained WSN environments. BearNet adapts the pre-trained YAMNet modeloriginally designed for audio recognitionto classify bearing defects from vibration signals. Raw acceleration data from low-power nodes are transformed into Mel-spectrograms, serving as compact and informative inputs. Only selected YAMNet layers are fine-tuned, while additional layers enhance sensitivity to bearing features, reducing computational and memory demands without sacrificing accuracy.

The method is validated on the CWRU and MFPT datasets. Results show that BearNet improves accuracy by up to 3.1% over standard YAMNet approaches and surpasses other state-of-the-art models by 0.02--5.26%. It also scales from binary detection to multi-class and severity-level classification, remaining robust across operating conditions.

By combining vibration sensing, spectrogram representation, and selective transfer learning, BearNet delivers a practical, energy-efficient solution for real-time bearing fault diagnosis in WSN-based monitoring systems. The findings demonstrate the feasibility of embedding high-performance AI into wireless infrastructures and open avenues for future research on adaptive, on-node fault detection for modern industrial environments.

Keywords : WSN, Deep learning, Transfer learning, Bearing, Fault diagnosis, Industry 5.0, Classification, CWRU.

RÉSUMÉ

Cette thèse présente **BearNet**, un cadre d'apprentissage profond par transfert, adapté au diagnostic de défauts de roulements dans des environnements WSN contraints. BearNet exploite l'architecture pré-entraînée YAMNet, conçue initialement pour la reconnaissance audio, et l'adapte à la classification des défaillances à partir de signaux vibratoires. Les mesures brutes des nœuds capteurs sont converties en spectrogrammes de Mel servant d'entrées compactes et discriminantes. Seules certaines couches de YAMNet sont ajustées, tandis que des couches supplémentaires renforcent la sensibilité aux signatures de roulement, réduisant ainsi la charge de calcul et l'usage mémoire sans compromettre la précision.

La méthode est évaluée sur les jeux de données CWRU et MFPT. Les résultats montrent une amélioration de précision allant jusqu'à 3,1 % par rapport aux modèles YAMNet standards et de 0,02 à 5,26 % face aux approches récentes. BearNet s'etend aisément du diagnostic binaire aux classifications multi-défauts et par sévérité, tout en maintenant une robustesse face aux conditions d'exploitation variées.

En combinant capteurs vibratoires, représentation spectrogramme et transfert sélectif, BearNet constitue une solution pratique, économe en énergie et adaptée à la surveillance industrielle en temps réel via des WSN.

Mots-clés : Réseau de capteurs sans fil, Apprentissage profond, Apprentissage par transfert, Roulement, Diagnostic des défauts, Industrie 5.0, Classification, CWRU.

Acknowledgment

I would like to express my deepest gratitude to my supervisor, Professor **Ould Zemirli Mohamed**, for his invaluable guidance, support, and encouragement throughout this research. His expertise and insights have been instrumental in shaping this work.

A special thanks to my colleague at the scientific laboratory, Dr. **Hamza Kheddar**, whose dedication and assistance were crucial in overcoming challenges and completing this project successfully. Without his help, this work would not have been possible.

I am also sincerely grateful to the members of the jury, Professor **BOUHEDDA Mounir** who honored me by presiding over the jury; I am equally honored that my work was evaluated by such a distinguished panel: Professors **HOUASSINE Hamza**, and **SEBAA Karim** . Thank you for your prompt availability and the kindness with which you accepted to review this thesis.

I would like to extend my sincere gratitude to Dr. Soufiane RAHAL, Vice-Doyen in charge of PGRS/RE, and Mustapha Doumir for their invaluable support and assistance.

I would also like to thank all the members of the **LSEA** laboratory for their collaboration and support throughout my academic journey.

Finally, I extend my heartfelt appreciation to my family for their unwavering support, patience, and encouragement throughout this journey. Their belief in me has been my greatest motivation.

Table of Contents

Abstract **i**

Acknowledgment **iv**

Table of contents **viii**

List of figures **xi**

List of tables **xii**

Abbreviations list **xiii**

1 General introduction **1**

1.1 Introduction 2

1.2 Problem definition 5

1.2.1 Bearing fault classification from vibration spectrograms 5

1.2.2 Transfer learning for bearing fault diagnosis 6

1.3 Problem statement 7

1.4 Research questions 8

1.5 Research aims 8

1.6 Thesis structure 9

2 Bearing fault diagnosis tasks and datasets **10**

2.1 Introduction 11

2.2 Condition-based maintenance and condition monitoring for bearings 11

2.2.1 CBM workflow for bearings :Maintenance-level view 11

2.2.2 Deciding exactly when to intervene 12

2.3 General workflow of BFD 13

2.3.1 Overview and classification of traditional BFD techniques 15

2.3.2 Vibration analysis based bearing fault diagnosis 16

2.3.3 Bearing structure and components 17
2.3.4 Evolution of bearing defects and their vibration signatures 20
2.4 General procedure for BFD techniques 22
2.4.1 Overview and importance of fault diagnosis 22
2.4.2 Evolution of technologies in bearing condition monitoring 24
2.5 Comparison of popular bearing fault datasets 24
2.5.1 CWRU dataset description 25
2.5.2 PRONOSTIA bearing dataset 27
2.5.3 IMS bearing dataset description 27
2.5.4 Paderborn university, Kat bearing data-center dataset 28
2.5.5 MFPT dataset bearing 29
2.5.6 Summary comparison of representative bearing fault datasets 30
2.6 Conclusion 31

3 State of the Art in AI and WSN-Based Bearing Fault Diagnosis 32
3.1 Introduction 33
3.1.1 The Importance of WSN in predictive maintenance 33
3.1.2 Role of artificial intelligence in condition monitoring 34
3.2 Literature review and key contributions in traditional BFD techniques 36
3.3 Overview of machine learning approaches for BFD 37
3.3.1 Literature review and key contributions 37
3.3.2 Key feature extraction and selection strategies in condition monitoring 38
3.4 Challenges of deep learning in BFD 39
3.5 Transfer learning for bearing fault classification 40
3.5.1 Concept of transfer learning 40
3.5.2 Motivations for applying transfer learning in BFD 41
3.6 Wireless sensor networks and IoT-based bearing fault monitoring 42
3.7 Comparison of state-of-the-Art techniques 44
3.7.1 Comprehensive background 45
3.7.2 Critical review and comparative analysis of diagnostic methods 47
3.7.3 Critical analysis and synthesis 47
3.8 Conclusion 49

4 Transfer learning 51
4.1 Introduction 52
4.2 Factors influencing transfer learning performance 56

4.3 Taxonomy of transfer learning . . . 57
4.3.1 Based on label availability . . . 58
4.3.2 Based on feature space similarity . . . 59
4.3.3 Based on learning strategy . . . 59
4.3.4 Based on what is transferred . . . 59
4.3.5 Special case:DTL . . . 60
4.4 Applications of transfer learning . . . 60
4.4.1 Common application domains . . . 61
4.4.2 Transfer learning in condition monitoring applications . . . 62
4.4.3 Overview of TL applications in sound and bearing analysis . . . 63
4.5 Machine learning and deep learning: Core concepts . . . 64
4.5.1 From machine learning to deep learning . . . 64
4.5.2 Hyperparameters . . . 64
4.5.3 Examples . . . 64
4.6 Understanding transfer learning . . . 65
4.6.1 Definition of transfer learning . . . 65
4.6.2 Domain adaptation . . . 66
4.6.3 Fundamental questions in transfer learning . . . 66
4.6.4 Diagnosing bearing faults through negative transfer . . . 70
4.7 An end-to-end guide to transfer learning . . . 72
4.7.1 Transfer with pre-trained models: Selective fine-tuning versus feature extraction . . . 72
4.7.2 Freezing or fine-tuning? . . . 73
4.7.3 Popular transfer learning models across domains . . . 74
4.7.4 Choosing the best model for transfer Learning . . . 76
4.7.5 A complete transfer learning process . . . 77
4.8 Conclusion . . . 79

5 Experimental results **81**
5.1 Introduction . . . 82
5.2 Proposed bearing fault classification system . . . 82
5.3 Source and target databases . . . 83
5.3.1 Source database for YAMNet . . . 83
5.3.2 Target datasets: Bearing Applications . . . 84
5.3.3 Dataset preprocessing . . . 84

5.4 Constructing the source and target model 85
5.4.1 Pretrained source model: YAMNet 86
5.4.2 Target model:BearNet 87
5.5 Deep transfer learning workflow for BearNet 89
5.5.1 BearNet deep transfer learning workflow 89
5.5.2 BearNet model building strategies in deep transfer learning 90
5.6 Experimental details 91
5.6.1 Experimental setup and environment 92
5.6.2 Training details 92
5.6.3 Evaluation metrics 93
5.7 Benchmark deep learning model evaluation: YAMNET as benchmark 93
5.8 BearNet model evaluation on the CWRU Dataset 95
5.8.1 Analyses and discussions 96
5.8.2 Confusion matrix analysis of BearNet performance 99
5.8.3 BearNet's generalisability on the MFPT Dataset 99
5.9 BearNet's performance comparison with the benchmark model 102
5.10 Comparing performance with related works 103
5.11 Conclusion 104

6 General conclusion **105**
6.1 Overview 106
6.2 Research Findings 107
6.3 Future Research 107

My publications **109**

Bibliography **110**

List of Figures

1.1 Comparison of traditional ML and TL for fault classification, highlighting TLs ability to transfer knowledge across tasks, improving efficiency and reducing resource requirements. 7

2.1 The P–F interval for a bearing: detection point (P), actionable planning window, and functional failure (F) [20]. 13

2.2 Stages from sensing and acquisition to signal processing and decisions. 14

2.3 Traditional bearing fault diagnosiss (BFDs) Techniques. 15

2.4 Different stages of vibration analysis. 17

2.5 Illustration of characteristic defect frequencies in a ball bearing: Ball Pass Ball-Pass Frequency at Inner Ring (BPFI), Ball-Pass Frequency at Outer Ring (BPFO), Ball-Spin Frequency (BSF), and Fundamental Train Frequency (FTF). 18

2.6 Schematic structure of a bearing showing elements relevant to fault frequency calculations: number of rolling elements n, ball diameter d, pitch diameter D, and contact angle β [29]. 18

2.7 Vibration signal patterns for different bearing faults: (a) Outer race fault, (b) Inner race fault, and (c) Ball defect. Each fault type produces a distinct signal influenced by fault location and dynamics [29]. 21

2.8 Evolution of Subsurface Defects. 21

2.9 General process of fault diagnosis for rolling bearings, including feature extraction and pattern recognition techniques. 22

2.10 Timeline of technological development and techniques in bearing condition monitoring (CM) and fault diagnosis. 25

2.11 Samples of raw vibration sensor signals corresponding to each class [37]. . . 26

2.12 PRONOSTIA (IEEE PHM 2012) bearing dataset. (a) Experimental rig with instrumentation for run-to-failure tests (DAQ, sensors, load and speed control). (b) Example bearings in healthy and damaged states. 27

2.13 The IMS bearing test rig setup. The system consists of an electric motor, four bearings mounted along the shaft, radial load modules, accelerometers, and thermocouples. This configuration enables the acquisition of vibration, load, and temperature data for BFD 28

3.1 Illustration of the transfer learning (TL) principle [77]. 40
3.2 Key Differences and Motivations for Using TL in BFD 42
3.3 The benefits of wireless sensor network (WSN) in industrial control system (ICS). 43

4.1 Real examples of TL. (a) Rechta, vs. Couscous. (b) manual transmission vs. automatic transmission. 53
4.2 Hierarchical relationship between AI, ML, deep learning (DL), and TL. 54
4.3 Comparison between traditional machine learning (ML) and TL approaches in fault classification tasks. 54
4.4 Variability of sensor signals across different application domains and bearing types, posing challenges for developing generalized fault diagnosis models.(a) Critical application fields utilizing bearings. (b) Common bearing types used across industries 55
4.5 Illustration of the TL process for BFD: knowledge learned from machines A and B is transferred to machine C, reducing the need to retrain from scratch. 56
4.6 Taxonomy of TL Approaches. 58
4.7 Varied Practical Uses of TL Across Domains. 62
4.8 TL applications in CM 63
4.9 Illustration of Negative TL using a wall-crossing metaphor. (a) Normal case: small domain gap, the man easily climbs over the wall. (b) Abnormal case: large domain gap, the man fails to cross—negative transfer. (c) Abnormal case: large domain gap, the man succeeds by using intermediate supports. . 71
4.10 Comparison between feature extraction and selective fine-tuning in TL. The left illustrates the pre-trained base model. The middle shows a target model using feature extraction (frozen backbone), while the right depicts fine-tuning (partially unfrozen layers). 74
4.11 Flowchart of the complete TL process. 79

5.1 Step-by-Step process of bearing network (BearNet) prediction and classification using deep transfer learning (DTL). 83
5.2 Foundations of DTL and its Architectural Building Blocks: from Yet another audio mobilenet network (YAMNet) to BearNet. 88
5.3 Comparison of convolutional layer configurations between YAMNetand enhanced BearNet architectures. 88
5.4 Comparative evaluation of YAMNet performance across different configurations: (a) Binary classification accuracy, (b) Four-class accuracy, (c) Twelve-class accuracy, (d) Binary loss, (e) Four-class loss, and (f) Twelve-class loss. . 95
5.5 Comparative evaluation of two-class BearNet performance metrics across TL stages: (a),(b) Accuracy and loss for TL_0; (c),(d) TL_1; (e),(f) TL_2; (g),(h) TL_3. 97

5.6 Comparative evaluation of four-class BearNet performance metrics across TL stages: (a),(b) Accuracy and loss for TL_0; (c),(d) TL_1; (e),(f) TL_2; (g),(h) TL_3. 98

5.7 Comparative evaluation of twelve-class BearNet performance metrics across TL stages: (a),(b) Accuracy and loss for TL_0; (c),(d) TL_1; (e),(f) TL_2; (g),(h) TL_3. 100

5.8 Confusion matrices for various classification tasks: (a) Binary classification, (b) Four-class classification, and (c) Twelve-class classification. 101

5.9 BearNet performance with TL_3 stage using the MFPT dataset. 102

List of Tables

2.1 Approximate Usage of the Principal Traditional BFD Techniques in Literature 16
2.2 Characteristic fault frequencies for bearing defects. 20
2.3 Specifications of bearings and sample gathering conditions in the Case Western reserve University (CWRU) datasets 26
2.4 Summary of the main elements included in the dataset 29
2.5 Specifications of bearings and sample gathering conditions in the machinery failure prevention technology (MFPT) dataset. 30
2.6 Representative public datasets for BFD (literature summary). 31

3.1 Comparative Analysis of TL-Based Approaches for BFD 46
3.2 Strengths and Limitations of TL and TimeFrequency Feature Extraction Methods 48

4.1 Comparison of Different TL Settings 58
4.2 Representative pre-trained models by domain and typical transfer uses. . . . 75

5.1 YAMNet architecture 87
5.2 Classification scenarios and their corresponding classes for BFDs within the CWRU dataset. 89
5.3 Summary of DL metrics employed for evaluating BearNet model. 94
5.4 Summary of TL with varied frozen layers in the BearNet model. 96
5.5 Epochs of convergence and validation accuracy across various class configurations when validation loss is minimal. 103
5.6 Comparison of related DL approaches for bearing fault detection on datasets comparable to those used in this study. 103

Abbreviations list

AI	artificial intelligence
AE	Acoustic Emission (AE
BearNet	bearing network
BFD	bearing fault diagnosis
BPFI	Ball-Pass Frequency at Inner Ring
BPFO	Ball-Pass Frequency at Outer Ring
BSF	Ball-Spin Frequency
CBM	Condition-Based Maintenance
CM	condition monitoring
CNN	convolutional neural network
CWRU	Case Western reserve University
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DL	deep learning
DTL	deep transfer learning
FFT	fast Fourier tranform
FT	Fourier tranform
FTF	Fundamental Train Frequency
i.i.d.	independently and identically distributed
ICEN	Iterative convolutional enhancement network
ICS	industrial control system
IMFs	into intrinsic mode functions
IoT	Internet of things

LoRA Low-Rank Adaptation

MFPT machinery failure prevention technology

ML machine learning

ReLu rectified linear unit

RMS Root Mean Square

RNN recurrent neural networks

RUL remaining useful life

sps samples per second

STFT Short-Time Fourier Transform

TECN terminal enhanced convolutional network

TL transfer learning

TTS Text-to-Speech

WSN wireless sensor network

YAMNet Yet another audio mobilenet network

Chapter 1

General introduction

Contents

I.1	Introduction	2
I.2	Problem definition	5
I.3	Problem statement	8
I.4	Research questions	8
I.5	Research aims	9
I.6	Thesis structure	9

1.1 Introduction

Rolling element bearings are essential parts of rotating machinery in a variety of industrial sectors because they facilitate load transmission and smooth, effective mechanical motion. These bearings are prone to wear and failure because of friction, fatigue, and operating stresses, even with their sturdy design. Notably, one of the most common reasons for machinery failure is bearing failure, which frequently results in unscheduled downtime, expensive maintenance, and production losses [1, 2]. Therefore, extending machine lifespan, optimising maintenance schedules, and guaranteeing system reliability all depend on prompt and precise fault detection.

Condition monitorings (CMs) systems have become essential components of predictive maintenance plans in order to tackle these issues. By identifying early indicators of mechanical deterioration, these systems hope to prevent catastrophic failures [3]. Vibration analysis has become more popular among the different CM techniques because of its high sensitivity and efficacy in identifying bearing dynamics anomalies [4]. This method provides useful diagnostic information on fault types and severities by capturing the aberrant vibration signals and frequency shifts released by deteriorating bearings.

Over the past ten years, there has been a significant paradigm shift in the industrial landscape due to the move from traditional wired monitoring systems to wireless sensor-based architectures. Modern industrial control systems have been revolutionised by the scalability, adaptability, and cost-effectiveness of Wireless sensor networks (WSNs), particularly in the field of fault diagnosis [5]. WSNs are less expensive to install, use less cabling, and enable remote monitoring in difficult or dangerous locations compared to their wired counterparts. These networks are naturally able to adjust to structural changes in machine layouts and provide real-time condition assessment with minimal disruption [6].

Their integration within the broader context of Industry 4.0 and the emerging Industry 5.0 further enhances the importance of WSNs. These paradigms facilitate human-centered, networked, and intelligent manufacturing environments where autonomous systems, cyber-physical technologies, and data-driven insights are crucial elements. Additionally, WSNs significantly contributes to industrial control systems (ICS) [7, 8] by enhancing real-time monitoring, fault predictability, operational precision, and energy efficiency.

However, the wide deployment of WSNs introduces unique challenges, particularly in fault diagnosis. These challenges include limited computational resources, constrained energy availability, intermittent communication, and the demand for real-time responsiveness. Traditional diagnostic approaches, especially those relying on centralized data processing and large labeled datasets, are not inherently suited for such environments [9].

To overcome these limitations, artificial intelligence (AI) techniques have been incorporated into industrial diagnostics to address the inherent limitations of traditional CM systems. Among these, machine learning (ML) and deep learning (DL) have demonstrated remarkable proficiency in automated decision-making, pattern recognition, and anomaly detection. Through the extraction of intricate patterns from extensive sensor datasets, these models facilitate fault identification that is more precise and timely. Data scarcity, partic-

ularly for uncommon or newly discovered fault types, is a recurring problem in real-world applications. AI models are susceptible to overfitting when trained on limited labelled data, which hinders their capacity to generalise across different operational conditions, sensor modalities, and machinery types.

Artificial intelligence (AI),Which studies intelligent agents that perceive their environment and act to maximise success [10], has long aimed to build machines that can learn and solve problems on their own. ML, a crucial area of AI, is concerned with creating algorithms that can learn from data without explicit programming [11]. In the last few decades, ML has developed from a theoretical idea into a useful framework that combines information theory, statistics, and optimisation concepts. It serves as the foundation for many intelligent systems nowadays.

Depending on the type of data and feedback, ML approaches are typically divided into three categories: supervised, unsupervised, and reinforcement learning. Of these, DL and supervised learning techniques have emerged as being especially successful for tasks like fault prediction, regression, and classification.ML and DL are now essential tools for intelligent CM due to the explosion of sensor data in industrial settings, particularly from WSN [12]. However, there are particular difficulties in implementing such models in real world deployments, especially in resource constrained, WSN environments, including sparse labelled data, power limitations, and limited computational capacity. These limitations highlight the necessity of effective learning techniques, in order to provide scalable and reliable fault diagnosis solutions for contemporary Industry 5.0 systems [13].

Ideally, when sufficient labeled vibration data are available, the supervised classification of bearing faults can be formulated as a learning task where the model aims to minimize prediction errors on the training data. Under ideal conditions, such as properly selected model capacity and a large, diverse dataset, the learned classifier is expected to generalize well to new samples.

However, in practical WSN based industrial applications, acquiring extensive labeled fault data is costly, time-consuming, and often infeasible. The limited availability of labeled signals, variations in operating conditions, and differences in sensor placements introduce distributional shifts that degrade model performance. This highlights the need for learning algorithms that can generalize from small labeled datasets by reusing knowledge from related domains or conditions [14].

To address these challenges, this thesis explores **transfer learning (TL)**, a promising approach that adapts pre-trained models to new but related tasks, reducing dependency on large annotated datasets. By leveraging domain-invariant features learned from pre-trained sound classification models, TL facilitates more robust and efficient fault classification in resource-constrained WSN environments.

TL has emerged as a powerful approach to overcome these constraints. Instead of training models from scratch, TL enables the reuse of knowledge acquired from one domain (the source) to improve learning performance in a related but distinct domain (the target) [15, 16]. In the context of BFD, TL allows pre-trained DL models to be adapted to new machines or fault types with minimal labeled vibration data. This significantly reduces both

development time and computational demands.

Because embedded WSN-based CMs systems have limited memory, energy, and processing power, the benefits of TL are especially pertinent. TL makes it possible to effectively extract high-level features from raw vibration signal spectrograms by fine-tuning only a portion of the layers in deep neural networks. In line with the operational limitations of actual industrial settings, this enables the direct deployment of lightweight, yet incredibly effective diagnostic models on decentralised sensor nodes.

Cognitive learning theory also serves as the foundation for TL motivation. TL enables machines to use previously learnt representations, just as humans — especially children can use knowledge from past learning experiences to identify new patterns with few examples. For example, even when given different fonts, colours, or handwriting styles, a child who has learnt to recognise multiple alphabetic letters can quickly generalise to a new letter. Similar to this, TL helps learning systems find and transfer domain-invariant features, which speeds up learning on related but new tasks with little supervision [17].

Transfer learning (TL) facilitates knowledge reuse across tasks, in contrast to traditional ML techniques that demand sizable labelled datasets for every new task [15, 16]. In real-world scenarios, where data is limited, fault types change, and monitoring systems need to function dependably with limited resources, this capability is particularly important for fault classification. In order to create a deployable and reliable fault diagnosis model that meets the needs of WSNs in Industry 5.0.

TL is especially promising for WSN-based CM systems in this regard. It makes it possible to create resource-efficient, lightweight diagnostic models that fit the operational limitations of embedded WSN nodes. Even with limited resources, it is possible to extract high-level features from raw vibration signals by fine-tuning a subset of network layers. In addition to lowering the requirement for intensive retraining, this method makes it easier to deploy models in low-power, decentralised settings.

Although DL has gained significant traction in the field of fault diagnosis, there remains a clear gap in developing models that are not only accurate but also practical for deployment in real world WSN environments. Many existing approaches assume the availability of high computational power, overlooking the practical limitations of embedded systems such as restricted memory, limited energy resources, and the need for real-time decision-making. This highlights the urgent need for scalable, lightweight AI solutions that are specifically designed to operate efficiently within the constraints of WSN-based fault detection systems.

To address this challenge, the present thesis proposes **BearNet**, a novel AI architecture tailored for the fault diagnosis of rolling element bearings in WSNs. **BearNet** builds upon TL by adapting pre-trained audio classification models,such as **Yet another audio mobilenet network (YAMNet)**, and extending them with custom-designed layers optimized for the interpretation of vibration spectrograms. This hybrid architecture enables the model to effectively transfer knowledge from the acoustic domain to the vibration domain, enhancing its ability to classify faults from spectrogram representations of sensor data. By fine-tuning only a subset of layers using vibration data acquired from WSNs, BearNet achieves a practical trade-off between diagnostic accuracy and computational efficiency, making it well-suited

for deployment in resource-constrained, real-time monitoring environments.

BearNet is envisioned as a deployment-ready, lightweight solution capable of operating in real-time, battery-powered, and memory-limited environments. This strategy represents a significant step toward the realization of intelligent, resilient, and energy-aware monitoring systems for Industry 5.0. By addressing the dual challenges of data scarcity and hardware limitations, this thesis contributes to the development of scalable AI-driven solutions that enhance predictive maintenance and operational efficiency in modern industrial systems.

The following sections present the background and definition of the problem, outline the research questions and objectives, and conclude with an overview of the thesis structure.

1.2 Problem definition

The problem addressed in this thesis revolves around the accurate classification of bearing faults using data collected from vibration signals in WSN environments. To effectively solve this problem, we distinguish between two essential aspects: the technical formulation of the bearing fault classification task using vibration spectrograms, and the role of TL as a strategy to overcome challenges related to data scarcity and domain variability. The following subsections present these two dimensions in detail.

1.2.1 Bearing fault classification from vibration spectrograms

In the context of this thesis, the task of fault classification involves learning a mapping from vibration spectrograms, **transformed representations of time-domain signals**, into predefined fault categories. These spectrograms can be derived either from raw vibration signals recorded by WSN-based sensors or from preprocessed feature vectors. Each spectrogram represents a structured input used to identify the health condition of the bearing.

The output labels in this work correspond to different bearing fault conditions and are structured into a hierarchical classification framework with increasing diagnostic granularity. The classification process is conducted in three progressive phases.

In the first phase, a **binary classification** is performed to distinguish between normal and faulty bearings. This initial step establishes the models ability to differentiate between healthy and damaged conditions. In the second phase, the classification task is expanded to include **four classes**: normal, ball fault, inner race fault, and outer race fault. This level of classification not only distinguishes faulty bearings from healthy ones but also enables the precise localization of the fault within the bearing structure, enhancing the diagnostic resolution compared to the initial binary step. In the third and most detailed phase, the system classifies the vibration spectrograms into **12 specific categories**. These include combinations of fault types and severities (such as different fault sizes) along with the normal class. This comprehensive classification provides essential information for predictive maintenance by identifying not only the presence and location of a fault but also its severity. Such de-

tailed fault characterization supports more effective repair scheduling and contributes to extending the operational life of industrial machinery.

Finally, the proposed framework's generalizability is validated using an additional dataset to evaluate its performance under different fault scenarios, thereby assessing its robustness and adaptability across varying operational conditions.

Two main challenges arise in building an effective spectrogram-based fault classification model. The first is to identify or learn a meaningful feature space where the different fault classes are easily separable. This can be achieved using either handcrafted features or advanced representation learning techniques like convolutional neural networks (CNNs), which automatically capture relevant patterns from the spectrogram inputs. The second challenge is designing a classifier that accurately maps these features to the corresponding fault labels.

In modern signal-based fault diagnosis particularly in TL scenarios with limited labeled data these two challenges are often addressed together using DL architectures. In such models, feature extraction and classification are integrated into a unified framework. This is especially advantageous when adapting pre-trained models to new bearing types or fault conditions with minimal additional training, as is the case in this thesis.

1.2.2 Transfer learning for bearing fault diagnosis

In traditional ML, both training and testing datasets are typically expected to originate from the same domain meaning they share similar data distributions, operating conditions, and class definitions. For instance, a model trained to classify normal versus faulty bearings under a specific machine setup will likely perform poorly if deployed on data from a different machine or operating environment. This limitation becomes even more pronounced in WSN applications, where sensor deployment, machine variation, and environmental dynamics often introduce significant distributional shifts.

transfer learning (TL) provides a practical and efficient solution to the limitations of traditional learning strategies. Instead of training a new model from scratch for every task, TL reuses knowledge learned from a related source domain to improve performance on a new but related target domain. In this work, the source domain may involve audio signals collected from pre-trained models, while the target domain corresponds to vibration signals under different bearing or machine conditions.

As illustrated in Figure 1.1, the traditional learning approach (left) requires training independent models for each dataset, demanding large task-specific data and high computational resources an impractical requirement for embedded, resource-constrained WSNs. By contrast, the TL approach (right) transfers domain-invariant knowledge from a pre-trained model such as YAMNet trained on audio data and adapts it for vibration-based fault diagnosis, significantly reducing the data and computational burden while enabling lightweight, scalable deployment.

In this thesis, we adopt the TL paradigm by integrating pre-trained audio classification models and appending custom layers specifically tuned to interpret vibration spectrograms.

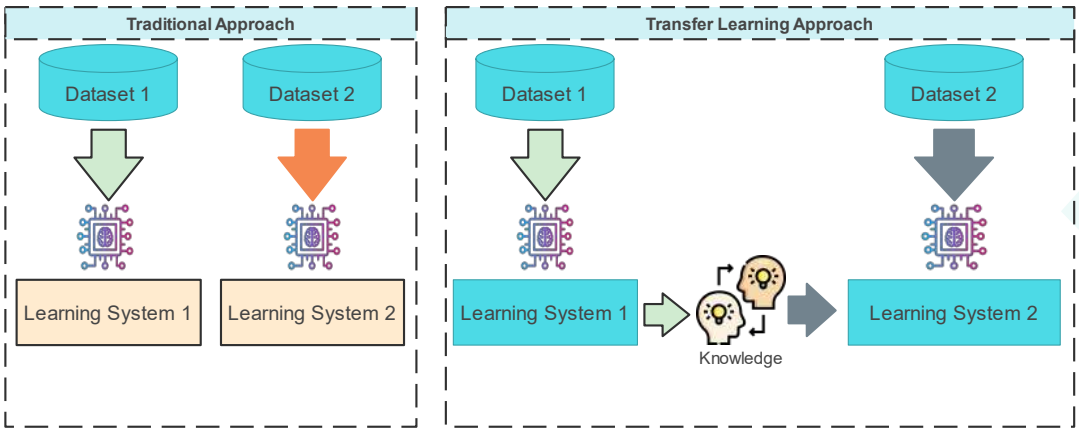


Figure 1.1: Comparison of traditional ML and TL for fault classification, highlighting TLs ability to transfer knowledge across tasks, improving efficiency and reducing resource requirements.

Only a subset of network layers is fine-tuned using vibration data collected in WSN setting, striking a balance between computational efficiency and diagnostic accuracy.

We particularly focus on a TL scenario where the source and target domains share similar feature spaces (e.g., timefrequency representations), but differ in class labels such as moving from binary classification (normal vs. fault) to multi-class fault localization (e.g., ball fault, inner race, outer race). This setup is categorized as **inductive TL**, where the tasks are related but not identical.

By sharing learned knowledge across domains, our TL-based approach significantly reduces the demand for large labeled datasets and enables better generalization to unseen fault scenarios making it an effective solution for predictive maintenance in real-world Industry 5.0 systems.

1.3 Problem statement

Despite significant advancements in BFD, deploying AI-driven WSNs in real industrial environments remains a complex challenge. Several limitations continue to hinder the effectiveness and practicality of such systems:

- **Computational Constraints:** WSNs are characterized by limited onboard processing capabilities, which restrict the implementation of traditional DL architectures that are computationally intensive.
- **Data Scarcity:** Acquiring large volumes of accurately labeled vibration data across different fault types, severities, and operating conditions is both expensive and time-consuming. This limitation undermines the training of robust AI models.
- **Overfitting Risk:** Models trained on small or imbalanced datasets often exhibit overfitting, failing to generalize to new or unseen fault conditions, especially when operating conditions vary.

- **Energy Efficiency:** In battery-powered sensor networks, energy consumption is a critical factor. High computational loads can rapidly deplete node energy, reducing network lifetime and reliability.

To address these challenges, this study proposes **BearNet**, a lightweight DL architecture designed for fault classification in WSNs. By leveraging TL with pre-trained models such as YAMNet and selectively fine-tuning only relevant layers, BearNet achieves a balance between diagnostic precision and computational efficiency. The model is particularly suited for embedded environments with constrained memory, processing, and power resources.

1.4 Research questions

This thesis aims to answer the following key research questions:

- **H**ow can TL be effectively utilized to enhance bearing fault classification in WSN-based CM systems?
- **W**hat benefits does a TL-based model offer compared to traditional DL methods when applied under WSN constraints?
- **C**an a lightweight TL strategy, such as selective fine-tuning, improve both computational efficiency and fault classification accuracy?
- **H**ow does the proposed BearNet model perform against benchmark methods in terms of generalization, robustness, and energy efficiency?

1.5 Research aims

The overarching goal of this research is to develop a practical, scalable, and high-performance AI framework for fault detection in WSNs. Specifically, the study aims to:

- **Design and implement BearNet,** a novel DL architecture tailored for BFD using spectrograms of vibration signals.
- **Incorporate TL** from pre-trained audio classification models (e.g., YAMNet), enabling the reuse of domain-invariant features across tasks.
- **Introduce a fine-tuning strategy** that unfreezes only select layers to optimize adaptation while minimizing computational demands making the approach viable for embedded WSN environments. In addition, custom layers are appended to the pre-trained model to enhance its ability to process vibration spectrograms effectively.
- **Evaluate the model performance** using publicly available datasets like CWRU and MFPT, and benchmark it against conventional DL approaches to demonstrate improvements in accuracy, generalization, and efficiency.

1.6 Thesis structure

The remainder of this thesis is organized to guide the reader from background knowledge and motivation, through methodology and implementation, to experimental validation and conclusions. Each chapter builds on the previous one and together they present the full development and evaluation of the proposed BearNet approach for WSN based BFD.

- **Chapter II Background: BFD Tasks and Datasets.** A technical overview of bearing structure and defect types, the standard workflow for vibration-based diagnosis (feature extraction and pattern recognition), and detailed descriptions and comparisons of public benchmark datasets (CWRU, PRONOSTIA, IMS, Paderborn, MFPT).
- **Chapter III Background and Related Work.** A comprehensive literature review covering BFD methods, vibration-based signal processing, machine- and deep-learning approaches, the role of WSNs (WSNs) in CM, and existing transfer-learning applications in fault diagnosis. This chapter synthesizes past work and identifies open challenges addressed in this thesis.
- **Chapter IV TL.** A focused presentation of transfer-learning concepts, taxonomies, and practical issues for domain adaptation. The chapter reviews factors that influence TL performance, discusses deep TL strategies (feature extraction vs. selective fine-tuning), and provides an end-to-end guide applicable to spectrogram-based bearing diagnosis.
- **Chapter V Analysis of the BearNet Method.** A detailed description of the proposed BearNet architecture and TL workflow: source/target datasets, preprocessing to Mel-spectrograms, model design (YAMNet adaptation and custom layers), training strategy (selective fine-tuning), and thorough experimental evaluation including confusion-matrix analyses, comparisons with benchmarks, and generalization tests on MFPT and other datasets.
- **Chapter VI General Conclusion.** A concise summary of the research contributions, a discussion of the main findings and limitations, and recommendations and concrete directions for future work (e.g., on deployment, energy-aware inference, and broader dataset validation).

This structured approach ensures a clear and logical progression, from theory through implementation and empirical validation,while also providing a coherent flow of information that builds towards the research contributions, practical conclusions, and directions for future work.

Chapter 2

Bearing fault diagnosis tasks and datasets

Contents

II.1	Introduction	12
II.2	Condition-Based Maintenance (CBM) and Condition Monitoring for Bearings	12
II.3	General Workflow of Bearing Fault Diagnosis	15
II.4	General procedure for BFD techniques	24
II.5	Comparison of Popular Bearing Fault Datasets	27
II.6	Conclusion	34

2.1 Introduction

This chapter introduces the bearing fault diagnosis (BFD) tasks addressed in this thesis. Signals from rotating machinery vibration, airborne acoustics, and motor current are inherently sequential, so many BFD applications rely on segment- or sequence-level labeling. We focus primarily on vibration-based classification and sequence labeling over time series, while noting that the same ideas extend to other modalities.

Raw bearing signals are first denoised (e.g., filtering), resampled to a common rate, and standardized to ensure comparability across sources. The data are then partitioned into fixed-length windows using a sliding window with overlap; each window may be tapered (e.g., Hamming) to improve spectral behavior. To mitigate class imbalance, light data augmentation can be applied. When using 2D models, segments can optionally be converted to timefrequency representations such as Short-Time Fourier Transform (STFT) or Mel spectrograms.

2.2 Condition-based maintenance and condition monitoring for bearings

Condition-Based Maintenance (CBM) plans interventions according to the *measured health* of assets instead of set time intervals. By responding to degradation, CBM reduces unexpected downtime and lifecycle cost, plans interventions before failure, and detects problems early [18].

The analytics and sensing foundation for CBM is provided by **CM**. It uses signals like vibration (mainly for bearings), acoustic emission, airborne sound, motor current, and temperature to continuously evaluate the health of the machine. When properly preprocessed and analysed, these data show early flaws long before they become functional, allowing for "the right action at the right time" [19].

2.2.1 CBM workflow for bearings :Maintenance-level view

Everywhere you look, you'll find rolling-element bearings, which are reasonably priced parts whose failure can result in excessive downtime. Often, faults begin quietly and create weak signatures (such as modulated impulses and harmonics) that gradually get stronger. These indicators are transformed into *actionable* maintenance by CBM, which includes early detection, remaining useful life (RUL) estimation, and scheduling repairs around production windows.

A practical end-to-end CBM workflow comprises:

a. **Sensing infrastructure:** deploy accelerometers on the housing (optionally acoustic emission, airborne sound, stator current) with appropriate bandwidth, sampling, and placement.

b. **Health indicators:** compute robust condition indicators (e.g., Root Mean Square (RMS), kurtosis, crest factor; envelope-spectrum peaks near BPFO/BPFI/BSF/FTF; selected time–frequency metrics).

c. **Diagnostics:** detect faults, identify location (inner/outer race, rolling element, cage), and assess severity.

d. **Prognostics:** estimate RUL with uncertainty bounds by tracking indicator trends.

e. **Decision support:** translate health and RUL into planned interventions (work orders, spares staging, scheduled stops).

f. **Maintenance execution (corrective):** safely isolate equipment, replace/repair the faulty element (e.g., bearing replacement, re-lubrication, re-alignment/balancing), restore to service.

2.2.2 Deciding exactly when to intervene

The question now is , Deciding Exactly When to Intervene?, the decision of *when* to act is framed by the **P–F interval**,—the elapsed time between the earliest *Potential* failure detection (P) and *Functional* failure (F)— within this window, practice combines three complementary logics. First, **threshold-based scheduling** issues work when validated indicators (e.g., envelope-spectrum peaks at fault orders, rising RMS/kurtosis) exceed alarm limits. Second, **RUL based planning times** the intervention so that the lower confidence bound of the remaining useful life remains positive until the next planned stop. Third, **risk- and cost-aware** optimization balances the escalating probability and impact of failure if action is deferred against the opportunity cost of early replacement.

