Fault Diagnosis of Wind Turbine Gearboxes Based on Transfer Learning

by

Dongdong Wei

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

Doctor of Philosophy

in

Engineering Management

Department of Mechanical Engineering

University of Alberta

Abstract

Operated under changing wind speed and harsh environment conditions, the rotating parts in wind turbine gearboxes, such as gears and bearings, will deteriorate and become faulty over time. By conducting real-time and accurate fault detection and diagnosis before significant failures occur, we can reduce the operation and maintenance costs of wind turbines. This is vital for the economic viability and stability of wind energy.

Vibration analysis based on deep learning technologies has emerged as a promising solution for fault diagnosis. Well-trained deep learning models can process large amounts of sensor data in real time and classify raw vibration signals into labels indicating different fault types, locations, and severity levels. However, these models typically require large amounts of labeled training data and may not generalize well to different working conditions or new fault modes. This limitation arises because these models are developed using the traditional supervised learning paradigm, which assumes a large and complete training dataset.

Other learning paradigms using the idea of transfer learning can be explored to address these limitations by leveraging knowledge gained from one diagnostic task or working environment to improve performance on another related task or environment. Models trained using transfer learning techniques show promise in 1) recognizing different fault classes under variable rotating speeds and load levels with high accuracy, 2) learning fault-discriminative knowledge with size-limited or incomplete dataset, and 3) providing improved

interpretability and trustworthiness in the diagnosis process.

This thesis includes three topics. Topic #1 focuses on the learning paradigm of domain adaptation. A weighted domain adaptation network is proposed to adapt the diagnostic knowledge from multiple labeled datasets to an unlabeled dataset which is collected under a different working condition than those labeled datasets. Domain adversarial training and transfer learning using Maximum Mean Discrepancy are applied to align the learned features from different datasets. Topic #2 studies the open-set recognition (or open-set fault diagnosis) setting and proposes an evidential abstaining classifier that can classify both known faults that are seen in the training dataset and unknown faults which are not included in the training set. Synthetic auxiliary training samples are used to form better features and classification boundaries. Evidential learning is used to better quantify the prediction uncertainty of the model. Topic #3 explores the continual learning paradigm considering the accumulation of data and fault classes through time. A continual learning model is fine-tuned through a sequence of diagnostic tasks each features a different fault class and a different working condition. A task balanced sampling scheme is proposed to select training samples to represent previously learned tasks, and a multi-way domain adaption is conducted to adapt to different working conditions in different tasks.

The novelties explored in this research advance the development of intelligent diagnostic systems for various industrial applications, including wind turbines. The learning paradigms studied in this thesis are useful for building diagnostic systems across multiple life stages of a machine, from the early stages with only a few fault classes to the later stages with many faults to remember. Future research could explore other learning paradigms and advanced models, such as few-shot learning and the Transformer model.

Preface

This thesis is an original work by Dongdong Wei. Parts of this thesis have been published or submitted in journals or conference proceedings. Dongdong Wei was responsible for concept formation, literature review, simulation, methods development, data collection, analysis, manuscript composition, and editing. Dr. Mingjian Zuo and Dr. Zhigang Tian, as the supervisory authors, were involved in conceptualization, reviewing, project administration, funding acquisition, and providing resources. Dr. Te Han and Dr. Fulei Chu are co-authors of some publications. Details of these works are shown below.

Materials presented in Chapter 3 have been published as a conference paper and a journal paper as follows.

- D. Wei, T. Han, F. Chu, and M. J. Zuo, "Adversarial Domain Adaptation for Gear Crack Level Classification Under Variable Load," in *2020 Asia-Pacific International Symposium on Advanced Reliability and Maintenance Modeling (APARM)*, Aug. 2020.

- D. Wei, T. Han, F. Chu, and M. J. Zuo, "Weighted domain adaptation networks for machinery fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 158, p. 107744, Sep. 2021.

Materials presented in Chapter 4 have been revised and re-submitted as a journal paper for possible publication as the following paper.

- D. Wei, Z. Tian, and M. J. Zuo, "Evidential abstaining classifier for

open-set fault diagnosis of rotating machines," in *IEEE Transactions on Industrial Cyber-Physical Systems*, May. 2024.

Materials presented in Chapter 5 have been published as a journal paper.

- D. Wei, M.J. Zuo, and Z. Tian, "Continual learning for fault diagnosis considering variable working conditions," in *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, p.1748006X241252469, Jun. 2024.

Credit is due to my colleagues at the University of Alberta, as well as the coauthors and collaborators at Tsinghua University, who collected the experimental datasets used in this thesis.

To my parents

For bring me to life.

Acknowledgements

A big 'thank you' to my supervisors Dr. Zhigang (Will) Tian and Dr. Mingjian Zuo. Thanks for guiding me thorough this PhD journey. Credit also due to my PhD candidacy committee and PhD defense committee members: Dr. Albert Vette, Dr. Di Niu, Dr. Martin Barczyk, Dr. Rafiq Ahmad, and external examiner Dr. Nan Wu. Also thanks to the chairs for my candidacy exam and my PhD defense: Dr. Ahmed Qureshi and Dr. Tian Tang.

Kudos to my co-authors and collaborators including but not limited to Drs. Te Han, Meng Rao, Xingkai Yang, Yuejian Chen, and Fulei Chu. Special thank to my Master's supervisor, Dr. Kesheng Wang. Appreciation goes to the amazing voluntary reviews and editors who helped polish my work.

Financial support from Future Energy Systems under Canada First Research Excellent Fund (FES-T11-P01, FES-T14-P02, FES-T14-T01), Natural Sciences and Engineering Research Council of Canada (Grant RGPIN-2015-04897), and China Scholarship Council (Grant 201806070147) is acknowledged.

My PhD has been the most exciting venture of my life so far. There were cloudy days and bluebird skies, pain and joy, confusion and clarity. There were office days when we share lunch in the MECE4-8 lounge and there were COVID times when we dressed in pajamas and ties in each other's screens. These will be my precious memories for life.

Finally, and most importantly, this thesis would not have been possible without the unlimited support and love from my parents and my girlfriend. I love you all.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the era of the 4th Industrial Revolution (Industry 4.0), the field of machine fault diagnosis takes on a pivotal role. This chapter provides an overview of the fundamentals, significance, and challenges inherent to this domain. Furthermore, it emphasizes the need for intelligent diagnostic tools, enabled by recent advancements in sensing and data-driven technologies, to unlock the full potential of future cyber-physical systems.