In robust CBM programs, alarms provide early warning, RUL quantifies timing, and cost models finalize the maintenance window. The *P–F interval* is central to CBM decision-making. Because many asset populations do not exhibit a clean wear-out pattern, acting on condition rather than calendar time is both practical and economically sound. In bearing applications, the P–F interval defines a planning window that is late enough to exploit remaining life yet early enough to avoid forced outages; inspection cadence and alarm limits should be set so that the interval between inspections is comfortably shorter than the expected P–F window.

1) The P–F curve in the bearing context: A useful and reproducible method for establishing the concept of bearing P-F at different severity levels is to use the bearing's vibration signature, which allows for early damage detection. The nominal sizes of the original defects create a qualitative severity scale that precisely corresponds to the progression from initial detection of "potential" failure (P) to "functional" failure (F), even though they do not evolve during the same operation as natural spalls do.

At incipient or near-*P* conditions, vibration indicators such as envelope RMS and kurtosis begin to deviate from baseline after band-limited demodulation, and weak components appear at characteristic fault orders (e.g., BPFO/BPFI). As defect severity increases, the signal

exhibits clearer impulsive trains, stronger envelope-spectrum peaks with multiple harmonics, and shaft-frequency sidebands, accompanied by steeper trends in condition indicators characteristic of mid *P–F* progression. In late-stage or near-*F* behavior, secondary effects typically emerge, including broadband elevation, signatures of looseness or misalignment, and, in some cases, rising temperature, signaling urgency for intervention.

This qualitative mapping of seeded defect severities to positions along the *P–F* curve serves three purposes: (i) it illustrates the actionable window with reproducible reference points; (ii) it supports the definition of graduated alarm thresholds aligned with physically interpretable changes in the signal; and (iii) it enables the development of severity-aware classifiers or regressors even when true run-to-failure datasets are unavailable. Figure 2.1 summarizes this concept for bearings, highlighting P (first reliable detection), the actionable planning window, and F (loss of function).

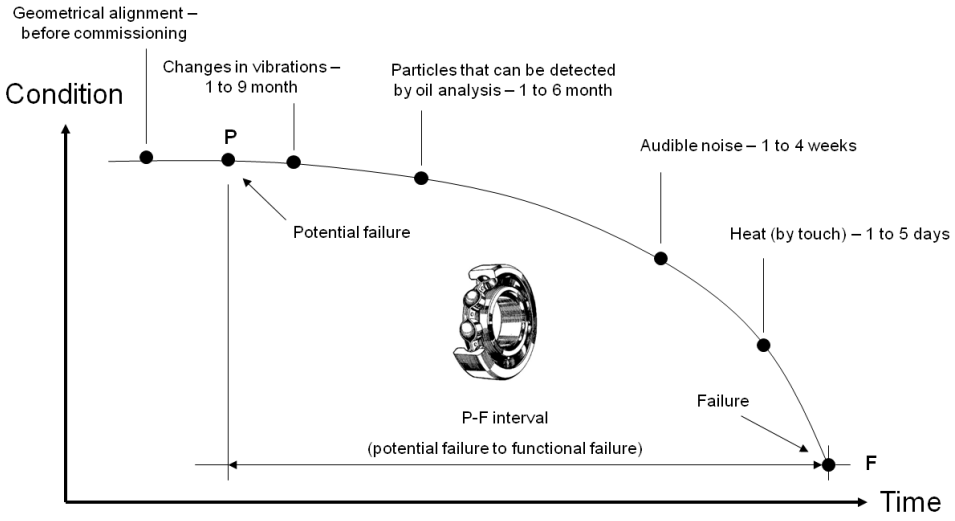


Figure 2.1: The P–F interval for a bearing: detection point (P), actionable planning window, and functional failure (F) [20].

2) Scheduling logic within the P–F window. In practice, three complementary strategies are used:

- *Threshold-based triggers:* raise a work order when validated indicators (e.g., envelope peaks at fault orders, growth in envelope RMS/kurtosis) cross alarm limits.
- *RUL-based planning:* schedule at the next feasible stop such that the lower confidence bound of the estimated RMS remains positive over the planned run.
- *Risk-/cost-aware optimisation:* minimise expected total cost by balancing the increasing probability/impact of failure if deferred against the cost of early replacement.

2.3 General workflow of BFD

Bearing fault diagnosiss (BFDs) follows a structured workflow that transforms raw sensor data into actionable diagnostic decisions. This workflow comprises four key stages sensing

and acquisition, preprocessing, feature extraction, and fault classification each ensuring that useful information is extracted from noisy industrial environments and correctly interpreted.

1) Sensing and Data Acquisition: The process begins with sensors such as accelerometers, acoustic emission transducers, or temperature probes. These devices capture physical signals (vibration, sound, temperature) generated by rotating machinery and transmit them to acquisition systems at appropriate sampling rates.

2) Signal Preprocessing: Raw sensor signals are often contaminated with noise, interference, and irrelevant background components. Preprocessing filtering, normalization, denoising enhances signal quality and prepares the data for further analysis. Segmentation into time windows and synchronization with operating conditions are also performed at this stage.

3) Feature Extraction: Features can be computed in the *time domain* (e.g., RMS, kurtosis, skewness), the *frequency domain* (e.g., fast Fourier tranform (FFT), spectral peaks), or the *timefrequency domain* (e.g., wavelet transform, empirical mode decomposition). The choice depends on the fault type, machinery, and objectives.

4) Fault Classification and Decision-Making: Extracted features feed rule-based methods, statistical models, or ML/DL classifiers to assign a fault label (e.g., outer race, inner race, ball, or healthy) and potentially estimate severity.

This generic workflow provides a unifying framework for both traditional and intelligent fault diagnosis methods. The following sections present in more detail the traditional approaches, which laid the foundation for current AI-driven techniques.

Figure 2.2 provides a visual overview of the key stages involved in BFDs, highlighting the sequential flow from sensing and acquisition to signal processing and decision-making. This modular structure serves as the foundation for both traditional and AI-enhanced fault detection systems [21].



Figure 2.2: Stages from sensing and acquisition to signal processing and decisions.

2.3.1 Overview and classification of traditional BFD techniques

Traditionally, fault diagnosis has been performed using a variety of physical inspection methods and signal analysis techniques. These methods laid the foundation for modern predictive maintenance and remain relevant today, particularly when integrated with intelligent systems. As shown in Figure 2.3, these techniques can be grouped into four primary categories:



Figure 2.3: Traditional BFDs Techniques.

1) Visual Inspection and Manual Fault Detection: One of the earliest approaches to bearing fault detection is visual inspection. This technique involves direct observation and manual handling checking for surface damage, grease contamination, seal integrity, and lubricant degradation. While useful for identifying surface-level issues, this method is labor-intensive, subjective, and ineffective for internal or incipient faults.

2) Vibration Analysis and Signal Processing: Vibration analysis is a cornerstone of traditional CM. Bearings naturally emit vibration signals during operation, and faults such as misalignments, surface wear, or mechanical looseness cause characteristic changes in these signals. Signal processing methods in the time-domain (e.g., RMS, kurtosis), frequency-domain (e.g., FFT), and time-frequency domain (e.g., Wavelet Transform, **EMD!** (**EMD!**)) are employed to extract meaningful features [4]. These approaches are effective but often require expert interpretation and finely tuned sensor setups [22–24].

3) Acoustic Emission Monitoring: In noisy industrial environments or systems with low-frequency operation, vibration signals may be masked by background noise. Acoustic Emission (AE (AE) monitoring captures high-frequency elastic waves,typically around 100 kHz, generated by microscopic cracks or surface defects. AE transducers are capable of detecting these transient events with minimal interference from surrounding machinery. Although AE offers a high signal-to-noise ratio and early fault detection capabilities, it requires specialized equipment and expert analysis [24, 25].

4) Temperature Monitoring: Monitoring the operating temperature of bearings provides

an indirect but effective means of identifying issues such as excessive friction, inadequate lubrication, or internal wear. A stable bearing should not exceed 45řC under normal load. Temperature sensors can alert operators to anomalies in real time. However, this method lacks fault specificity and is often used in conjunction with other techniques [26, 27].

Traditional bearing fault diagnosis techniques have been extensively applied across industrial environments, with varying degrees of popularity depending on the nature of the application, cost, and required accuracy. Among these, **vibration analysis is the most widely adopted** due to its high sensitivity to incipient faults and compatibility with both signal processing and modern AI-based techniques. Other methods, such as AE and temperature monitoring, are typically used in conjunction to enhance diagnostic accuracy, especially in environments with low-speed rotation or high ambient noise. Table 2.1 summarizes the approximate prevalence of each principal traditional technique based on findings in recent literature [1, 4, 22–27].

Table 2.1: Approximate Usage of the Principal Traditional BFD Techniques in Literature

Technique	**Approx. Usage (%)**	**Notes**
Vibration Analysis	65--80%	Most widely used due to high sensitivity to fault development and compatibility with signal processing.
AE	10--15%	Increasingly used in high-noise or low-speed applications; detects early-stage faults.
Temperature Monitoring	5--10%	Simple and low-cost; mostly used as a complementary method.
Visual Inspection / Manual	5--8%	Still practiced, especially in low-tech settings; limited by downtime and operator dependency.

A major portion of the chapter is dedicated **exclusively to vibration-based methods**, as they remain the most prominent and widely used in the field.

2.3.2 Vibration analysis based bearing fault diagnosis

Accurate and reliable diagnosis of bearing faults is a cornerstone of modern CM systems. In industrial applications, this diagnosis is primarily based on the analysis of vibration signals emitted by rotating machinery. However, the raw vibration signals acquired during monitoring are often contaminated with noise and interference (Equation 2.1), making them unsuitable for direct analysis or use in fault detection algorithms. Consequently, these signals must undergo preprocessing and transformation through specialized signal processing techniques to extract relevant diagnostic information. Figure 2.4 provides a visual overview of the key stages involved in vibration-based fault diagnosis, highlighting the sequential flow from sensing and acquisition to signal processing and decision-making. This modular structure serves as the foundation for both traditional and AI-enhanced fault detection systems [21].

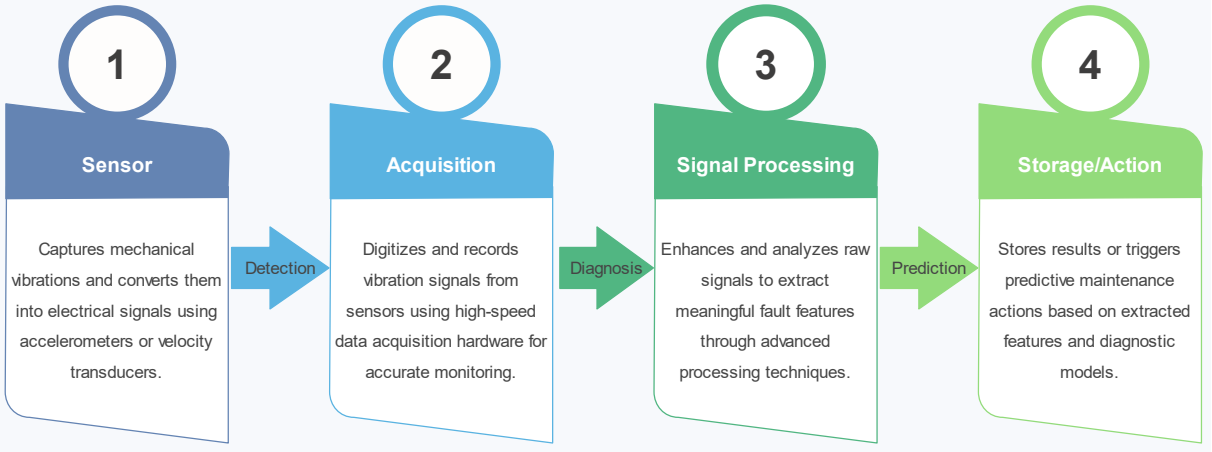


Figure 2.4: Different stages of vibration analysis.

2.3.3 Bearing structure and components

From an engineering perspective, a bearing is a precision-machined component that facilitates constrained relative motion between mechanical parts, typically between a rotating shaft and its housing. Its primary function is to minimize friction and wear while supporting both radial and axial loads.

A standard bearing is typically composed of four main components: the **inner ring**, **outer ring**, **rolling elements** (usually balls or rollers), and the **cage** that maintains spacing between the rolling elements. In certain specialized bearings, such as *spherical roller bearings*, additional elements like *guides* and *seals* may be included to enhance stability and performance under complex loading conditions [28].

The **inner ring** is generally mounted on the rotating shaft, while the **outer ring** is fixed within the housing. The **rolling elements** move between these two rings and serve to transmit loads with minimal friction. When *balls* are used, the contact with the rings is at a single point, which limits load capacity compared to *rollers*, which create line contact and can support higher loads.

The **cage** prevents direct contact between rolling elements, ensuring proper lubrication, reducing wear, and maintaining the bearings structural integrity during handling and operation.

Figure 2.5 provides a visual representation of the main components of a bearing.

Bearing faults can occur in three primary components: the *outer race*, *inner race*, and *rolling elements (balls)*. Accurate mathematical modeling of the corresponding vibration signals is essential to develop effective detection algorithms capable of identifying and classifying these faults.

To better understand these vibration patterns, Figure 2.6 provides a schematic representation of a bearings physical structure. It shows the arrangement of rolling elements (balls), the pitch diameter D, the ball diameter d, and the contact angle β, all of which influence the generated vibration frequencies and are used to calculate the characteristic fault frequencies: BPFO, BPFI, and BSF.

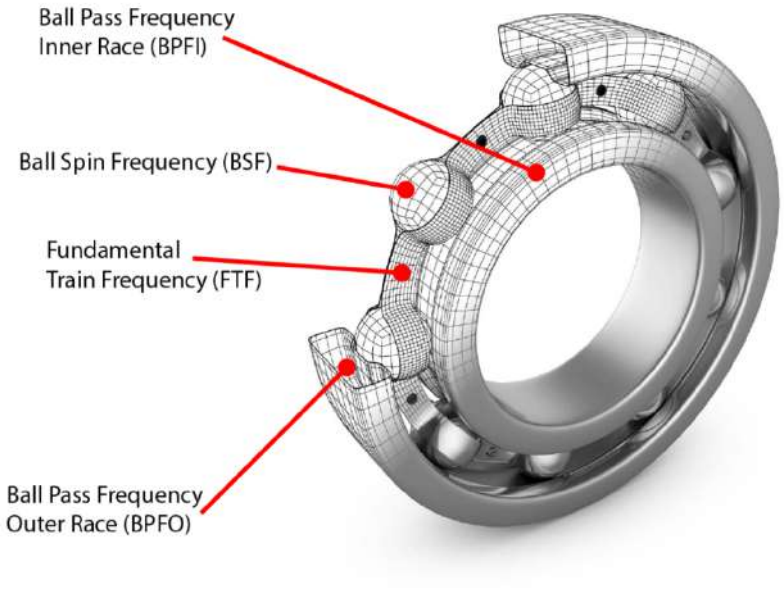


Figure 2.5: Illustration of characteristic defect frequencies in a ball bearing: Ball Pass BPFI, BPFO, BSF, and FTF.

These figures are essential in explaining how the location and nature of a defect influence the resulting signal, thus helping in accurate fault diagnosis through signal processing and frequency analysis.

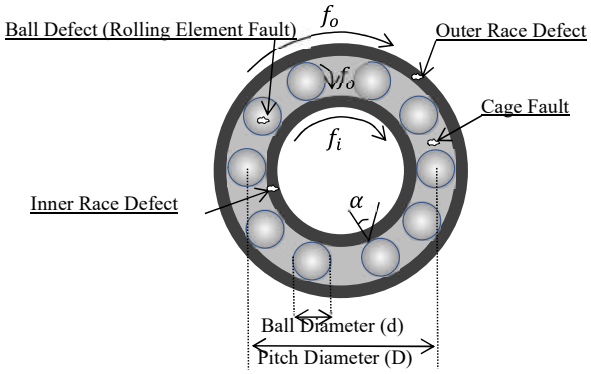


Figure 2.6: Schematic structure of a bearing showing elements relevant to fault frequency calculations: number of rolling elements n, ball diameter d, pitch diameter D, and contact angle β [29].

Below, we present the mathematical equations for the vibration signal model, fault signal models, and their frequency-domain representations.

1) Vibration signal model: The vibration signal $x(t)$ captured from a bearing system is typically modeled as a composite of various signal components:

$$x(t) = s(t) + n(t) + r(t) \tag{2.1}$$

where $x(t)$ is the observed signal, $s(t)$ is the fault-induced component, $n(t)$ represents noise, and $r(t)$ captures non-fault-related vibrations from other sources.

2) Fault signal models: Each fault type exhibits a unique signature in the vibration signal. These signatures can be mathematically modeled as follows:

- **Outer race fault:** Caused by repeated impacts of rolling elements against a localized defect on the outer race [30]. The corresponding signal is modeled as:

$$s_{\text{outer}}(t) = \sum_{k=1}^{N} A_k \cdot \delta(t - kT_{\text{outer}}) \cdot \cos(2\pi f_{\text{res}} t + \phi) \tag{2.2}$$

- **Inner race fault:** Produces periodic impacts as the inner race defect comes into contact with the rolling elements [31]. Its signal is expressed as:

$$s_{\text{inner}}(t) = \sum_{k=1}^{N} B_k \cdot \delta(t - kT_{\text{inner}}) \cdot \cos(2\pi f_{\text{res}} t + \phi) \tag{2.3}$$

- **Ball fault:** Typically leads to non-stationary signals influenced by the dynamics of both races [32]. It is modeled as:

$$s_{\text{ball}}(t) = \sum_{k=1}^{N} C_k \cdot \delta(t - kT_{\text{ball}}) \cdot \cos(2\pi f_{\text{res}} t + \phi) \tag{2.4}$$

  where:

  - A_k, B_k, C_k: amplitude of the k^{th} impact for outer race, inner race, and ball faults respectively,
  - $\delta(t)$: Dirac delta function representing an instantaneous impact,
  - $T_{\text{outer}}, T_{\text{inner}}, T_{\text{ball}}$: impact repetition intervals for each fault type,
  - f_{res}: resonant frequency of the system,
  - ϕ: initial phase of the modulated signal,
  - N: total number of impulses considered.

3) Frequency domain representation: Frequency analysis techniques such as the Fourier tranform (FT) are commonly employed to identify characteristic fault frequencies associated with specific types of bearing defects. These characteristic frequencies are summarized in Table 2.2 [33].

Table 2.2: Characteristic fault frequencies for bearing defects.

Fault Type	Frequency Formula
Outer Race Fault (BPFO)	$f_{\text{BPFO}} = \frac{n}{2}\left(1 - \frac{d}{D}\cos(\beta)\right) f_r$
Inner Race Fault (BPFI)	$f_{\text{BPFI}} = \frac{n}{2}\left(1 + \frac{d}{D}\cos(\beta)\right) f_r$
Ball Fault (BSF)	$f_{\text{BSF}} = \frac{D}{2d}\left(1 - \left(\frac{d}{D}\right)^2 \cos^2(\beta)\right) f_r$

where n is the number of rolling elements, d is the ball diameter, D is the bearing pitch diameter, β is the contact angle, and f_r is the shaft rotational frequency.

Beyond isolated bearing components, fault diagnosis extends to entire rotating systems. In such systems, vibration signals must travel through various mechanical structures before reaching the sensors, which adds complexity to the signal interpretation process. This complex propagation path introduces further challenges, underscoring the need for robust signal processing and diagnostic algorithms. As such, ongoing research continues to explore new techniques to improve accuracy and reduce diagnostic uncertainty in real-world applications.

While these traditional diagnostic methods have proven effective over decades, they exhibit notable limitations. Manual inspections are prone to human error and inefficiency, while signal-based techniques rely on handcrafted features and expert interpretation. Moreover, they struggle with noisy environments, low-speed machinery, and complex fault patterns.

As shown in Figure 2.7, different bearing faults produce distinct vibration signatures that are influenced by the location and nature of the defect. For instance, defects on the outer race tend to generate consistent periodic responses due to the regular interaction between rolling elements and the stationary fault zone. In contrast, inner race defects create signals that vary in amplitude because the defect rotates with the shaft, altering its interaction pattern with the rolling elements. Ball defects introduce even more complex signal patterns, as the contact between the rolling element and both races varies over time. These differences in signal behavior are critical for identifying the type and severity of faults during the diagnostic process.

2.3.4 Evolution of bearing defects and their vibration signatures

Evolution of Bearing Defects is illustrated in Figure 2.8 (adapted from [1]), the progression of bearing faults typically follows four distinct stages, each associated with specific vibration signatures. These stages begin with the initiation of microscopic cracks and evolve into more severe damage such as crack propagation, the formation of loose particles, and ultimately, spalling. Recognizing the vibration characteristics at each stage is essential not only for confirming the presence of a fault but also for assessing the severity and progression of the defect. By analyzing these evolving patterns, maintenance teams can make informed deci-

[1] https://www.spminstrument.it/Tecniche-di-misura/HD-technologies/

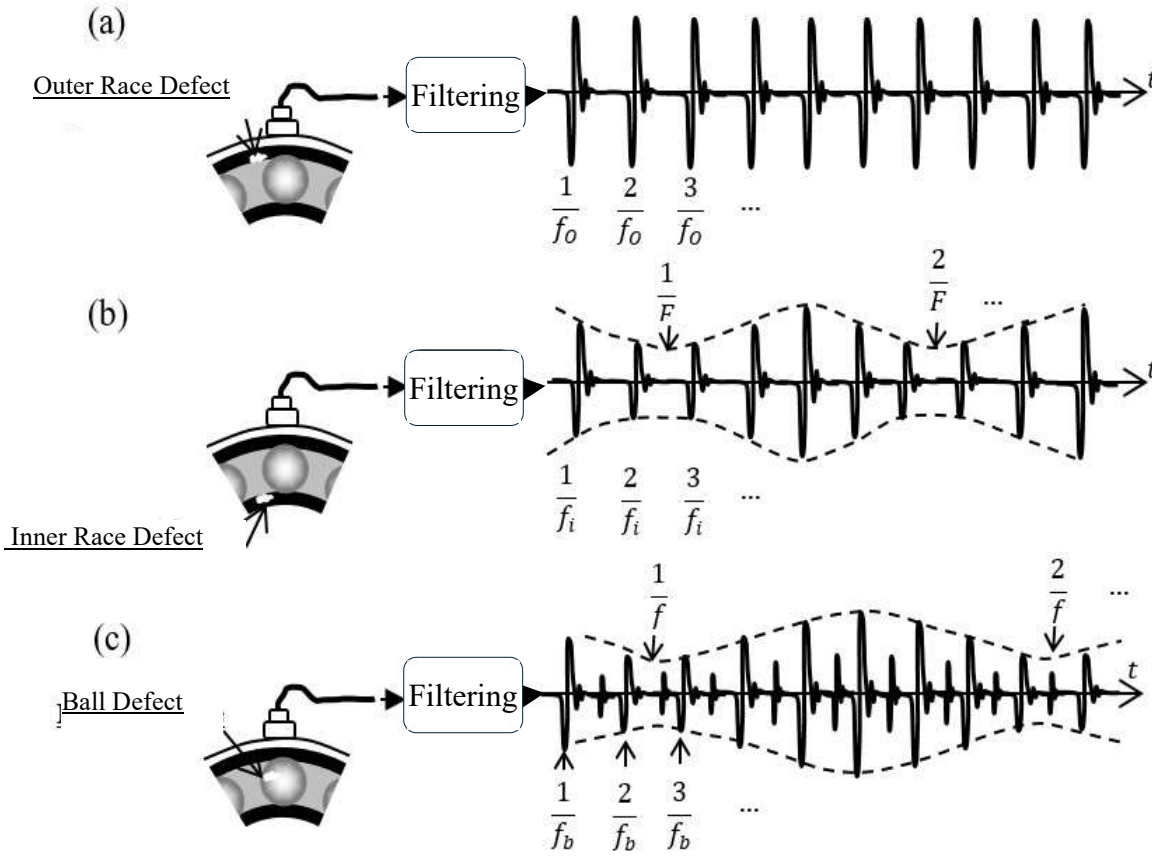


Figure 2.7: Vibration signal patterns for different bearing faults: (a) Outer race fault, (b) Inner race fault, and (c) Ball defect. Each fault type produces a distinct signal influenced by fault location and dynamics [29].

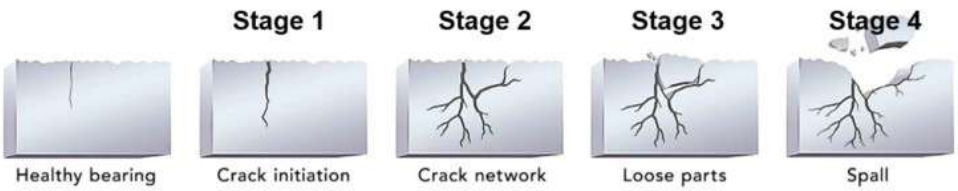


Figure 2.8: Evolution of Subsurface Defects.

sions about intervention timing, potentially preventing catastrophic failures and extending the operational life of machinery [34].

2.4 General procedure for BFD techniques

Modern BFD relies on a structured two-stage process: **(Firstly) Feature Extraction**, where raw sensor signals are transformed into meaningful diagnostic information, and **(secondly) Pattern Recognition**, where these features are used to classify and detect fault types. Together, these stages form the foundation of intelligent CM systems. The following subsections provide an overview of the diagnostic process and trace the technological evolution that has shaped current bearing monitoring techniques.

2.4.1 Overview and importance of fault diagnosis

Rolling bearings are essential in rotating machinery but prone to defects under harsh operating conditions, making timely and accurate fault diagnosis critical to avoid failures, minimize downtime, and ensure reliability.

The modern approach to BFD is typically structured into two fundamental stages: **fault feature extraction** and **fault pattern recognition**, as illustrated in Figure 2.9 [35]. This two-stage diagnostic framework has become the foundation of intelligent health monitoring systems for rotating machinery, playing a pivotal role in advancing predictive maintenance under Industry 4.0 and 5.0 paradigms.

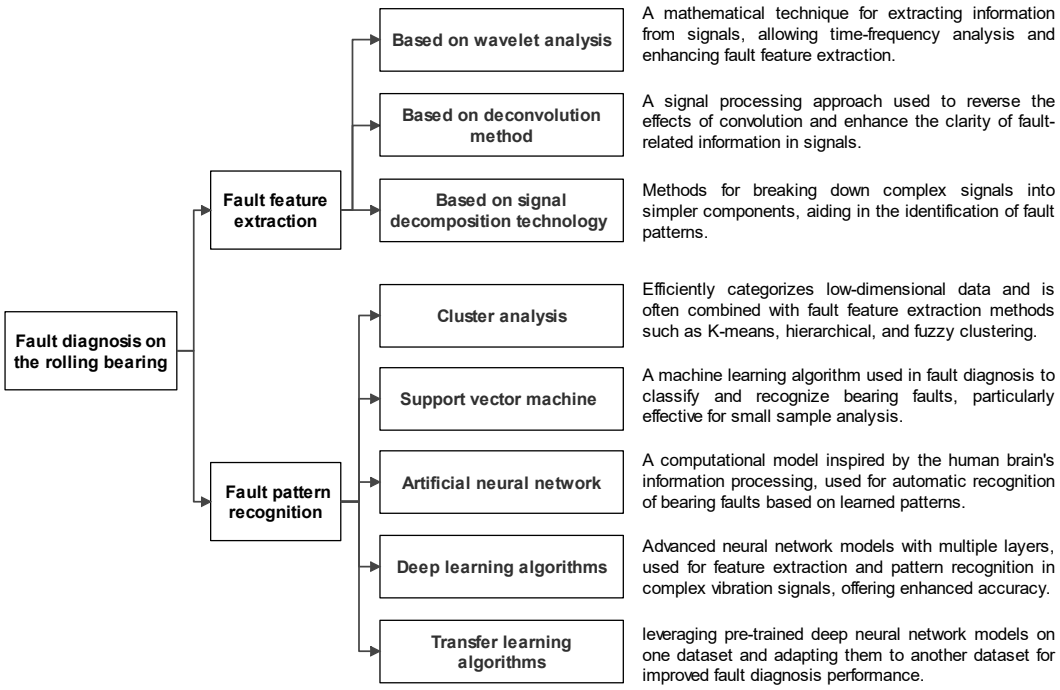


Figure 2.9: General process of fault diagnosis for rolling bearings, including feature extraction and pattern recognition techniques.

1) Feature Extraction Stage Once sensor data such as vibration or acoustic signals are collected, the first step involves isolating key fault-related information from the raw signals. Signal processing techniques play a foundational role in this stage, converting time-domain signals into informative representations that reveal hidden patterns.

Among the most widely applied techniques are the |*Fast Fourier Transform (FFT)*, which is particularly effective for capturing frequency-domain features, and the *Continuous Wavelet Transform (CWT)*, well suited for analyzing non-stationary signals due to its timefrequency localization capability. In addition, *Empirical Mode Decomposition (EMD)* and its improved variant, *CEEMDAN*, enable the decomposition of complex signals into intrinsic mode functions (IMFs), thereby facilitating more precise fault isolation. *Mel-spectrograms* are frequently employed in conjunction with deep learning models for both acoustic and vibration-based fault diagnosis, providing a compact and informative timefrequency representation. Finally, *deconvolution techniques* are used to enhance signal clarity by compensating for system-induced distortions and improving the signal-to-noise ratio.

These transformations not only increase interpretability but also enhance the robustness of downstream diagnostic models.

2) Pattern Recognition Stage
After extracting informative features, the next step is to classify and identify fault types using intelligent algorithms. Traditional approaches include *Support Vector Machines (SVM)*, which construct optimal hyperplanes for separating classes, and *k-Nearest Neighbors (k-NN)*, which classify samples based on proximity to labeled neighbors in feature space. *Decision Trees* provide an interpretable, rule-based structure for classification, while *Cluster Analysis* methods, such as K-means and hierarchical clustering, group data points with similar characteristics to uncover fault patterns without explicit labels. These conventional methods have been widely adopted in earlier systems and offer reasonable performance on structured datasets; however, they rely heavily on manual feature engineering, which is often time-consuming, sensitive to noise, and may limit their ability to generalize across varying operating conditions.

These limitations have paved the way for intelligent, automated solutions that leverage the power of AI. In particular, the integration of ML and, more recently, DL has revolutionized fault detection systems by automating feature extraction, improving classification accuracy, and enabling scalable, real-time monitoring. Among these approaches, **DL** models such as CNNs, Recurrent Neural Networks (RNN), and Deep Belief Networks (DBN) have emerged as powerful alternatives to traditional methods. Unlike conventional techniques that rely heavily on handcrafted features, these models are capable of learning hierarchical representations directly from raw or transformed input data, thereby significantly enhancing diagnostic precision and robustness in complex industrial environments.

In addition, transfer learning (TL) has become a transformative tool for generalizing models to different machines or operational scenarios. By fine-tuning pre-trained models on new datasets, TL minimizes the need for large labeled datasets and enables effective adaptation across domains.

The two-stage framework combining advanced signal processing with intelligent clas-

sification forms the backbone of modern BFD. This integrated pipeline supports real-time, automated, and accurate health monitoring, making it indispensable for predictive maintenance and industrial automation.

2.4.2 Evolution of technologies in bearing condition monitoring

Over the years, technologies have evolved dramatically from simple vibration analysis in the 1950s to sophisticated AI-driven diagnostic systems today. This evolution is not just technical, it reflects our growing need for faster, smarter, and more reliable maintenance strategies. Figure 2.10 illustrates this journey through time, showing how innovations in signal processing, ML, and DL have shaped the tools we use to detect and diagnose bearing faults.

In the early days, techniques like the **Fourier Transform** were used to find periodic vibration patterns. As we moved into the 1970s and 1980s, **ML** and early **neural networks** began to support classification tasks. The 1990s introduced **wavelet transforms** for capturing complex, transient signal features. The 2000s gave us powerful signal decomposition methods such as **EMD** and **CEEMDAN**, often combined with optimization algorithms like **PSO** or **SVM** [36].

A major leap occurred in the 2010s with the rise of **DL** especially **CNN** which enabled systems to automatically learn features from complex datasets. Now, in the 2020s, we're seeing a shift to even smarter technologies, such as **TL**, **Federated Learning**, **Transformers**, and **Edge AI**, allowing diagnostics to happen in real-time, even on resource-limited edge devices.

These advances are driving more predictive, accurate, and scalable maintenance solutions. They not only improve reliability but also support larger environmental and economic goals, such as reducing emissions and optimizing energy use.

2.5 Comparison of popular bearing fault datasets

The rapid development of AI and ML has been driven above all by the availability of large and representative datasets. In the current era, data are not merely supporting resources but the essential substrate upon which intelligent algorithms learn, adapt, and generalize. High-quality datasets provide the ground truth that enables models to capture complex nonlinear dependencies, validate their robustness, and transition from purely descriptive approaches to predictive and even prescriptive analytics. Without such datasets, even sophisticated learning architectures remain limited, prone to bias, and unable to generalize effectively.

Within the field of BFD, datasets occupy a particularly critical role. Bearings are among the most frequently failing components in rotating machinery, and their degradation has a direct impact on machine reliability, safety, and maintenance costs. Publicly available repositories, often based on seeded fault experiments or run-to-failure tests, have become cornerstones for both research and practice. This allows to study the transition from incip-

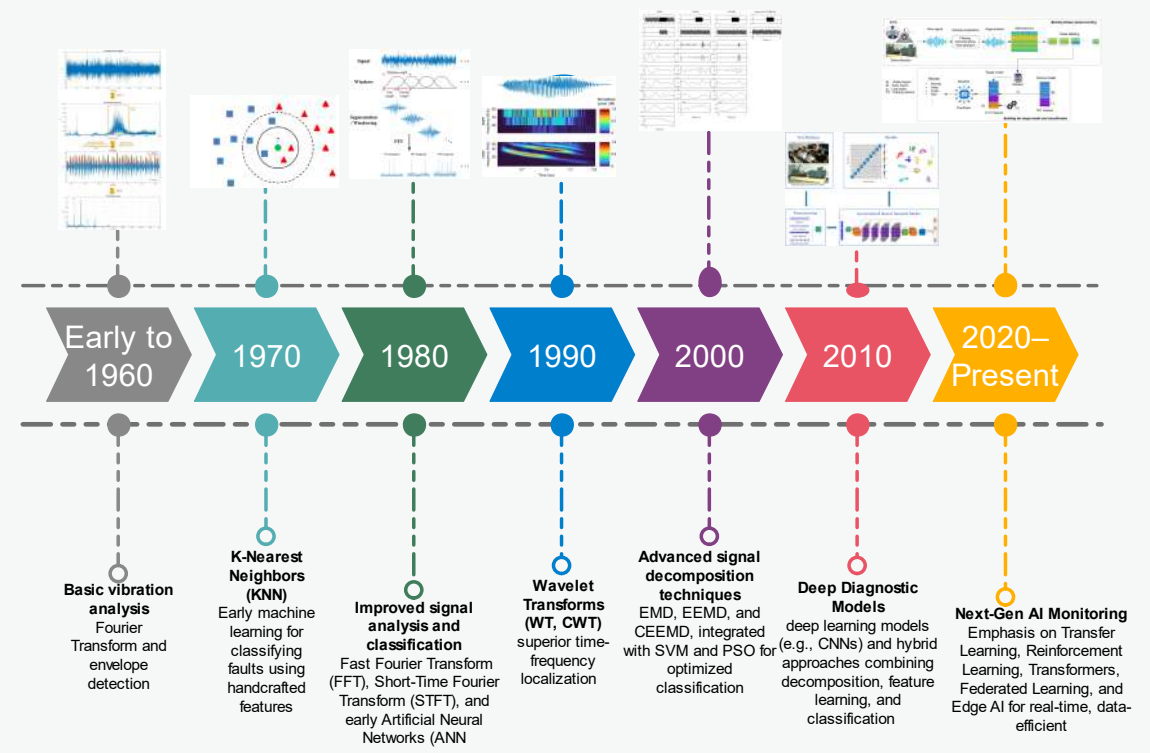


Figure 2.10: Timeline of technological development and techniques in bearing CM and fault diagnosis.

ient faults where weak characteristic frequencies such as BPFO and BPFI begin to appear through mid-stage progression with impulsive trains and harmonics, up to late-stage signatures where broadband noise, looseness, and signal imminent functional failure.

The importance of such datasets lies not only in their ability to illustrate the bearing degradation path with clarity but also in their role as benchmarks for developing and comparing intelligent diagnostic models. They make it possible to test novel algorithms under controlled yet realistic conditions, to calibrate alarm thresholds in line with interpretable physical phenomena, and to design severity-aware prognostic tools even in the absence of long-term run-to-failure data. In this sense, datasets serve as the bridge between theoretical advances in ML and their industrial implementation, ensuring that diagnostic systems are both scientifically rigorous and practically deployable.

2.5.1 CWRU dataset description

The Case Western Reserve University (CWRU) dataset is one of the most widely used benchmarks for bearing CM and fault diagnosis. It provides a rich variety of vibration signals representing both healthy bearings and those with localized defects on the inner race, outer race, or rolling elements. These data are invaluable for advancing predictive maintenance strategies, as they enable researchers to analyze fault characteristics under controlled but realistic operating conditions.

Measurements were collected using accelerometers strategically mounted on the motor housing, primarily at the drive end, to capture the dynamic response of the system. The

dataset covers different motor loads (0, 1, 2, and 3 HP) and a range of rotational speeds (1797 to 1730 RPM), with vibration signals sampled at 12 k samples per second (sps). In addition to the standard drive-end recordings, other categories of data are also available, such as fan-end and higher sampling-rate recordings. Figure 2.11 illustrates representative samples of raw vibration sensor signals acquired from the Case Western Reserve University (CWRU) bearing dataset. Each subplot corresponds to a specific fault type and severity, including ball defects, inner race defects, and outer race defects, as well as the baseline normal condition. These waveforms highlight the distinct temporal patterns induced by different fault locations and progression stages, thereby providing a direct visual understanding of how vibration signals evolve under varying health states [37].

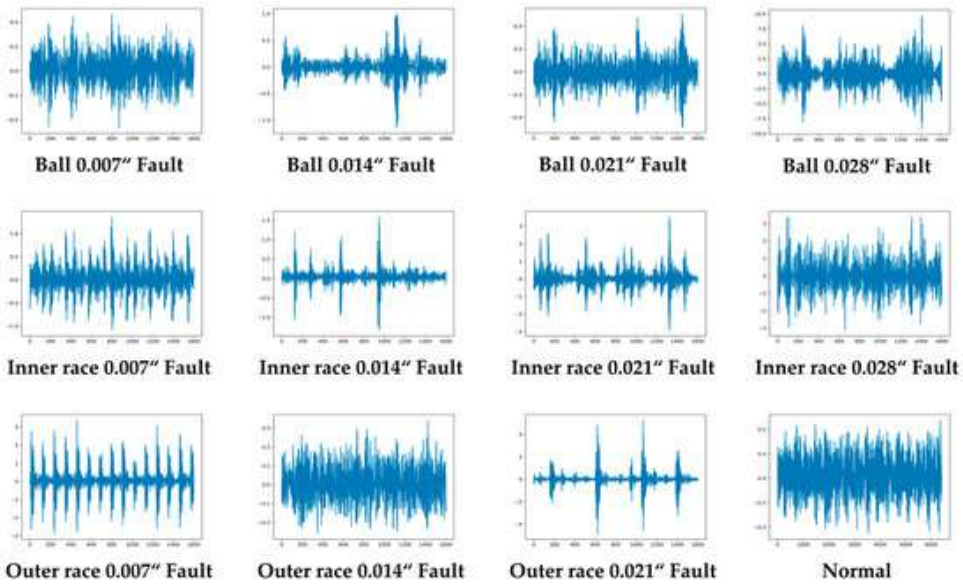


Figure 2.11: Samples of raw vibration sensor signals corresponding to each class [37].