In this opening chapter, Section 1.1 introduces foundational concepts in machine fault diagnosis, wind turbine gearboxes, and deep learning. Following this, Section 1.2 identifies current research gaps and elucidates the motivations driving the research in this thesis. Transitioning to Section 1.3, the research contributions of this thesis are outlined. Finally, Section 1.4 provides an overview of the structure and content of the upcoming chapters.

1.1 Background

1.1.1 Fault Diagnosis

Machines play a vital role in providing us with a comfortable and safe environment, from smooth transportation to dependable energy supplies. However, as machines age, their mechanical components will wear down, leading to faults. In our daily lives, mechanical faults may manifest as jerking transmissions,

unusual noises, oil leakages, etc. Some seemingly minor issues like gear tooth chirp or bearing abrasions may not trigger immediate shutdowns. If left unaddressed, a minor fault can evolve into a malfunction and eventually lead to system failure [1]. To name a catastrophic incident, in 2016, a helicopter carrying 13 personnel crashed due to an undetected fatigue fracture in its main rotor gearbox [2].

Prognostics and Health Management (PHM) is a field focusing on anomaly detection, fault diagnosis, and degradation prognosis for machines and equipment [3]. The goals of PHM include enhancing reliability, reducing maintenance costs, maximizing production availability, and avoiding catastrophic failures [4]. Its significance is especially pronounced in industries such as manufacturing, power generation, and mining, where the smooth operation of machinery is crucial to safety and productivity. Unplanned downtime can cost up to $250,000 per hour in the process manufacturing industry [5] and cost around $647 billion per year for industrial manufacturers across all industry segments [6].

A good maintenance planning strategy is the key to reducing unplanned downtime. Ideally, an industrial asset should be maintained just before it fails to perform its designated functions to maximize availability while minimizing costs. Maintenance planning strategies can be classified into these three categories: run-to-failure maintenance, scheduled maintenance, and condition-based maintenance (CBM) [7], [8]. Figure 1.1 illustrates how the three different strategies work and the potential costs and failures they are exposed to.

Run-to-failure maintenance, also known as breakdown maintenance, is a strategy where the machine is allowed to run until it fails. This approach is only suitable when the costs associated with breakdowns are low. Scheduled maintenance, often referred to as preventive or time-based maintenance in the literature [9], involves regular maintenance activities based on the expected

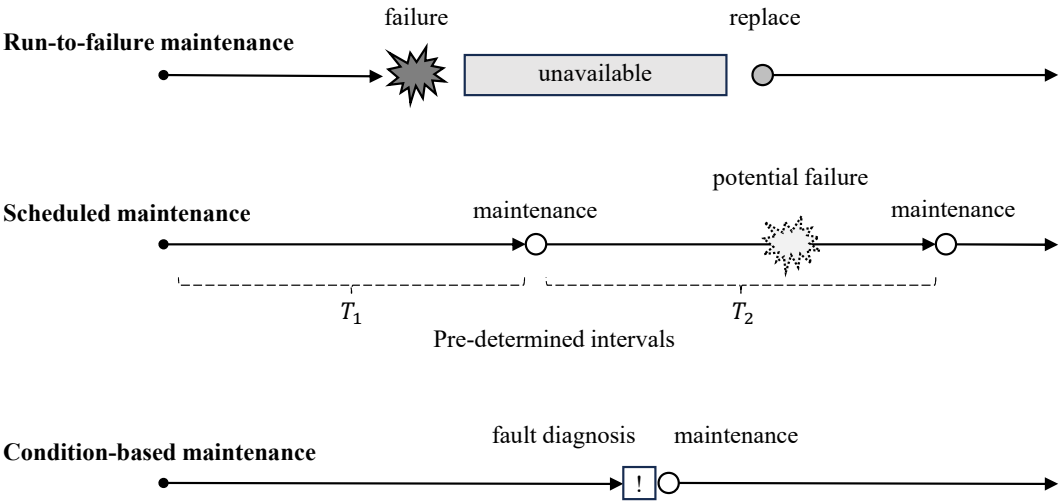


Figure 1.1: Three different maintenance planning strategies.

time between failures, which is calculated using historical failure data or simulation models. Scheduling maintenance too frequently can lead to wasted time and resources, while a loose maintenance schedule increases the risk of failures occurring between maintenance events.

To reduce unnecessary maintenance activities and avoid failure, PHM techniques can be applied to facilitate CBM strategies. A CBM strategy is to plan maintenance based on the health conditions of machines. Maintenance activities can be executed after a fault is detected or, preferably, classified and accessed with a severity level. Fault prognosis methods may also be used to predict the remaining useful life (RUL) [10] before faults occur. These techniques help operators determine the urgency of maintenance and prepare for upcoming activities, thereby reducing maintenance costs.

Fault diagnosis is an indispensable step to the success of CBM. It is the process of identifying the specific fault that has occurred and determining the underlying root causes of undesirable operating status. An effective fault diagnosis system is expected to reduce 50% to 80% of the maintenance cost and increase 20% to 30% of the total production [11]. The key to achieving these benefits lies in accurately detecting and identifying warning signs as

early as possible. Three fundamental requirements should be imposed on fault diagnostic tools:

1. Sensitive detection: the ability to detect faults at their early stages. For example, detect root cracks in gear teeth before the whole transmission train fails.

2. Multi-class inclusion: the ability to recognize multiple fault types, and isolate fault locations. For example, both bearing abrasions and gear tooth cracks located at either stage of a two-stage gearbox should be detected and classified.

3. Accurate recognition: the ability to report all fault occurrences and avoid false alarms. Failing to report a fault may lead to system failure while false alarms waste maintenance work.

The most basic form of fault diagnosis involves field inspections conducted by skilled operators and technicians who rely on their observations, experience, and manual examinations of machines to make diagnoses. However, as machines become increasingly widespread around the world, industries are seeking automated and remote solutions.

In recent decades, various technological advancements have computerized fault diagnosis. On one hand, digital measuring instruments and sensors, such as accelerometers and acoustic emission sensors, are widely deployed in the field to collect firsthand data. On the other hand, various computer programs, including computer-aided models and digital signal processing algorithms, are increasingly being utilized for fault diagnosis. In this modern context, fault diagnosis can be viewed as the process of mapping the sensor's measurements to the fault space [12]. Generally, a fault diagnosis program comprises three essential steps, as illustrated in Figure 1.2).

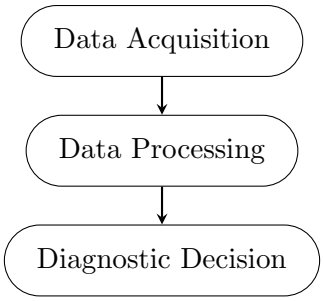


Figure 1.2: Three steps in fault diagnosis.