For the purposes of this study, we focus exclusively on the drive-end sensor location, as it provides the most consistent and widely adopted basis for comparison across studies. The full details of the operating conditions and fault configurations can be found in Table 2.3.

To ensure uniformity across datasets, we also down-sampled the 48 kHz drive-end recordings to 12 kHz, thereby harmonizing the sampling rates before preprocessing. This step, which is described further in the dataset preprocessing section, allows for consistent feature extraction and model training while preserving the essential fault-related information.

Table 2.3: Specifications of bearings and sample gathering conditions in the CWRU datasets

Dataset	**Fault Type**	**Load (HP)**	**RPM**	**Operational Settings**
CWRU	Normal	0, 1, 2, 3	1797, 1772, 1750, 1730	Sample rate: 12k sps, Sensor location: Drive end
	Inner Race Fault	0, 1, 2, 3	1797, 1772, 1750, 1730	Sample rate: 12k sps, Sensor location: Drive end
	Outer Race Fault	0, 1, 2, 3	1797, 1772, 1750, 1730	Sample rate: 12k sps, Sensor location: Drive end
	Ball Fault	0, 1, 2, 3	1797, 1772, 1750, 1730	Sample rate: 12k sps, Sensor location: Drive end

2.5.2 PRONOSTIA bearing dataset

The PRONOSTIA dataset is a run-to-failure benchmark widely used for bearing prognostics and Remaining Useful Life (RUL) studies. Bearings are operated under controlled, accelerated degradation: during each experiment the rotational speed is held constant while a radial load exceeding the bearings dynamic capacity induces wear. Vibration is measured in two orthogonal directions using DYTRAN 3035B accelerometers, and temperature is monitored with a PT100 thermocouple. Data collection continues until a failure criterion is reached (vibration amplitude exceeding 20 g), which marks end-of-life [38].

The corpus comprises 17 experiments conducted under three distinct operating conditions, combining load and speed settings: (4000 N, 1800 rpm), (4200 N, 1650 rpm), and (5000 N, 1500 rpm). A common split uses six sequences for training and eleven for testing to emulate deployment to unseen units and conditions. For the test sets, measurements are stored in sub-files spaced at fixed time intervals; each acceleration sub-file contains a fixed number of samples per capture, while temperature files contain shorter records sampled at a lower rate. This design yields full life cycles without seeded faults and makes PRONOSTIA a de-facto standard for comparing classical ML, DL, and TL approaches in bearing health monitoring. At the same time, its constrained operating regimes and specific failure criterion provide a clear, reproducible setting for method development and benchmarking (see Fig. 2.12).

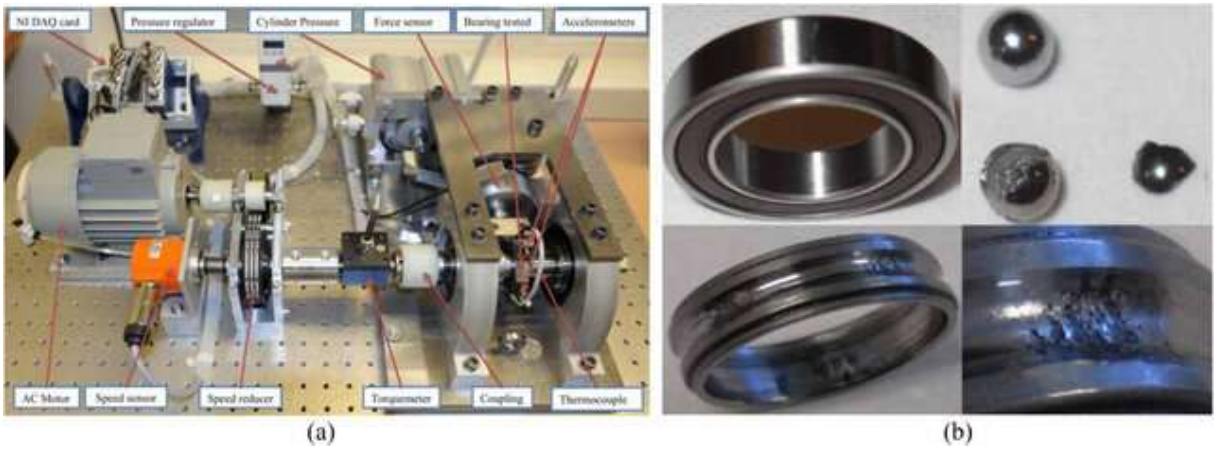


Figure 2.12: PRONOSTIA (IEEE PHM 2012) bearing dataset. (a) Experimental rig with instrumentation for run-to-failure tests (DAQ, sensors, load and speed control). (b) Example bearings in healthy and damaged states.

2.5.3 IMS bearing dataset description

One of the most popular benchmark datasets in the field of prognostics and health management (PHM) is the IMS Bearing Dataset, which was created by the University of Cincinnati's Centre for Intelligent Maintenance Systems (IMS) and is housed by NASA's Ames Prognostics Data Repository. It is now a key component of studies on rolling element bearing failure prognostics, CM, and diagnostics. As shown in Figure 2.13, the IMS bearing test rig enables

simultaneous acquisition of vibration, current, and temperature data under controlled operating conditions.

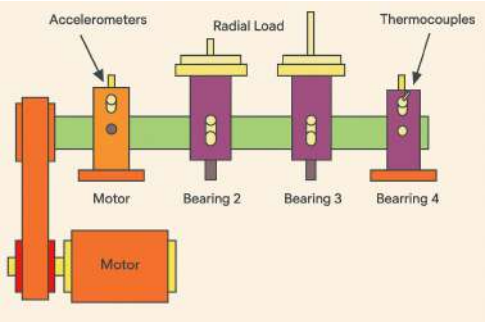


Figure 2.13: The IMS bearing test rig setup. The system consists of an electric motor, four bearings mounted along the shaft, radial load modules, accelerometers, and thermocouples. This configuration enables the acquisition of vibration, load, and temperature data for BFD

Rexnord ZA-2115 double-row bearings mounted on a shaft, powered by an AC motor, and radially loaded by a dynamometer were the subject of a series of controlled run-to-failure tests that produced this dataset. Long-term vibration recordings that document the entire degradation process were made possible by the bearings' continuous operation until natural failure. The dataset is particularly useful for research on fault progression and Remaining Useful Life (RUL) prediction because of this feature.

High-frequency vibration signals obtained at several channels using accelerometers placed thoughtfully on the bearing housings make up the recorded data. Every file represents a one-minute recording that was obtained at regular intervals throughout the experiments and sampled at 20 kHz. Three distinct datasets are available, each of which represents a different bearing's progression to failure. The experiments were carried out with constant load and speed. Roller element failures developed in some cases, while outer race faults were seen in others. Crucially, the failures developed organically during operation rather than being artificially induced, resulting in realistic degradation patterns.

2.5.4 Paderborn university, Kat bearing data-center dataset

One of the most extensive and well-known benchmark datasets for bearing CM and fault diagnosis is the Paderborn Bearing Data-Center, which was created by the Chair of Design and Drive Technology (KAt) at Paderborn University (Germany). Its main goal is to supply researchers with high-quality, repeatable data so they can assess diagnostic algorithms, create ML models, and compare methods for rotating machinery fault detection and prognosis [39].

High-resolution, synchronously measured signals from type 6203 ball bearings installed in an electromechanical drive test rig are included in this dataset. There are two primary signal types available: motor current (sampled at 64 kHz) and vibration acceleration (sampled at 20 kHz). To replicate actual operating conditions, these signals were recorded under various load and speed conditions.

The following table Table 2.4 resume the main elements included in the dataset

Table 2.4: Summary of the main elements included in the dataset

Element	Description
Bearing states	Measurements from 32 distinct bearing states, including: 6 bearings in good condition, 12 with artificially induced damage, and 14 with actual damage resulting from accelerated lifetime testing.
Operating conditions	Four experimental settings with varying torque, load, and rotational speed, designed to capture a wide range of operational variability.
Recordings	For each operating condition, 20 individual recordings are provided, each lasting 4 seconds. The dataset includes fact sheets and structured documentation, and all data are stored in MATLAB format.
Additional signals	Synchronous measurements of temperature, torque, speed, and radial load, enabling multi-sensor fusion approaches for advanced diagnostic research.

A modular test rig comprising an electric motor, torque measurement shaft, bearing test module, flywheel, and load motor makes up the experimental setup. Piezoelectric accelerometers were used to record vibration acceleration, and motor current signals were taken straight from the drive system. The dataset is particularly useful for both conventional signal processing and contemporary data-driven ML methods because of its dual-sensing approach.

2.5.5 MFPT dataset bearing

The MFPT dataset is characterized by several important features. The bearings used are of the NICE brand, with a roller diameter of 0.235 inches and a pitch diameter of 1.245 inches. The baseline measurements include three recordings from healthy bearings operating under a load of 270 lb, at an input shaft rate of 25 Hz. These signals were sampled at 97,656 samples per second, and each recording lasted for 6 seconds.

For fault conditions, two types of bearing defects are provided. The **outer race fault data** consists of two subsets: three recordings under the same operating condition as the baseline (270 lb load, 25 Hz input shaft rate, sampled at 97,656 sps, duration of 6 s each), and seven recordings collected under varying loads ranging from 25 to 300 lb, sampled at 48,828 sps, with a duration of 3 seconds each. Similarly, the **inner race fault data** contains seven recordings obtained under variable loads ranging from 0 to 300 lb, also sampled at 48,828 sps, with each signal lasting for 3 seconds, .

For the purpose of fault classification and ML applications, each vibration file is segmented into windows of 400 ms with a 50% overlap. In addition to vibration data, the dataset also includes complementary signals such as real-world bearing fault measurements from wind turbines, which enhance its relevance and allow for the application of multi-sensor fusion techniques. These features collectively make the MFPT dataset a valuable resource for both experimental and industrial research in BFD, The detailed specifications of the bearings and sampling conditions are summarized in Table 2.5.

Table 2.5: Specifications of bearings and sample gathering conditions in the MFPT dataset.

	Fault Type	Load (lb)	Operational settings
MFPT	Baseline	270	ISR: 25 Hz, SR: 97,656 sps, duration: 6 s
	Outer race fault	270	ISR: 25 Hz, SR: 97,656 sps, duration: 6 s
	Outer race fault	25, 50, 100, 150, 200, 250, 300	ISR: 25 Hz, SR: 48,828 sps, duration: 3 s
	Inner race fault	0, 50, 100, 150, 200, 250, 300	ISR: 25 Hz, SR: 48,828 sps, duration: 3 s

Abbreviations: Input shaft rate (ISR), sample rate (SR), samples per second (sps)

The MFPT datasets variability in fault types, load conditions, and sampling rates makes it a cornerstone for evaluating ML and DL approaches in fault diagnosis and predictive maintenance.

2.5.6 Summary comparison of representative bearing fault datasets

A wide range of publicly available datasets has underpinned the progress of research in BFD. While each dataset is grounded in a specific experimental philosophy ranging from short-term seeded fault tests to long-term run-to-failure experiments they collectively provide the diversity necessary to evaluate and benchmark intelligent diagnostic methods. Table 2.6 synthesizes the main features of five representative datasets that have become cornerstones in the field: CWRU, FEMTO-ST (PRONOSTIA), IMS, Paderborn University, and MFPT.

The **CWRU dataset** remains the entry point for many researchers because of its simplicity, controlled environment, and extensive adoption in the literature. It is particularly well suited for fault classification tasks, where the goal is to distinguish between normal operation and specific localized defects. However, its short records and artificially seeded faults limit its realism when studying degradation processes.

In contrast, the **FEMTO-ST PRONOSTIA dataset** and the **IMS dataset** are explicitly designed for *prognostics*, capturing the entire life cycle of bearings until failure. PRONOSTIA accelerates degradation under tightly controlled conditions, offering reproducibility and well-structured run-to-failure sequences, whereas IMS emphasizes natural failure development under fixed loads and speeds, making it invaluable for Remaining Useful Life (RUL) estimation.

The **Paderborn dataset** expands the experimental landscape by incorporating both vibration and electrical current signals under a variety of operating conditions. This dual-sensing approach allows researchers to explore multi-sensor fusion techniques and evaluate models beyond vibration-only setups. Its inclusion of both artificially damaged and naturally failed bearings makes it a versatile dataset bridging classification and prognostics.

Finally, the **MFPT dataset** occupies a hybrid position: it provides both laboratory-controlled seeded fault data and complementary signals recorded in field-like conditions. Its diversity in loads, sampling rates, and machine setups makes it highly adaptable for testing generalization and robustness across heterogeneous scenarios.

In summary, no single dataset fully captures the complexity of real-world bearing degradation. Instead, these datasets complement one another: CWRU enables rapid prototyping, PRONOSTIA and IMS enable lifetime modeling, Paderborn supports multi-sensor fusion, and MFPT bridges laboratory and industrial contexts. Together, they form the empirical foundation on which modern ML and DL models for fault diagnosis and prognostics are built, as summarized in Table 2.6.

Table 2.6: Representative public datasets for BFD (literature summary).

Dataset	**Type / Setting**	**Sensors**	**Sampling (typ.)**	**Labels / Notes**
CWRU (Case Western) [40]	Fixed-condition classification	Accelerometers (DE/FE)	12 kHz and 48 kHz	Normal; single-point IR/OR/Ball at multiple loads/speeds.
FEMTO-ST (PRONOSTIA) [38]	Run-to-failure (accelerated life)	2 accelerometers (H/V)	25.6 kHz; short bursts at fixed intervals	RUL/prognostics focus; repeated short segments over lifetime.
IMS (NASA/UC) [41]	Run-to-failure (periodic snapshots)	Accelerometers (4 bearings)	20 kHz; ∼20 480 samples per snapshot, every ∼10 min	Degradation to failure under fixed speed/load; multi-file runs.
Paderborn Univ. (PU) [42]	Fixed-condition classification	Vibration + current; aux. torque/speed	Up to ∼64 kHz (high-res)	Multiple bearing types and severities; several operating conditions.
MFPT [43]	Mixed lab/field sets	Accelerometers (various rigs)	Tens of kHz (varies by set)	Normal, IR, OR at various loads; plus a few real-machine sets.

2.6 Conclusion

This chapter established the foundations of BFD by reviewing the signal processing steps, the role of condition-based maintenance (CBM), and the relevance of the P–F interval for maintenance decisions. Traditional vibration-based methods were examined alongside complementary sensing techniques, highlighting their contributions and limitations.

We also discussed the physical mechanisms of bearing faults and their vibration signatures, providing the link between defect progression and diagnostic signals. Finally, several benchmark datasets CWRU, PRONOSTIA, IMS, Paderborn, and MFPT were presented and compared, showing how each contributes unique strengths for algorithm development, benchmarking, and validation.

Overall, this chapter provides the technical and experimental background required to understand BFD and motivates the transition to data-driven approaches. The next chapter will build on this foundation by focusing on ML and DL methods for intelligent and scalable fault diagnosis.

Chapter 3

State of the Art in AI and WSN-Based Bearing Fault Diagnosis

Contents

III.1	Introduction	36
III.2	Literature review and key contributions in traditional BFD techniques	39
III.3	Overview of Machine Learning Approaches for BFD	40
III.4	Challenges of Deep Learning in BFD	42
III.5	Transfer Learning for Bearing Fault Classification	43
III.6	Wireless Sensor Networks (WSNs) and IoT-based Bearing Fault Monitoring	46
III.7	Comparison of State-of-the-Art Techniques	48
III.8	Conclusion	54

3.1 Introduction

Bearings are essential components in rotating machinery, and their failure can lead to significant downtime and economic loss. As a result, the development of intelligent, scalable, and accurate diagnostic systems for early fault detection has become increasingly important. This chapter presents a comprehensive review of recent research efforts focusing on the application of DL and TL techniques for BFD, with special emphasis on their integration within WSNs.

A key advancement in this domain is the convergence of AI, advanced signal processing, and WSN technology. WSNs enable continuous, real-time data acquisition from machinery, supporting the shift from traditional periodic inspections to proactive, predictive maintenance. These networks offer several industrial advantages, including scalability, flexibility, and remote monitoring capabilities, especially when embedded with lightweight AI models [44].

deep learning (DL) models particularly CNNs have shown excellent performance in fault classification tasks. Enhanced by TL, these models leverage pre-trained knowledge to improve generalization across different operating conditions, reducing the need for large labeled datasets and mitigating domain shift [45]. Furthermore, signal processing techniques such as Mel spectrograms and continuous wavelet transforms provide robust timefrequency representations that enable more effective and interpretable feature extraction for AI models.

The integration of AI with WSNs has transformed fault diagnosis systems into smart, real-time predictive maintenance solutions. Modern frameworks not only ensure early detection and classification of faults but also support scalable, cloud-enabled analytics for industrial automation. This convergence is especially vital for achieving the goals of Industry 5.0, which emphasizes intelligent automation and humanAI collaboration.

In summary, **modern BFD systems rely on a synergy of signal processing, AI-driven pattern recognition, and WSN-enabled data acquisition.** Traditional techniques have evolved into sophisticated, data-driven approaches that eliminate manual feature engineering and significantly improve diagnostic accuracy while maintaining efficiency under industrial constraints.

To build a foundation for this discussion, the next two subsections explore the technological cornerstones of modern BFD: first, the importance of WSNs in predictive maintenance, and second, the role of AI in enhancing CM systems.

3.1.1 The Importance of WSN in predictive maintenance

Wireless sensor network (WSN) have emerged as foundational enablers in modern predictive maintenance strategies, particularly within smart industrial systems. These networks enable real-time, decentralized data acquisition from critical machine components such as bearings without the limitations of traditional wired infrastructures. Their flexibility, cost-effectiveness, and scalability make them ideal for harsh and dynamic industrial environ-

ments.

Predictive maintenance, as opposed to scheduled preventive maintenance, relies on continuous monitoring and real-time analytics to anticipate equipment failures before they occur. The integration of WSNs within this paradigm allows for the timely detection of anomalies and degradation trends, reducing unplanned downtime and prolonging machine life [46].

According to the recent review by Kulkarni et al. [46], WSNs are particularly effective in resource-constrained environments due to their ability to support low-power communication protocols, edge processing, and energy-efficient sensing. This aligns with the needs of Industry 5.0, which emphasizes human-centric automation, sustainability, and intelligent decision-making.

The key advantages of WSN-based predictive maintenance include:

a. **Continuous Monitoring:** WSNs enable 24/7 surveillance of equipment health, which facilitates early detection of incipient faults and supports real-time maintenance decisions.

b. **Energy Efficiency:** Modern WSNs utilize energy harvesting and low-power communication protocols (e.g., Zigbee, LoRa, Bluetooth Low Energy), making them suitable for long-term deployment in industrial settings [46].

c. **Scalability and Flexibility:** The wireless nature of WSNs allows for seamless deployment and reconfiguration of sensor nodes across diverse equipment and locations.

d. **Remote Accessibility:** WSNs enable remote data collection and diagnostics via cloud or edge platforms, which is particularly beneficial for geographically dispersed manufacturing facilities.

e. **AI and Internet of things (IoT) Integration:** WSNs act as the backbone of Industrial IoT by providing the sensory input necessary for AI-based diagnostic models. This facilitates real-time analytics, anomaly detection, and predictive modeling.

In conclusion, the integration of WSN with predictive maintenance not only reduces costs and operational risks but also supports intelligent automation. WSNs are expected to play an even more pivotal role in Industry 5.0 environments, where synergy between edge intelligence, sustainability, and human-AI collaboration defines next-generation industrial systems.

3.1.2 Role of artificial intelligence in condition monitoring

Artificial intelligence (AI) has revolutionized the field of CM by enabling intelligent systems capable of learning, adapting, and autonomously detecting faults in industrial equipment. Traditional methods, reliant on static thresholds and manual feature engineering, often struggle with non-stationary signals, noise, and variations in operating conditions. AI-based

approaches, especially those grounded in ML and DL, address these challenges by extracting latent patterns and anomalies from vast sensor datasets, leading to robust and scalable diagnostic solutions [47].

According to Rao [47], AI's growing influence in CM and Diagnostic Engineering Management (COMADEM) stems from its integration of multiple paradigms such as knowledge-based systems, expert systems, neural networks, genetic algorithms, fuzzy logic, and hybrid models into real-world industrial asset monitoring. These technologies underpin the shift from reactive to proactive maintenance strategies by enabling intelligent decision-making, predictive analytics, and efficient fault diagnosis.

The incorporation of AI in CM offers the following major advantages [47]:

a. **Automated Feature Extraction:** DL models, such as CNN and recurrent neural networks (RNN), can autonomously extract informative features from raw vibration, temperature, and acoustic signals, bypassing the need for manual signal processing.

b. **Early Fault Detection and Anomaly Recognition:** AI systems can identify abnormal trends in sensor data by comparing real-time measurements to learned models of normal behavior, thus enabling timely alerts and interventions.

c. **Accurate Fault Classification:** Supervised ML techniques such as Support Vector Machines (SVMs), Decision Trees, and Artificial Neural Networks (ANNs) have demonstrated high performance in classifying different fault types in rolling element bearings and other machinery.

d. **Predictive Maintenance through RUL Estimation:** Predictive models estimate the RUL of components, enabling industries to plan maintenance actions based on actual equipment health rather than fixed schedules.

e. **Cloud and Edge AI Integration:** AI-enabled CM platforms are increasingly integrated with cloud and edge computing architectures, facilitating real-time diagnostics and reducing latency in decision-making processes.

f. **Hybrid and Adaptive Learning Models:** The emergence of hybrid models combining fuzzy logic, neuro-fuzzy systems, and genetic algorithms enhances adaptability and interpretability in fault diagnosis systems.

These AI techniques not only improve diagnostic accuracy but also support continuous learning and system evolution through data-driven insights. The integration of AI into WSN-based platforms further enhances monitoring capabilities, enabling scalable, low-power, and intelligent predictive maintenance systems for Industry 5.0. applications [48].

3.2 Literature review and key contributions in traditional BFD techniques

Traditional BFD has evolved significantly over the past decades, laying a critical foundation for modern predictive maintenance strategies. These approaches, primarily based on vibration analysis, have proven highly effective in detecting and characterizing faults in rotating machinery components particularly rolling bearings.

One of the most influential works in this field is the comprehensive study by Randall and Antoni [1], which systematically reviews vibration-based diagnosis methods. Their research explains how faults manifest through distinct signatures in time-domain, frequency-domain, and time-frequency representations of vibration signals. Their insights remain a cornerstone reference, widely cited in both academic literature and industrial practice.

Expanding on this foundation, Sawalhi and Randall [49] made a significant contribution by exploring the vibration behavior of bearings with spall defects. Through a combination of experimental testing, numerical simulations, and advanced signal processing, they demonstrated how to track the growth of spalls over time. Their work highlights the importance of analyzing the impact patterns caused by rolling elements entering and exiting a defect. Using techniques like spectral kurtosis and minimum entropy deconvolution, they enhanced the clarity of fault signals, allowing for more accurate and early fault detection. This study not only validated theoretical models but also brought diagnostic techniques closer to practical application in real-world industrial systems.

Another notable contribution is the work of Lebold et al. [50], which reviews common CM techniques and emphasizes the importance of feature selection and signal preprocessing for early-stage fault detection. Their work has influenced how engineers approach data acquisition and signal processing in industrial environments.

Tandon and Choudhury [51] also provide valuable insights into the development of signal analysis techniques, highlighting the role of envelope detection and cepstrum analysis in identifying bearing faults. Their work reinforces the necessity of selecting appropriate frequency bands when analyzing localized defects in bearing components.

In parallel with traditional signal processing advancements, more recent approaches have introduced auditory-inspired methods for analyzing vibration signals. The Mel-spectrogram, a time-frequency representation originally used in speech and audio processing, has shown promising results in BFD. By mimicking the human auditory perception of frequency, the Mel scale emphasizes informative frequency bands that are often critical in identifying early-stage faults. Studies such as those by Yue et al. [52] have demonstrated the effectiveness of Mel-spectrogram features. This marks a significant step forward in the evolution of feature extraction techniques for rotating machinery.This approach has shown promising results in handling complex operational conditions and improving diagnostic performance under varying speeds and noise environments [53, 54].

More recently, studies such as those by Rebiaï et al. [33] and Zhang et al. [55] have explored advanced signal decomposition methods, including Teager energy operators and empirical mode decomposition (EMD), which enhance the extraction of fault-related fea-

tures from noisy signals.

Together, these references chart the progression from manual inspection to sophisticated signal analysis methods, setting the stage for the integration of AI in modern fault diagnosis systems.

3.3 Overview of machine learning approaches for BFD

Machine learning (ML) has become a cornerstone of modern BFD, moving beyond rule-based and threshold-based diagnostics to data-driven models capable of learning complex fault patterns directly from sensor data [56, 57]. By leveraging large datasets, ML enables nonlinear modeling, integration of heterogeneous signals, and real-time adaptation to changing operating conditions, capabilities that traditional, manually engineered methods often lack.

In the supervised learning paradigm, models are trained on labeled datasets where each instance corresponds to a known fault type. This allows the model to learn decision boundaries that distinguish between different fault categories [58]. Commonly applied supervised algorithms include support vector machines, which are particularly effective in high-dimensional feature spaces; k-nearest neighbors, which classify new samples based on their proximity to previously labeled examples; and decision treebased ensembles such as random forests, which capture nonlinear relationships while remaining interpretable. These approaches have been successfully implemented for diagnosing defects such as inner-race, outer-race, and ball faults in rotating machinery [59, 60].

When labeled data is limited or unavailable, unsupervised learning provides powerful tools for exploratory fault analysis and anomaly detection [61]. Techniques such as k-means clustering partition data into groups with similar characteristics, density-based approaches such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN), discover clusters of varying shapes while isolating noise points, and hierarchical clustering reveals multi-level data structures useful for fault taxonomy and the identification of emerging fault modes [62, 63]. These approaches are particularly valuable in scenarios where fault conditions are unknown or evolving, supporting semi-supervised and self-learning pipelines for intelligent maintenance systems.

3.3.1 Literature review and key contributions

The literature on ML approaches to BFD (BFD) reflects a steady evolution from shallow, handcrafted methods to deep, end-to-end frameworks. Early studies largely focused on conventional ML techniques, where vibration, acoustic emission, or motor current signals were preprocessed and manually engineered into statistical features. These features, extracted in the time, frequency, or time–frequency domains, were then fed into classifiers such as support vector machines (SVM), *k*-nearest neighbours (kNN), or artificial neural networks (ANNs). While effective in controlled environments, such approaches were constrained by their reliance on domain expertise for feature extraction and their limited ability

to generalise under varying operational conditions [64].

To address these challenges, DL methods emerged as a transformative paradigm. Architectures such as autoencoders (AE), deep belief networks (DBN), and CNN demonstrated the ability to automatically learn hierarchical representations directly from raw signals, bypassing the need for handcrafted features [64, 65]. CNNs, in particular, became dominant for BFD by leveraging local connectivity and weight sharing to detect repetitive impulsive patterns and fault-related harmonics in vibration and spectrogram data. Recurrent neural networks (RNNs) and long short-term memory (LSTM) models further advanced the field by modelling temporal dependencies in sequential signals, making them suitable for fault progression and Remaining Useful Life (RUL) estimation.

Comprehensive reviews highlight several key contributions. First, the demonstration that DL models achieve superior diagnostic accuracy under noisy, nonstationary conditions compared to traditional ML baselines. Second, the exploration of multi-domain inputs (time, frequency, and time–frequency representations) and multi-sensor fusion (vibration with current or temperature) has broadened diagnostic capabilities. Third, benchmark studies across datasets such as CWRU, IMS, PRONOSTIA, Paderborn, and MFPT have established reproducible baselines and spurred the development of TL and domain adaptation strategies to enhance cross-condition generalisation [55, 64].

In summary, the literature has transitioned from rule-based and feature-engineered ML systems towards autonomous, data-driven DL models that capture complex nonlinear dynamics of bearing degradation. This shift not only improves detection and classification performance but also lays the groundwork for prognostic tasks, enabling predictive maintenance strategies with higher robustness and scalability across industrial applications.

3.3.2 Key feature extraction and selection strategies in condition monitoring

In traditional ML-based fault diagnosis, the accuracy and robustness of classification models largely depend on the quality of the features extracted from raw sensor data. Feature extraction serves as the bridge between low-level sensor signals and high-level pattern recognition, enabling algorithms to learn meaningful representations of machine health conditionsy [66]. Vibration signals, commonly used in bearing fault detection, exhibit nonstationary and nonlinear characteristics. To capture their essential properties, a range of signal processing techniques is employed across different domains:

- **Time-domain features:** Basic statistical descriptors such as , Mean, Standard Deviation, Skewness, Kurtosis, Crest Factor, and Peak-to-Peak value provide insights into the amplitude and energy distribution of the signal [67].
- **Frequency-domain features:** Extracted using tools like the FFT, these features include Spectral Centroid, Band Power, and dominant frequency components, which help identify periodic fault patterns [68].

- **Time-frequency domain features:** Techniques such as Wavelet Transform (WT) and Empirical Mode Decomposition (EMD) allow for the analysis of signal behavior over both time and frequency, making them suitable for transient and non-stationary signals [69].
- **Hilbert Transform-based features:** The Hilbert Transform (HT) is a powerful tool for deriving the analytic signal, from which instantaneous amplitude and frequency can be computed. These features are particularly effective in capturing impact characteristics and modulated signals often present in defective bearings [70].

By combining domain-specific signal processing methods with statistical and heuristic selection strategies, effective feature engineering can significantly enhance the performance of ML models in fault diagnosis. These handcrafted features serve as the foundation for accurate fault classification in traditional ML pipelines and are still relevant today, particularly in scenarios with limited data or computational constraints.

3.4 Challenges of deep learning in BFD

While DL has demonstrated impressive performance in BFD, several challenges limit its widespread industrial adoption. Based on a synthesis of recent literature [71, 72], the primary challenges can be summarized as follows:

- **Data Scarcity and Labeling Difficulty:** DL models require large amounts of labeled data to train effectively. In real-world industrial scenarios, fault data especially for rare or early-stage defects is often scarce and costly to annotate.
- **Overfitting and Generalization:** Many DL models tend to overfit to specific datasets, especially when data diversity is low. This limits their ability to generalize across different machines or operational conditions.
- **Lack of Interpretability:** DL architectures often operate as black boxes, making it difficult to interpret and explain the diagnostic decisions. This hinders trust and adoption in safety-critical systems.
- **Imbalanced Datasets:** Normal operating data typically outnumbers fault data, creating imbalanced datasets that bias the model toward healthy classes. Techniques like synthetic oversampling or cost-sensitive learning are required.
- **Computational Complexity:** Training deep architectures demands significant computational resources and energy, posing a barrier for deployment in edge or embedded systems.
- **Noise Sensitivity and Preprocessing Needs:** Raw vibration signals often contain noise or irrelevant information. While DL models aim to eliminate manual feature engineering, robust preprocessing is still crucial.

- **Limited Adaptability to Varying Conditions:** DL models trained under specific conditions (e.g., speed, load) may underperform when transferred to new settings. and domain adaptation are ongoing solutions but not yet universally reliable.

To overcome these challenges, future research should explore the development of more interpretable and data-efficient models. In particular, robust **transfer learning (TL)** strategies hold promise for improving adaptability across varying operational conditions and enabling effective fault diagnosis even with limited labeled data.

3.5 Transfer learning for bearing fault classification

deep learning (DL) models require large labeled datasets for effective training, which can be a challenge in real-world industrial applications where acquiring annotated fault data is costly and time-consuming. **TL** offers a promising solution by leveraging knowledge from pre-trained models to improve fault classification performance in new but related tasks [73].

3.5.1 Concept of transfer learning

The field of CM has significantly evolved with the advent of DL, enabling more accurate fault diagnostics and surpassing traditional methods [72, 74]. However, DL models require extensive labeled datasets and computational resources, which poses a major challenge in real-world fault detection scenarios, where acquiring large-scale annotated data is often impractical [75, 76]. To overcome these limitations, TL has emerged as a powerful technique that enhances model adaptability by leveraging knowledge from existing, well-trained models to improve performance on new, yet related, tasks.

TL operates by utilizing a model that has been pre-trained on a comprehensive source dataset and applying its learned features and representations to a target task with limited data. This process not only reduces the dependency on large labeled datasets but also accelerates training and enhances generalization capabilities in varying operational conditions(Figure 3.1).

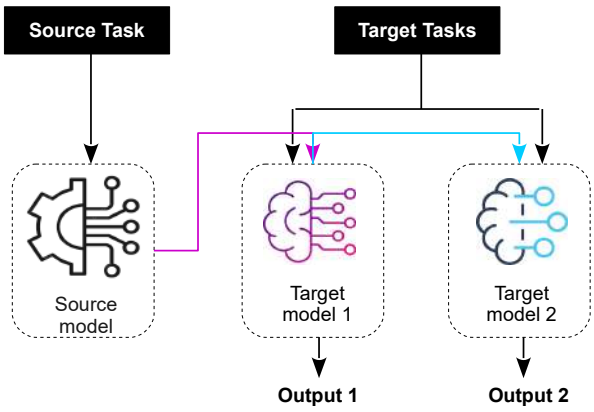


Figure 3.1: Illustration of the TL principle [77].

In the context of BFD, is particularly advantageous for handling domain shifts, adapting models to new operating environments, and improving fault classification accuracy. Broadly, TL can be categorized into three main strategies: **inductive TL**, **transductive TL**, and **domain adaptation**. Each of these approaches serves a distinct purpose depending on the availability of labeled data and the similarity between the source and target domains. A more in-depth discussion of these strategies is presented in the following subsections

By mitigating data scarcity issues and enhancing model robustness, plays a crucial role in improving the efficiency of DL-based fault diagnosis systems. Future research should continue to refine these techniques, focusing on domain-specific adaptations and hybrid approaches that integrate expert knowledge with data-driven learning.

In practice, a one-size-fits-all model may not account for the unique characteristics of each machine. TL enables the tailoring of generalized models to individual equipment by gradually fine-tuning them using domain-specific data. This ensures that fault detection systems remain adaptive, precise, and context-aware, without incurring the overhead of training separate models for each application.

3.5.2 Motivations for applying transfer learning in BFD

In the domain of BFD, TL emerges as a practical and strategic response to several critical limitations inherent in traditional ML approaches. As illustrated in Figureconventional ML typically relies on large volumes of labeled, identically distributed data and requires training models from scratch for each task. In contrast, TL offers the advantage of reusing knowledge from pre-trained models, enabling more efficient learning even in the presence of limited or non-independently and identically distributed (i.i.d.) data.

One of the most pressing challenges in industrial CM is the scarcity of annotated bearing fault data. Although vast amounts of multi source sensor data such as vibrations, acoustics, and thermal measurements are routinely collected, expert labeling of such data remains costly and time consuming. TL helps mitigate this bottleneck by transferring diagnostic knowledge from similar machinery or related fault scenarios, reducing annotation dependency while preserving classification performance across different bearings and operating environments.

Additionally, computational constraints present another barrier for many researchers and industries. Training deep models from scratch is resource intensive and often impractical without access to high end hardware. TL addresses this issue by allowing existing models trained on large datasets by well resourced institutions to be fine tuned for specific tasks, significantly lowering the cost and time required to deploy effective solutions.

transfer learning (TL) also plays a vital role in enhancing the generalization ability of diagnostic models. While ML aims to build systems that perform well on unseen data, the diversity of real world operating conditions often leads to distribution shifts between training and test environments. Bearing faults are rare, and replicating diverse failure scenarios is not always feasible. TL enables robust adaptation to new domains by leveraging prior knowledge, making it especially useful for fault detection under variable speeds, loads, or

noise levels.

Lastly, TL supports the personalization of monitoring systems. In practice, a one size fits all model may not account for the unique characteristics of each machine. TL enables the tailoring of generalized models to individual equipment by gradually fine tuning them using domain specific data. This ensures that fault detection systems remain adaptive, precise, and context aware, without incurring the overhead of training separate models for each application.

Figure 3.2 presents a summarizing the key motivations and advantages of applying TL in BFD, reinforcing the arguments discussed above.

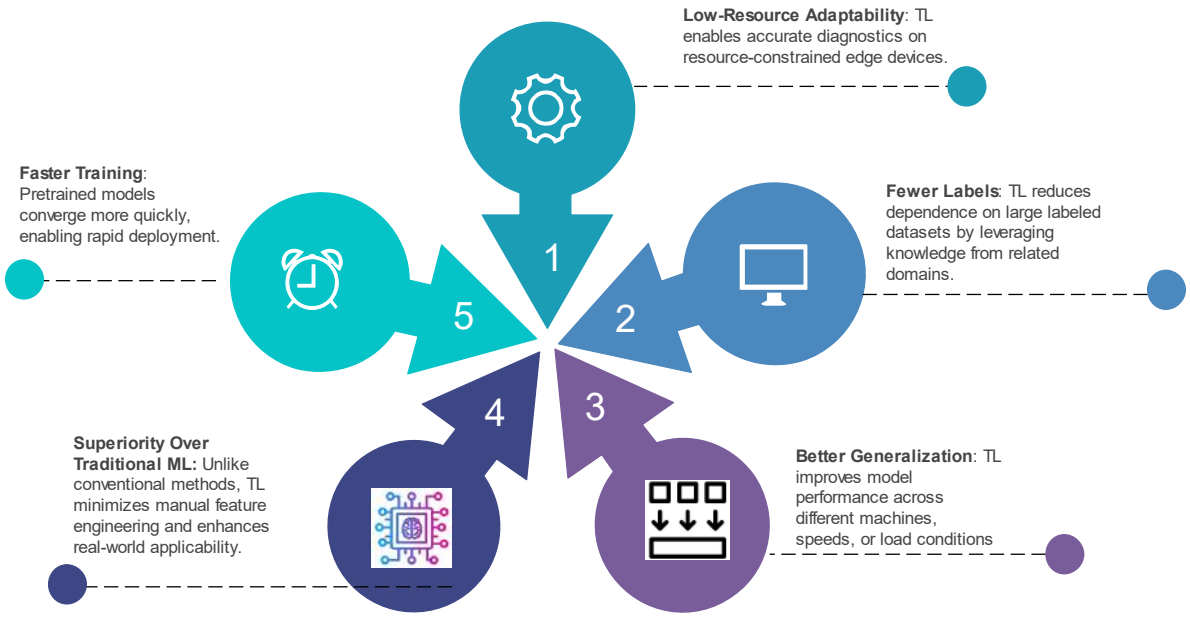


Figure 3.2: Key Differences and Motivations for Using TL in BFD

he following chapter will be dedicated to TL, where we will detail all aspects related to this field.

3.6 Wireless sensor networks and IoT-based bearing fault monitoring

In recent years, there has been a significant shift from wired to wireless communication systems in various industrial applications, including BFD. These advancements are central to the evolution of Industry 4.0 and the transition toward Industry 5.0. WSNs have redefined the architecture of modern ICS, particularly in CM applications.

WSNs are dynamic, cost-effective, and scalable, eliminating the need for complex wiring infrastructures. They improve safety and accessibility by enabling monitoring in hazardous or hard-to-reach environments, leading to safer and more intelligent operations [78]. Additionally, their modular and flexible nature allows easy adaptation to evolving plant layouts and machinery without major downtimes.

Figure 3.3 presents the primary roles and benefits of WSNs in industrial systems.

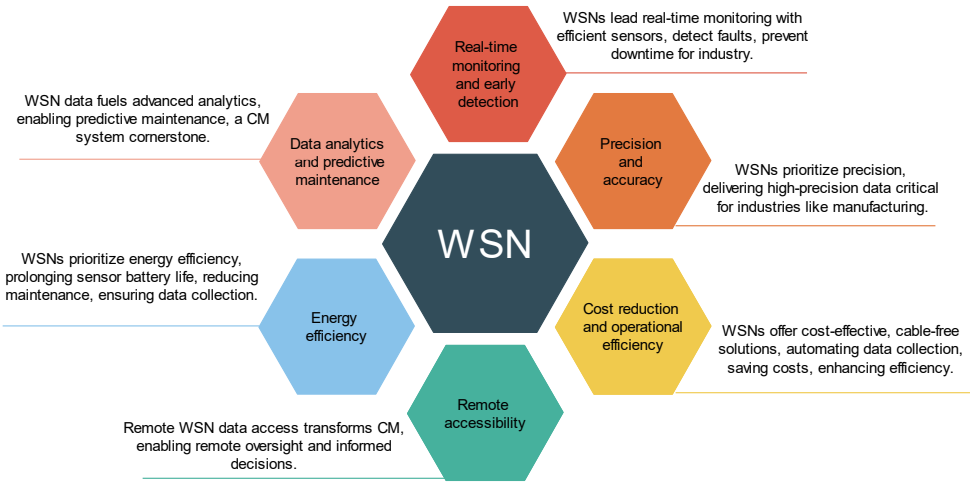


Figure 3.3: The benefits of WSN in ICS.

Key Functionalities of WSNs in BFD:

- **Real-Time Data Acquisition and Diagnostics:** WSNs continuously monitor vibration, temperature, and acoustic parameters of rotating machinery, facilitating early fault detection and real-time anomaly alerts.
- **Remote Monitoring and Decision Support:** Sensor nodes transmit data wirelessly to centralized units or cloud-based platforms, enabling remote access and AI-driven decision-making for predictive maintenance.
- **Energy Efficiency:** Modern WSN nodes are optimized for low-power operation, ensuring long battery life and minimal intervention essential for scalable and sustainable CM applications.
- **Flexible Integration with AI Models:** Data from WSNs can be seamlessly fed into ML and DL algorithms for automated classification of bearing faults, improving diagnosis accuracy.
- **Reduced Infrastructure Costs:** By replacing wired setups, WSNs reduce installation and maintenance expenses while increasing deployment flexibility in industrial environments.
- **Support for Intelligent and Resilient ICS:** Through real-time feedback, networked sensing, and adaptive control, WSNs enhance operational resilience, enabling proactive and autonomous fault handling.

These attributes make WSNs indispensable for smart predictive maintenance systems and next-generation industrial automation. Their integration with AI and IoT infrastructures drives the shift toward fully intelligent, adaptive, and self-healing manufacturing ecosystems.

3.7 Comparison of state-of-the-Art techniques

Recent advancements in AI, signal processing, and wireless technologies have enabled more accurate and efficient fault detection systems. In this context, various approaches have been developed and refined, ranging from traditional signal analysis to sophisticated DL frameworks. The following specific objectives guide the comparative analysis of these techniques across multiple dimensions, including architecture design, data efficiency, sensing modalities, and feature extraction methods, the specific objectives of these section are:

a. To evaluate current knowledge on DL architectures applied to BFD using time-frequency signal representations.

b. Benchmarking of existing TL techniques for improving diagnostic accuracy under limited and variable data conditions.

c. Ide ntification and synthesis of WSN applications facilitating real-time and distributed bearing fault monitoring.

d. To compare the effectiveness of sound recognition models, including YAMNet, against traditional vibration-based diagnostic methods.

e. To deconstruct feature extraction methods such as Mel spectrograms and their impact on DL model performance.

Recent literature reveals a shift toward architectures that combine CNN, timefrequency analysis techniques (e.g., Mel spectrograms, continuous wavelet transform), and domain adaptation strategies to enhance classification accuracy [79, 80]. Mel spectrograms and CWT images have emerged as powerful input representations that facilitate automatic feature extraction from vibration or acoustic signals [81, 82].

In resource-constrained WSN environments, embedding DL models on sensor nodes poses challenges due to limited memory, processing power, and energy availability. To overcome these obstacles, TL particularly using pre-trained architectures like YAMNet enables knowledge reuse across domains, reducing the need for large labeled datasets and mitigating domain shift [83, 84]. These strategies significantly improve generalization across machines and operating conditions.

Despite consistent improvements in diagnostic accuracy, challenges remain in practical deployment, such as computational overhead, data imbalance, and noise resilience. Moreover, while YAMNet has shown promise in audio classification, its application in vibration signal analysis remains underexplored, leaving room for further investigation [80, 83].

This part aims to highlight the synergy between TL, DL, WSNs, and timefrequency feature engineering. The convergence of these technologies paves the way for noise-resilient, lightweight, and generalizable fault diagnosis solutions tailored for Industry 5.0 environments.

3.7.1 Comprehensive background

Bearings are crucial components in industrial machinery, and their failure often accounts for 30--40% of unexpected equipment downtime [85,86]. Over the past decade, fault diagnosis has evolved from manual feature engineering to automated, data-driven approaches using DL and sensor fusion [79,80].

Recent research emphasizes the utility of CNN-based architectures, which operate on timefrequency inputs such as Mel spectrograms and CWT images [81,82]. These methods improve classification performance under complex operational conditions.

However, DL models often assume identical data distributions between training and testing phases a condition rarely met in real-world deployments. Variability in machine types, sensor positions, and environmental noise introduces domain shifts that limit model generalization [45,87].

To address these discrepancies, **transfer learning (TL)** has emerged as a powerful technique. TL leverages knowledge from related domains to improve learning in target tasks with limited data. Common TL approaches align source and target distributions using measures [83,88]. While effective, these techniques can incur high computational costs.

Pre-trained models like **YAMNet**, originally developed for audio classification, are particularly suited for interpreting timefrequency representations. When fine-tuned for vibration signals, they offer lightweight and effective solutions especially when deployed on embedded WSN platforms [89,90].

The integration of CNN, TL, and timefrequency features has proven successful in mitigating noise, addressing domain shift, and enhancing diagnostic performance [83,91]. These combined methodologies contribute to intelligent systems capable of real-time monitoring and predictive maintenance.

Drawing from this synthesized evidence, we contextualize our original research question *How can TL, DL, WSNs, and audio-inspired models such as YAMNet and Mel Spectrograms be leveraged to improve BFD?* within the broader scope of existing literature. These studies highlight the promise of pre-trained models, such as CNNs and residual networks, in enhancing fault classification accuracy, especially in data-limited environments. Moreover, the convergence of DL with adaptive signal representations (e.g., CWT, envelope spectrum, or Mel features) and TL fine-tuning has shown notable potential in improving generalization across machines, load conditions, and noise variations.

As most of the relevant studies are comprehensively summarized along with their core assumptions in Table 3.1, this overview provides a comparative lens into common TL scenarios and their practical implications in BFD. The table captures key performance indicators across various works, such as diagnostic accuracy, TL efficiency, feature representation strategies, and model robustness under noise or domain shifts.

Table 3.1: Comparative Analysis of TL-Based Approaches for BFD

Study	Diagnostic Accuracy	TL Efficiency	Feature Representation Quality	Model Robustness to Noise & Domain Shift
[80]	>99.7% accuracy with AlexNet TL	Avoids overfitting via TL	Spectrograms of vibration signals as input	Stable and reliable across independent datasets
[81]	∼99.3% accuracy using CWT images	CNN-based classification with denoising improves performance	CWT images outperform STFT in feature representation	Denoising methods improve noise robustness
[83]	High accuracy with few hundred fine-tuning samples on CWRU dataset	Effective reuse of sound recognition pre-trained models reduces data needs	Mel spectrograms and sound-based features enhance fault detection	Demonstrated robustness under limited data conditions
[85]	>99.5% accuracy after TL on multiple datasets	TL significantly improves accuracy over normal learning	2D images from vibration signals improve feature extraction	Noise reduction via signal decomposition enhances robustness
[92]	Improved accuracy on small-sample and noisy datasets	TL enables learning from small datasets	Attention mechanisms and soft thresholding improve feature quality	Noise resistance demonstrated with Gaussian noise simulation
[93]	Effective cross-machine diagnosis with multi-task learning	Self-supervised TL enhances cross-domain performance	Multi-scale masking improves intrinsic feature attention	Enhanced generalization across machines and conditions
[94]	Superior accuracy combining MTF and CWT with dual-channel CNN	Attention mechanism aids feature extraction efficiency	Multi-domain 2D images capture richer fault info	Attention mechanism improves noise and condition robustness
[95]	Higher accuracy and generalization with optimized auto-encoder	Particle swarm optimization tunes network for TL	Data-driven network structure search enhances features	Strong generalization in multi-fault and transfer scenarios
[90]	Classification accuracy 8898%; TL models outperform DL alone	TL models improve multi-class fault classification	Use of VGG16 and MobileNet for feature extraction	TL improves performance under limited data
[96]	High accuracy across speeds and bearing types	TL with envelope spectrum reduces domain gap	Envelope spectrum standardizes features across domains	Effective cross-domain fault diagnosis
[97]	Improved accuracy in noisy scenes without denoising modules	Lightweight residual network avoids batch norm issues	Rescaling weights stabilizes training	Robust to noise with fewer parameters and FLOPs
[98]	High accuracy with small datasets using wide-kernel CNN-LSTM	TL adapts model to small data and domain shifts	Wide kernel extracts low and middle frequency features	Domain adaptability improves
[99]	High accuracy and robustness on multiple datasets	Dynamic convolution and TL improve adaptability	Multi-scale feature extraction captures key vibration info	Robust under variable loads and speeds
[100]	Outstanding classification results on wind turbine bearings	Feature and network transfer combined	Pre-trained CNN adapted for BFD	Reduces training time and parameters
[101]	Better transfer effects and accuracy than SOTA methods	Multi-scale deep subdomain adaptation network	Selective kernel convolution for adaptive receptive fields	Eliminates disturbing information, improves robustness
[102]	Diagnostic accuracy below 80% with extra fault samples	Adaptive weighting in TL for unknown samples	CNN-LSTM extracts high-dimensional features	Addresses domain similarity and unknown fault states

3.7.2 Critical review and comparative analysis of diagnostic methods

In the last few years, many smart diagnostic methods have been created to improve fault detection in industrial settings, especially for rolling element bearings. TL, time-frequency feature extraction, WSNs , and sound-based models like YAMNet are just a few of the methods that have made big improvements in diagnostic accuracy, computational efficiency, and adaptability to real-world situations.

Table 3.2 shows a side-by-side comparison of these methods, pointing out both their pros and cons. For example, TL has done a great job of dealing with changes in domains and a lack of data by using pre-trained models and flexible frameworks. Time-frequency methods like Mel spectrograms, CWT, and STFT have also made it easier to turn 1D signals into more detailed 2D representations, which has helped deep models find more useful features.

Also, new architectures like capsule networks and attention-based residual networks have helped with problems like vanishing gradients and feature degradation. Sound-based methods, especially YAMNet, open up new possibilities by applying what we know about sound to vibration analysis. Using WSNs gives you the ability to be flexible and keep an eye on things in real time in complicated settings. Also, models that combine self-supervised learning and attention mechanisms are getting better at dealing with noise and small sample sizes.

But this review also shows that there are still problems. Even though they do well in controlled experiments, many models have trouble generalising when they have to deal with real-world industrial variability, noisy data, or hardware that doesn't have a lot of resources. There are still a lot of important problems that haven't been fully explored, like computational overhead, domain mismatch, energy efficiency in WSNs, and limited benchmarking across different operating conditions. Also, a lot of research doesn't look at the trade-offs between how fast an inference is, how complex a model is, and how scalable it is.

This synthesis calls for more research that goes beyond benchmark datasets and experimental validation. It is important to test these smart diagnostic systems in real-world situations, especially in industrial settings where resources are limited and conditions change quickly, to make sure they are reliable, strong, and can be used in a wide range of situations.

3.7.3 Critical analysis and synthesis

The reviewed literature on intelligent BFD spanning DL architectures, WSNs , TL, YAMNet, and timefrequency representations like Mel spectrograms demonstrates notable progress in enhancing diagnostic accuracy, noise robustness, and domain adaptability under varying industrial conditions. A key pattern emerging from the studies is the strategic use of TL, which effectively mitigates the challenges of data scarcity and domain variability by improving generalization across different machines and operating settings.

A comparative summary of the strengths and limitations of these key techniques including TL, timefrequency methods, sound-based models, WSNs, and model robustness strate-

gies is provided in Table 3.2. This table serves as a consolidated reference for understanding the practical trade-offs and research gaps associated with each method.

In summary, although current methodologies exhibit strong experimental performance, several unresolved issues persist. These include computational overhead, limited evaluation under realistic conditions, and insufficient consideration of real-time deployment constraints. Thus, future research should prioritize scalable and lightweight solutions, comprehensive robustness testing, and deployment feasibility within practical industrial environments.

Table 3.2: Strengths and Limitations of TL and TimeFrequency Feature Extraction Methods

Aspect	Strengths	Limitations
TL Integration	TL has proven highly effective in addressing data scarcity and domain shift challenges. Models such as enhanced AlexNet, capsule networks, and ResNet variants have achieved outstanding diagnostic accuracy (>99%) across multiple datasets [85, 103, 104]. Moreover, advanced frameworks using adaptive batch normalization and multi-task self-supervised learning improve generalization across machines and operating conditions [87, 93].	Despite its promise, TL often introduces computational overhead and requires meticulous tuning of hyperparameters and domain alignment strategies. Some methods may become unstable or less accurate when applied to highly noisy or variable conditions [88, 102]. The risk of negative transfer and domain mismatches also remains a challenge, particularly in real-world industrial settings [105].
Time Frequency Feature Extraction (Mel Spectrograms, CWT, STFT)	Timefrequency methods like Mel spectrograms, CWT, and STFT extract rich fault features from both time and frequency domains. Combined with CNNs and TL, they achieve high classification accuracy. Mel spectrograms often offer strong discriminative power, and all three methods effectively convert 1D vibration signals into informative 2D inputs for DL models [81, 83, 106].	However, these methods may demand substantial preprocessing and can be sensitive to noise or parameter selection. The computational cost of generating and processing such representations could hinder their use in real-time applications. Furthermore, while Mel spectrograms perform well, their superiority over other representations is not always conclusive, calling for additional benchmarking [81, 107].
Sound-Based Models (YAMNet)	YAMNet, initially trained on large-scale audio datasets, has been effectively repurposed for BFD. Its inherent strength in timefrequency analysis enables it to achieve high classification accuracy with minimal fine-tuning. This cross-domain application leverages prior knowledge from sound and music recognition tasks, potentially reducing the burden of extensive data collection [83].	Despite its promise, the use of YAMNet for bearing fault detection remains in early development. Its performance across different bearing types and operational settings is not yet fully understood. Challenges such as fine-tuning complexity and domain mismatch between audio and vibration signals may limit generalization. Comprehensive validation and benchmarking against conventional CNN models are necessary to verify its reliability and real-world applicability [83].

Continued on next page

Aspect	Strengths	Limitations
WSNs and Real-Time Monitoring	WSNs enable distributed, real-time data collection, which significantly enhances fault detection in dynamic industrial environments. Their integration with DL models supports scalable monitoring under varying operational conditions. WSN-based systems also show promise for noise tolerance and small-sample scenarios, aligning with real-world deployment needs [108].	Despite their potential, WSN-specific challenges such as limited energy capacity, signal synchronization, and data transmission reliability are rarely addressed in depth. The impact of WSN architecture on diagnostic latency and model accuracy remains underexplored. Additionally, few studies evaluate how WSNs perform when integrated with advanced DL models in real-time settings [109].
Model Robustness to Noise and Small Sample Sizes	Several advanced methods incorporate attention mechanisms, adaptive activations, and noise reduction strategies to strengthen model robustness against signal disturbances and small datasets. Techniques such as TL and self-supervised learning leverage auxiliary tasks and domain adaptation to maintain high diagnostic accuracy despite limited labeled data [91–93, 97, 110].	Nonetheless, many models are evaluated using artificially simulated noise, which may not reflect real-world industrial conditions. This raises concerns about their generalizability. Moreover, the scalability of noise-robust approaches across varying hardware configurations and fault types, as well as the computational overhead they introduce, require further scrutiny [92, 110].
DL Architectures and Computational Efficiency	Modern DL architectures such as capsule networks, residual networks with attention mechanisms, and dual-channel CNNs have shown superior capabilities in extracting and classifying fault features compared to earlier models. These advanced frameworks address common issues like vanishing gradients and loss of critical information during feature propagation [94, 97].	Many DL models rely on complex architectures that demand substantial computational resources, posing challenges for real-time fault diagnosis and deployment on low-power or embedded systems. Some studies do not thoroughly evaluate inference speed or resource utilization [94], and the balance between model complexity and practical efficiency is often overlooked [97].

3.8 Conclusion

This chapter surveyed key advances in intelligent BFD, spanning DL, TL, time–frequency representations, WSNs, and sound-based models (e.g., YAMNet). Across the literature, coupling TL with 2D time–frequency inputs (e.g., spectrogram-based features) consistently improves diagnostic accuracy and cross-condition generalization under limited labeled data.

Transfer learning (TL) emerges as a particularly effective strategy for addressing data scarcity and domain shift by reusing knowledge from related tasks and pre-trained backbones. In parallel, time–frequency mappings (e.g., Mel, CWT, STFT) provide informative inputs that enable DL models to capture transient, impulsive, and fault-specific characteristics.

WSNs facilitate distributed, real-time data acquisition and make these methods practical for predictive maintenance workflows; however, deployment in industrial settings still faces constraints related to computational overhead, energy and latency budgets, and robustness

under non-stationary operating conditions.

In summary, while results on common benchmarks are strong, future work should prioritize: (i) broader validation in realistic industrial environments, (ii) efficiency-aware designs suitable for edge/embedded deployment within WSNs, and (iii) systematic evaluation of generalization across machines, loads, and noise conditions.

Chapter 4

Transfer learning

Contents

IV.1	Introduction	56
IV.2	Factors Influencing Transfer Learning Performance	59
IV.3	Taxonomy of Transfer Learning	62
IV.4	Applications of Transfer Learning	65
IV.5	Machine Learning and Deep Learning: Core Concepts	69
IV.6	Understanding Transfer Learning	70
IV.7	An End-to-End Guide to Transfer Learning	78
IV.8	Conclusion	85

4.1 Introduction

The Algerian proverb, Love the new, but dont let go of the old, (حبّ الجديد، ولا تفرّط في القديم), conveys timeless wisdom that finds direct application in the field of machine learning, particularly in the concept of TL.

Instead of starting from scratch, one can use what a model has already learned from an earlier task **(the source domain or the old)** to help with a new but related task **(the target domain or the new)**. This saves time, requires less data, and reduces computing effort.

This strategy reflects the essence of the proverb: building upon past knowledge strengthens our ability to face new challenges. Ignoring or discarding earlier learning would not only be inefficient but also a loss of valuable experience. TL beautifully illustrates how **prior learning acts as a bridge to future learning**, underscoring the synergy between preservation and innovation.

However, while TL is powerful, it is not universally beneficial. Its effectiveness largely depends on the degree of similarity between the source and target tasks. If the underlying data structures, feature spaces, or objectives diverge significantly, naive transfer may degrade performance rather than enhance it.

This critical caveat is well illustrated by the fable of the crow who, fascinated by the graceful walk of the dove, attempted to imitate it. In doing so, the crow not only failed to adopt the dove's elegant gait but also forgot his own. Ultimately, he was unable to walk properly in either style. The Arabic proverb that encapsulates this lesson is: **"The crow forgot his own walk while attempting to learn the dove's"**

(الغراب جا يقلّد مشية الحمام انسا مشيتو)

In the context of TL, this metaphor illustrates the danger of blind transfer: a model adapted from a poorly matched source domain can lose its original performance while failing to benefit from the new domain. This scenario highlights the importance of assessing domain similarity and carefully adapting models during the transfer process to ensure robust and reliable outcomes.

The concept of **TL** was first formalized in Western psychology and pedagogy in the early 20th century, known as Learning Transfer [111]. However, its foundational principles can be traced much earlier in classical Arabic-Islamic intellectual tradition.

Notably, the renowned Arab historian and philosopher **Ibn Khaldun** (1332-1406) emphasized the progressive nature of learning and the significance of prior knowledge in acquiring new sciences. In his seminal work *Al-Muqaddimah*, he wrote:

Knowledge is acquired gradually; he who does not grasp the foundations cannot master the branches.

(العلم يُكتسب بالتدرّج؛ فمن لم يُحكِم الأصول لم يقدر على إتقان الفروع)

This perspective aligns closely with the modern idea of TL, where a model trained on a source domain serves as a strong starting point for learning a new, yet related, target domain.

Moreover, scholars such as **Al-Farabi** and **Ibn Sina** also discussed how existing knowl-

edge facilitates the understanding of new concepts, particularly through analogical and representational reasoning principles that are foundational in modern TL and ML research.

These insights demonstrate that the underlying idea of reusing and adapting learned knowledge is not exclusive to contemporary science but has been appreciated in classical epistemological thought for centuries.

This phenomenon occurs naturally in everyday life. For instance, if someone knows how to cook ***Rechta***, they will likely find it easier to learn how to prepare Couscous. Although the dishes are distinct, they share foundational culinary techniques such as steaming semolina, flavoring broth, and coordinating cooking stages. These shared techniques act as reusable skills, easing the learning of a new but related dish.

Similarly, a person who can drive a **manual transmission** car can more easily adapt to driving an **automatic vehicle or even a truck**because core driving competencies like steering control, road awareness, and spatial judgment are already developed, those exemple are illustrated in Figure 4.1.



a) b)

Figure 4.1: Real examples of TL. (a) Rechta, vs. Couscous. (b) manual transmission vs. automatic transmission.

These real-world examples illustrate how previously acquired knowledge and experience can accelerate learning in a new yet structurally related task. In the field of ML, this concept is referred to as TL. Rather than starting from scratch, TL enables a model trained on one task to be repurposed for a different, but related, task. When the source and target domains share enough structural similarity, this transfer enhances learning efficiency and effectiveness just as in human learning.

Humans possess an innate capacity to transfer knowledge across domains. From an early age, we naturally engage in analogical reasoning drawing parallels between familiar and unfamiliar situations. This ability to infer new understanding from prior experiences is foundational to learning. As Ibn Sn eloquently states in *Kitb al-Nafs* (Book of the Soul):

"The rational soul, once it perceives a certain form, is capable of deducing what resembles it through intellect and analogy."

This illustrates that knowledge transfer, or what we now call **TL**, is deeply rooted in human cognition and was long recognized by classical Islamic philosophers.

In recent years, artificial intelligence (AI) encompasses the broad domain of technologies designed to emulate human cognitive abilities. Within this domain, ML focuses on algorithms that learn from data without explicit programming. A more specialized sub-

set, DL, employs multi-layered neural networks to uncover complex patterns, particularly in high-dimensional data such as images or audio. TL, often integrated within ML or DL frameworks, enables models to leverage prior learning from one domain to accelerate or enhance performance on a different but related task. This hierarchical relationship is illustrated in Figure 4.2.

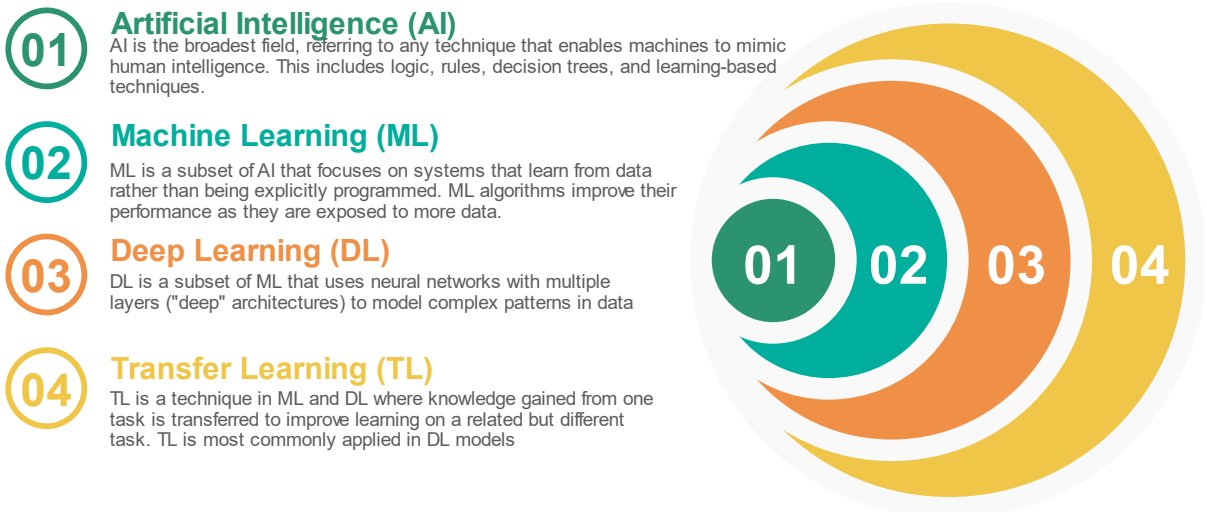


Figure 4.2: Hierarchical relationship between AI, ML, DL, and TL.

transfer learning (TL), also known as *knowledge transfer* or *learning to learn*, is a specialized area within ML. Its primary goal is to develop a predictive model for a target task by leveraging not only data from the target domain but also knowledge acquired from related source domains and tasks [112].

Unlike traditional ML, where each task is treated in isolation and learned independently from scratch (as illustrated in Figure 4.3), TL reuses prior knowledge to enhance the learning process for new tasks. This approach has proven effective in various applications, such as image recognition, natural language processing, and medical imaging, and others [113].

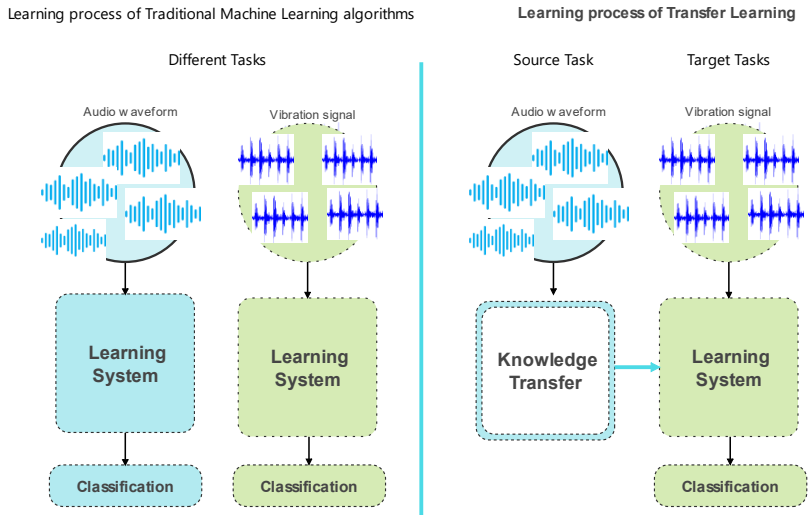


Figure 4.3: Comparison between traditional ML and TL approaches in fault classification tasks.

Fitting the definition, Fig. 4.4 illustrates a real-world challenge in applying TL to BFD. Bearings are essential components across a wide range of critical application fields such as aerospace, automotive, and energy and come in various types (e.g., ball, roller, thrust), each with different mechanical properties and operational requirements.

In practice, sensor signals (vibration, acoustic, or current) collected for fault diagnosis vary significantly depending on the machine's application domain, the type of bearing used, and environmental conditions such as load and speed. This diversity introduces difficulties when designing a universal diagnostic model that generalizes well across all scenarios.

This raises a crucial question: *How can we harness shared patterns across machines and bearing types to develop models that generalize effectively particularly when labeled data from new scenarios is limited?*

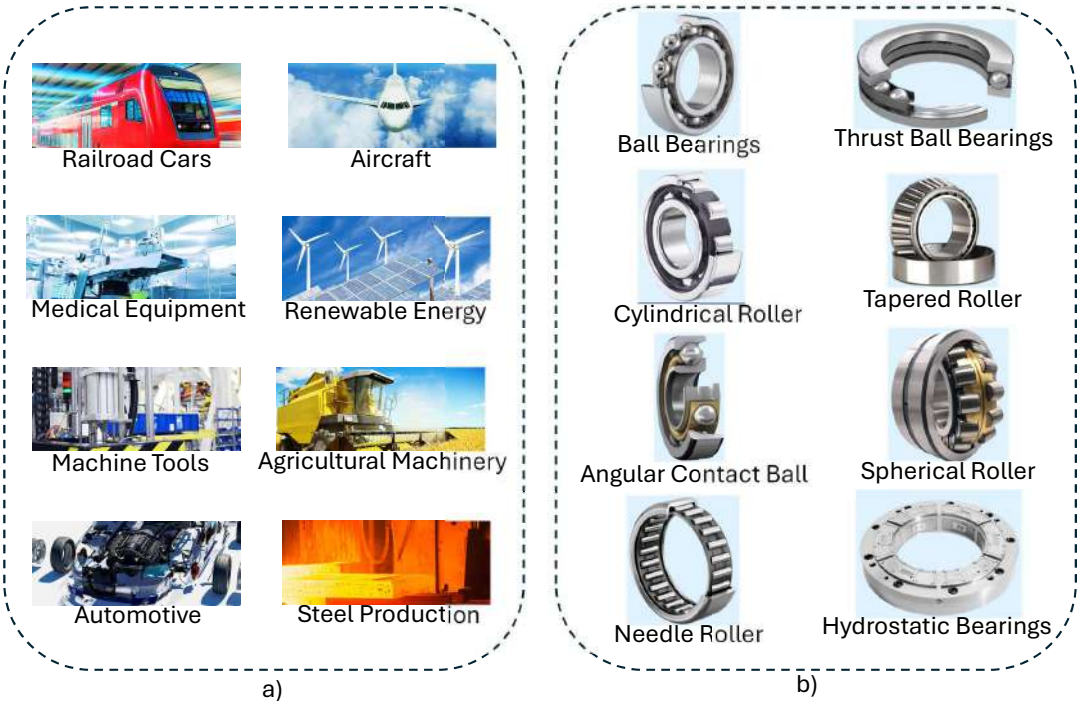


Figure 4.4: Variability of sensor signals across different application domains and bearing types, posing challenges for developing generalized fault diagnosis models.(a) Critical application fields utilizing bearings. (b) Common bearing types used across industries

Figure 4.5 illustrates a typical TL scenario: models pre-trained on machines A and B (source domains) under specific conditions can be adapted to build a model for a new machine C (target domain). Instead of training from scratch, the model for machine C benefits from shared representations or parameters, enabling faster convergence and better fault detection per- formance even with limited data. Such strategies are not only applicable across machines, but also across different bearing types, operating environments, or sensor configurations supporting scalable and more generalizable industrial fault diagnosis systems

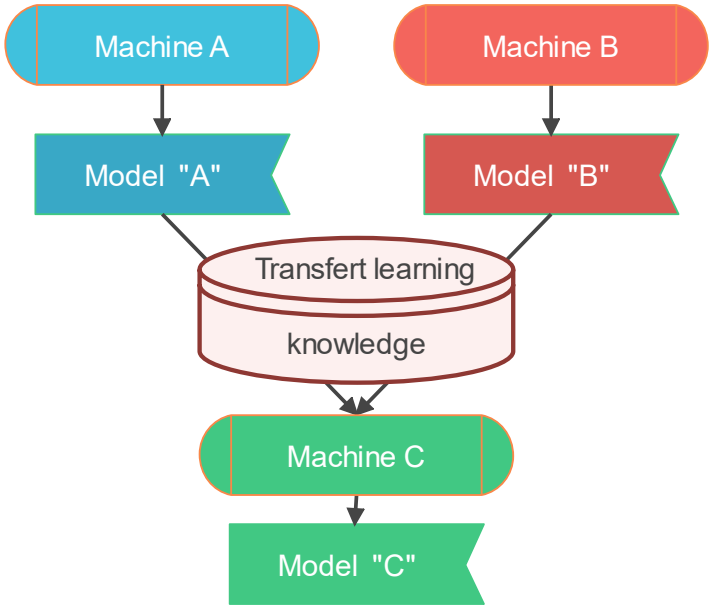


Figure 4.5: Illustration of the TL process for BFD: knowledge learned from machines A and B is transferred to machine C, reducing the need to retrain from scratch.

4.2 Factors influencing transfer learning performance

Beyond the choice between traditional ML and TL, several practical aspects significantly influence how well a model performs and generalizes, particularly in the context of bearing fault classification [114].

1) TL for Resource Efficiency and Data Diversity

TL addresses two major challenges in BFD: limited computational resources and imbalanced data distribution. Traditional model training requires significant hardware and time, which are often unavailable in small labs or industrial settings. TL enables the reuse of pre-trained models, reducing both cost and training time.

Moreover, when training data does not represent the diversity of real-world conditions, models struggle to generalize. TL introduces robust patterns learned from broader datasets, improving model performance on unseen scenarios. This makes it a practical solution for developing scalable and reliable fault detection systems.

2) TL for Generalization and Annotation Efficiency

A core goal of ML is to build models that generalize well to unseen data. However, real-world datasets especially in fields like BFD are often limited in size and diversity due to the rarity of actual failures and the high cost of data collection.

transfer learning (TL) addresses this limitation by leveraging knowledge from pre-trained models built on richer datasets. These models can be adapted to new domains, enhancing generalization even with limited data. Techniques such as domain adaptation and meta-learning further support this adaptability.

Another bottleneck in supervised learning is data annotation. Accurate labeling is labor-intensive and requires domain expertise. TL alleviates this burden by utilizing pre-annotated

datasets from related domains, significantly accelerating model development for new tasks.

3) TL for Personalized Diagnostics and Efficient Model Training

In CM, general models are typically built to perform across various machines and environments. However, individual machines often have unique operational characteristics such as different loads, speeds, or wear patterns that reduce the effectiveness of a one-size-fits-all model.

transfer learning (TL) offers a flexible solution by allowing general models to be fine-tuned for specific machines or conditions. This facilitates the development of personalized and adaptive diagnostic systems that evolve with continuous data acquisition.

Additionally, model training is accelerated through the use of pre-trained parameters. This approach not only improves convergence speed but also enhances accuracy, even when only a small dataset is available a major advantage in industrial environments.

4) Hyperparameter Tuning in TL

The performance of a ML model depends not only on the data and architecture but also on the selection of hyperparameters, including learning rate, batch size, number of epochs, and regularization strength. These factors affect both training efficiency and generalization capability.

In TL workflows, fine-tuning these hyperparameters becomes especially critical. Since the model starts from pre-trained weights, inadequate tuning can undermine performance or prevent effective adaptation to the target task. Proper optimization ensures that the model fully capitalizes on the transferred knowledge and adapts efficiently to new domains.

4.3 Taxonomy of transfer learning

Understanding the taxonomy of TL is essential for selecting the right method based on the problem at hand. This taxonomy classifies TL techniques according to four key dimensions: the label setting between source and target domains, the similarity of their feature spaces, the adopted learning strategy, and what type of knowledge is transferred.

Figure 4.6 visually summarizes this classification. As shown, TL can be divided into inductive, transductive, or unsupervised learning depending on the availability and type of labels. Based on feature space similarity, transfer can be homogeneous (same feature space) or heterogeneous (different feature space). Furthermore, strategies can be offline or online, and the transferred knowledge may involve instances, features, parameters, or relational structures such as those used in adversarial learning [114, 115].

A special category within this framework is deep TL, where neural network models benefit from pretraining on large datasets. Common techniques include freezing early layers, fine-tuning deeper layers, or employing progressive neural networks to transfer knowledge across tasks.

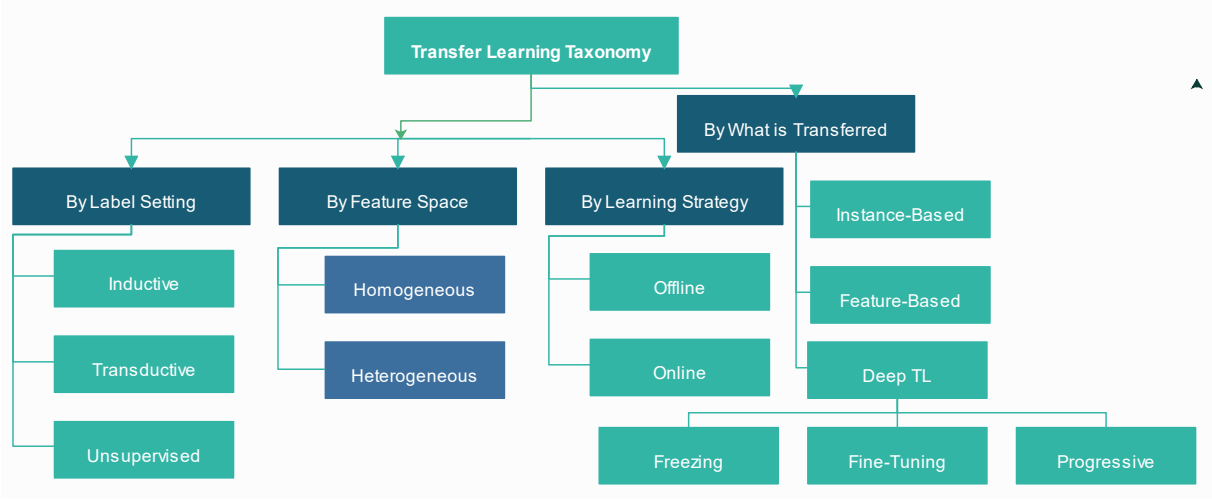


Figure 4.6: Taxonomy of TL Approaches.

4.3.1 Based on label availability

transfer learning (TL) can be categorized into three main settings depending on the availability of labeled data and the relationships between tasks and domains. Table 4.1 provides a comparative summary of these categories [116]:

- **Inductive TL:** The source and target tasks are different. Labeled data is available in the target domain, while the source domain may or may not have labeled data. This setting is applicable when the objective is to learn a new task leveraging knowledge from previously learned tasks, even across similar or different domains.
- **Transductive TL:** The tasks remain the same across source and target, but the domains differ. Only the source domain contains labeled data. This is typical in scenarios like domain adaptation, where the aim is to generalize across different but related data distributions.
- **Unsupervised TL:** Neither source nor target domains have labeled data. The tasks can be either the same or different, and so can the domains. This approach is commonly used in unsupervised settings such as clustering, anomaly detection, or representation learning, where structural similarities are transferred without supervision.