Data acquisition encompasses the installation of sensors and the collection of condition monitoring data from the machines of interest. It also involves the storage and transfer of this data. A diverse range of measurements, including vibrations, acoustic emissions, currents, flow, speed, pressure, temperature, and more, can be employed to monitor the health condition of these machines. These data can be collected either continuously or intermittently, respectively resulting in permanent or intermittent monitoring methods. The former generates much more data to be processed but can react quickly to sudden faults. In either case, the collected data will be time series.

The most popular measurement used in the field of fault diagnosis is vibration signals [13]. A healthy machine operates with a certain vibration signature. As faults develop, this signature undergoes changes that can be correlated with the fault [9]. Figure 1.3 shows how vibration analysis can serve as one of the earliest tools to detect the degradation of a mechanical component during the process of mechanical failure development [14]. Vibration sensors also offer the advantage of conducting online monitoring without the need for production shutdowns.

In addition to vibration sensors, acoustic emission sensors also possess the capability for online monitoring and demonstrate sensitivity to early faults. However, they require complex mounting involving slip rings and necessitate high sampling frequencies, resulting in much larger datasets to handle when

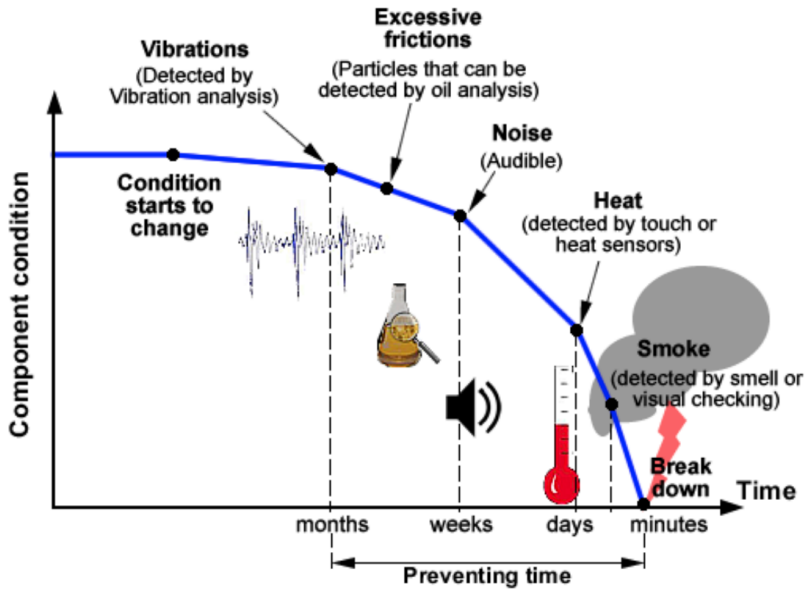


Figure 1.3: Typical development of a mechanical failure [14].

compared to vibration sensors [9]. Oil analysis is another popular method that accesses mechanical wear based on analyzing the content of elements and the viscosity of the oil. However, an oil analysis process typically takes a few hours or even days, making it unsuitable for online monitoring. It is also challenged in isolation of different fault types as multiple components may have the same chemical composition. For rotating machines such as gearboxes, to track their working conditions and to aid in extracting fault-indicative information, tachometers and torque sensors are also commonly used to monitor the rotating speeds and load levels. The research works in this thesis are based on commonly accessible measurements including vibration, rotating speed, and load level.

In many research studies, laboratory data collected from experimental test rigs are often used to study the behaviors of machines and to test diagnostic methods. For example, the Case Western Reserve University (CWRU) bearing dataset has been widely used in many research studies including refs. [15], [16]. To obtain experimental datasets, electrical discharge machining is often

employed to create artificial faulty components, such as bearings with pitted outer races and gears with cracked teeth. The goal is to mimic the natural faults and to recreate mechanical behaviors such as meshing stiffness changes [17]. However, these artificial faults may fail to capture some details, such as the crack closure phenomenon, due to the limitation of machining precision. The machined faulty component is then installed into the test rig to simulate specific health conditions of the machine. This approach allows for the collection of data representing different machine health conditions under various working conditions by design. Experiment datasets presented in Chapters 3, 4, and 5 are all collected using seeded artificial faults.

Data processing is the main research question in the field of fault diagnosis. Decoding complex patterns of sensor data and mining fault-related information out of machines' noisy working environments can be challenging. It often involves signal processing, feature extraction and selection, and machine learning techniques. They are also usually categorized into physics-based, data-driven, and hybrid methods. A review of existing data processing methods for vibration-based fault diagnosis will be presented as Section 2.1 in Chapter 2.

1.1.2 Wind turbine gearboxes

In recent years, the world has seen significant advancements in response to climate change and global greenhouse gas emission reduction policies. Wind turbines (WTs) are being installed all over the globe to harness the kinetic energy of the wind and convert it into electricity. As of 2022, wind energy accounted for 10.2% of the electricity generated in the United States, and more turbines and wind farms are being constructed to increase this percentage [18]. In Canada, wind energy capacity grew by 7.1% in the same year [19], and according to the Canada Energy Regulator, it is expected to increase nine-fold

from its current levels in the Global Net-zero Scenario [20].

A turbine is a complex electro-mechanical system consisting of rotor blades, a hub, a tower, and a nacelle that encloses its drive train. As shown in Figure 1.4, the drive-train typically consists of the rotor, main bearing, main shaft, gearbox, and generator.

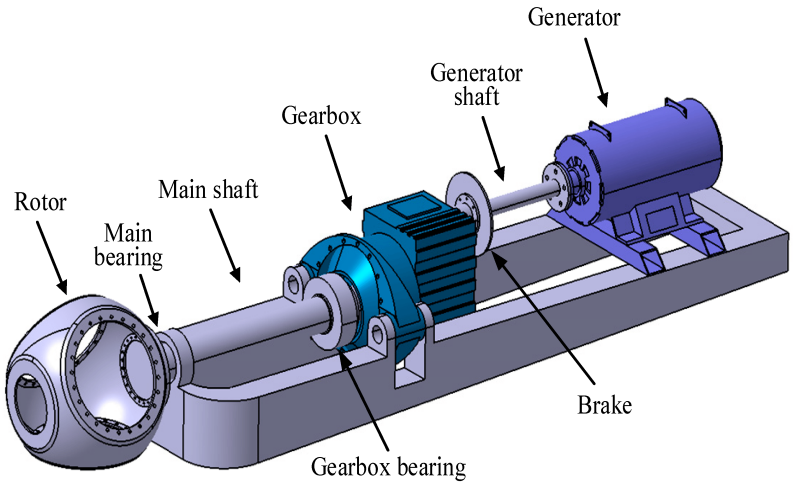


Figure 1.4: A typical drive-train and its components in a wind turbine [21].

The gearbox plays a crucial role in wind turbines by converting the low-speed, high-torque rotation of the rotor into high-speed rotation to drive the generator. Multi-stage gear transmissions, including both fixed-axis gearboxes and planetary gearing (also known as epicyclic gearing), are commonly used to achieve high gear ratios, resulting in complex and compact gearbox structures [22]. These gearboxes are subjected to all the stresses and vibrations generated by the wind, turbine-side components, and fluctuations in generator load. Moreover, wind turbine gearboxes must adapt to variable working conditions based on changes in wind speed and electricity demand. As a result, the gearbox is often the weakest link of the WT drive train and typically fails within 5 years, significantly shorter than the design lifetime of a WT, which is usually around 20 years [23]. To make it worse, the downtime and maintenance costs associated with gearbox failure are substantial [24]. Therefore, monitoring the

health of WT gearboxes is crucial for the efficient operation of wind farms.

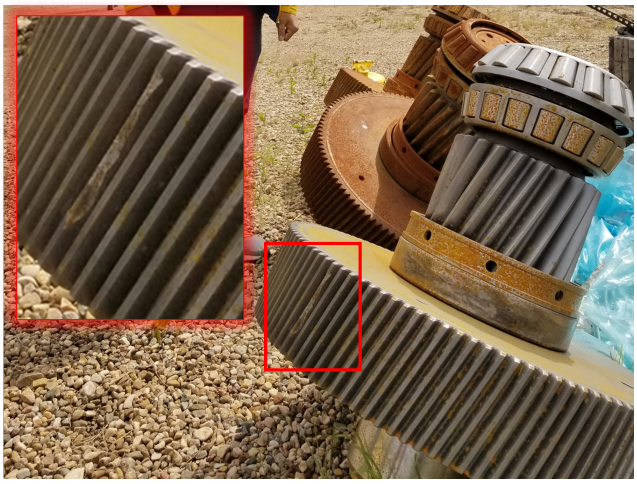


Figure 1.5: Example faulty gear with a broken tooth in Halkirk wind farm, Alberta, Canada. Photo taken by the author in 2019.

Bearings and gears are the two of the most common elements to develop faults in a gearbox. In 2014, the U.S. National Renewable Energy Laboratory reported that about 70% of the WT gearbox failures are cause by the bearings and 26% are caused by gears [25]. Fatigue cracking, macropitting, micropitting, corrosion, and scuffing are the typical failure modes for bearings and gears [26]. Figure 1.5 shows an example faulty gear that has a broken tooth due to bending fatigue in Halkirk wind farm, Alberta, Canada.

Many studies have been conducted to detect early faults, such as gear tooth crack [27] and bearing race defects [28]. Faults are often classified based their location and size. For example, the CWRU bearing dataset is commonly described as having nine different fault classes for the drive end bearing, including three sizes of defects at the outer race, three sizes of defects at the inner race, and three sizes of defects at the roller.

Detecting incipient faults, such as cracking in a gear tooth and pitting in bearing races, within complex mechanical systems, particularly (WT) gearboxes, presents a significant challenge due to several factors:

1. Weak vibration signatures: The vibrations produced by early faults are often very faint compared to other signal components, such as those generated by gear meshing, making their detection challenging.

2. Non-stationary vibration signals: Vibration signals from WT gearboxes exhibit non-stationary behavior, complicating their modeling and analysis [29]. This complexity is made worse by the time-varying working conditions of the gearboxes.

3. Complicated signal transmission path: Fault-induced vibrations undergo damping and modulation as they travel through the signal transmission path from the location of the fault to the sensor. In the case of WT gearboxes, which typically involve multi-stage and planetary gearing, the transmission path is intricate and time-varying [22], [30].

4. Sensor-related challenges: Sensors, including accelerometers used in Wind Turbines, are susceptible to various sources of noise. This noise can arise from electromagnetic interference, temperature variations, and the impact of raindrops.

These factors collectively contribute to a low signal-to-noise ratio, making the early detection of faults in WT gearboxes a complex and intricate task that requires specialized techniques and methodologies. Many research advances have been made revolving around the diagnostic problem of WT gearboxes, including the development of dynamic modeling [17], signal models [31], signal processing [22], feature extraction and selection [32], machine learning [33], and deep learning [34]. A detailed review will be presented in Section 2.1, Chapter 2.

1.1.3 Deep learning

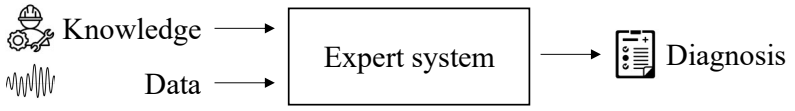
The aspiration to create intelligent machines has been a longstanding dream for inventors. It traces back to Alan Turing's seminal question in 1950, 'Can machines think?' [35]. Artificial Intelligence (AI) has undergone transformative developments since then. Particularly in the last decade, there has been a significant surge in AI advancements driven by deep learning research [36]. Deep learning has found extensive applications in our daily lives, including autonomous driving [37], image recognition [38], and interactive chatbot [39].

Intelligent Fault Diagnosis (IFD) is the application of artificial intelligence to fault detection and diagnosis [40]. In the era of Industry 4.0, marked by the increased deployment and connectivity of sensors through the Industrial Internet of Things (IIoT), the popularity of IFD is on the rise.

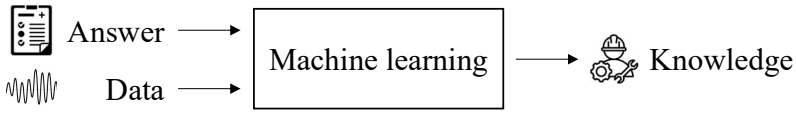
Typically, IFD systems follow a common approach. Initially, these systems extract features from high-dimensional raw vibration signals. Subsequently, the dimensionality of the feature space is further reduced by selecting features that are particularly discriminative for faults. Finally, diagnostic decisions are made based on the analysis of the selected features. This process involves a systematic extraction and refinement of relevant information from the initial vibration signals to enhance the accuracy and efficiency of fault diagnosis.

In the realm of IFD, the term 'knowledge' is often employed to encapsulate the computational processes and rules involved in translating raw data into the fault space [41]. The expert system illustrated in Figure 1.6a, as an approach to IFD, aims to emulate human diagnosticians based on specific rules set by experts [42]. As the machines to diagnose become more and more complicated, it becomes cumbersome to exhaust and implement all the rules required. However, as the complexity of the machines to be diagnosed increases, it becomes challenging to exhaustively define and implement all the necessary rules. In

contrast, Machine Learning (ML) shown in Figure1.6b leverages algorithms to learn diagnostic rules and knowledge directly from data, reducing the reliance on explicit expert knowledge [43].