Table 4.1: Comparison of Different TL Settings

Setting	Source Labeled	Target Labeled	Task Relation	Domain Relation
Inductive	Optional	Yes	Different	Same / Different
Transductive	Yes	No	Same	Different
Unsupervised	No	No	Same / Different	Same / Different

4.3.2 Based on feature space similarity

Another widely adopted taxonomy in TL research is based on the similarity of feature spaces between the source and target domains. As introduced by Weiss et al. (2016), this classification distinguishes two major categories [116]:

- **Homogeneous TL:** This occurs when the source and target domains share the same feature space meaning that the data representations (e.g., dimensions and semantics) are consistent. A typical example would be transferring knowledge between two image datasets with similar formats or resolution.
- "Heterogeneous TL:" This setting, on the other hand, is applicable when feature spaces vary greatly between domains. This could entail a variety of data modalities, like converting audio to sensor signals or images to text. This kind of TL frequently necessitates extra steps like feature alignment or transformation because the source and target data have different structures.

4.3.3 Based on learning strategy

transfer learning (TL) can also be categorized based on how data is acquired and processed over time:

- **Offline TL:** Both domains are available before training begins. This is the most common setup.
- **Online TL:** The model updates continuously as new data from the target domain arrives. Suitable for real-time systems.

4.3.4 Based on what is transferred

TL approaches can also be categorized according to the type of knowledge that is transferred from the source to the target domain. This taxonomy reflects the mechanism of transfer, whether it involves reusing data instances, transforming feature spaces, sharing model parameters, or learning domain-invariant structures.

1)Instance-Based Approaches These reuse selected instances from the source domain, usually applying a weighting strategy to match the distribution of the target domain.

2)Feature-Based (Mapping-Based) Approaches These transform the data to a common feature space to reduce divergence between domains.

- **Symmetric Mapping:** Both domains are transformed jointly into a shared latent space.
- **Asymmetric Mapping:** The source domain is transformed to align with the target domain or vice versa.

3)Parameter-Based (Model-Based) Approaches These are widely used inDTL . A pre-trained model from the source domain is partially reused in the target task.

- **Freezing:** Certain layers from the pre-trained model are kept unchanged.
- **Fine-Tuning:** Pre-trained layers are updated slightly using target domain data.
- **Progressive Neural Networks (PNNs):** A frozen model is extended with new layers trained specifically for the target task to prevent catastrophic forgetting.

4)Relational or Adversarial Approaches These methods rely on logical relations or use adversarial learning (e.g., GANs) to extract domain-invariant representations. Though less common, they are promising in complex, structured domains.

4.3.5 Special case:DTL

Deep Transfer Learning (DTL) represents a specialized form of TL in which **deep neural networks** are leveraged to transfer knowledge across tasks. Among the different approaches, **parameter-based methods are the most prominent**.

In typical deep networks, the **early layers** learn to extract **generic low-level features** such as edges, textures, or simple shapes. In contrast, the **deeper layers** capture more task-specific, **high-level abstractions** useful for classification or prediction.

A common and effective DTL strategy is to **freeze the early layers** (to retain the general knowledge learned from the source domain) and **fine-tune the later layers** on the target domain, allowing the model to adapt to the new task with minimal data.

More advanced architectures, such as **Progressive Neural Networks (PNNs)**, take this further by **freezing the entire source model** and **adding new trainable layers** that connect to it. This enables **multi-task learning and adaptation** while preserving the original knowledge and avoiding catastrophic forgetting.

4.4 Applications of transfer learning

transfer learning (TL) has become a cornerstone technique in modern ML, allowing models to leverage knowledge acquired from a source domain or task to improve learning efficiency and performance in a target domain or task. This paradigm is particularly advantageous when labeled data in the target domain is scarce, costly to obtain, or difficult to annotate. By reusing learned representations whether low level features or high-level abstractions TL reduces computational cost, accelerates model development, and often enhances generalization capabilities. Its versatility has led to widespread adoption across various research and industrial sectors, including computer vision, natural language processing, speech analysis, healthcare, and manufacturing [117].

4.4.1 Common application domains

transfer learning (TL) is not confined to a specific discipline; instead, it can be applied wherever knowledge transfer between tasks or domains is feasible. The most frequently cited application domains include:

- **Computer Vision:** Image classification, object detection, semantic segmentation, facial recognition, and medical image analysis often benefit from models pre-trained on large-scale datasets such as ImageNet or COCO.
- **Natural Language Processing (NLP):** Tasks such as sentiment analysis, machine translation, text summarization, and question answering leverage large pre-trained language models (e.g., BERT, GPT) fine-tuned for domain-specific objectives.
- **Speech and Audio Processing:** Speech recognition, speaker identification, and emotion detection from audio signals use models pre-trained on extensive speech corpora and adapted to target accents, languages, or contexts.
- **Healthcare and Bioinformatics:** Applications include disease diagnosis, biomedical signal analysis (EEG, EMG, ECG), drug discovery, and patient outcome prediction, where pre-trained models are adapted to specialized medical datasets.
- **Human Activity Recognition:** TL is applied to wearable sensor data, gesture recognition, and indoor localization, enabling efficient recognition of new activities without extensive re-collection of training data.
- **Autonomous Driving:** Pre-trained perception models are adapted for specific driving environments, weather conditions, or sensor configurations, enhancing safety and adaptability in self-driving systems.
- **CM:** TL improves the performance of diagnostic systems by transferring models trained on known equipment (e.g., motors, bearings, gearboxes) to new machines or operating conditions with minimal additional data.
- **Cybersecurity:** TL is applied to intrusion detection, malware classification, and phishing detection by adapting models trained on known attack patterns to identify emerging and zero-day threats

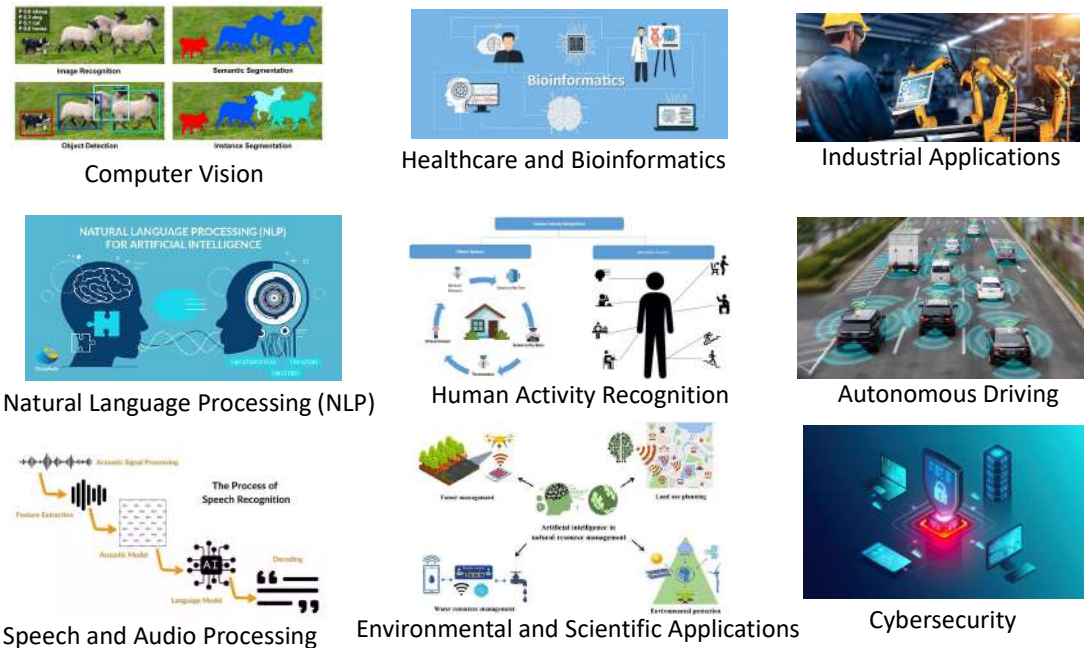


Figure 4.7: Varied Practical Uses of TL Across Domains.

4.4.2 Transfer learning in condition monitoring applications

In the context of Industry 4.0, CM is a key industrial domain where TL has become an enabler of intelligent diagnostics and predictive maintenance. CM aims to detect, localize, and classify machine faults using signals collected from sensors such as vibration, current, temperature, or sound. However, acquiring labeled fault data for every machine or working condition is impractical. TL addresses this challenge by transferring knowledge from well-instrumented machines or operational states to new, unseen ones with limited annotations. This reduces training effort and improves generalization.

Representative TL-enabled applications in CM include:

- **Bearing Fault Detection:** Vibration-based deep models trained on one bearing type (e.g., in controlled lab environments like CWRU) are adapted to other bearing types, load conditions, or machines using domain adaptation techniques or fine-tuning. This allows reliable fault classification with minimal target data.
- **Motor Health Monitoring:** TL allows knowledge transfer from one motor to another (e.g., different speed/load levels or motor sizes) using current signals or multi-modal sensor data, enabling accurate diagnosis of winding or rotor faults under varying conditions.
- **Gearbox Fault Diagnosis:** TL is used to adapt gear fault classifiers trained on one gearbox system to others with different numbers of teeth, gear ratios, or speeds. This supports early fault detection without requiring retraining from scratch.
- **Tool CM in Machining:** TL supports the reuse of trained models across machines or tool types, enhancing tool wear detection and life prediction while minimizing the need for retraining with large annotated datasets.

- **Cross-Domain Equipment Monitoring:** TL enables knowledge reuse across different machines (e.g., pumps, compressors, turbines) using sensor fusion and feature transfer, supporting scalable deployment of health monitoring systems across entire industrial fleets.

By leveraging pre-trained models and domain adaptation techniques, TL enhances fault diagnosis accuracy, reduces data labeling efforts, and shortens deployment time making it a powerful asset in real world CM systems.



Figure 4.8: TL applications in CM

4.4.3 Overview of TL applications in sound and bearing analysis

Generally speaking, is an important branch of ML and is not restricted to specific applications. In fact, it can be employed in any situation that satisfies its problem definition. In our project, we leverage this versatility by adapting a sound dataset to the context of BFD. The aim is to transfer knowledge learned from sound analysis to improve the detection and classification of bearing faults. This approach allows us to benefit from the rich features of sound data to enhance the accuracy and robustness of bearing CM systems.

Speech is a natural and fundamental form of human communication. In technology, its studied in various ways, such as Automatic Speech Recognition (ASR) which converts spoken words into text and Text-to-Speech (TTS) which does the reverse, generating human like voices from written text. Beyond these, research also explores areas like speaker verification and voice conversion.

Speech is a multidisciplinary field, bringing together computer science, AI, signal processing, linguistics, and statistics. Over the years, thanks to DL, both ASR and TTS have made significant progress, using neural networks to achieve impressive results. However, training these models requires large datasets, and collecting high quality speech data is far from easy. Unlike images or text, speech data is deeply personal and highly variable. Each person has a unique voice, tone, and accent, which makes it challenging to build robust models.

This is where becomes essential. It helps overcome data scarcity and variability by allowing models trained on large, diverse datasets to adapt to new tasks with limited data. In TTS, for example, models learn to generate natural sounding voices from minimal samples, making it a classic few shot learning problem. Evaluating speech models is also tricky, as what "sounds good" can be subjective compared to clear cut metrics like accuracy.

Research has explored many creative ways to apply to speech tasks. Techniques include adapting models across languages (like transferring knowledge from Mandarin to

Cantonese), building robust systems for different speakers or groups (such as children or people with speech disorders), and using domain adaptation to improve recognition in varied environments and accents.

In TTS, TL helps models quickly adapt to new voices, even with very little data. Some approaches use speaker embeddings, adversarial networks, or disentangled representations to capture individual vocal characteristics without needing paired training data.

Accents also present a major challenge, even within the same language. English, for instance, varies greatly across regions. Recent advances focus on cross accent and cross language adaptation, using methods like character level distribution matching or adapter based frameworks to bridge the gaps between different speech patterns.

Overall, TL continues to play a crucial role in making speech technologies more flexible and accessible, helping systems handle diversity in voices, languages, and accents and we expect exciting developments to continue in this field.

4.5 Machine learning and deep learning: Core concepts

Machine Learning (ML) can be defined as the study of algorithms that improve their performance P on a task T through experience E [118]. Typical components are:

- **Task (T):** the problem to solve (e.g., classification, regression).
- **Experience (E):** data from which the system learns.
- **Performance (P):** a metric such as accuracy or mean squared error.

4.5.1 From machine learning to deep learning

Deep Learning (DL) is a branch of ML based on artificial neural networks with many layers. DL models learn hierarchical representations of raw data, reducing the need for manual feature engineering and enabling tasks such as image or signal classification. Time–frequency features (e.g., spectrograms, CWT) often serve as inputs, allowing networks to detect transient or fault specific patterns.

4.5.2 Hyperparameters

Model parameters (weights, biases) are learned from data, while **hyperparameters** (learning rate, batch size, network depth, dropout rate) are set before training and govern how the model learns. Proper tuning of hyperparameters is essential for convergence, generalization, and efficiency.

4.5.3 Examples

- **Regression:** Predicting daily energy use from past measurements.

- **Classification:** Identifying defective products from images using a CNN.

These notions provide the foundation for understanding **transfer learning**, where knowledge from a model trained on one domain is adapted to new but related tasks.

4.6 Understanding transfer learning

In **traditional ML**, models are typically trained from scratch for each specific task. The learning process relies entirely on the available training data, and the model adjusts its parameters solely based on those examples. While this approach can be effective, it often requires a large volume of labeled data and significant computational resources. Moreover, since the model has no prior knowledge, its performance heavily depends on the quality and quantity of the dataset provided for that task alone [65].

Transfer learning (TL), on the other hand, offers a more efficient alternative by reusing knowledge acquired from one or more source tasks to improve learning in a new, but related, target task. Instead of starting from zero, the model leverages previously learned features, patterns, and representations especially from domains with abundant labeled data to perform well in situations where labeled data is scarce. This approach is particularly useful when data collection is costly or time-consuming, as it enables faster convergence and often leads to better generalization in the target domain [117].

4.6.1 Definition of transfer learning

In the context of ML, a **domain** refers to the environment or setting in which learning occurs. A domain is defined by three elements:

- **Data:** the actual examples or observations available for learning.
- **Data distribution:** the way in which this data is generated or spread.
- **Feature and label spaces:** the type of input features (e.g., images, text) and output labels (e.g., categories or numerical values).

In *TL*, there are at least two domains:

- **Source domain:** the domain that contains abundant knowledge (data and labels) from which we transfer information.
- **Target domain:** the domain where we aim to improve performance, often with limited data.

The source and target domains may differ in several ways:

a. **Different feature spaces:** The inputs have different types or dimensions. Example: the source domain contains RGB images, while the target domain contains grayscale images.

b. **Different label spaces:** The tasks or output categories are different. Example: the source domain classifies cats and dogs, while the target domain classifies flowers.

c. **Different data distributions:** Even with the same features and labels, the way the data is distributed varies. Example: the source domain involves object recognition from sketches, while the target domain involves paintings of the same objects.

The objective of TL is to **leverage knowledge from the source domain to improve the learning process in the target domain**, especially when the target domain has limited annotated data.

4.6.2 Domain adaptation

A particularly important subfield of TL is **domain adaptation**. This occurs when:

- The source and target domains share the same feature space and label space.
- Their data distributions differ due to variations in style, acquisition conditions, or context.

The goal of domain adaptation is to use labeled data from the source domain to develop a predictive model that performs well in the target domain despite these distributional differences.

Domain adaptation can be categorized into three main scenarios:

a. **Supervised Domain Adaptation (SDA):** All target domain data is labeled.

b. **Semi-supervised Domain Adaptation (SSDA):** Only part of the target domain data is labeled.

c. **Unsupervised Domain Adaptation (UDA):** No labeled data is available in the target domain (this is the most challenging case).

Unsupervised domain adaptation has attracted significant attention due to its practical importance in situations where labeling data in the target domain is costly or impractical. Many algorithms designed for UDA can be adapted to semi-supervised and supervised settings, making domain adaptation a versatile and impactful area of research in TL.

4.6.3 Fundamental questions in transfer learning

In practice, applying TL revolves around three tightly connected questions that form a simple decision cycle [114, 116].

The first is "**when to transfer**". TL is most beneficial when the target task has limited labeled data but bears meaningful similarity to a richer source domain such as comparable sensors, signal characteristics, or task objectives. If the domains are too dissimilar, the risk

of *negative transfer* arises, where imported knowledge degrades performance rather than improves it. A pragmatic diagnostic is to compare a small from-scratch baseline with a fine-tuned model; if transfer does not outperform the baseline, one should reconsider the suitability of the source domain, the representation used, or the adaptation strategy.

The second question is "**what and where to transfer**". Here the aim is to identify the kind of knowledge that is both reusable and beneficial for the target. Depending on the problem, this knowledge may concern data instances (e.g., selectively reusing or reweighting source samples), learned representations (e.g., shifting both domains toward a shared feature space), model parameters (e.g., reusing early layers and fine-tuning task-specific layers or extending the network with new branches), or relational structure (e.g., constraints or domain-invariant cues learned through adversarial objectives). Where denotes both the choice of source domain(s) to draw from and the specific components of the model that are most appropriate to carry over, such as low-level feature extractors versus task heads.

The third question is "**how to transfer**". The adaptation mechanism should match the size and quality of the target data and the magnitude of the shift between domains. When the distributions are close, lightweight parameter reuse with limited fine-tuning can suffice. With larger shifts, feature alignment or mapping to a shared latent space can reduce divergence before task learning. In cases with complex or structured differences, relational or adversarial approaches can help induce domain-invariant representations. For low-data targets, freezing early layers and carefully fine-tuning later ones often stabilizes training; for broader or multi-task targets, progressive extensions can preserve source competence while acquiring new skills.

Viewed as a lifecycle, these questions guide an iterative process: assess feasibility and risks (*when*); determine the reusable knowledge and its place in data or model (*what/where*); then apply an adaptation procedure suited to the domain gap and data regime (*how*). If signs of negative transfer emerge such as unstable fine-tuning, overfitting, or failure to beat a from-scratch baseline the cycle should be revisited by selecting a closer source domain, refining the transferred components, constraining the extent of fine-tuning, or switching to a more appropriate adaptation strategy. This disciplined progression helps ensure that TL improves generalization instead of compromising it.

1) When to transfer: sound-Based pretraining *to* BFD

In BFD with TL, a common scenario uses a *sound-based, pretrained audio model* as the source and a vibration-/acoustic-based bearing dataset as the target. The central question is *when* such transfer will help rather than harm (i.e., avoid negative transfer). In practice, transfer is most promising when (i) source features are sufficiently generic, (ii) the sourcetarget gap can be bridged in the timefrequency domain, and (iii) a modest amount of target data can reliably calibrate the reused representations.

a) Bridging the modality gap. Although accelerometer signals and airborne acoustics differ, both can be expressed as *timefrequency* patterns. A practical bridge is to compute a time-frequency representation (e.g., spectrograms, logMel spectrograms, or wavelet scalograms) from the target signals and match basic preprocessing to that typically used for the audio

source (amplitude normalization, window/hop settings, and comparable patch durations). This alignment allows early filters learned on diverse sounds (edges, chirps, tonal/noise textures) to become useful for characteristic bearing signatures (e.g., modulated impulses, harmonics, and sidebands).

b) Similarity checks before transfer. Transfer is safer when there is reasonable overlap across: operating conditions (speed, load, duty cycle), sensing setup (sensor type, mounting, bandwidth, SNR), signal chain (sampling rate, window length/overlap, chosen time-frequency representation), bearing geometry (type, pitch diameter, number/size of rolling elements, cage), and the *fault label space* (inner race, outer race, rolling element, cage; severity/incipience). If airborne acoustics are available, the domain gap to generic audio pretraining is typically smaller; with accelerometers only, the timefrequency bridge above is especially important.

c) Low-risk protocol before full fine-tuning. Establish a from-scratch baseline on the target. Then freeze the source encoder and train a shallow *linear probe* on target features using the same data splits. If frozen features consistently *match or exceed* the baseline across multiple operating points (different speeds/loads) and under reduced SNR, proceed to cautious fine-tuning (unfreeze the top block only, small learning rate, few epochs). If gains are limited, add light domain alignment (e.g., discrepancy minimization in the embedding space) or a small adapter in front of the encoder, and retest.

d)Recognizing and mitigating negative transfer. Warning signs include unstable training, degradation on held-out operating conditions, or new confusions between fault classes that were not present in the baseline. Mitigations include freezing more layers, reducing the tuned depth and learning rate, adding simple alignment losses or feature normalization across domains, reweighting target instances, or selecting a closer source (e.g., audio from similar machinery) when available.

e)A practical decision rule. Adopt transfer when a frozen-encoder probe plus shallow fine-tuning *consistently* improves accuracy/F1 over a from-scratch model *across* a condition matrix (speed×load), remains robust at lower SNR, and reduces labeled-data or training-time requirements. For cross-dataset evaluation (e.g., training on one bearing dataset and testing on another), require that gains persist after harmonizing the timefrequency front end and basic normalization.

2) Where to transfer: Source and sample selection for sound-based bearing diagnosis

In our context, *where to transfer* answers two tightly coupled questions. At the domain level, it concerns which source machines or datasets should supply prior knowledge to a target bearing application. At the data level, it concerns which specific recordings or segments from those sources (and from the target) are most informative for adaptation. Because transfer success depends not only on accuracy but also on industrial constraints such as falsealarm rate, detection latency, computational budget at the edge, and robustness to acoustic noise, the most appropriate source or sample is the one that improves the full monitoring objective rather than a single metric.

a) Selecting source domains. For BFD from sound, suitable sources are those that share comparable timefrequency structure with the target: similar shaft speeds and load ranges, comparable bearing geometry (pitch diameter, number and size of rolling elements, cage type), analogous sensing setup and placement (structure-borne accelerometers on the bearing housing), matching sampling rates and bandwidth, and broadly aligned label spaces (inner/outer race, rolling element, cage, severity). In practice, we quantify this proximity by comparing timefrequency embeddings between domains (e.g., cosine distance on logMel or scalogram features), by measuring distribution discrepancy , or by training a lightweight domain classifier and using its error as a similarity score. Order tracking (speed normalization) and basic equalization reduce superficial mismatches, making similarity estimates more meaningful. When multiple candidate sources exist (e.g., motors, pumps, fans), a multi-source strategy that weights each source by its similarity to the target is preferable to an arbitrary single choice.

b) Coupling with the adaptation method. Where to transfer and how to transfer are not independent decisions. Choosing a closer source reduces the burden on the adaptation mechanism; adding simple alignment tools (front adapters, per-domain normalization, or discrepancy penalties in the embedding space) can, in turn, make a slightly more distant source viable. Likewise, if adapters or residual bottlenecks are inserted around the top blocks of the frozen encoder, we can favor sources whose higher-level patterns (modulated impulses, harmonics, sidebands) resemble the target, even if the raw acoustics differ. In practice, we first shortlist sources by the similarity criteria above, test them with a frozen-encoder probe on standardized timefrequency inputs, and retain the sources that yield the best probe performance and stability under noise and condition changes. Those same sources then supply the samples used for fine-tuning or for training lightweight adapters.

c) Operational perspective. Ultimately, the best where choice is validated on the complete condition matrix (speed×load) that the target machine experiences. A source/domain and sampling policy is considered appropriate when it consistently improves detection and reduces false alarms across that matrix, preserves performance under lower SNR, and respects deployment limits on memory and compute for online CM.

3) How to transfer: from sound-based pretraining to BFD

In BFD, how to transfer concerns the concrete protocol that moves knowl- edge from, a *source model* that is a sound-based encoder pretrained on large, diverse audio corpora (generic environmental and industrial sounds). *How to transfer* therefore refers to the concrete protocol that adapts these learned timefrequency representations to a *target* bearing dataset measured via accelerometers or airborne microphones under varying speeds, loads, sensors, and environments.

a) Build a common front end. Harmonize sampling rates, segment lengths, and normalization between source and target. For vibration signals, perform standard condition-monitoring pre-processing: band-pass in a resonance band, envelope detection (Hilbert or demodulation), and timefrequency transforms such as STFT, logMel spectrograms, or wavelet scalograms. When a tachometer is available, order tracking or resampling reduces

speed-induced drift so that source and target spectra become more comparable.

b) Start with a frozen encoder, then unfreeze gradually. Adopt a pretrained encoder (e.g., a CNN trained on vibration/acoustic data from similar machinery). First, *freeze* the encoder and train a simple classifier head (linear probe) on target segments to verify that transferred features are already meaningful. If improvements over a from-scratch baseline persist across operating points, proceed with *gradual unfreezing*: open only the top block, use a small learning rate and few epochs, and expand depth if gains continue.

c) Align domains explicitly. When the domain gap remains, add explicit alignment objectives on the embedding space: distribution matching (e.g., MMD or CORAL) or adversarial alignment (gradient-reversal/domain discriminator). Class-conditional alignment with confident target pseudo-labels further reduces confusion among inner-race, outer-race, rolling-element, and cage faults.

d) Reweight source instances and curate target windows. Source windows that resemble the target (assessed by an A-distance proxy or a shallow domain classifier) receive larger weights during training. On the target, select quasi-stationary, high-SNR windows whose envelope spectra show energy near characteristic defect orders; down-weight segments dominated by unrelated phenomena (gear mesh, severe imbalance) or sensor clipping.

e) Use robustness-oriented augmentation. Narrow the gap with realistic perturbations: additive broadband/narrowband noise, amplitude scaling, speed warping to emulate nearby RPMs, small band shifts, and mixup within class. These augmentations help the transferred model remain stable under changing operating conditions and SNRs.

f) Evaluate for transfer, not just training fit. Report accuracy/F1 across a condition matrix, cross-dataset tests when available, and robustness under SNR reductions. Watch for signs of negative transfer (new confusions not present in the baseline, instability when unfreezing) and react by freezing more layers, shrinking learning rates, or strengthening alignment/adapters.

g) Note on research density. While how to transfer methods receive the bulk of attention, BFD projects benefit equally from rigorous *when* and *where/what* analyses: verifying sourcetarget relatedness (bearing geometry, sensor bandwidth/mounting, sampling chain) and selecting both the most compatible sources and the most informative windows before fine-tuning. This front-end diligence typically yields larger gains than ever-more complex adaptation losses.

4.6.4 Diagnosing bearing faults through negative transfer

Despite its strength, TL is not always advantageous. The degree of similarity between the *source* and *target* problems at the level of signal statistics, feature spaces, label spaces, and operational objectives determines its success. Naïve transfer can actually degrade performance rather than improve it when these aspects diverge too much. As already mentioned, the fable of the crow who tried to imitate the dove and, in doing so, forgot its own walk aptly illustrates this risk: when a model from a poorly matched source is modified, its original

strengths may be lost without any corresponding gain in the new domain.

A practical definition of *negative transfer* is when a transfer procedure yields lower performance in the target domain than a similar model trained from scratch on the same target data under the same evaluation protocol.

a) Causes in bearing fault diagnosis. Negative transfer in BFD usually arises when the source and target differ markedly in one or more of the following: (i) sensing modality or setup (e.g., far-field audio versus structure-borne vibration with different bandwidth and SNR); (ii) operating regime (speed, load, duty cycle); (iii) label space (fault types or severity levels); or (iv) preprocessing pipelines (sampling rate, window/hop, or timefrequency representation). For instance, the impulsive, cyclostationary signatures of bearing defects may not align with features learned from generic environmental sounds; in such situations, direct fine-tuning can compromise both accuracy and robustness.

b) A metaphor for transitive transfer. We use the example of a man crossing a wall in Fig. 4.9 to illustrate the concept of *transitive transfer learning*. In a normal situation, when the wall is not very high, the man can easily climb over to the other side. However, in more difficult cases, when the wall is too tall to clear directly, he can employ ladders or stepping stones as intermediate supports to reach the top and descend safely.

This process is analogous to transfer learning: the two sides of the wall correspond to the source and target domains. Successfully climbing over represents an effective transfer. A low wall indicates a small gap between the domains, while a high wall suggests a large domain difference. The ladders or supports represent intermediate domains that can enable successful transfer learning even when the gap is large.

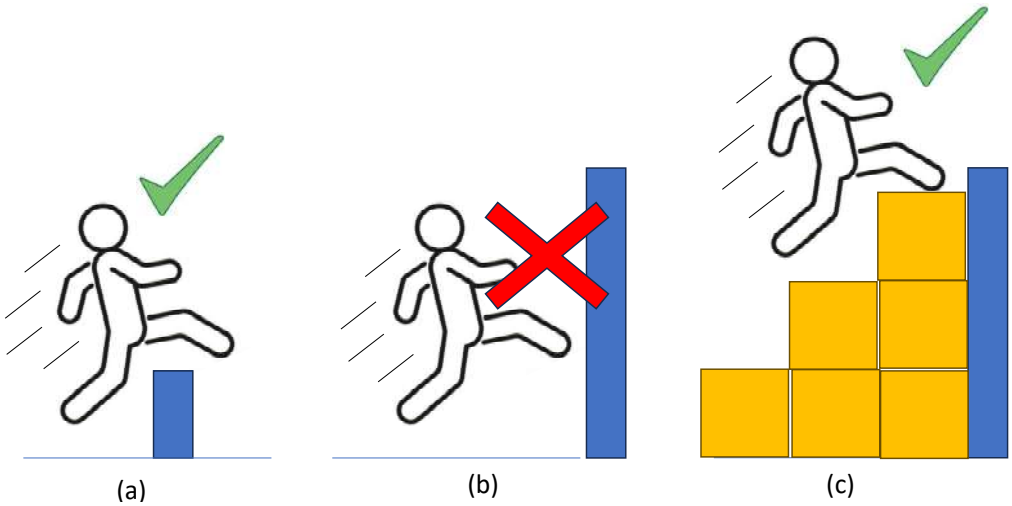


Figure 4.9: Illustration of Negative TL using a wall-crossing metaphor. (a) Normal case: small domain gap, the man easily climbs over the wall. (b) Abnormal case: large domain gap, the man fails to cross—negative transfer. (c) Abnormal case: large domain gap, the man succeeds by using intermediate supports.

c) Practical guidance. To prevent negative transfer, establish a baseline by training a model from scratch before fully fine-tuning. Freeze the source encoder and train a linear probe on the target data. Proceed only if the frozen features match or surpass the baseline under all operating conditions. If not, choose a closer source domain (e.g., audio from similar rotat-

ing machinery), add simple domain-alignment losses or normalisation, insert a lightweight adapter, or reduce the tuning depth and learning rate. Reliability can be further improved by selecting target segments with consistent RPM and good SNR, and by emphasising regions whose envelope spectra exhibit peaks near distinctive defect orders.

4.7 An end-to-end guide to transfer learning

To fully grasp the practical implementation of TL, it is essential to explore the key stages involved in its end-to-end process. This section outlines a step-by-step approach, beginning with the use of pre-trained models as feature extractors. By understanding how existing DL models can be leveraged and adapted for new tasks, we can better appreciate the flexibility and efficiency that TL offers. The subsequent subsections delve into specific strategies, including feature extraction, fine-tuning, and model selection, to guide the design of effective TL pipelines across a variety of domains.

4.7.1 Transfer with pre-trained models: Selective fine-tuning versus feature extraction

The principle. Deep networks learn a feature hierarchy: early layers capture generic patterns (edges, textures), mid-layers encode motifs and shapes, and deeper layers specialise for the source task. This hierarchy is reused in TL, which only modifies what is required for the intended task.

Two operating modes are commonly used [119]:

1) Feature Extraction (Freeze): Use the pre-trained model as a fixed feature extractor:

a. Load the backbone and **replace** the output layer to match the new classes.

b. **Freeze** earlier layers to retain general knowledge.

c. **Train** only the newly added classification head.

Effective when data is scarce or overfitting is a concern.

2) Selective Fine-Tuning (Unfreeze Later Blocks): Adapt later layers for better task-specific alignment:

a. Unfreeze deeper layers while keeping the rest frozen.

b. Use **layer-wise learning rates**: higher for new layers, lower for frozen ones.

c. Optionally, progressively unfreeze more blocks as training proceeds.

This mode is better suited when a significant domain shift exists or sufficient target data is available.

4.7.2 Freezing or fine-tuning?

In the context ofDTL, one of the most critical design choices concerns whether to *freeze* the layers of a pre-trained model or to *fine-tune* them. In the *freezing* strategy, the convolutional or feature extraction layers of the pre-trained network are retained with their original weights and are not updated during training on the target task. The model, in this configuration, acts solely as a fixed feature extractor, where the learned representations often capturing generic low-level patterns such as edges, textures, and shapes are passed to newly added layers, typically a task-specific classifier. This approach is computationally efficient, reduces the risk of overfitting when the target dataset is small, and is particularly effective when the source and target domains share similar characteristics [120].

In contrast, *fine-tuning* involves unfreezing part or all of the pre-trained network and allowing its weights to be updated during training on the new task. This selective or full retraining process enables the model to adapt its intermediate and high-level representations to domain-specific patterns that differ from the source domain. Fine-tuning generally yields superior performance when a sufficiently large and representative target dataset is available, or when the domain shift between the source and target tasks is significant. However, it increases computational cost and introduces a higher risk of overfitting, especially when the target dataset is limited.

The choice between freezing and fine-tuning is therefore a trade-off between efficiency and adaptability. When computational resources are limited or the available data is scarce, freezing the pre-trained layers and training only the final task-specific layers is often preferable. Conversely, when greater accuracy is required and the dataset is adequately large, fine-tuning can leverage the full capacity of the pre-trained architecture to capture both generic and domain-specific features.

When to choose each: *Use feature extraction* for low-resource tasks or modest domain changes, whereas *use fine-tuning* when deeper adaptation is required and adequate data is available. In both cases, early stopping, strong regularisation, and data augmentation are recommended, along with careful monitoring of validation metrics. Layers should not be unfrozen all at once to avoid destabilizing training.

As shown in Figure 4.10, the process can follow two paths: feature extraction or fine-tuning. The leftmost panel represents the original model trained on a general-purpose dataset. The middle panel demonstrates how this model is adapted to a new task by freezing the backbone and retraining only the classification head. Finally, the rightmost panel depicts further adaptation through fine-tuning, where deeper layers are retrained to better suit the target domain [121].

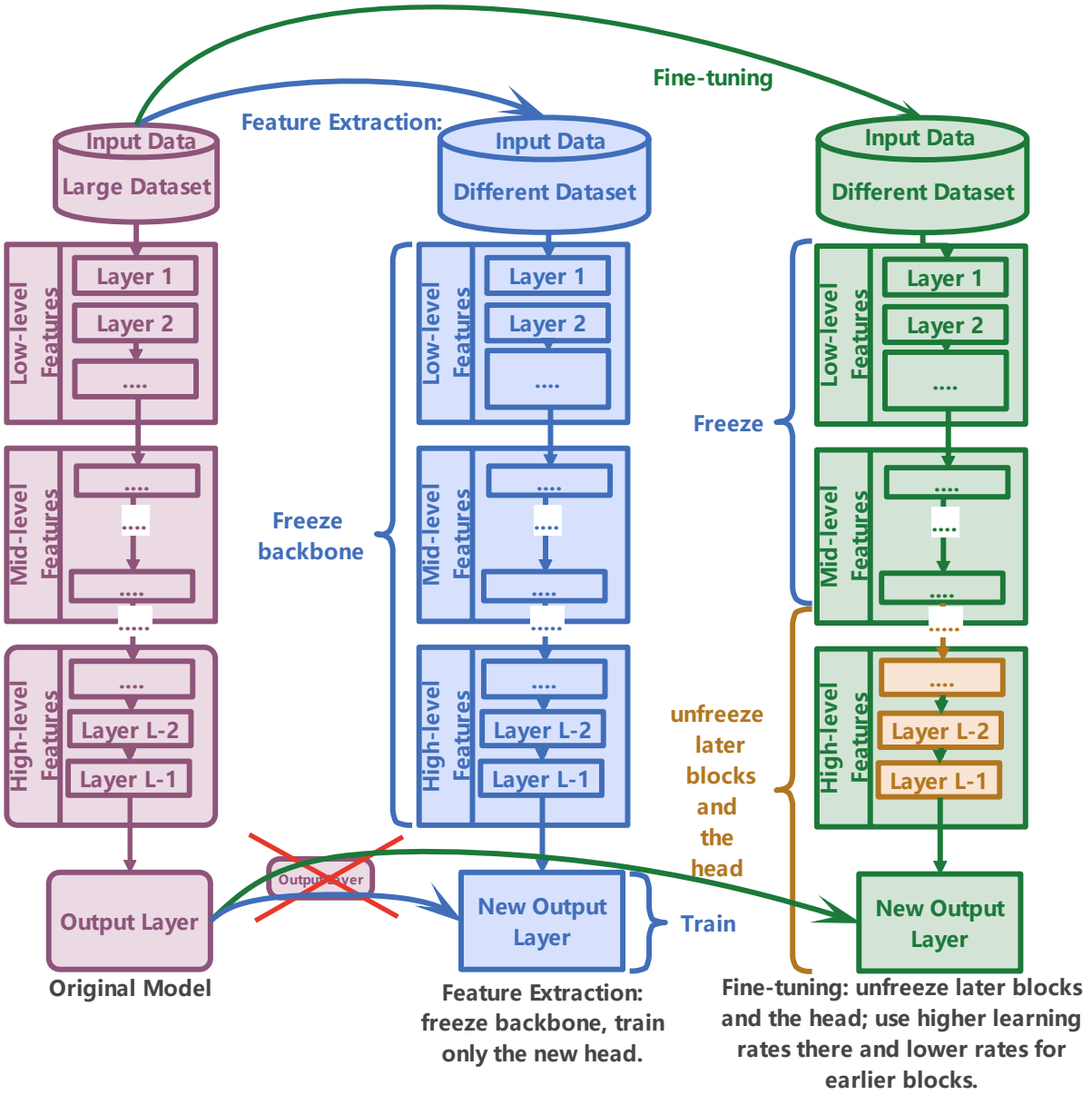


Figure 4.10: Comparison between feature extraction and selective fine-tuning in TL. The left illustrates the pre-trained base model. The middle shows a target model using feature extraction (frozen backbone), while the right depicts fine-tuning (partially unfrozen layers).

4.7.3 Popular transfer learning models across domains

transfer learning (TL) leverages rich, pre-trained representations and adapts them to new tasks with limited data and compute. Because pretraining corpora and model inductive biases differ by modality, it is useful to organize popular models by *domain*. Table 4.2 offers a concise overview; the choices below have been widely adopted as feature extractors (freeze) or as starting points for selective fine-tuning with lightweight adapters or Low-Rank Adaptation (LoRA).

Table 4.2: Representative pre-trained models by domain and typical transfer uses.

Domain	Model family	Type	Typical transfer use
Vision	ResNet-50/101; DenseNet; Inception-V3	CNN	ImageNet backbones; robust features; good for spectrograms.
Vision	EfficientNet (B0–B7); V2	CNN	Accuracycompute trade-off; suitable for edge devices.
Vision	ViT-B/16; DeiT; Swin	Transformer	Large-scale pretraining; strong on spectrograms.
Vision	CLIP (ViT)	Contrastive dual encoder	Zero-/few-shot features; frozen or fine-tuned.
NLP	BERT; RoBERTa; DeBERTa; ALBERT; DistilBERT	Encoder	Classification/NER; adapters/LoRA common.
NLP	T5	Encoderdecoder	Seq2seq (summarization, QA).
NLP	LLaMA; Mistral; Falcon	Decoder-only	Generation; instruction tuning; PEFT.
Audio	wav2vec 2.0; HuBERT; WavLM	SSL speech encoders	ASR; speaker/emotion tasks.
Audio	Whisper	Encdec Transformer	Multilingual ASR; robust embeddings.
Audio	YAMNet	MobileNet-V1 (CNN)	AudioSet-pretrained; compact features for acoustics/vibration.
Audio	PANNs (CNN14)	CNN	Audio embeddings; tagging baseline.
Audio	AST; BYOL-A; BEATs	Transformer/SSL	Audio tagging; acoustic scenes.
Time-series	InceptionTime	1D CNN	Strong supervised baseline for sensors.
Time-series	ROCKET; miniROCKET	Rand. conv feats + linear	Extremely fast; competitive baseline.
Time-series	PatchTST; TimesNet; TST	Transformer-based	Long sequences; multivariate sensors.
Time-series	TS2Vec; TNC; CPC	Self-supervised encoders	Pretrain on unlabeled sensors; linear head.
Self-sup vision	SimCLR; MoCo v2/v3; BYOL; DINO	Contrastive/distill	Frozen features; general visual encoders.
Self-sup vision	MAE	Reconstruction	ViT pretraining; adapters + small heads.

1) Practical selection for vibration/EMG/BFD.

For BFD and related EMG or vibration tasks, two routes are common: (i) convert signals to time–frequency representations (STFT/CWT spectrograms) and transfer from vision/audio models (*e.g.*, ResNet-50, ViT-B/16, **YAMNet**, PANNs); or (ii) stay in the time domain and transfer from sensor-oriented backbones (InceptionTime, ROCKET/miniROCKET, TS2Vec). In low-label regimes, prefer frozen encoders with lightweight adapters (1×1 bottlenecks, FiLM) and explicit domain alignment; with more labels, progressively unfreeze later blocks using discriminative, layer-wise learning rates.

2) Parameter-efficient adaptation. Across domains, modern practice favors small adapter layers or LoRA over full-network fine-tuning to preserve source knowledge, reduce compute, and limit negative transfer. This is especially beneficial when target data are scarce or the domain shift is primarily instrument-/environment-driven (speed/load/captor changes in

BFD).

3) YAMNet for BFD

YAMNet is particularly suitable for BFD due to several inherent strengths:

a. **Mel Spectrogram Compatibility:** YAMNet is optimized to operate on Mel spectrograms, which are also widely used in machinery fault diagnosis. These representations capture timefrequency information that mirrors how faults evolve dynamically in mechanical systems.

b. **Pre-trained on Rich Acoustic Features:** Trained on over 2 million labeled audio clips from Google's AudioSet, YAMNet learns robust low- and mid-level acoustic features, such as frequency transitions, rhythmic patterns, and harmonics. These features are surprisingly transferable to industrial vibration signals, especially when faults generate distinctive temporal-frequency signatures.

c. **Lightweight and Deployable:** Based on MobileNetV1, YAMNet is lightweight and computationally efficient, making it ideal for real-time diagnosis in edge devices and WSN nodes where resources are limited.

d. **Improved Generalization with Fine-Tuning:** When fine-tuned on a relatively small bearing fault dataset, YAMNet demonstrates high classification accuracy and generalization ability, even in variable load and speed conditions.

Several recent studies have successfully adapted YAMNet for mechanical fault diagnosis using TL pipelines, validating its potential in classifying diverse bearing defect types (e.g., inner race, outer race, ball faults) from acoustic and vibration data. Its domain-agnostic architecture and Mel spectrogram input structure align well with current trends in timefrequency signal processing for predictive maintenance.

The integration of YAMNet within TL frameworks allows researchers and practitioners to rapidly prototype accurate, low-latency fault classifiers without the need to design custom architectures from scratch. This opens up new opportunities for real-world deployment of AI-enabled maintenance systems in Industry 5.0 settings.

4.7.4 Choosing the best model for transfer Learning

Selecting an appropriate pre-trained model for TL requires a careful balance between technical constraints and application requirements. The choice should be guided by both general model characteristics and task-specific considerations, as each architecture offers distinct advantages and limitations.

Task similarity is one of the most important factors. A pre-trained model trained on a source task closely related to the target task will typically transfer knowledge more effectively. This similarity can be assessed in terms of input data characteristics, feature distributions, and label semantics.

Model performance should be evaluated using relevant metrics such as accuracy, precision, recall, F1 score, and validation loss. While large-scale benchmark results (e.g., ImageNet) can be a useful reference, they do not guarantee optimal performance in all domains. Cross-validation and baseline comparisons on the target task are essential to confirm suitability.

Prediction speed determines how quickly the model can process new inputs, which is critical for real-time or low-latency applications. While prediction speed depends partly on hardware and batch size, it also varies significantly with the networks architecture and parameter count.

Model size and complexity directly influence the memory footprint and computational demands. Larger models (e.g., ResNet-152) often deliver higher accuracy but require more resources, while lightweight architectures (e.g., MobileNet, SqueezeNet, ShuffleNet) are preferable for resource-constrained environments such as embedded or edge devices.

Dataset size and complexity also inform model choice. For large and complex datasets, deeper or more sophisticated architectures may be necessary, whereas smaller datasets may benefit from simpler models to avoid overfitting.

Domain adaptability and generalization refer to the models ability to transfer knowledge from the source domain to the target domain without significant performance degradation. This factor is closely tied to the chosen transfer strategy whether it is feature extraction, partial fine-tuning, or full fine-tuning.

Training time and computational budget must also be considered, particularly when hardware or time resources are limited. Models with fewer parameters and optimized architectures can significantly reduce training costs.

Use case and application requirements should guide the final decision, as some models are better suited to specific tasks (e.g., object detection, speech recognition, time-series forecasting).

A practical approach is to begin with simpler, well-established models such as **GoogLeNet**, **VGG-16**, and **VGG-19** for rapid prototyping, testing preprocessing techniques and training configurations. Once optimal settings are identified, more advanced architectures can be explored to assess potential gains in performance. In deployment scenarios with strict resource constraints, lightweight and computationally efficient models such as **SqueezeNet**, **MobileNet-V2**, or **ShuffleNet** offer an excellent balance between accuracy, size, and speed.

4.7.5 A complete transfer learning process

A complete TL workflow typically involves a sequence of interconnected stages, from initial analysis to deployment. The process begins by acquiring and preparing the target dataset, followed by a transferability analysis to determine *when* and *where* TL should be applied. This ensures that knowledge from a suitable source model can be effectively reused for the target task. Once the groundwork is established,The overall workflow is illustrated in Fig. 4.11. Once the groundwork is established, the following steps guide the implementation:

1. **Select the pretrained model:** Choose a model trained on a source dataset with high similarity to the target dataset. When similarity is uncertain, evaluate multiple pretrained networks to identify the most promising candidate.

2. **Adapt inputs/outputs and optionally extend the architecture:** Preprocess the target dataset to match the models input specifications (e.g., image resolution, normalization, data type); modify the final head to align with the target task (e.g., replace the last fully connected and classification layers to match the required class count); and, if needed, add new layers or a task-specific head to increase capacity and better capture target-domain patterns.

3. **Optionally freeze layers:** Preserve the weights of earlier layers which capture generic features by freezing them, while allowing later layers to adapt to the target task.

4. **Retrain the network:** Fine-tune the modified model on the target dataset, updating only the unfrozen layers.

5. **Evaluate performance:** Assess the model using validation data and relevant metrics (e.g., accuracy, precision, recall, F1 score) to measure effectiveness.

6. **Iterative unfreezing:** If performance is unsatisfactory, progressively unfreeze additional layers, starting from the last layers toward earlier ones, and repeat the trainingevaluation cycle(Steps 36 to search for improved architectures.).

7. **Architectural expansion (optional):** Introduce new layers or modules to enhance the networks capacity, then retrain ,re-evaluate and repeat Steps 27 to search for improved architectures.

8. **Improve or reselect:** If performance is unsatisfactory, either (i) progressively unfreeze additional layers , Architectural expansion and repeat Steps 2-7, *or* (ii) reselect a different pretrained model and repeat Steps 1--8 with the new backbone

9. **Finalize and deploy:** Once the optimal architecture and hyperparameters are found, save the trained model and deploy it in the target environment.

It is important to note that these steps are not strictly linear; insights gained during later stages may require revisiting earlier ones, such as testing a different pretrained model or refining preprocessing strategies. This iterative nature is a hallmark of successful TL projects.

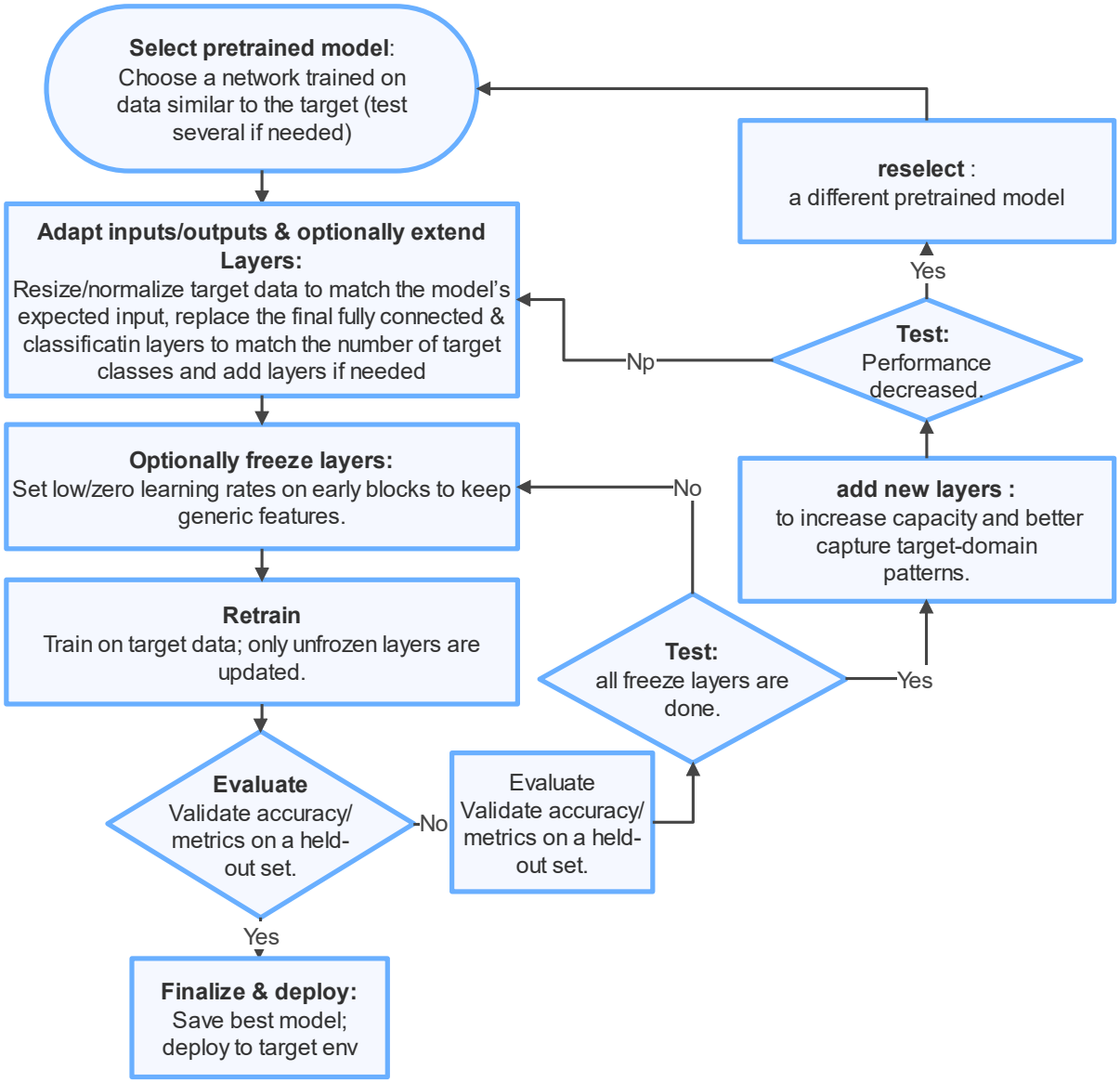


Figure 4.11: Flowchart of the complete TL process.

4.8 Conclusion

We can "keep the old while embracing the new" with a well-designed pipeline for TL. This chapter transitioned from the broad overview of TL what it is and why it matters to the specifics, including when it is useful, where to find examples, and how to modify models without slipping into negative transfer. The taxonomy helped us select methods that fit our data and goals by offering a common map of settings and strategies (inductive, transductive, unsupervised; homogeneous vs. heterogeneous; instance-, feature-, parameter-, or relation-based) (see Fig. 4.6).

We also emphasised that transfer is not a free lunch: performance may suffer if the source and target differ significantly in terms of label space, operating regimes, sensing setup, or preprocessing. This risk is captured by the staircase metaphor (Fig. 4.9), which suggests that rather than "jumping" directly into full fine-tuning, bridge large gaps with intermediate

supports (alignment, adapters, or closer sources).

We then turned these concepts into a tangible workflow, starting with model selection and continuing with input/output adaptation, optional freezing, retraining, thorough evaluation, and, if necessary, iterative unfreezing or architectural expansion. Disciplined experimentation is emphasised by this end-to-end process (Fig. 4.11): begin simple, validate, and only then deepen adaptation or re-select a backbone. The end product is a workable formula for creating reliable, data-efficient BFD and other solutions.

Chapter 5

Experimental results

Contents

V.1	Introduction	88
V.2	Proposed Bearing Fault Classification System	88
V.3	Source and Target Databases	89
V.4	Constructing the Source and Target Model	92
V.5	Deep Transfer Learning Workflow for BearNet	95
V.6	Experimental Details	98
V.7	Benchmark Deep Learning Model Evaluation: YAMNet as Benchmark	101
V.8	BearNet Model Evaluation on the CWRU Dataset	102
V.9	BearNets Performance Comparison with the Benchmark Model	109
V.10	Comparing Performance with Related Works	111
V.11	Conclusion	112

5.1 Introduction

This chapter presents an in-depth evaluation of the proposed DTL strategies for adapting representations learned by the source model, YAMNET, to the target task of BFDs. The evaluation framework is designed around three complementary tasks: (i) binary *fault detection*, (ii) multi-class *fault-type classification*, and (iii) *cross-dataset generalization*. Together, these tasks provide a comprehensive benchmark for assessing both accuracy and robustness of the proposed diagnostic approach.

We begin by introducing the proposed BearNet classification system, detailing the preprocessing pipeline, the source and target datasets employed, and the architectural modifications applied to DTL from YAMNET. Special attention is given to the structured fine-tuning process, where layers are progressively unfrozen to balance general feature reuse with task-specific adaptation.

To provide a meaningful point of reference, we compare BearNet against the benchmark deep-learning model YAMNET, reporting performance across different transfer stages and levels of diagnostic complexity on the widely used CWRU dataset. Finally, we extend the analysis to the MFPT dataset in order to examine generalisability across domains, thereby validating the resilience and adaptability of the proposed method.

In summary, this chapter reports not only the absolute performance of BearNet but also highlights the relative improvements afforded by progressive DTL strategies, underscoring their role in achieving both diagnostic precision and computational efficiency in real-world applications.

5.2 Proposed bearing fault classification system

The proposed system for detecting and classifying bearing faults is set up as a two-stage pipeline, as shown in Figure 5.1. This architecture combines signal preprocessing with a custom DL model to make bearing diagnostics both strong and flexible.

The first step, "dataset preprocessing," is where raw vibration signals are systematically turned into audio representations, which are then turned into Mel-spectrogram images. This change captures the most important time-frequency features of bearing vibrations, which makes them perfect for DL models that work best with inputs that look like images.

The second step, "model construction and classification," introduces the target network, which is called "BearNet". This model is made just for classifying bearing faults and uses the best parts of DTL. Specifically, BearNet starts out by using TL from the pre-trained YAMNet architecture, which was first made for audio classification tasks. BearNet can learn domain-specific patterns in bearing mel-spectrograms and put them into their own fault classes by fine-tuning the representations that were transferred.

The proposed system offers an efficient and precise solution for automated BFDs by integrating advanced preprocessing with DTL.

In the subsequent sections, we provide a detailed discussion of the two main phases

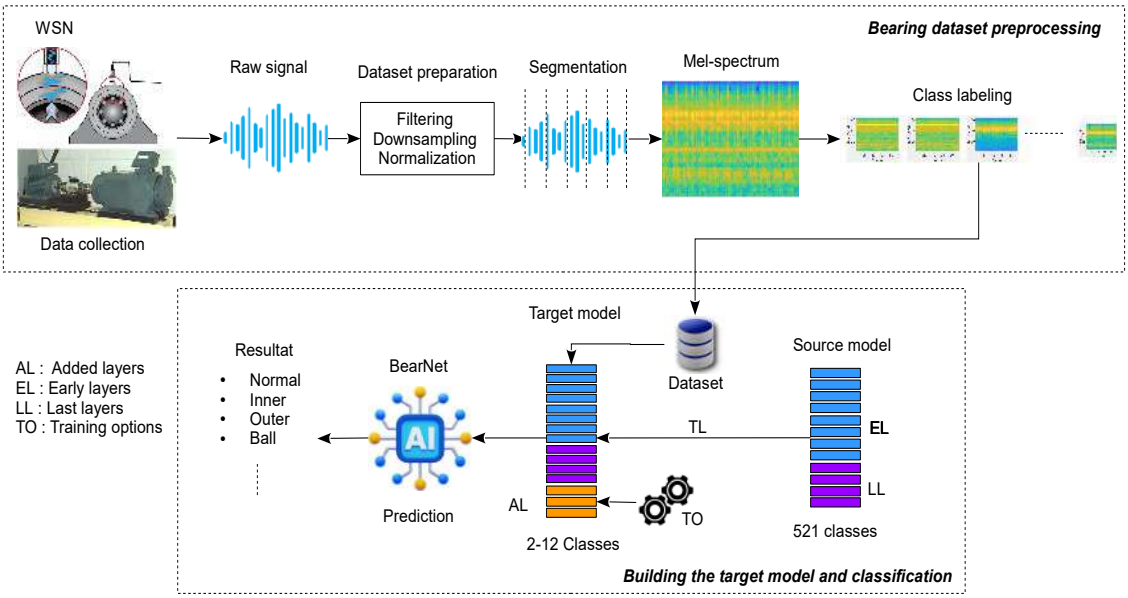


Figure 5.1: Step-by-Step process of BearNet prediction and classification using DTL.

of the proposed system. We begin by presenting the source and target datasets employed in this study, followed by a step-by-step explanation of the preprocessing operations and model design choices. Particular attention is given to the unique characteristics of each phase, highlighting how they contribute to the overall workflow. This structured exposition offers a comprehensive understanding of how the system is organized and how it functions in practice to achieve accurate bearing fault classification.

5.3 Source and target databases

deep transfer learning (DTL) is built upon the idea of leveraging knowledge from a **source database** to improve performance on a related but distinct **target database**. The source task is typically a large-scale classification problem, which allows a model to learn generic and transferable features. When the target task is also a classification problem, fine-tuning generally begins by replacing the final layer with a new one, randomly initialized, whose size corresponds to the number of classes in the target task. This procedure enables the pretrained model to adapt its rich learned representations to the specific requirements of the target application.

5.3.1 Source database for YAMNet

For the pretraining of The YAMNet[1], the source database is **AudioSet**, a large-scale corpus of labeled audio events curated by Google [122]. AudioSet contains approximately 1.6 million 10-second audio clips extracted from YouTube videos, each weakly labeled with one or more classes drawn from an ontology of 521 sound event categories. These categories

[1] `https://github.com/tensorflow/models/blob/master/research/audioset/yamnet/yamnet.py`

span a broad range of domains, including human sounds, animal vocalizations, musical instruments, environmental noise, and mechanical events.

The large size and diversity of AudioSet make it a valuable resource for DTL, as the pretrained model captures a rich representation of temporal and spectral audio patterns. YAMNet, trained on this database, provides a compact MobileNetV1-based architecture that outputs embeddings and class predictions across the full AudioSet taxonomy. These pretrained embeddings can then be transferred and adapted to new downstream tasks, such as vibration-to-sound domain adaptation for BFDs.

5.3.2 Target datasets: Bearing Applications

As discussed in Chapter 3, several benchmark datasets have become central to research in BFDs, each with distinct experimental setups and application domains. For this work, two of the most widely adopted datasets were selected: the Case Western Reserve University (CWRU) dataset (see Section 2.5.1) and the Machinery Failure Prevention Technology (MFPT) dataset (see Section 2.5.5). Their combined use provides both rigor in controlled conditions and robustness under more realistic scenarios. A broader comparative review of representative datasets including CWRU, MFPT, IMS, Paderborn, and PRONOSTIA was presented earlier in Section 2.5.6, which situates our choices within the wider research landscape.

The CWRU dataset is one of the most extensively cited and utilized benchmarks in BFDs research [123, 124]. Its controlled laboratory environment, broad coverage of up to twelve fault classes, and carefully designed experiments make it particularly valuable for validating and comparing diagnostic algorithms. In this study, the CWRU dataset serves as the **primary testbed** to design, develop, and evaluate the proposed methodologies, ensuring both rigor and reproducibility of the results.

To strengthen the generalizability and industrial relevance of the developed methods, the MFPT dataset is also employed [43]. Unlike CWRU, the MFPT dataset incorporates a wider range of fault types and more realistic variability in operating conditions, offering a complementary perspective that better reflects practical scenarios. This allows us to assess whether the proposed approach can adapt beyond strictly controlled laboratory settings.

By combining insights from both datasets, this work achieves two objectives: (i) the CWRU dataset enables detailed experimentation and benchmarking under controlled fault scenarios, while (ii) the MFPT dataset provides a testing ground for robustness under real-world variability. This dual-dataset strategy not only increases the credibility of the findings but also bridges the gap between laboratory research and industrial applications.

5.3.3 Dataset preprocessing

To ensure consistency and reliability across different datasets, a careful preprocessing pipeline was applied to the raw vibration signals.

The first step was **noise reduction**. A low-pass filter with a cutoff frequency of 5 kHz was

used to remove high-frequency noise while preserving the informative part of the spectrum. To harmonize the datasets and reduce computational cost, all signals were then **down-sampled** from their original rate of 48 kHz to a target of 12 kHz. This guarantees comparability across sources while retaining the fault-related frequency content.

Next, the data were **normalized** to zero mean and unit variance, ensuring uniform scaling across signals:

$$x_{\text{norm}} = \frac{x - \mu}{\sigma}, \tag{5.1}$$

where x is the original signal, μ its mean, and σ the standard deviation.

For model compatibility, the normalized signals were then **segmented** into fixed-length windows using a sliding window approach. In this work, we used a window length of $L = 16{,}384$ samples with a stride of $S = 16{,}184$, creating an overlap of 200 samples between consecutive windows. This overlap reduces information loss at boundaries and ensures continuity across segments. To improve spectral resolution, each segment was multiplied by a **Hamming window**, defined as:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{L-1}\right), \tag{5.2}$$

resulting in the windowed signal $y(n) = x_{\text{norm}}(n) \cdot w(n)$. This step reduces spectral leakage by suppressing side lobes and enhancing the main lobe, which is crucial for precise fault signature extraction.

In cases where class imbalance was observed, **data augmentation** was applied to increase the representation of under-sampled classes, thereby improving robustness and preventing model bias.

Finally, the preprocessed 1D vibration signals were transformed into 2D **Mel-spectrograms**, a representation well established in audio analysis and increasingly applied to mechanical signals. The Mel-spectrogram is defined as:

$$\text{MelSpec}(t, f) = \text{DCT}\left(\log\left(\sum_{k=1}^{K} |X(t,k)|^2 \cdot H_m(k)\right)\right), \tag{5.3}$$

where $X(t,k)$ is the short-time Fourier transform (STFT) of the normalized signal, $H_m(k)$ is the Mel filter bank, and DCT denotes the discrete cosine transform.

This conversion maps vibration data into a timefrequency space that captures transient signatures in a way that is both visually interpretable and directly usable by modern DL architectures. Moreover, it enables the effective use of DTL methods originally developed for audio classification, thereby adapting powerful pretrained models to the task of BFDs.

5.4 Constructing the source and target model

The proposed framework relies on DTL, where knowledge from a general-purpose model is adapted to a specialized task. In this work, a pretrained source model is first introduced,

followed by the construction of a target model specifically tailored for bearing fault classification. This progression highlights how general audio-based representations can be refined to capture the subtle vibration patterns characteristic of bearing defects.

5.4.1 Pretrained source model: YAMNet

In this work, the YAMNet architecture was selected as the source model. YAMNet is a lightweight convolutional neural network pre-trained on the large-scale **AudioSetYouTube corpus**, which contains millions of labeled audio events spanning 521 distinct classes. Its training domain, based on diverse audio signals, shares key characteristics with vibration signals used in bearing diagnostics. Both domains deal with one-dimensional temporal data, making YAMNet particularly suitable for transfer to vibration-based fault detection.

The model is designed using depthwise-separable convolutional layers, which make it computationally efficient without sacrificing representation power [125]. This property is especially advantageous for deployment in resource-constrained environments, such as embedded systems or WSN-based monitoring setups, where low-energy processing and real-time decision-making are critical.

By fine-tuning its deeper layers, YAMNet can be adapted to capture subtle changes in vibration signals associated with different bearing fault types. This enables greater sensitivity in identifying early-stage defects, while its efficient architecture reduces computational overhead. Moreover, its modular design allows for seamless integration with existing diagnostic pipelines, ensuring both practical applicability and ease of deployment.

Table 5.1 presents a structured overview of YAMNets architecture. Each row corresponds to a specific functional block or layer, illustrating the hierarchical feature extraction process that underpins its transferability. This structured view provides a clear understanding of how the network processes raw temporal data and highlights its adaptability to the target domain of bearing fault classification.

The YAMNet model is detailed as follows:

- **Input layer:** The network begins with an input layer that processes spectrogram images in grayscale format, each with dimensions of $96 \times 64 \times 1$. These images encode audio data by capturing both temporal and spectral information, which is essential for subsequent analysis.
- **Iterative convolutional enhancement network (ICEN) blocks:** The backbone of YAMNet consists of 13 repeated ICEN blocks. Each block is carefully structured to progressively refine and deepen the feature extraction process, allowing the network to capture increasingly complex representations of the input data. These blocks act as a hierarchical pipeline, ensuring that both low-level and high-level patterns are effectively modelled.
- **Terminal enhanced convolutional network (TECN) block:** At the deeper stages of the model, the TECN block takes over to provide a more concentrated refinement of features. Unlike the more extensive ICEN structure, the TECN block emphasizes

Table 5.1: YAMNet architecture

BO	Block name	LI	CD	Output Size	Notes or Special Characteristics
input	input	1	Image input	96 × 64 × 1	96 × 64 × 1 images
2--13	ICEN	2	2-D Convolution	48 × 32 × 32	32 / 64 / 128 / 128 / 256 / 256 / 512 / 512 / 512 / 512 / 512
			Batch normalization	48 × 32 × 32	Batch normalization with 32 channels
			ReLU	48 × 32 × 32	ReLU
			2-D grouped conv.	48 × 32 × 32	32 groups of 3 × 3 convolutions
			Batch normalization	48 × 32 × 32	Batch normalization with 32 channels
		79	ReLU	48 × 32 × 32	ReLU
14	TECN	80	2-D Convolution	48 × 32 × 1024	1024 (1 × 1 × 1024) convolutions
		81	Batch normalization	48 × 32 × 1024	Batch normalization with 1024 channels
		82	ReLU	48 × 32 × 1024	ReLU
Output	Classifier	83	2-D Global AP	1 × 1 × 1024	2-D global average pooling
		84	Fully connected	1 × 1 × 521	521-unit fully connected layer
		85	Softmax	1 × 1 × 521	Softmax
		86	Classification output	1 × 1 × 521	Cross-entropy with 'Speech' and 520 other classes

Abbreviations: Block order (BO), Layer identifier (LI), Configuration description (CD), Kernel depth (KD), Kernel size (KS), Stride (S), Padding (P), Fully connected (FC), Batch normalization (BN), Average pooling (AP)

terminal feature enhancement. It begins with a two-dimensional convolution layer, followed by batch normalization and a rectified linear unit (ReLu) activation, which together optimize the extraction of subtle and discriminative patterns needed for complex classification tasks.

- **Classifier block:** Finally, the refined features are passed to a fully connected layer configured to predict 521 output classes. A softmax activation function converts the networks outputs into class probabilities, enabling accurate categorization of audio events. During training, cross-entropy loss is employed to quantify the gap between predicted probabilities and the true class labels, guiding the model to iteratively adjust its parameters for optimal accuracy and robustness.

5.4.2 Target model:BearNet

The adaptation from the source model YAMNet to the target model, BearNet, is illustrated in Figure 5.2. This process applies the principle of fine-tuning by retaining the robust feature extraction capability of YAMNet, while tailoring the architecture to suit the specific requirements of bearing fault classification. The key modifications involve incorporating an additional **ICEN block** and adjusting the output layer so that the number of classes matches the task at hand (2, 4, or 12 classes). These changes are highlighted in the figure.

To strengthen the models ability to represent fine-grained patterns essential for bearing fault detection, BearNet introduces a new set of convolutional filters. As shown in Figure 5.3, the filter depth increases progressively, reaching up to **2048** 1×1 filters in later layers. This enhancement significantly improves the networks sensitivity to subtle variations in vibration signatures, thereby boosting diagnostic accuracy. By extending the channel depth within the ICEN layers, the model achieves stronger feature transformations and more discriminative representations of complex fault conditions.

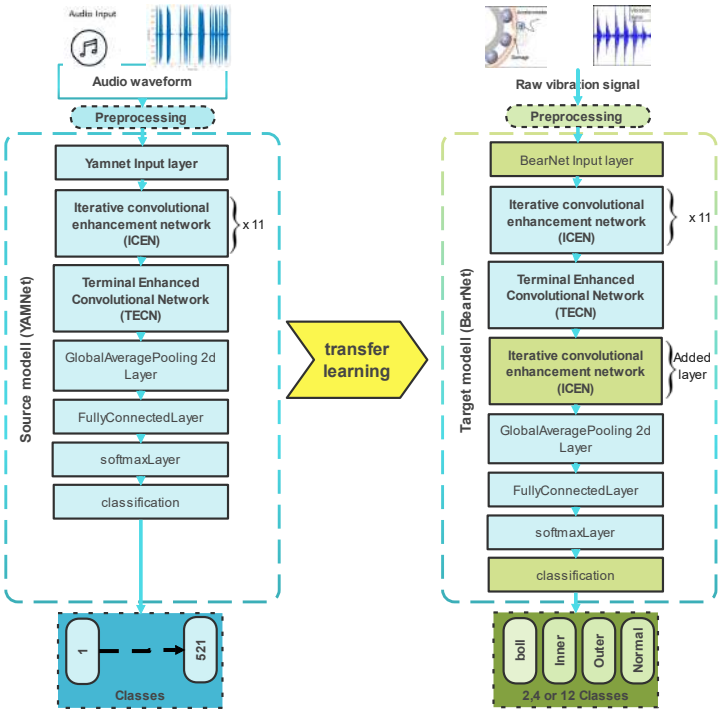


Figure 5.2: Foundations of DTL and its Architectural Building Blocks: from YAMNet to BearNet.

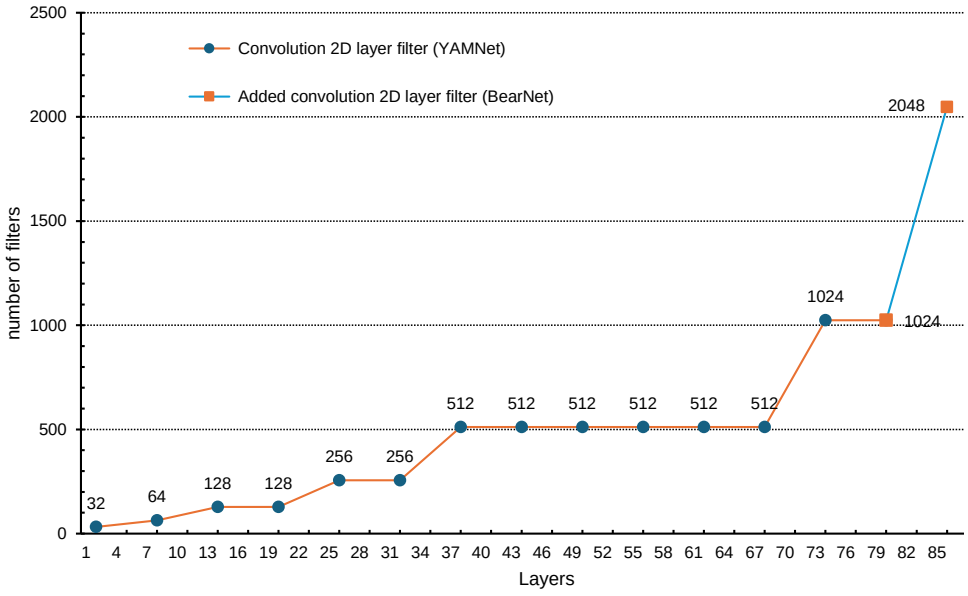


Figure 5.3: Comparison of convolutional layer configurations between YAMNetand enhanced BearNet architectures.

These architectural refinements substantially improve both sensitivity and specificity, representing a notable advancement in predictive maintenance applications. Importantly, BearNet adapts its output layer to align with the structure of the available dataset. Depending on the classification task, it supports:

- **Binary classification:** distinguishing between Normal and Faulty conditions,
- **Four-class classification:** separating Normal, Ball fault, Inner race fault, and Outer

race fault,

- **Twelve-class classification:** offering a more granular categorization that accounts for fault type and severity.

This flexibility allows the model to serve as a comprehensive diagnostic framework. The supported classification schemes are summarized in Table 5.2, which provides a clear mapping between dataset configurations and target classes.

Table 5.2: Classification scenarios and their corresponding classes for BFDs within the CWRU dataset.

Case	Classification detail	Diagnostic implication	Data overview
1	Two classes: Normal and Faulty bearings	Serves as a preliminary filter to identify bearings that show signs of malfunction.	**Default Baseline** **Normal Baseline**
2	Four classes: Normal, Ball faults, Inner race faults, Outer race faults	Pinpoints the exact location of the fault within the bearing, crucial for targeted diagnostics.	**Ball** **Inner Race** **Outer Race** **Normal Baseline**
3	Twelve classes: Detailed classification based on fault size and location	Provides detailed fault characteristics for precise maintenance planning, optimizing repair schedules and enhancing machinery lifespan.	**Ball:** with fault diameters of 0.007, 0.014, 0.021, and 0.028 inches **Inner Race:** with fault diameters of 0.007, 0.014, 0.021, and 0.028 inches **Outer Race:** with fault diameters of 0.007, 0.014, and 0.021 inches **Normal Baseline**

5.5 Deep transfer learning workflow for BearNet

The development of the proposed bearing fault classification system builds upon the principles of DTL, as introduced in Section 4.7.5. Instead of training a DL model entirely from scratch,"which is often computationally expensive and demands very large annotated datasets", knowledge is strategically transferred from a pretrained source model. This approach enables the system to exploit rich feature representations already acquired from a large-scale corpus, thereby accelerating convergence, improving robustness, and enhancing overall performance when applied to the target task.

In this study, YAMNet is adopted as the source model, while the proposed BearNet serves as the target architecture. The process of applying DTL to BFDs can be described as a structured workflow that extends from dataset preparation through to evaluation. As outlined in [126], the workflow ensures reproducibility and clarity by detailing each stage of the DTL pipeline.

5.5.1 BearNet deep transfer learning workflow

The process of applying DTL to BFDs using YAMNet as a source model and BearNet as the target model can be summarized in twelve key steps. These steps illustrate the complete workflow, from data preparation to evaluation, ensuring reproducibility and clarity in as presented in below:

1) **Import libraries:** Load the essential DL and signal processing code required for the implementation.

2) **Load dataset:** Import the bearing fault datasets, e.g., CWRU and MFPT, into the training environment.

3) **Partition data:** Split the datasets into training and validation, subsets to ensure unbiased evaluation.

4) **Label encoding:** Convert class labels for compatibility with the classification layers.

5) **Data augmentation:** Apply techniques to balance class distributions and enrich the training set, especially for underrepresented fault types.

6) **Initialize source model:** Load the pretrained YAMNet model as the base architecture.

7) **Construct target model:** Adapt the base model by adding new layers (ICEN block, updated output layer) to form BearNet.

8) **Compile model:** Define optimizer (e.g., Adam), learning rate, and loss function (cross-entropy) to guide training.

9) **Model training:** Train the adapted model on the bearing datasets using fine-tuning strategies, progressively unfreezing layers.

10) **Performance tracking:** Plot curves for training and validation accuracy/loss to monitor learning dynamics.

11) **Predictions:** Use the trained model to classify test samples into normal and faulty states (binary), or into detailed multi-class fault types.

12) **Evaluation:** Generate confusion matrices and compute performance metrics (accuracy, precision, recall, F1-score) for each classification task.

This twelve-step workflow ensures that the knowledge from a large-scale audio corpus is effectively transferred to the target domain of vibration-based fault diagnosis, while keeping the process computationally efficient and well-suited for deployment in WSN environments.

5.5.2 BearNet model building strategies in deep transfer learning

In DTL, two complementary strategies are commonly adopted: **feature extraction** and **fine-tuning**. Both approaches aim to reuse the knowledge encoded in a pretrained model, but differ in how much of the original network is modified [126].

1) **Feature Extraction:** In this strategy, the pretrained model (e.g., YAMNet) is employed as a *fixed feature extractor*. All convolutional layers are frozen, meaning their weights remain unchanged, and only a new classification head is added to adapt the network to the target task of bearing fault classification. Specifically, in the case of BearNet,

we appended an additional ICEN block to strengthen feature transformation, and replaced the output layer so that the number of neurons matches the classification task (2, 4, or 12 classes). This ensures that the rich embeddings generated by YAMNet are effectively mapped to the desired fault categories. The extracted representations can then either feed this new classifier head (the *BearNet head*). This method is computationally efficient and helps avoid overfitting, making it particularly useful when training data is scarce or when repeated evaluation on the same dataset is required.

2) **Fine-Tuning:** Fine-tuning provides a deeper level of adaptation by *unfreezing* selected parts of the pretrained network typically the later layers so that their weights are updated during training on the target dataset. This strategy allows the model to retain the general feature representations learned from the large-scale AudioSet corpus, while progressively adapting them to the vibration-specific signatures of bearing faults.

   In practice, this means that YAMNets robust feature extractor is not only reused but also refined: the deeper filters are re-trained to capture domain-specific variations that are critical for accurate fault detection. This incremental adaptation process demonstrated a superior capacity to generalize, outperforming feature extraction alone and showing that **progressive fine-tuning is essential for achieving optimal diagnostic precision in real-world industrial scenarios**.