(a) Expert system.



(b) Machine learning.

Figure 1.6: Comparison of expert system and machine learning in fault diagnosis.

ML models undergo training, validation, and testing stages before deployment for real applications. The available data are typically divided into a training set, a validation set, and a testing set. During the training stage, an optimization algorithm, such as Stochastic Gradient Descent (SGD) or Adam [44], will minimize a predefined cost function (e.g., cross-entropy loss [45]) by adjusting model parameters (e.g. weights and biases in ANNs). Multiple models are often trained to find the optimal model design and training setting. Hyperparameters, variables related to model design and training settings, play a crucial role. They are tuned to maximize accuracy or minimize the loss function on the validation set. Tuning can be conducted based on experience, theoretical analysis, or algorithms like grid search or Bayesian search [46]. Once the hyperparameters are tuned, and the model parameters trained, the model proceeds to the testing stage, where its performance (typically test accuracy) is evaluated.

For ML models, there are different learning paradigms (or learning settings) given different availabilities of data and answers (labels). If both the

data and labels are available during the training stage, the model undergoes supervised learning. This is one of the easiest and most commonly studied learning settings, and the resulting models are often reliable. If the labels are not available, learning must be conducted in the more challenging unsupervised setting, leading to less reliable models. If the training set is partly labeled, the learning goes semi-supervised. Most existing IFD studies are focused on supervised learning and semi-supervised learning but few success has been found using unsupervised learning for fault diagnosis. A more detailed discussion of learning paradigms will be presented in Section 2.2.1.

Regression and classification are fundamental tasks in the field of ML. Regression focuses on predicting continuous numerical outputs, while classification involves assigning discrete labels or categories to input data. For fault prognosis or RUL prediction, regression is often used. In the context of fault diagnosis, the typical formulation is a classification problem. The objective is to detect and identify which fault has occurred. Detecting the presence of a fault or anomaly constitutes a binary classification problem, while identifying different fault classes involves multi-class classification.

Various classification models including K-Nearest Neighbors (KNN) [47], Support Vector Machines [48], and Artificial Neural Networks (ANN) [16] have been used for fault diagnosis. Most ML models still rely on human experts to design and select relevant features as their input. In fault diagnosis, numerous research studies have focused on signal processing techniques, feature extraction, and selection methods [22], [32]. Various mathematical and computational models have been developed to help us understand the behavior of faulty machines and assist in the design of features [17], [31]. Models that use manually crafted features as input are commonly called shallow models in the literature [16]. Dealing with high-dimensional data, such as raw vibration signals, poses a challenge for these shallow models.

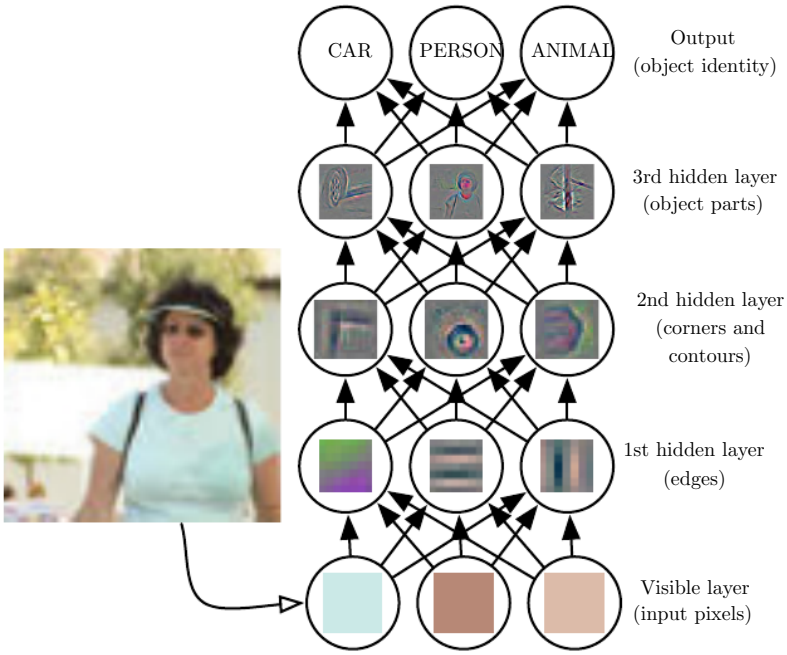


Figure 1.7: Illustration of a deep learning model [45].

Deep learning (DL), as a subset of ML, has emerged as a promising tool for handling high-dimensional data. Its initial success was observed in the field of image recognition, where deep Convolutional Neural Networks (CNN) surpassed human performance in classifying images into different categories [38]. DL models are essentially multi-layer ANNs and are also referred to as deep neural networks (DNNs) in the literature [36]. The term 'deep' is used to describe the high number of layers. The first (lowest) layer is the input layer, and the subsequent hidden layers form representations of the inputs, also known as features. Typically, as illustrated in Figure 1.7, lower layers extract simpler representations such as edges in images [45]. The higher layers build more complex concepts, such as car wheels, by combining the simpler representations formed in the lower layers. With such a hierarchical structure and representations, DL models can extract features from high-dimensional raw inputs without human intervention, enabling end-to-end solutions that directly map raw data to the label space.

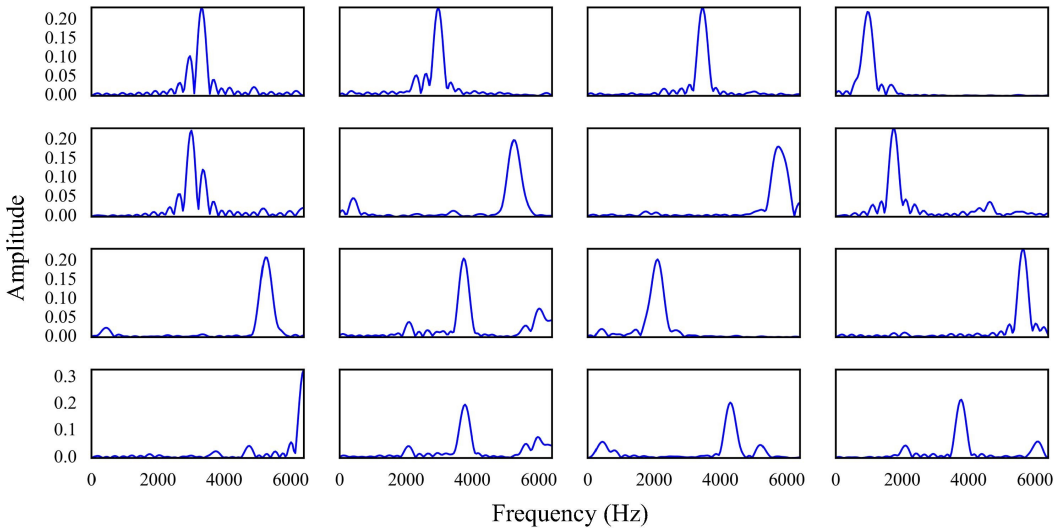


Figure 1.8: Convolutional kernels learned by CNN displayed in frequency domain [51].

In recent years, DL has shown great potential in the fault diagnosis of various industrial assets including gearboxes, induction motors, and electric power transformers [34]. Deep autoencoders (DAE) have reported near-perfect diagnostic accuracy in many standard datasets including the CWRU bearing dataset [16] and the gearbox fault dataset released by the Prognostics and Health Management Society in 2009 [49]. CNNs are proven to be able to learn effective filters, as shown in Figure 1.8, to extract fault-related frequency components from vibration signals. A long short-term memory (LSTM) model was developed to extract rotating speed and diagnose faults in engines, rotors, and gearboxes [50].

1.2 Research motivations

Successful DL models for fault diagnosis typically rely on two key conditions: (1) access to a large amount of labeled training data and (2) powerful computing resources for running the training algorithm. However, meeting these conditions can be challenging in many fault diagnosis applications.

In fault diagnosis applications, a crucial aspect to consider is the presence

of time-varying working conditions and health conditions. Firstly, changes in working conditions, such as variations in rotating speed and load levels, present additional challenges. DL models are often trained with the assumption that the training and testing data have the same data distribution. However, changes in working conditions can induce different modes of mechanical vibrations, leading to a shift in the data distribution to be tested. These shifts in data distribution present the challenge of domain adaptation (DA) in developing IFD models [49]. Secondly, the training set may initially contain only a limited number of fault classes, with more types of faults emerging gradually as machines age. On the one hand, the model needs to recognize new fault classes that are unseen in the training stage. This is termed open-set recognition (OSR) [52] or open-set fault diagnosis (OSFD) [53] in the literature. On the other hand, as more training data become available, the model needs continual learning (CL) ability to update its diagnostic knowledge [54].

Acquiring large amounts of labeled data may be difficult and expensive. Fault-related data can be obtained from field operations or experimental test runs. In real-world scenarios, data becomes faulty when a fault occurs, but it remains unlabeled until the fault is detected and diagnosed. Determining the exact time of fault occurrence and label all the collected data is often challenging. Most existing deep learning models in fault diagnosis are based on on supervised learning and require fully labeled training data. Utilizing unlabeled data presents an opportunity for more efficient solutions and the development of more powerful models in real-world fault diagnosis applications [43].

Training DL models also requires data collected under different faulty conditions. Waiting for different faults to naturally occur for data collection is inefficient. An alternative approach is to create a training dataset in experimental settings, where faults of interest are simulated using experimental test

rigs. Faulty components can be obtained through accelerated life testing or electrical discharge machining. Various working conditions can be tested by controlling rotating speed and load levels. However, it can be expensive to set up experiments that cover all possible faults and working conditions. Models built using test rig data may not perform well when applied to real machines in service due to differences in machine structures, sensor installations, and working conditions [55]. Adaptation towards different machines and working conditions is a critical step in DL-based fault diagnosis.

DL models also face significant challenges when tested with fault types not included in the training set. These challenges manifest in the model failing to accurately recognize the true fault class and potentially providing misleading high-confidence predictions for unseen classes [56]. Misclassification, especially in failing to detect an existing fault, can lead to substantial operation costs. Rather than treating fault diagnosis as a close-set problem, where models are only trained to recognize a fixed set of fault classes, it is essential to explore OSR solutions [52]. In the context of fault diagnosis, this approach is often referred to as OSFD [53]. OSFD requires the model to recognize both the classes seen in the training dataset and indicate the presence of data from other unseen classes. This enables the detection of newly occurring faults.

DL-based methods are known to be computationally intensive. Large memory processing units are usually required to run the training algorithm and large storage space is needed to host the training dataset. To monitor the health condition of machines in remote areas, the system often needs to operate in an edge computing fashion, where computing speed and storage are limited [57]. In common practices, DL models are trained offline with powerful computers and then deployed on the microcomputers in the field. However, to adapt to new fault classes and working conditions unseen in the previous training stage(s), the deployed model must be continuously improved. A DL

model may undergo multiple training stages to assimilate advancements from new data while retaining knowledge acquired from previous training data [58]. As the size of the training dataset grows larger, the computation cost of implementing multiple training stages to keep the model up-to-date can become overwhelming.

In the CL process, the stability-plasticity dilemma poses a significant challenge [59]. The plasticity of a model allows it to form new knowledge efficiently, while stability plays a crucial role in maintaining previously learned knowledge. DNNs, however, are susceptible to the 'catastrophic forgetting' (CF) problem during multi-stage training [60]. They forget the knowledge learned from previous tasks while allocating neurons to learn new knowledge for new tasks. Storing all the historical training data may prevent CF but comes at the cost of substantial data storage and cumbersome model training. An efficient solution for preserving diagnostic knowledge across multiple training stages is to be studied.

Overall, the limited and time-varying availability of training data emphasizes the importance of the designs of training algorithms for DL-based fault diagnosis. Essentially, IFD models are to be developed in many non-conventional settings including semi-supervised learning, DA, OSR, and CL.

Transfer learning (TL) is the core concept revolving around these topics. As shown in Figure 1.9, unlike traditional Machine Learning, which builds separate learning systems for different domains, TL leverages knowledge acquired in source domains to enhance the learning system in a target domain [61]. This concept has been extensively explored in computer vision [62], natural language processing [63], and fault diagnosis [55]. In fault diagnosis, the domains to transfer across may involve different working conditions, various fault types, and more. The target domain is where the diagnostic task is executed, while the source domains contribute relevant knowledge about the task. TL

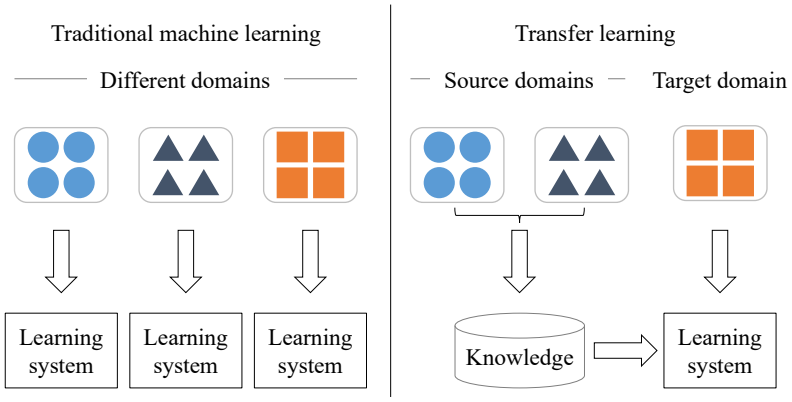


Figure 1.9: Comparison between traditional machine learning and transfer learning.

becomes indispensable when the target domain has a small dataset, lacks data for some classes, or only produces unlabelled data.