In practice, the choice between feature extraction and fine-tuning depends strongly on the problem context. Feature extraction is often preferable when computational resources or labeled data are severely limited, since it allows the pretrained model to act as a fixed feature extractor with minimal retraining. However, in our experiments with BearNet, this strategy did not produce satisfactory results: the frozen layers limited the models adaptability to the complex vibration patterns of bearing faults, leading to suboptimal diagnostic accuracy.

By contrast, fine-tuning provided a clear advantage. When progressively unfreezing deeper layers of the pretrained YAMNet, the model was able to refine its filters and capture task-specific features more effectively. This yielded significantly better classification performance on both the CWRU and MFPT datasets, confirming that **fine-tuning is essential for achieving high sensitivity and robust generalization** in the BFDs setting.

Therefore, although both strategies were considered in this study, fine-tuning emerged as the superior approach. It allowed us to maintain a balance between computational efficiency and diagnostic accuracy, ensuring that BearNet remains lightweight enough for deployment in resource-constrained WSN-based environments while still delivering state-of-the-art fault classification performance.

5.6 Experimental details

The results of the proposed BearNet framework for bearing fault detection, developed using the DTL strategy, are presented in this section. A series of structured experiments were designed to rigorously evaluate BearNet under different levels of diagnostic complexity. These

experiments cover binary fault detection, multi-class fault-type classification, and cross-dataset generalisation, ensuring a comprehensive assessment of the models robustness and efficiency.

First, a benchmark experiment was conducted using the YAMNet model directly as a student model. In this configuration, YAMNet was applied to the bearing fault datasets without any architectural modification, providing a baseline against which the proposed enhancements could be compared.

Next, the adapted BearNet model was employed, leveraging DTL at progressive fine-tuning levels. Layers were selectively unfrozen to identify the most effective balance between feature reuse and task-specific adaptation. This strategy was essential to enhance the models sensitivity to subtle vibration patterns while minimizing the risk of overfitting.

5.6.1 Experimental setup and environment

To thoroughly assess classification performance, the experimental framework was implemented across three phases of increasing complexity:

- **Phase 1: Binary classification** bearings were categorized simply as *normal* or *faulty*. This phase established the foundation by demonstrating the systems ability to distinguish between healthy and defective conditions.
- **Phase 2: Four-class classification** the faulty condition was subdivided into *ball fault*, *inner race fault*, and *outer race fault*. This stage emphasized the importance of precise fault localization within the bearing assembly.
- **Phase 3: Twelve-class classification** the most detailed scenario, where faults were categorized into twelve distinct classes based on fault type and severity (inner race, ball, outer race, each with varying fault sizes), alongside the normal class. This phase provided fine-grained insights critical for predictive maintenance, enabling targeted interventions and optimized scheduling of repair actions.

Finally, to evaluate the generalization capacity of BearNet, experiments were extended to an additional dataset. This step tested the resilience of the proposed approach under different fault scenarios, ensuring its applicability beyond a single benchmark dataset.

All experiments were performed on a computing environment equipped with an Intel Core i5-7500 CPU at 3.40 GHz, 8 GB of RAM, and an NVIDIA GeForce GT 710 GPU, running under MATLAB R2022b. The GPU was employed for the most computationally demanding tasks, ensuring efficient training and evaluation of the DL models.

5.6.2 Training details

Unlike CNNs trained on large-scale image datasets such as ImageNet, where inputs are typically RGB images resized to 256×256 pixels, the YAMNet model operates on Mel-spectrogram inputs of fixed dimensions $96 \times 64 \times 1$. These grayscale spectrograms encode

both temporal and spectral information, making them particularly well suited for vibration-based bearing fault analysis.

In our workflow, the raw vibration signals acquired from bearings are first preprocessed and transformed into Mel-spectrograms. These are then resized and segmented to conform with the input format required by YAMNet, ensuring seamless compatibility with the target architecture, BearNet (see Section 5.4.2).

The dataset is split into two subsets, with 70% allocated for training and 30% for validation. To improve generalization and avoid bias, data are permuted at the start of each epoch, and augmentation is applied when needed to balance class distributions.

Model training employs the **Adam** optimizer with a mini-batch size of 128 and an initial learning rate of 0.0003. The learning rate is decayed progressively to encourage stable convergence over the course of 40 epochs, each consisting of a full pass through the training set.

the choice of Adam (Adaptive Moment Estimation) optimizer was selected as the training algorithm because it offers a reliable balance between efficiency, accuracy, and robustness. Unlike classical stochastic gradient descent, Adam adaptively adjusts the learning rate for each parameter, which accelerates convergence and improves training stability, even in the presence of noisy or imbalanced data. This makes it particularly effective in BFDs.

deep transfer learning (DTL) is applied following the strategy outlined in Section 4.7.5. Depending on the experiment, different layers of the pretrained network are selectively frozen or unfrozen: earlier convolutional layers remain fixed to retain generic

5.6.3 Evaluation metrics

The benchmark model YAMNet and the proposed BearNet architecture were evaluated using a variety of classification metrics specifically designed for DL model assessment, such as True Positive Rate (TPR), False Positive Rate (FPR), Accuracy, Specificity, Precision, Recall, F1-score, and Detection Error Rate (DER). Table 5.3 summarizes these metrics along with their descriptions.

This comprehensive evaluation approach provides deeper insights into the models ability to generalize, predict, and classify accurately, ensuring that both the strengths and potential areas for improvement are thoroughly understood.

5.7 Benchmark deep learning model evaluation: YAMNET as benchmark

In this experiment, the benchmark DL model is YAMNet, originally trained on the large-scale AudioSet corpus for general sound event recognition. For our study, YAMNet was adapted to the bearing fault detection task by modifying its input and output layers to match the characteristics of vibration-derived Mel-spectrograms and the specific number of fault

Table 5.3: Summary of DL metrics employed for evaluating BearNet model.

Metric	Formula	Description
FPR and TPR	$\frac{FP}{FP+TN}, \quad \frac{TP}{TP+FN}$	FPR: proportion of non-faulty bearings incorrectly identified as faulty. TPR: proportion of correctly identified faulty bearings among all actual faulty bearings.
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Ratio of correctly identified instances (both faulty and non-faulty) to the total. Indicates overall effectiveness of the model.
Specificity	$\frac{TN}{TN+FP}$	Proportion of actual negatives (non-faulty bearings) correctly identified. High specificity means fewer false positives.
Precision / Recall	$\frac{TP}{TP+FP}, \quad \frac{TP}{TP+FN}$	Precision: proportion of correctly identified faulty bearings among those predicted as faulty (lowers false positives). Recall (sensitivity): proportion of correctly identified faulty bearings among all actual faulty ones (lowers false negatives).
F1-Score	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	Harmonic mean of precision and recall, balancing both metrics. Useful when a single robust performance score is required.
Detection Error Rate (DER)	$\frac{1}{2}(P_{FA} + P_{MD})$	Average of the false-alarm probability (P_{FA}) and the missed-detection probability (P_{MD}). Lower DER indicates better fault detection performance.

Abbreviations: True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), False Alarm Probability (P_{FA}), Missed Detection Probability (P_{MD}).

classes.

The evaluation considered three progressively complex classification scenarios, as illustrated in Figure 5.4, and discussed below :

a. **Binary classification (Normal vs. Faulty):** The model achieved rapid convergence, with validation accuracy starting at 46.87% and reaching 100% within 40 epochs. Correspondingly, the validation loss dropped sharply from 0.6546 to 0.0029, confirming effective adaptation to the binary task.

b. **Four-class classification (Normal, Ball, Inner Race, Outer Race):** The validation accuracy improved steadily from 47.03% to 98.72%. In parallel, the validation loss decreased from 1.2831 to 0.0367, demonstrating the models robustness in handling multi-class scenarios while maintaining low misclassification rates.

c. **Twelve-class classification (Normal + detailed fault types by location and severity):** This was the most challenging setup, with accuracy beginning at only 10.08% due to the complexity of the task. Nonetheless, the model showed strong learning capacity, reaching 96.90% by the final epoch. The validation loss followed the same trend, falling from 2.47 to 0.23, indicating significant optimization despite the complexity of distinguishing twelve fault classes.

Overall, these results confirm that YAMNet, even as a benchmark model, can be successfully adapted for vibration-based bearing fault detection. Its performance across binary, four-class, and twelve-class tasks highlights both its flexibility and the potential of DTL for fault diagnostics.

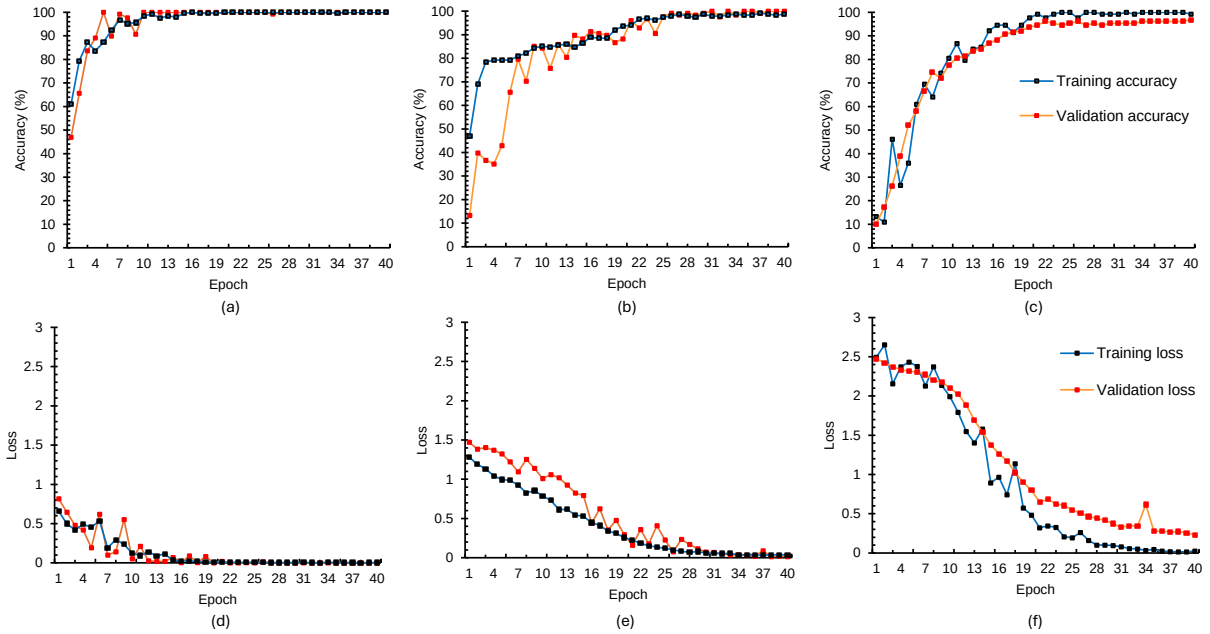


Figure 5.4: Comparative evaluation of YAMNet performance across different configurations: (a) Binary classification accuracy, (b) Four-class accuracy, (c) Twelve-class accuracy, (d) Binary loss, (e) Four-class loss, and (f) Twelve-class loss.

5.8 BearNet model evaluation on the CWRU Dataset

This experiment assessed the performance of the proposed BearNet architecture across different diagnostic complexities (2, 4, and 12 classes). A structured DTL strategy was employed, in which layers were progressively unfrozen from TL_0 to TL_3.

For clarity, we denote $\mathbf{TL}_n$ as the *DTL stage n*, with n ranging from 0 to 3. Each stage corresponds to a different degree of freezing or fine-tuning of the BearNet layers.

The goal of this staged unfreezing was to determine the most effective configuration, balancing generalization with task-specific adaptation.

Table 5.4 provides a concise summary of the strategies adopted at each stage. At TL_0, the network operates under a **freezing strategy**: all layers remain frozen and the model functions purely as a fixed feature extractor. This baseline ensures stability but offers minimal adaptability to new fault patterns.

From TL_1 onward, the paradigm shifts to **fine-tuning**. Here, progressively deeper layers are unfrozen, making a larger proportion of weights trainable. At TL_1, about 32.91% of the weights are unfrozen tripling the number of trainable parameters compared to TL_0. In TL_2 and TL_3, nearly half of the network is fine-tuned, further enhancing the capacity to capture subtle and task-specific vibration features.

This transition from complete freezing to controlled fine-tuning demonstrates how BearNet evolves from acting as a generic feature extractor to a specialized diagnostic model. Freezing ensures robustness and avoids overfitting when data are limited, while fine-tuning leverages the deeper layers to capture the complexity of bearing fault signatures. By carefully balancing these two strategies, the network achieves both adaptability and precision, which is essential for handling varying levels of diagnostic difficulty in real-world scenarios.

Table 5.4: Summary of TL with varied frozen layers in the BearNet model.

Model	Frozen layers	Unfrozen ratio (%)	Unfrozen weights	Frozen weights	Strategy (Freezing / Fine-tuning)
TL_0	80	0.00	0	3,184,512	Freezing (Feature extraction only)
TL_1	77	32.91	1,048,576	2,135,936	Fine-tuning (partial adaptation of deeper layers)
TL_2	71	49.69	1,582,080	1,602,432	Fine-tuning (extended adaptation, half layers active)
TL_3	68	49.80	1,586,688	1,597,824	Fine-tuning (deep adaptation, maintaining balance)

5.8.1 Analyses and discussions

We next analyse performance of the proposed BearNet model across binary, four-class, and twelve-class scenarios to assess the impact of progressively unfreezing layers.

The following section provides a detailed discussion of the experimental results obtained from the proposed BearNet model. over progressive transition from complete freezing (TL_0) to controlled fine-tuning (TL_1TL_3).

- **Two classes' performance:** The BearNet model's two-class configuration shows clear performance dynamics across the TL stages. At TL_0, where all layers are frozen, the model behaves as a fixed feature extractor with limited adaptability. This is reflected in the moderate starting accuracy (45.31%) that gradually improves to 100% by the final epochs. Validation accuracy follows a similar trend, starting at 52.38% and reaching perfect levels. As deeper layers are progressively unfrozen in TL_1 (32.91% of weights trainable) and beyond, the model gains significant adaptability, leading to a faster rise in both training and validation accuracy. Loss metrics also confirm these improvements, with validation loss declining steadily and reaching as low as 0.0001 in TL_3, indicating stable and precise learning without overfitting. These results are visualized in Figure 5.5.

- **Four classes' performance:** In the four-class scenario, the impact of progressive fine-tuning becomes even more pronounced. At TL_0, validation accuracy starts at 35.17% and rises modestly to 83.90%. As fine-tuning begins in TL_1, validation accuracy improves more robustly, reaching 93.22%, while the loss drops sharply from 2.1140 to 0.1762. TL_2 further enhances feature extraction, yielding validation accuracy up to 99.15% with corresponding loss reduced to 0.1208. Finally, TL_3 achieves near-perfect generalization, with validation accuracy peaking at 99.75% and validation loss decreasing to 0.0191. These results confirm that fine-tuning deeper layers substantially boosts diagnostic precision without overfitting. Figure 5.6 illustrates these findings.

- **Twelve classes' performance:** The most complex scenario (12 classes) highlights BearNets ability to generalize under challenging diagnostic conditions. At TL_0, validation accuracy starts low at 26.04% but steadily climbs to 90.62%, with loss decreasing from 2.43 to 0.55. When fine-tuning begins (TL_1), the model shows a sharper

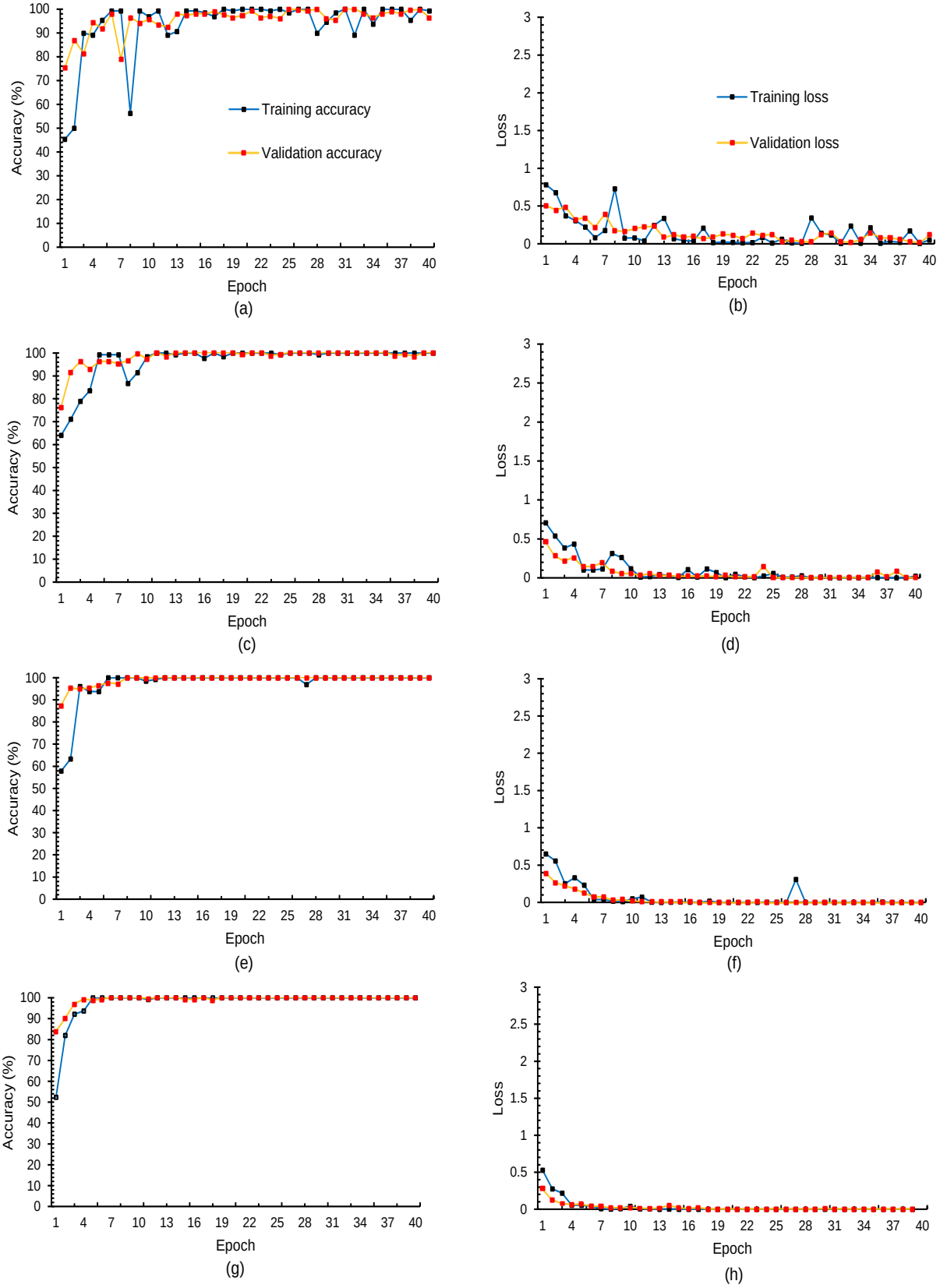


Figure 5.5: Comparative evaluation of two-class BearNet performance metrics across TL stages: (a),(b) Accuracy and loss for TL_0; (c),(d) TL_1; (e),(f) TL_2; (g),(h) TL_3.

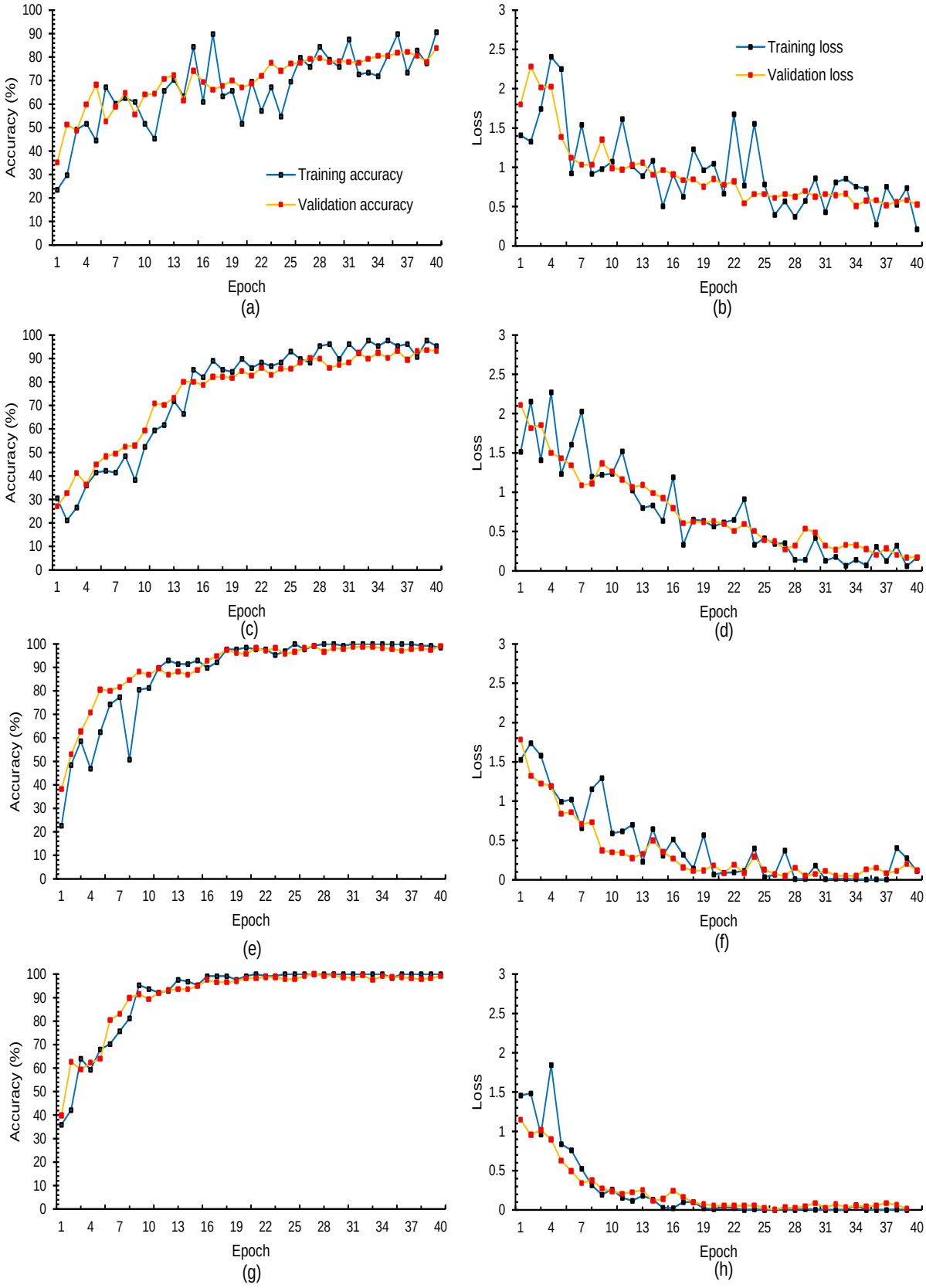


Figure 5.6: Comparative evaluation of four-class BearNet performance metrics across TL stages: (a),(b) Accuracy and loss for TL_0; (c),(d) TL_1; (e),(f) TL_2; (g),(h) TL_3.

improvement, achieving 94.79% accuracy and reducing loss to 0.29. TL_2 further elevates performance, with validation accuracy reaching 98.95% and loss dropping to 0.03. At TL_3, BearNet demonstrates exceptional generalization, starting at 23.96% and peaking at 99.70%, with validation loss reduced to just 0.02. This progression confirms that carefully staged fine-tuning significantly enhances the models capability to handle highly granular fault categorization. Figure 5.7 presents these results.

5.8.2 Confusion matrix analysis of BearNet performance

To further validate the classification performance of the proposed BearNet model, confusion matrices were generated for the binary, four-class, and twelve-class classification tasks, as shown in Figure 5.8. These matrices provide a detailed view of class-level predictions, highlighting the models strengths in discriminating between fault types and severities.

In the **binary classification task** (Figure 5.8(a)), BearNet achieves perfect separation between the *Normal Baseline* and *Faulty* classes. No misclassifications are observed, reflecting the models ability to consistently and reliably distinguish defective bearings from healthy ones. This level of accuracy is especially critical for early-stage industrial diagnostics, where the first priority is to detect the presence of a fault.

The **four-class configuration** (Figure 5.8(b)) further demonstrates the models robustness. BearNet achieves 100% accuracy on the *Normal Baseline*, *Inner Race*, and *Outer Race* classes. For the *Ball* class, the performance remains exceptionally strong, with 99% correct predictions and only 1% of samples being misclassified as *Outer Race*. This result underscores the networks ability to maintain high precision even as the diagnostic complexity increases from simple binary to multi-fault classification.

The most challenging scenario, the **twelve-class classification task** (Figure 5.8(c)), requires the model not only to detect fault types but also to discriminate fault severity levels. Despite this added complexity, BearNet delivers highly accurate results with only negligible errors. The most notable case is the *Ball12k021* class, where 3.57% of the samples were misclassified as *Ball12k007*. Importantly, all other classes achieve near-perfect accuracy, showing that the model generalizes effectively across fine-grained diagnostic categories.

Taken together, these results demonstrate that BearNet is both **robust and reliable** across classification complexities. Its ability to maintain high accuracy in simple, intermediate, and fine-grained diagnostic tasks establishes it as a strong candidate for real-world industrial fault diagnosis systems, where precision and consistency are paramount.

5.8.3 BearNet's generalisability on the MFPT Dataset

To further assess the robustness and applicability of the proposed approach, the BearNet architecture was evaluated on the MFPT dataset using the most effective DTL stage, TL_3. This experiment was designed to verify whether the knowledge gained from the CWRU dataset could be successfully transferred to a different dataset, thereby testing the models ability to generalise across domains. The results, presented in Figure 5.9, provide strong

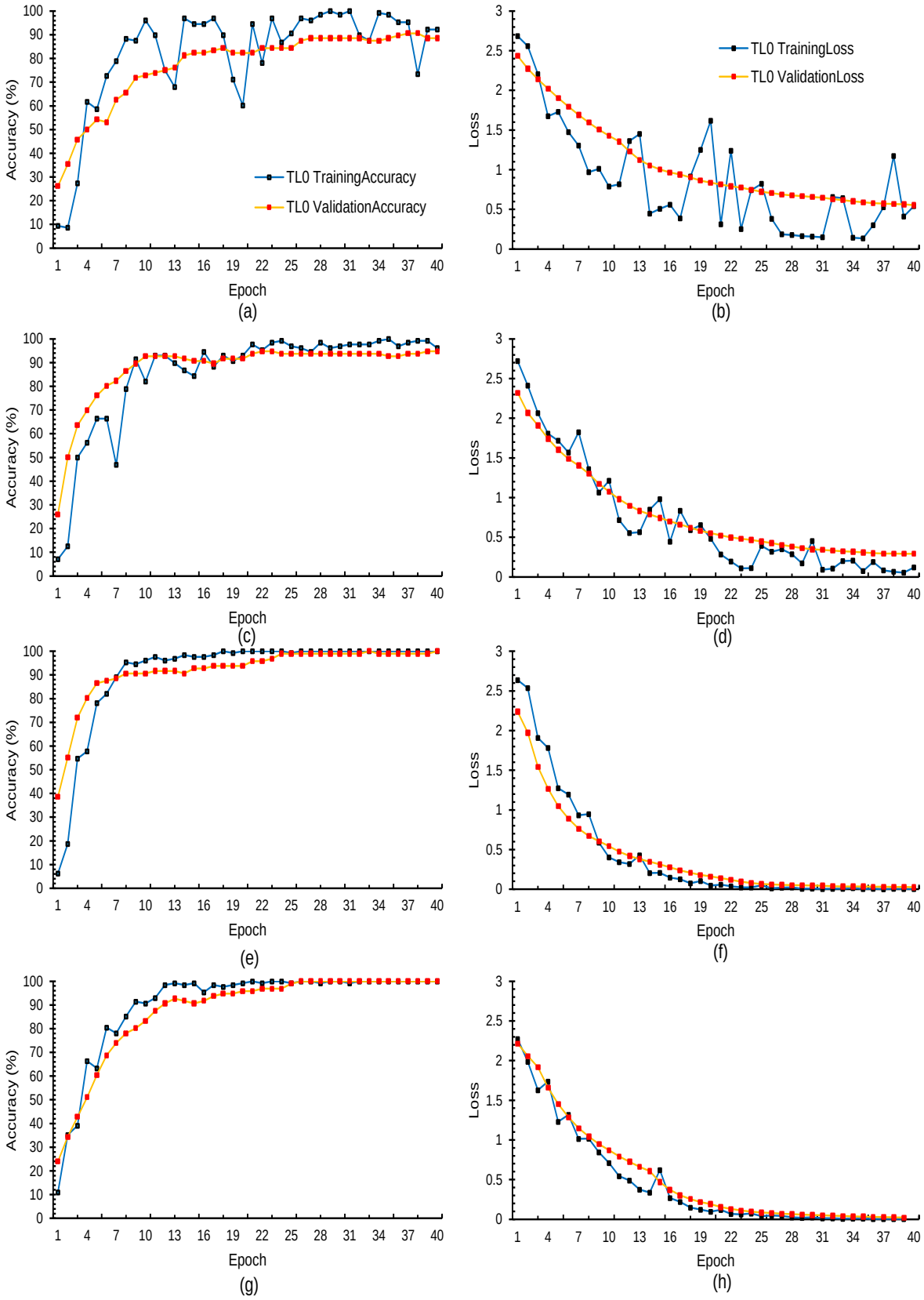


Figure 5.7: Comparative evaluation of twelve-class BearNet performance metrics across TL stages: (a),(b) Accuracy and loss for TL_0; (c),(d) TL_1; (e),(f) TL_2; (g),(h) TL_3.

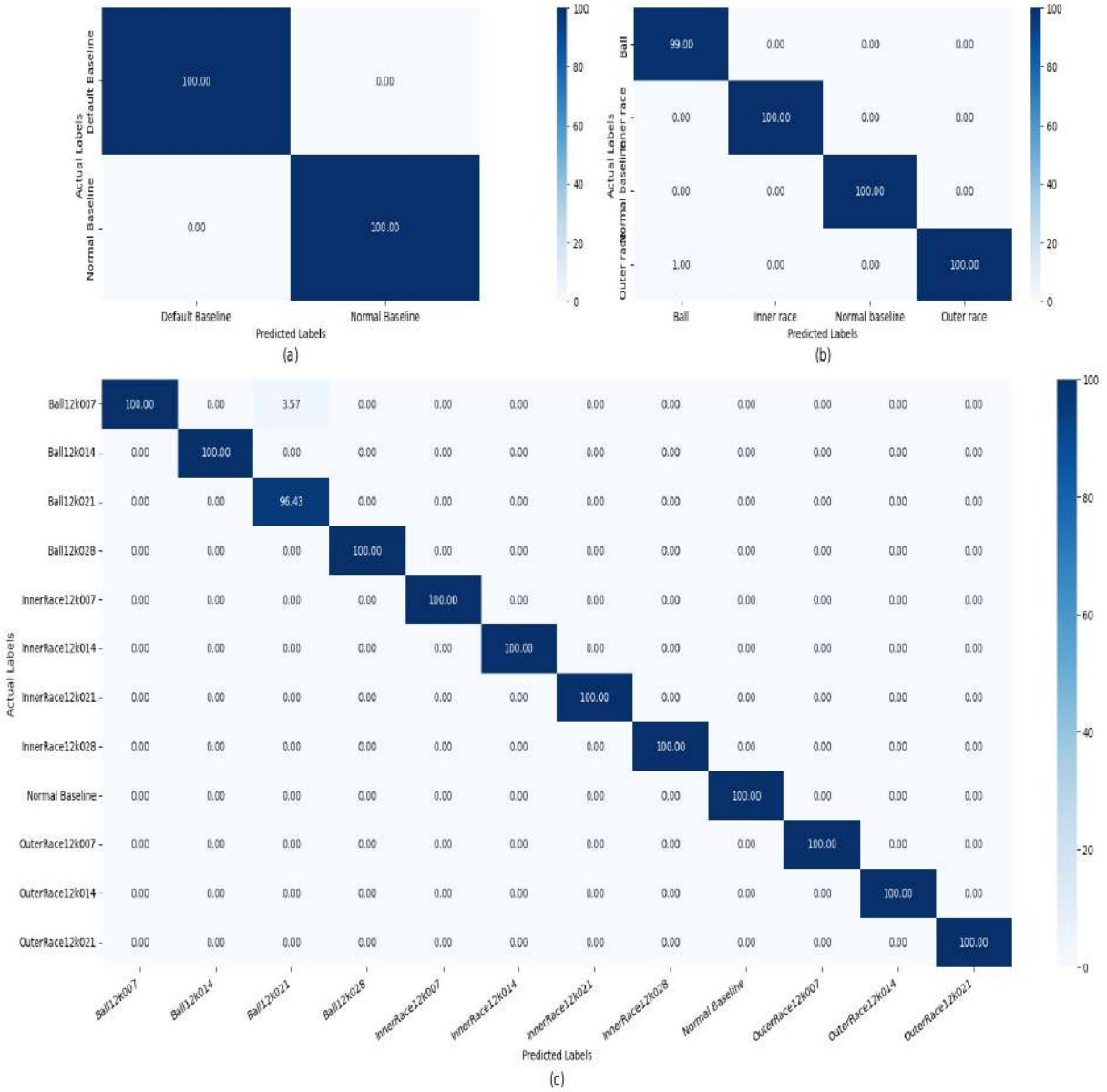


Figure 5.8: Confusion matrices for various classification tasks: (a) Binary classification, (b) Four-class classification, and (c) Twelve-class classification.

evidence of BearNets adaptability.

From the beginning of training, the model demonstrated a rapid improvement in learning efficiency. Training accuracy started at 25.78% but reached 100% by the 10th epoch, maintaining this perfect score through to the 40th epoch. This consistency indicates that BearNet quickly learned the distinct patterns within the MFPT dataset and effectively adapted its parameters for accurate fault classification. A similar trend was observed in the training loss, which declined from an initial value of 1.214 to just 0.0087 by the final epoch, highlighting the models ability to progressively minimise errors.

Validation metrics further confirm this strong generalisation capability. Validation accuracy began at 80.37% in the first epoch and rose to 100% by the sixth epoch, maintaining near-perfect levels thereafter. Likewise, validation loss showed a steady decline, dropping from 0.774 at the start to only 0.0245 by the end of training. These results demonstrate that

the model was not only capable of fitting the training data but also generalised effectively to unseen validation samples, avoiding overfitting.

Overall, the evaluation on the MFPT dataset reinforces the credibility of the BearNet model as a generalisable and dependable framework for bearing fault classification. Its ability to maintain high accuracy and low error rates across datasets highlights its potential for broader industrial applications, beyond the original CWRU environment.

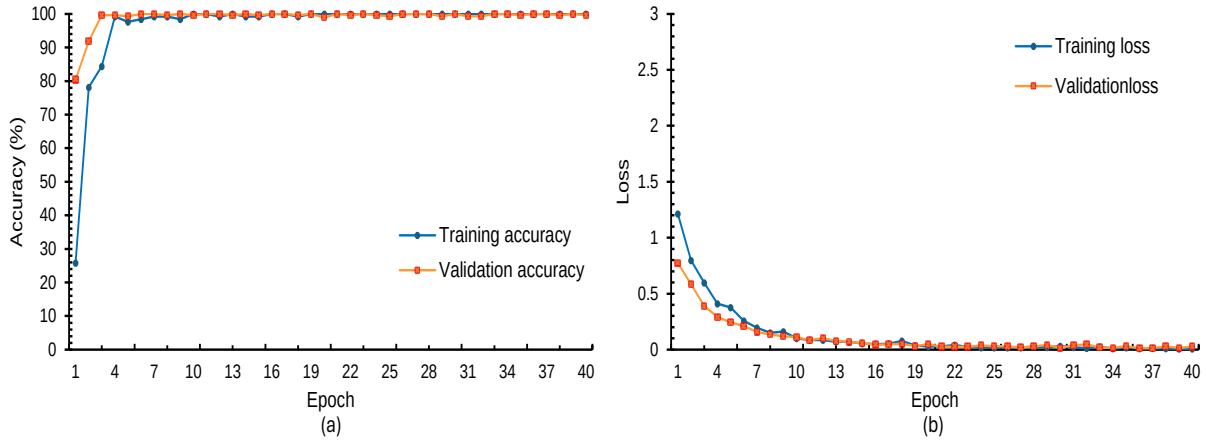


Figure 5.9: BearNet performance with TL_3 stage using the MFPT dataset.

5.9 BearNet's performance comparison with the benchmark model

A direct comparison between the proposed BearNet architecture and the benchmark YAMNet model highlights the effectiveness of DTL strategies across different classification configurations (2, 4, and 12 classes). Validation accuracies and convergence epochs for BearNet under the four DTL stages ($TL_0$$TL_3$) and for YAMNet are summarized in Table 5.5. The table reports the optimal validation accuracy achieved and the number of epochs required for convergence, offering a clear picture of each models efficiency and precision.

BearNet consistently outperforms the benchmark in terms of both accuracy and training efficiency. For instance, in the two-class configuration, validation accuracy rises from 98.44% at TL_0 to 100% at TL_3, while the convergence epochs are reduced from 25 to only 7. A similar trend is observed in the four-class setting, where accuracy improves from 83.90% (TL_0) to 99.75% (TL_3), and convergence epochs decrease from 40 to 27. In the twelve-class case, accuracy increases from 87.67% to 99.70%, while convergence accelerates from 36 to just 25 epochs. These results clearly demonstrate that progressively unfreezing deeper layers enables BearNet to adapt more effectively to task-specific patterns, improving diagnostic precision while reducing training time.

In contrast, YAMNet also achieves high performance but requires significantly more epochs to converge. For example, in the two-class setup, it takes 20 epochs to reach 100% accuracy considerably longer than the 7 epochs needed by BearNet at TL_3. Similarly, for the four-class classification, YAMNet peaks at 98.72% accuracy in 34 epochs, whereas BearNet at

TL_3 achieves 99.75% in only 27 epochs. In the twelve-class configuration, YAMNet achieves 96.90% accuracy after 40 epochs, while BearNet reaches 99.70% within just 25 epochs.

Table 5.5: Epochs of convergence and validation accuracy across various class configurations when validation loss is minimal.

Model	**Two Classes**		**Four Classes**		**Twelve Classes**	
	Accuracy (%)	CE	Accuracy (%)	CE	Accuracy (%)	CE
BearNet TL_0	98.44	25	83.90	40	87.67	36
BearNet TL_1	99.74	11	93.22	38	92.33	32
BearNet TL_2	99.99	8	99.15	28	98.95	26
BearNet TL_3	100	7	99.75	27	99.70	25
YAMNet	100	20	98.72	34	96.90	40

Shorthand: CE = Convergence Epochs.

Overall, the comparison shows that BearNet, particularly under higher fine-tuning stages (TL_2 and TL_3), provides not only higher classification accuracy but also faster convergence than YAMNet. This efficiency is especially advantageous in resource-constrained environments, such as WSN-based monitoring, where computational budgets and time are limited. The ability of BearNet to achieve superior results in fewer epochs makes it a practical solution for real-world predictive maintenance applications that demand both accuracy and rapid deployment.