1.3 Research topics and contributions

The overall objective of this thesis is to develop accurate classification models for identifying component faults in WT gearboxes, with a focus on designing learning algorithms to address the challenges posed by time-varying working conditions and time-varying health conditions of the machine. The research works are structured around three distinct topics, each introducing a novel TL algorithm to aid model learning under specific problem settings in fault diagnosis. These three stated research topics stem from real-world engineering applications in deploying next-generation PHM systems in the era of Industry 4.0. The primary focuses of each topic are outlined as follows:

Topic #1 addresses the challenge of learning from multiple labeled datasets and a single unlabeled dataset obtained under different working conditions. The model will be tested under target working conditions that produce unlabeled data. This mirrors the real-world scenarios where machines operate in varying conditions and obtaining labeled data for every possible working con-

dition is impractical. The primary focus is on developing diagnostic knowledge that can be efficiently applied in a target working condition where only unlabeled data is accessible. This research aims to enhance the adaptability of fault diagnosis models to the dynamic nature of working conditions in practical industrial settings.

Topic #2 centers on the challenge of recognizing new faults without access to their training data. The training dataset exclusively comprises data from known fault types, while the testing set introduces a novel fault class previously unseen by the model during training. The goal is to develop models capable of utilizing a training dataset with a limited set of fault types to extrapolate and identify unknown or previously unseen faults. This research scenario simulates real-world situations where novel faults may emerge, requiring the model to generalize its understanding and accurately diagnose both the seen and unseen fault types. The focus is on enhancing the model's ability to handle unforeseen faults in practical fault diagnosis applications.

Topic #3 presents the model with the challenge of addressing a sequence of diagnostic tasks involving new fault classes and changing working conditions. A multi-stage training scheme is considered, aiming to update the model each time training data for a new fault type becomes available. This progressive training strategy empowers the model to accumulate and retain diagnostic knowledge, enabling it to recognize new fault classes and adapt to evolving working conditions over time. With such an ability, the model can progressively accumulate and preserve diagnostic knowledge as the fault history of the machine of interest evolves.

Table 1.1 summarizes the differences and connections among the three studied topics. All the topics involve transfer learning across different domains. Topic #1 focuses on adapting to different working conditions, while Topics #2 and #3 deal with variations in fault classes alongside changes in working

Table 1.1: Summary of the three research topics in this thesis.

	Topic #1	Topic #2	Topic #3
Source data	Labeled	Labeled	Labeled
Target data	Unlabeled	Not available	Labeled
Domain differences	Working conditions	Fault classes & working conditions	Fault classes & working conditions
Testing domain(s)	Target	Target	Source & Target

conditions. In Topics #1 and #2, data from the target domain is unlabeled and not available, respectively. For these two topics, the model will only be tested with data from the target domain(s). For Topic #3, although both the source and the target can produce labeled training data, the model must be tested in both the source and target domains. The problem settings of these three topics are termed unsupervised DA, OSR (or OSFD), and CL, respectively.

In general, this thesis aims to contribute to the development of advanced ML algorithms for machine fault diagnosis. By taking a progressive view of the life stages of machines and phases of data availability, multiple novel TL algorithms will be proposed to learn fault diagnostic knowledge more effectively and efficiently. The proposed algorithms will be optimized to make the best use of available data, whether labeled or unlabeled, and data collected from multiple machines and working conditions. They will also be designed to mitigate the influence of changes in working conditions, missing fault categories, and incremental training data.

This research also aims to advance machine learning methods in the PHM context. By investigating transfer learning in relation to working conditions, fault types, and fault levels, the research seeks to uncover meaningful insights into the generalization capabilities of ANNs. Additionaly, considering the physical context of these factors, the study may reveal significant interpreta-

tions regarding the transferability of neural networks.

The expected contributions of the three research topics are as follows:

In topic #1, multi-source DA algorithms will be developed to leverage unlabeled data, addressing the challenges posed by variable working conditions. This research will focus on investigating the appropriate methods for assigning different weights to various source domains during model training. The expected contributions include (1) investigating and proposing effective methods for assigning weights to different source domains during model training and (2) applying statistical metrics, such as Maximum Mean Discrepancy (MMD), to quantify discrepancies between different domains. This metric will facilitate the balanced assignment of weights to diverse source domains and help prevent negative transfer.

Topic #2 aims to enhance OSFD methodologies by developing advanced DL models capable of identifying fault classes not encountered during training. The research focuses on designing effective auxiliary training samples to establish fault-discriminative features and optimized decision boundaries, improving OSFD performance. Additionally, the study addresses uncertainties in diagnostic models by tuning the loss function during DL model training, enhancing the accuracy of classification uncertainty estimation. This research contributes to more robust OSFD methodologies, advancing fault diagnosis under varying working conditions.

In topic #3, the anticipated contributions involve the development of CL algorithms for DL models to manage sequential diagnostic tasks with evolving fault classes and working conditions (domains). Multi-stage training approach will be considered with the goal of systematically enhancing the model's capabilities as it encounters new fault types. This research is expected to contribute efficient CL algorithms that enable the model to assimilate knowledge from new data while retaining knowledge gained from prior training stages. The

result is a more adaptive and efficient fault diagnosis approach that accommodates changes in machine conditions over time.

1.4 Thesis Structure

The thesis adheres to the dissertation requirements outlined by the Faculty of Graduate Studies and Research (FGSR) at the University of Alberta. Organized into six chapters, the thesis is structured as follows:

Chapter 2 of the thesis reviews related works, encompassing both a categorical review of fault diagnostic methods and an overview of deep learning methods. This comprehensive review aims to provide a detailed exploration of the existing literature and methodologies in the field of machine fault diagnosis, particularly in the context of vibration analysis and deep learning techniques.