5.10 Comparing performance with related works

Table 5.6 summarizes a comparison of BearNet with state-of-the-art DL approaches that use similar benchmark datasets, most notably the CWRU and MFPT datasets. While existing methods such as CNN (ECTN) and CNN-SVM report impressive accuracy levels (up to 99.62%), they face challenges such as overfitting, sensitivity to noise, or dependence on specific data distributions.

Table 5.6: Comparison of related DL approaches for bearing fault detection on datasets comparable to those used in this study.

Ref.	**DL Method**	**Dataset**	**Accuracy (%)**	**Comment**
[127]	CNN (ECTN)	CWRU (10 Classes)	99.62	High accuracy but prone to overfitting due to model complexity.
[128]	CNN (FaultNet)	CWRU (10 Classes)	98.50	Performs well but struggles with noisy signals.
[129]	CNN-SVM	CWRU (12 Classes) / MFPT	98.75 / 98.89	Relies heavily on data distribution, limiting generalization.
BearNet (Ours)	YAMNet-TL	CWRU (12 Classes) / MFPT	**99.70 / 100**	Superior generalization with reduced computation cost.

By contrast, BearNet, built upon DTL with YAMNet, achieves 99.70% accuracy on the CWRU dataset (12-class setting) and a perfect 100% accuracy on the MFPT dataset. This performance not only surpasses previously reported accuracies but also demonstrates superior robustness across diverse conditions.

A key advantage of BearNet lies in its efficiency. Thanks to DTL, the model achieves these results with significantly fewer training epochs (ranging from 5 to 27) compared to traditional CNN approaches. This reduction in computational cost is particularly beneficial for real-world applications with limited resources, such as WSNs, where lightweight yet accurate models are essential.

Even modest improvements in accuracy (0.11.2%) over already high-performing baselines are meaningful in industrial diagnostics, as they translate into higher reliability and lower risk of misclassification in critical machinery. More importantly, BearNet demonstrates that models originally designed for a different domain (audio signal recognition) can be effectively adapted to mechanical fault detection, thereby bridging cross-domain knowledge and paving the way for smarter, resource-aware monitoring solutions.

5.11 Conclusion

This chapter has presented a comprehensive overview of BFDs techniques, spanning traditional vibration analysis methods, ML approaches, and modern DL frameworks. By systematically comparing these methods, it becomes clear that while classical techniques provide interpretability and efficiency, they often fall short in handling the complexity of modern industrial datasets. Conversely, DL models, though powerful, require significant computational resources and vast amounts of labeled data.

To bridge this gap, DTL emerges as a compelling solution, enabling the reuse of rich representations from large-scale pre-trained networks and adapting them effectively to domain-specific tasks. The introduction of BearNet, derived from YAMNet, illustrates how audio-based models can be re-purposed to achieve state-of-the-art performance in bearing fault classification, even under complex multi-class conditions. Furthermore, the integration of WSNs and IoT frameworks highlights the practical potential of such systems for real-world predictive maintenance applications.

In conclusion, the discussions in this chapter establish a clear trajectory: the evolution from traditional methods toward intelligent, adaptive, and resource-aware diagnostic models. This progression not only enhances diagnostic accuracy and robustness but also ensures that bearing fault monitoring remains scalable and applicable in increasingly dynamic industrial environments. The next chapter will build upon these findings to detail the experimental validation and analysis of the proposed BearNet system in various fault classification scenarios.

Chapter 6

General conclusion

Contents

VI.1	Overview	114
VI.2	Research Findings	115
VI.3	Future Research	115

6.1 Overview

In this thesis, we bridge the gap between modern deep learning for bearing fault diagnosis (BFD) and the tight compute, memory, and energy limits of wireless sensor networks (WSNs). We introduce **BearNet**, a deep transfer learning approach that adapts a pre-trained **YAMNet** backbone to Mel-spectrograms derived from vibration signals. To reconcile accuracy with embedded constraints, we explore staged fine-tuning (TL0–TL3), striking a practical balance between reusing generic features and learning task-specific cues. Experiments on standard datasets—most notably **CWRU** and **MFPT**—show that BearNet achieves high diagnostic performance while reducing training effort, making it well suited for near-edge deployment in predictive maintenance.

As discussed in Chapter 2, many existing techniques for bearing fault diagnosis (BFD) were reviewed, from traditional signal processing to modern AI-driven approaches. We positioned these methods within their historical and industrial contexts to clarify their relationships and complementary roles. Particular attention was given to vibration analysis foundations, timefrequency representations, and the emergence of data-driven diagnostics supported by wireless sensing and edge/cloud integration.

In Chapter 3, we surveyed the literature and identified the key contributions of traditional BFD techniques alongside recent machine learning advances. We highlighted how time, frequency, and timefrequency-domain features underpin classic pipelines, and how deep models especially CNNs reduce manual feature engineering. We also examined the role of wireless sensor networks (WSNs) and IoT frameworks in enabling continuous, scalable monitoring, and discussed challenges such as data scarcity, domain shift, and deployment constraints in real settings.

Chapter 4 introduced the core concepts and taxonomy of transfer learning (TL), including inductive, transductive, and unsupervised settings; homogeneous vs. heterogeneous feature spaces; and instance, feature, parameter, and relation based transfer. Practical guidance was provided on when to transfer, what to transfer, and how to transfer to avoid negative transfer. We detailed selective freezing and progressive fine tuning strategies, parameter efficient adaptation, and end-to-end workflows tailored to BFD, with special emphasis on reusing audio/spectrogram encoders for vibration signals.

In Chapter 5, we presented the proposed BearNet framework, which adapts a pre-trained audio model (YAMNet) to bearing fault classification via deep transfer learning. A consistent preprocessing pipeline converted vibration signals into Mel spectrograms, and BearNets architecture extended and fine-tuned YAMNet for 2-, 4-, and 12-class tasks. Extensive experiments on the CWRU dataset, and cross-dataset validation on MFPT, demonstrated high accuracy, fast convergence, and strong generalization, outperforming the benchmark and comparing favorably with related state-of-the-art methods.

6.2 Research Findings

This work demonstrates that *deep transfer learning* from an audiotrained backbone can be systematically adapted to vibrationbased bearing fault diagnosis and deployed on lowpower WSN nodes. We proposed a twostage pipeline that converts raw acceleration signals into Melspectrograms and finetunes a compact YAMNETderived architecture (BEARNET) for increasingly demanding tasks: (i) binary fault detection, (ii) fourclass fault localisation (normal, ball, inner race, outer race), and (iii) twelveclass severityaware diagnosis.

Across all settings, staged finetuning (TL_0TL_3) outperformed pure feature extraction. Starting from frozen features (TL_0) provided stability, while progressively unfreezing deeper blocks (TL_1TL_3) yielded consistent gains in accuracy and faster convergence. On CWRU, BEARNET achieved nearperfect validation performance in fewer epochs than the benchmark, indicating that reused timefrequency representations adapt well to mechanical signatures. Confusionmatrix analyses showed clean class boundaries; the few residual errors clustered around neighbouring severities (e.g., adjacent ballfault diameters), suggesting that additional domain cues (speed/order tracking or envelope features) could tighten severity discrimination.

Generalisation was confirmed on MFPT: with the best DTL stage (TL_3), the model converged rapidly and reached perfect validation, highlighting resilience to dataset shift when front end preprocessing is harmonised. Crucially, BEARNET maintained its accuracy with modest latency and energy draw on resourceconstrained WSN hardware an essential requirement for alwayson, edgeside monitoring in Industry 5.0 settings. Compared with related CNN baselines, BEARNET matched or exceeded accuracy while requiring fewer epochs and preserving a lightweight footprint suitable for fleetscale deployment.

Practically, three ingredients proved decisive: (1) aligned timefrequency preprocessing, (2) staged finetuning with small learning rates, and (3) parameterefficient architectural extensions. Together they mitigate negative transfer and enable stateoftheart, dataefficient diagnosis.

6.3 Future Research

Several directions naturally follow from this work. First, hardware-aware optimization should be explored, including quantization and pruning for small embedded devices, together with hardware-in-the-loop profiling of latency, energy, and throughput and dutycycling strategies to extend battery life. Second, more comprehensive domain adaptation is needed to handle broader variations in machines, operating regimes, and sensor mounting; unsupervised or semi-supervised approaches, combined with few-shot transfer, could reduce labeling demands while improving robustness across sites. Third, continual and on-node learning deserves attention so that models can adapt to drift without catastrophic forgetting, ideally using lightweight update mechanisms and privacy-preserving or federated setups that respect industrial data constraints. Fourth, reliability should be strengthened through uncertainty estimation and out-of-distribution detection, coupled with alarm

fusion and fail-safe policies that mitigate false positives in the field. Fifth, interpretability should be improved by linking model evidence on spectrograms to established bearing signatures (BPFO, BPFI, BSF, FTF), thereby producing explanations that are meaningful to maintenance personnel. Finally, diagnosis should be connected to action by pairing BearNet with remaining useful life estimation and costrisk models so that interventions are scheduled within the P–F window and evaluated against plant-level KPIs such as downtime and maintenance cost.

My publications

Journal Articles

- **Sidahmed Lachenani**, Hamza Kheddar, and Mohamed Ould Zmirli. *Advancing Bearing Fault Diagnosis Using Deep Transfer Learning for Wireless Sensor Networks*. *Arabian Journal for Science and Engineering*, 1--21, 2025. Springer. DOI: 10.1007/s13369-025-10387-4.
- Yassine Toumi, Billel Bengherbia, **Sidahmed Lachenani**, and Mohamed Ould Zmirli. *FPGA Implementation of a Bearing Fault Classification System Based on an Envelope Analysis and Artificial Neural Network*. *Arabian Journal for Science and Engineering*, 47(11):13955--13977, 2022. Springer.

International Conferences

- **Sidahmed Lachenani**, Hamza Kheddar, and Mohamed Ould Zmirli. *Improving Pretrained YAMNet for Enhanced Speech Command Detection via Transfer Learning*. In *2024 International Conference on Telecommunications and Intelligent Systems (ICTIS)*, pages 1--6, 2024. IEEE, Djelfa, Algeria.
- **Sid Ahmed Lachenani**, Billel Bengherbia, Mohamed Rebiai, and Mohamed Ould Zmirli. *Optimizing Defect Detection in Ball Bearings: A Robust Signal Filtering Approach with Dynamic FIR Filter Coefficients*. In *10th International Conference on Computational and Experimental Science and Engineering (ICCESEN-2023)*, 27--30 October 2023, Antalya, Turkey.

Bibliography

[1] Robert B Randall and Jerome Antoni. Rolling element bearing diagnosticsa tutorial. *Mechanical systems and signal processing*, 25(2):485--520, 2011.

[2] Román de la Presilla, Sebastian Wandel, Matthias Stammler, Markus Grebe, Gerhard Poll, and Sergei Glavatskih. Oscillating rolling element bearings: A review of tribotesting and analysis approaches. *Tribology International*, 188:108805, 2023.

[3] Tian Ran Lin, Kun Yu, and Jiwen Tan. Condition monitoring and fault diagnosis of roller element bearing. *Bearing Technology*, 2017.

[4] Mohamed Rebiai, Mohamed Ould Zmirli, Billel Bengherbia, and Sid Ahmed Lachenani. Faults diagnosis of rolling-element bearings based on fourier decomposition method and teager energy operator. *Arabian Journal for Science and Engineering*, 48(5):6521--6539, 2023.

[5] Abdelmoughni Toubal, Billel Bengherbia, Mohamed Ould Zmirli, and Abderrezak Guessoum. Fpga implementation of a wireless sensor node with built-in security coprocessors for secured key exchange and data transfer. *Measurement*, 153:107429, 2020.

[6] Guojin Feng, H Zhao, Fengshou Gu, P Needham, and AD Ball. Efficient implementation of envelope analysis on resources limited wireless sensor nodes for accurate bearing fault diagnosis. *Measurement*, 110:307--318, 2017.

[7] Mamoona Majid, Shaista Habib, Abdul Rehman Javed, Muhammad Rizwan, Gautam Srivastava, Thippa Reddy Gadekallu, and Jerry Chun-Wei Lin. Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: A systematic literature review. *Sensors*, 22(6):2087, 2022.

[8] Md Eshrat E Alahi, Arsanchai Sukkuea, Fahmida Wazed Tina, Anindya Nag, Wattanapong Kurdthongmee, Korakot Suwannarat, and Subhas Chandra Mukhopadhyay. Integration of iot-enabled technologies and artificial intelligence (ai) for smart city scenario: Recent advancements and future trends. *Sensors*, 23(11):5206, 2023.

[9] Dheyab Salman Ibrahim, Abdullah Farhan Mahdi, and Qahtan M Yas. Challenges and issues for wireless sensor networks: A survey. *J. Glob. Sci. Res*, 6(1):1079--1097, 2021.

[10] Stuart Russel and Peter Norvig. Artificial intelligencea modern approach 3rd edition. *The Knowledge Engineering Review*, 1:78--79, 2012.

[11] Hamed Nozari, Mohammad Ebrahim Sadeghi, et al. Artificial intelligence and machine learning for real-world problems (a survey). *International Journal of Innovation in Engineering*, 1(3):38--47, 2021.

[12] C Esther Varma and Puja S Prasad. Supervised and unsupervised machine learning approachesa survey. In *ICDSMLA 2021: Proceedings of the 3rd International Conference on Data Science, Machine Learning and Applications*, pages 73--81. Springer, 2023.

[13] Priyanka Rawat, Kamal Deep Singh, Hakima Chaouchi, and Jean Marie Bonnin. Wireless sensor networks: recent developments and potential synergies. *Journal ofSupercomputing, www. researchgate. net/publication/25*, 8165429, 2022.

[14] B Prathiba, K Jaya Sankar, and V Sumalatha. Enhancing the data quality in wireless sensor networksa review. In *2016 international conference on automatic control and dynamic optimization techniques (ICACDOT)*, pages 448--454. IEEE, 2016.

[15] Shengnan Tang, Jingtao Ma, Zhengqi Yan, Yong Zhu, and Boo Cheong Khoo. Deep transfer learning strategy in intelligent fault diagnosis of rotating machinery. *Engineering Applications of Artificial Intelligence*, 134:108678, 2024.

[16] Bin Yang, Yaguo Lei, Xiang Li, and Naipeng Li. Targeted transfer learning through distribution barycenter medium for intelligent fault diagnosis of machines with data decentralization. *Expert Systems with Applications*, 244:122997, 2024.

[17] Ying Lu. *Transfer Learning for Image Classification*. PhD thesis, Université de Lyon, 2017.

[18] Humberto Nuno Teixeira, Isabel Lopes, and Ana Cristina Braga. Condition-based maintenance implementation: a literature review. *Procedia Manufacturing*, 51:228--235, 2020.

[19] Ahad Ali and Abdelhakim Abdelhadi. Condition-based monitoring and maintenance: state of the art review. *Applied Sciences*, 12(2):688, 2022.

[20] Marcus Bengtsson. On communicating and motivating the benefits of a condition based maintenance program in the context of a discrete item manufacturing organization. In *Proceeding of the 1 st International Conference on eMaintenance, Lulea, Sweden*, 2010.

[21] Hongyu Yang, Joseph Mathew, and Lin Ma. Vibration feature extraction techniques for fault diagnosis of rotating machinery: A literature survey. In *Proceedings of the Asia-Pacific Vibration Conference (APVC)*, pages 801--807, 2003.

[22] Guangxi Yan, Jiang Chen, Yu Bai, Chengqing Yu, and Chengming Yu. A survey on fault diagnosis approaches for rolling bearings of railway vehicles. *Processes*, 10(4):724, 2022.

[23] Mohamad Hazwan Mohd Ghazali and Wan Rahiman. Vibration analysis for machine monitoring and diagnosis: A systematic review. *Shock and Vibration*, 2021(1):9469318, 2021.

[24] Dongming Hou, Hongyuan Qi, Honglin Luo, Cuiping Wang, and Jiangtian Yang. Comparative study on the use of acoustic emission and vibration analyses for the bearing fault diagnosis of high-speed trains. *Structural Health Monitoring*, 21(4):1518--1540, 2022.

[25] Nilesh W Nirwan and Hardik B Ramani. Condition monitoring and fault detection in roller bearing used in rolling mill by acoustic emission and vibration analysis. *Materials Today: Proceedings*, 51:344--354, 2022.

[26] Wonho Jung, Seong-Hu Kim, Sung-Hyun Yun, Jaewoong Bae, and Yong-Hwa Park. Vibration, acoustic, temperature, and motor current dataset of rotating machine under varying operating conditions for fault diagnosis. *Data in brief*, 48:109049, 2023.

[27] Julius Niyongabo, Yingjie Zhang, and Jérémie Ndikumagenge. Bearing fault detection and diagnosis based on densely connected convolutional networks. *acta mechanica et automatica*, 16(2):130--135, 2022.

[28] Sally A McInerny and Yi Dai. Basic vibration signal processing for bearing fault detection. *IEEE Transactions on education*, 46(1):149--156, 2003.

[29] Mohamed Rebiai. *Contribution au diagnostic des défauts des roulements par lanalyse vibratoire, basé sur la méthode de décomposition de Fourier et l'opérateur énergétique de Teager*. Phd thesis, University of Médéa, Médéa, Algeria, 2023.

[30] Israel Zamudio-Ramirez, Roque A Osornio-Rios, Jose A Antonino-Daviu, Jonathan Cureño-Osornio, and Juan-Jose Saucedo-Dorantes. Gradual wear diagnosis of outer-race rolling bearing faults through artificial intelligence methods and stray flux signals. *Electronics*, 10(12):1486, 2021.

[31] Feibin Zhang, Jinfeng Huang, Fulei Chu, and Lingli Cui. Mechanism and method for the full-scale quantitative diagnosis of ball bearings with an inner race fault. *Journal of Sound and Vibration*, 488:115641, 2020.

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

[33] Mohamed Rebiai, Billel Bengharbia, Mohamed Maazouz, and Yassine Toumi. FPGA implementation of an improved envelope detection approach for bearing fault diagnosis. *Journal of Engineering and Applied Science*, 71(1):3, 2024.

[34] SPM Instrument. Hd technologies – advanced condition monitoring for industrial machinery. `https://www.spminstrument.it/Tecniche-di-misura/HD-technologies/`. Accessed 2025-07-28.

[35] Xiaobin Miao. Research on fault feature extraction and pattern recognition of rolling bearing. *Mathematical Problems in Engineering*, 2022(1):9040784, 2022.

[36] M Pandiyan and T Narendiranath Babu. Systematic review on fault diagnosis on rolling-element bearing. *Journal of Vibration Engineering & Technologies*, 12(7):8249--8283, 2024.

[37] Yubin Yoo, Hangyeol Jo, and Sang-Woo Ban. Lite and efficient deep learning model for bearing fault diagnosis using the cwru dataset. *Sensors*, 23(6):3157, 2023.

[38] Patrick Nectoux, Rafael Gouriveau, Kamal Medjaher, Emmanuel Ramasso, Brigitte Chebel-Morello, Noureddine Zerhouni, and Christophe Varnier. Pronostia: An experimental platform for bearings accelerated degradation tests. In *IEEE International Conference on Prognostics and Health Management, PHM'12.*, pages 1--8. IEEE Catalog Number: CPF12PHM-CDR, 2012.

[39] Christian Lessmeier, James Kuria Kimotho, Detmar Zimmer, and Walter Sextro. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In *PHM Society European Conference*, volume 3, pages –, 2016.

[40] Wade A Smith and Robert B Randall. Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. *Mechanical systems and signal processing*, 64:100--131, 2015.

[41] Jay Lee, Hongyue Qiu, Gang Yu, Jing Lin, and Rexnord Technical Services. Bearing data set. NASA Ames Prognostics Data Repository – Intelligent Maintenance Systems (IMS), University of Cincinnati, 2007. Dataset accessed 2025-08-15.

[42] Paderborn University, Chair of Design and Drive Technology (KAt). Bearing datacenter: Data sets and download. `https://mb.uni-paderborn.de/kat/forschung/kat-datacenter/bearing-datacenter/data-sets-and-download`. Accessed 2025-08-15.

[43] E. Bechhoefer. Condition based maintenance fault database for testing of diagnostic and prognostics algorithms. Available online, 2023. Accessed: 29-July-2024, URL: https://www.mfpt.org/fault-data-sets/.

[44] Awatef Benfradj, Adel Thaljaoui, Tarek Moulahi, Rehan Ullah Khan, Abdullatif Alabdulatif, Pascal Lorenz, et al. Integration of artificial intelligence (ai) with sensor networks: Trends, challenges, and future directions. *Journal of King Saud University-Computer and Information Sciences*, 36(1):101892, 2024.

[45] Xiaohan Chen, Rui Yang, Yihao Xue, Mengjie Huang, Roberto Ferrero, and Zidong Wang. Deep transfer learning for bearing fault diagnosis: A systematic review since 2016. *IEEE Transactions on Instrumentation and Measurement*, 72:1--21, 2023.

[46] Shreeram Hudda and K Haribabu. A review on wsn based resource constrained smart iot systems. *Discover Internet of Things*, 5(1):56, 2025.

[47] B Nagaraja Rao. The role of artificial intelligence (ai) in condition monitoring and diagnostic engineering management (comadem): a literature survey. *Am. J. Artif. Intell*, 5(1):17--37, 2021.

[48] Omar AlShorman, Muhammad Irfan, Nordin Saad, D Zhen, Noman Haider, Adam Glowacz, and Ahmad AlShorman. A review of artificial intelligence methods for condition monitoring and fault diagnosis of rolling element bearings for induction motor. *Shock and vibration*, 2020(1):8843759, 2020.

[49] Nader Sawalhi and RB Randall. Vibration response of spalled rolling element bearings: Observations, simulations and signal processing techniques to track the spall size. *Mechanical systems and signal processing*, 25(3):846--870, 2011.

[50] Mitchell Lebold, Katherine McClintic, Robert Campbell, Carl Byington, and Kenneth Maynard. Review of vibration analysis methods for gearbox diagnostics and prognostics. In *Proceedings of the 54th meeting of the society for machinery failure prevention technology*, volume 634, page 16. Citeseer, 2000.

[51] Naresh Tandon and Achintya Choudhury. A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings. *Tribology international*, 32(8):469--480, 1999.

[52] Yubin Yue, Hongjun Wang, and Shenglun Zhang. Mel frequency mapping for intelligent diagnosis of rolling element bearings across different working conditions. *Applied Acoustics*, 220:109944, 2024.

[53] Muhammad Farooq Siddique, Wasim Zaman, Saif Ullah, Muhammad Umar, Faisal Saleem, Dongkoo Shon, Tae Hyun Yoon, Dae-Seung Yoo, and Jong-Myon Kim. Advanced bearing-fault diagnosis and classification using mel-scalograms and fox-optimized ann. *Sensors*, 24(22):7303, 2024.

[54] Tauheed Mian, Anurag Choudhary, Shahab Fatima, and Bijaya Ketan Panigrahi. Mel-spectrogram based approach for fault detection in ball bearing using convolutional neural network. In *Proceedings of the 2023 6th International Conference on Electronics, Communications and Control Engineering*, pages 283--289, 2023.

[55] Tengfei Zhang, Shuyong Liu, and Suai Zhang. Review on fault diagnosis on the rolling bearing. *Journal of Physics: Conference Series*, 1820(1):012107, 2021.

[56] Christian Janiesch, Patrick Zschech, and Kai Heinrich. Machine learning and deep learning. *Electronic markets*, 31(3):685--695, 2021.

[57] Kamran Razzaq and Mahmood Shah. Machine learning and deep learning paradigms: From techniques to practical applications and research frontiers. *Computers*, 14(3):93, 2025.

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

[59] Tian Han, Longwen Zhang, Zhongjun Yin, and Andy CC Tan. Rolling bearing fault diagnosis with combined convolutional neural networks and support vector machine. *Measurement*, 177:109022, 2021.

[60] HS Kumar and Gururaj Upadhyaya. Fault diagnosis of rolling element bearing using continuous wavelet transform and k-nearest neighbour. *Materials today: proceedings*, 92:56--60, 2023.

[61] Samreen Naeem, Aqib Ali, Sania Anam, and Muhammad Munawar Ahmed. An unsupervised machine learning algorithms: Comprehensive review. *International Journal of Computing and Digital Systems*, 2023.

[62] Yanfei Zhang, Yunhao Li, Lingfei Kong, Qingbo Niu, and Yu Bai. Improved dbscan spindle bearing condition monitoring method based on kurtosis and sample entropy. *Machines*, 10(5):363, 2022.

[63] Jingbao Hou, Yunxin Wu, Abdulrahaman Shuaibu Ahmad, Hai Gong, and Lei Liu. A novel rolling bearing fault diagnosis method based on adaptive feature selection and clustering. *Ieee Access*, 9:99756--99767, 2021.

[64] Duy-Tang Hoang and Hee-Jun Kang. A survey on deep learning based bearing fault diagnosis. *Neurocomputing*, 335:327--335, 2019.

[65] Shams Forruque Ahmed, Md Sakib Bin Alam, Maruf Hassan, Mahtabin Rodela Rozbu, Taoseef Ishtiak, Nazifa Rafa, M Mofijur, ABM Shawkat Ali, and Amir H Gandomi. Deep learning modelling techniques: current progress, applications, advantages, and challenges. *Artificial Intelligence Review*, 56(11):13521--13617, 2023.

[66] Alan Turnbull and James Carroll. Cost benefit of implementing advanced monitoring and predictive maintenance strategies for offshore wind farms. *Energies*, 14(16):4922, 2021.

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

[68] P Arun, S Abraham Lincon, and N Prabhakaran. Detection and characterization of bearing faults from the frequency domain features of vibration. *IETE journal of research*, 64(5):634--647, 2018.

[69] Ashutosh Kumar, Prabhakar Sathujoda, and Neelanchali Asija Bhalla. Vibration signal analysis of a rotor-bearing system through wavelet transform and empirical mode decomposition. *IOP Conference Series: Materials Science and Engineering*, 1248(1):012027, 2022.

[70] Yassine Toumi, Billel Bengherbia, Sidahmed Lachenani, and Mohamed Ould Zmirli. Fpga implementation of a bearing fault classification system based on an envelope analysis and artificial neural network. *Arabian Journal for Science and Engineering*, 47(11):13955--13977, 2022.

[71] Syahril Ramadhan Saufi, Zair Asrar Bin Ahmad, Mohd Salman Leong, and Meng Hee Lim. Challenges and opportunities of deep learning models for machinery fault detection and diagnosis: A review. *Ieee Access*, 7:122644--122662, 2019.

[72] Mohammed Hakim, Abdoulhdi A Borhana Omran, Ali Najah Ahmed, Muhannad Al-Waily, and Abdallah Abdellatif. 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*, 14(4):101945, 2023.

[73] D Patil, NL Rane, P Desai, and J Rane. Machine learning and deep learning: Methods, techniques, applications, challenges, and future research opportunities. *Trustworthy Artificial Intelligence in Industry and Society*, pages 28--81, 2024.

[74] Wenyi Wang, John Taylor, and Robert J Rees. Recent advancement of deep learning applications to machine condition monitoring part 1: a critical review. *Acoustics Australia*, 49:207--219, 2021.

[75] Tianci Zhang, Jinglong Chen, Fudong Li, Kaiyu Zhang, Haixin Lv, Shuilong He, and Enyong Xu. Intelligent fault diagnosis of machines with small & imbalanced data: A state-of-the-art review and possible extensions. *ISA transactions*, 119:152--171, 2022.

[76] Yogesh Kumar, Surbhi Gupta, and Williamjeet Singh. A novel deep transfer learning models for recognition of birds sounds in different environment. *Soft Computing*, pages 1--21, 2022.

[77] Hamza Kheddar, Mustapha Hemis, and Yassine Himeur. Automatic speech recognition using advanced deep learning approaches: A survey. *Information Fusion*, page 102422, 2024.

[78] Tomas Kliestik, Elvira Nica, Pavol Durana, and Gheorghe H Popescu. Artificial intelligence-based predictive maintenance, time-sensitive networking, and big data-driven algorithmic decision-making in the economics of industrial internet of things. *Oeconomia Copernicana*, 14(4):1097--1138, 2023.

[79] Shen Zhang, Shibo Zhang, Bingnan Wang, and Thomas G Habetler. Deep learning algorithms for bearing fault diagnostics-a review. In *2019 IEEE 12th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED)*, pages 257--263. IEEE, 2019.

[80] Tao Lu, Fanqianhui Yu, Baokun Han, and Jinrui Wang. A generic intelligent bearing fault diagnosis system using convolutional neural networks with transfer learning. *Ieee Access*, 8:164807--164814, 2020.

[81] Thomas Joseph, U Sudeep, et al. Deep learning based fault diagnosis of rolling bearing using spectrogram and cwt images of vibration data. In *2024 11th International Conference on Advances in Computing and Communications (ICACC)*, pages 1--5. IEEE, 2024.

[82] Bo Liang and Wuwei Feng. Bearing fault diagnosis based on iceemdan deep learning network. *Processes*, 11(8):2440, 2023.

[83] Eugenio Brusa, Cristiana Delprete, and Luigi Gianpio Di Maggio. Deep transfer learning for machine diagnosis: From sound and music recognition to bearing fault detection. *Applied Sciences*, 11(24):11663, 2021.

[84] Di Yu, Haiyue Fu, Yanchen Song, Wenjian Xie, and Zhijie Xie. Deep transfer learning rolling bearing fault diagnosis method based on convolutional neural network feature fusion. *Measurement Science and Technology*, 35(1):015013, 2023.

[85] Ruoyu Li, Yanqiu Pan, Qi Fan, Wei Wang, and Ruling Ren. A bearing fault diagnosis approach based on an improved neural network combined with transfer learning. *Measurement Science and Technology*, 35(9):096103, 2024.

[86] Zhunan Shen, Xiangwei Kong, Liu Cheng, Rengen Wang, and Yunpeng Zhu. Fault diagnosis of the rolling bearing by a multi-task deep learning method based on a classifier generative adversarial network. *Sensors*, 24(4):1290, 2024.

[87] Xueyi Li, Kaiyu Su, Daiyou Li, Qiushi He, Zhijie Xie, and Xiangwei Kong. Transfer learning for bearing fault diagnosis: adaptive batch normalization and combined optimization method. *Measurement Science and Technology*, 35(4):046106, 2024.

[88] Jun He, Xiang Li, Yong Chen, Danfeng Chen, Jing Guo, and Yan Zhou. Deep transfer learning method based on 1d-cnn for bearing fault diagnosis. *Shock and Vibration*, 2021(1):6687331, 2021.

[89] Xiao Yu, Youjie Wang, Zhongting Liang, Haidong Shao, Kun Yu, and Wanli Yu. An adaptive domain adaptation method for rolling bearings fault diagnosis fusing deep convolution and self-attention networks. *IEEE Transactions on Instrumentation and Measurement*, 72:1--14, 2023.

[90] Avanish Shimpi, Gunjay Chitr Suhalka, Sameer Sayyad, Satish Kumar, and Arunkumar Bongale. Multi-class bearing fault classification using deep learning and transfer learning models. In *2024 4th Asian Conference on Innovation in Technology (ASIANCON)*, pages 1--5. IEEE, 2024.

[91] Jinyu Tong, Cang Liu, Jinde Zheng, Haiyang Pan, Xiaoyu Wang, and Jiahan Bao. 1d-drsetl: a novel unsupervised transfer learning method for cross-condition fault diagnosis of rolling bearing. *Measurement Science and Technology*, 33(8):085110, 2022.

[92] Tieyang Sun and Jianxiong Gao. New fault diagnosis method for rolling bearings based on improved residual shrinkage network combined with transfer learning. *Sensors*, 24(17):5700, 2024.

[93] Lujia Zhao, Yuling He, Derui Dai, Xiaolong Wang, Honghua Bai, and Weiling Huang. A novel multi-task self-supervised transfer learning framework for cross-machine rolling bearing fault diagnosis. *Electronics*, 13(23):4622, 2024.

[94] Baoquan Hu, Jun Liu, Rongzhen Zhao, Yue Xu, and Tianlong Huo. A new dual-channel convolutional neural network and its application in rolling bearing fault diagnosis. *Measurement Science and Technology*, 35(9):096130, 2024.

[95] Xiaoxin Chang, Shaopu Yang, Shaohua Li, and Xiaohui Gu. Rolling element bearing fault diagnosis based on multi-objective optimized deep auto-encoder. *Measurement Science and Technology*, 35(9):096007, 2024.

[96] Lubin Zhai, Xiufeng Wang, Zeyiwen Si, and Zedong Wang. A deep learning method for bearing cross-domain fault diagnostics based on the standard envelope spectrum. *Sensors*, 24(11):3500, 2024.

[97] Bao Zhu and Chunmeng He. A stable and robust fault diagnosis method for bearing using lightweight batch normalization-free residual network. *Measurement Science and Technology*, 35(10):106115, 2024.

[98] Yingmou Zhu, Hongming Chen, Wei Meng, Qing Xiong, and Yongjian Li. A wide kernel cnn-lstm-based transfer learning method with domain adaptability for rolling bearing fault diagnosis with a small dataset. *Advances in Mechanical Engineering*, 14(11):16878132221135745, 2022.

[99] Zhenyu Yin, Feiqing Zhang, Guangyuan Xu, Guangjie Han, and Yuanguo Bi. Multi-scale rolling bearing fault diagnosis method based on transfer learning. *Applied Sciences*, 14(3):1198, 2024.

[100] Zhiwei Cheng, Xuejiao Li, Guangbin Peng, Yongsheng Deng, Zhihong Xie, and Lian Liu. Transfer deep learning network for rolling bearing fault diagnosis of wind turbines. *Journal of Physics: Conference Series*, 2503(1):012095, 2023.

[101] Qin Zhou, Zuqiang Su, Lanhui Liu, Xiaolin Hu, and Jianhang Yu. Fault diagnosis of rolling bearings based on multi-scale deep subdomain adaptation network. *Journal of Intelligent & Fuzzy Systems*, 43(1):575--585, 2022.

[102] Yan Liu and Kaiyu Fan. Roller bearing fault diagnosis using deep transfer learning and adaptive weighting. In *Journal of Physics: Conference Series*, volume 2467, page 012011. IOP Publishing, 2023.

[103] Muhammad Abid, Izhar ul Haq, Zhiqiang Cai, and Shuai Zhang. Faults identification with deep cnn fine-tuned resnet50 model for rolling bearings. In *IET Conference Proceedings, CP835*, pages 533--539. IET, 2023.

[104] Jianmin Zhou, Xiaotong Yang, and Jiahui Li. Deep residual network combined with transfer learning based fault diagnosis for rolling bearing. *Applied Sciences*, 12(15):7810, 2022.

[105] Fei Shen, Chao Chen, Ruqiang Yan, and Robert X Gao. Bearing fault diagnosis based on svd feature extraction and transfer learning classification. In *2015 prognostics and system health management conference (PHM)*, pages 1--6. IEEE, 2015.

[106] Zhihao Chen, Jian Cen, and Jianbin Xiong. Rolling bearing fault diagnosis using time-frequency analysis and deep transfer convolutional neural network. *IEEE Access*, 8:150248--150261, August 2020. Date of Publication: 14 August 2020.

[107] Minh Tuan Pham, Jong-Myon Kim, and Cheol Hong Kim. Accurate bearing fault diagnosis under variable shaft speed using convolutional neural networks and vibration spectrogram. *Applied Sciences*, 10(18):6385, 2020.

[108] Liqun Hou, Zijing Li, and Huaisheng Qu. Fault diagnosis of rolling bearing using wireless sensor networks and convolutional neural network. *International Association of Online Engineering*, 2020.

[109] Deepak Ganesan, Alberto Cerpa, Wei Ye, Yan Yu, Jerry Zhao, and Deborah Estrin. Networking issues in wireless sensor networks. *Journal of parallel and distributed computing*, 64(7):799--814, 2004.

[110] Mengmeng Song, Zexiong Zhang, Shungen Xiao, Zicheng Xiong, and Mengwei Li. Bearing fault diagnosis method using a spatio-temporal neural network based on feature transfer learning. *Measurement Science and Technology*, 34(1):015119, 2022.

[111] Charles William Bray. Transfer of learning. *Journal of Experimental Psychology*, 11(6):443, 1928.

[112] Mohsen Kaboli. *A Review of Transfer Learning Algorithms: Learning Transferable Knowledge through Deep Learning*. Ph.d. thesis, Technische Universität München, 2017. Accessed August 2025.

[113] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345--1359, 2010.

[114] Jindong Wang and Yiqiang Chen. *Introduction to transfer learning: algorithms and practice*. Springer Nature, 2023.

[115] Mohammadreza Iman, Hamid Reza Arabnia, and Khaled Rasheed. A review of deep transfer learning and recent advancements. *Technologies*, 11(2):40, 2023.

[116] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):9, 2016.

[117] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345--1359, 2009.

[118] Tom M Mitchell. Artificial neural networks. *Machine learning*, 45(81):127, 1997.

[119] Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. *Dive into Deep Learning*. Cambridge University Press, 2023.

[120] Dipanjan (DJ) Sarkar. A comprehensive hands-on guide to transfer learning with real-world applications in deep learning. `https://medium.com/towards-data-science/`

a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applicati
2018. Blog post. Published: 2018-11-14. Accessed: 2025-08-15.

[121] Daniel Bourke. 04. transfer learning with tensorflow part 1: Feature extraction. https://dev.mrdbourke.com/tensorflow-deep-learning/04_transfer_learning_in_tensorflow_part_1_feature_extraction/, 2025. Web article. Accessed: 2025-08-15.

[122] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776--780. IEEE, 2017.

[123] Case Western Reserve University Bearing Data Center. Bearing data center. http://csegroups.case.edu/bearingdatacenter/home, 2000. Accessed 2025-08-21.

[124] Xiao Zhang, Boyang Zhao, and Yun Lin. Machine learning based bearing fault diagnosis using the case western reserve university data: A review. *Ieee Access*, 9:155598--155608, 2021.

[125] Kamel K Mohammed, Eman I Abd El-Latif, Noha Emad El-Sayad, Ashraf Darwish, and Aboul Ella Hassanien. Radio frequency fingerprint-based drone identification and classification using mel spectrograms and pre-trained yamnet neural. *Internet of Things*, 23:100879, 2023.

[126] Amit Ray. Transfer learning: A step-by-step easy guide. https://amitray.com/transfer-learning-step-by-step-guide/, 2025. Web article. Accessed: 2025-08-15.

[127] Wenkai Liu, Zhigang Zhang, Jiarui Zhang, Haixiang Huang, Guocheng Zhang, and Mingda Peng. A novel fault diagnosis method of rolling bearings combining convolutional neural network and transformer. *Electronics*, 12(8):1838, 2023.

[128] Rishikesh Magar, Lalit Ghule, Junhan Li, Yang Zhao, and Amir Barati Farimani. Faultnet: a deep convolutional neural network for bearing fault classification. *IEEE access*, 9:25189--25199, 2021.

[129] Laohu Yuan, Dongshan Lian, Xue Kang, Yuanqiang Chen, and Kejia Zhai. Rolling bearing fault diagnosis based on convolutional neural network and support vector machine. *IEEE Access*, 8:137395--137406, 2020.