Chapters 3, 4, and 5 of the thesis focus on three different research topics. These chapters meticulously unfold the methodologies, experimental setups, and findings associated with each research topic, namely unsupervised domain adaptation (Topic #1), open-set fault diagnosis (Topic #2), and continual learning (Topic #3). The detailed exploration within these chapters offers a comprehensive understanding of the unique challenges addressed, the methodologies employed, and the valuable contributions made within each research topic. Chapter 3 has been published as a journal paper in Mechanical Systems and Signal Processing. Chapter 4 has been submitted to IEEE Transactions on Industrial Cyber-Physical Systems for possible publication, and Chapter 5 has been accepted for publication by the Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability.

Chapter 6 concludes the thesis by summarizing key findings and contributions from the previous chapters. It provides a holistic overview of advancements in fault diagnosis through proposed methodologies. The conclu-

sions emphasize the research's significance, addressing real-world challenges and contributing to transfer learning, open-set fault diagnosis, and continual learning in industrial applications. The chapter closes by outlining potential avenues for future research and highlighting the broader impact on machine learning in industrial settings.

Chapter 2

Related works

This chapter starts with a review of fault diagnostic methods including physics-based methods, data-driven methods, and hybrid methods. Then, deep learning technologies including different learning paradigms, models, and training algorithms will be introduced with example applications in the field of machine fault diagnosis.

2.1 Fault diagnosis methods

Three primary categories of fault diagnostic methods are commonly recognized: physics-based, data-driven, and hybrid methods [64], [65].

2.1.1 Physics-based methods

Physics-based fault diagnosis methods are sometimes referred to as model-based methods [3]. It involves the use of mathematical modeling or computer simulation to replicate the physical behavior of the machine under investigation. Diagnosis is then carried out by comparing the actual system behavior with the expected model behavior. In the case of WT gearboxes, the vibration responses from gear meshing are commonly analyzed [24], [31]. Expert knowledge of gear dynamics and signal modulation is involved in physics-based methods. Some simplifications are generally applied to focus on key components and/or emphasize fault symptoms.

The rotating and meshing motions of the gear, on one hand, follow certain rules of dynamics, and on the other hand, result in the amplitude, frequency,

and phase modulation characteristics of its vibration responses. Dynamics-based models essentially decompose the gearbox into many mass-spring-damper systems, while modulation-based models describe the modulation characteristics in the vibration responses [17].

In dynamic modeling, a gearbox system can be either simulated as a lumped mass model (LMM) or a finite element modeling (FEM) [27]. A LMM divides the structure into discrete points, known as lumped masses. The connections between these masses are modeled using springs and dampers, representing the stiffness and damping characteristics of the structure. These characteristics are known to change due to faults, enabling potential diagnoses. For example, ref. [66] proposed a simple one-stage gearbox model and showed that two broken teeth can have a significant impact on the dynamics of the gearbox. Essentially, the broken tooth will cause changes in the meshing stiffness of the gear pair and lead to abnormal vibrations. Based on this model, ref. [67] used statistical indicators and the discrete wavelet transform to identify possible gear tooth crack propagation levels. To deal with more complicated gearboxes and to get more accurate models, ref. [68] proposed a 6-degree-of-freedom (DoF) model that considers four angular rotations and two translations for one-stage gearboxes. Ref. [69] constructed a 26-DoF model for a gearbox with two pairs of gears in mesh, and ref. [70] developed a 21-DoF model for a planetary gearbox. Considering the complex structures of gearboxes, however, all the reported LMMs should be considered over-simplified.

FEM, on the other hand, divides a structure into numerous small elements and considers the dynamic behavior of each element. FEM can deal with complex mechanical structures and can provide more detailed information about stress, strain, and deformation. Ref. [71] developed two FEMs of spur gears to study their non-linear dynamic response. Ref. [72] extended this work to model planetary gear systems. Using FEM techniques, ref. [73] investigated possible crack propagation paths along the root of the tooth. However, FEMs are known to be computationally intensive, especially when modeling complex systems with high mesh density. Considering faults including localised spalling and crack damage, ref. [69] developed FEMs to calculate the torsional

stiffness and tooth load sharing ratio of the gears and integrate these results into a 26-DoF LLM. By using lumped masses to represent the main structures of the system and finite element analysis to determine the detailed changes induced by faults, the dynamic responses of the system can be simulated more efficiently.

Modulation-based modeling is to reconstruct the vibration responses using the understanding of the frequency components of the measured signal. Amplitude, frequency, and phase modulations caused by the meshing of faulty gears can be observed and used to diagnose the fault [69]. Ref. [74], as one of the early works on this topic, pointed out that local gear faults can give rise to frequency components over a very wide range while distributed faults mainly raise the sidebands around the tooth-meshing harmonics. Ref. [75] studied the modulation effects in planetary gearboxes and pointed out that the mounting of the vibration transducers may also have an impact on the frequency component of the measured signals. The relative motions of the sun gear and the planet gears will further complicate this issue [30], [31]. Changes in rotating speed will also introduce amplitude and frequency modulations to vibration signals [76].

Advanced signal processing methods such as time synchronous averaging (TSA) [77], frequency analysis [30], and signal decomposition [68] are often employed to extract key signatures from the vibration signals, facilitating a more interpretative comparison between actual and expected behaviors.

2.1.2 Data-driven methods

Unlike physics-based methods that require expert knowledge of the structure and mechanism of machines, data-driven methods leverage the inherent patterns and information present in the data collected from machines. Few domain knowledge about machines and their faults is needed. These methods use statistical and machine learning techniques to learn patterns directly from historical data. Data-driven methods are also referred to as Intelligent fault diagnosis (IFD) in the literature [43].

Traditionally, a data-driven method includes the steps of signal processing,

feature extraction, feature selection, and, finally, classification/clustering. At first, the vibration signals are typically treated with signal processing techniques such as filtering [77], decomposition [68], auto-correlation [78], and so on, to remove noise and highlight possible fault-related signal components. Then, feature extraction and feature selection are carried out to convert these useful signal components into a low-dimensional feature vector with the key features to identify possible faults [32]. Finally, a clustering or classification model will be constructed to map the selected features into the fault space.

For example, ref. [79] proposed a data-driven method to detect and identify different levels of gear cracks. Firstly, the raw vibration signals are decomposed using wavelet packet transform. Then statistical features such as mean, standard deviation, and kurtosis of the decomposed signals are extracted. Principal Component Analysis (PCA) is used to reduce the number of features to seven. The K-nearest neighbors (KNN) model is finally used to classify the signals into different crack levels based on these seven features.

Other commonly used classification models include Support Vector Machines (SVMs) [48] and Artificial Neural Networks (ANNs) [80]. These models fall under the category of shallow models, relying significantly on hand-crafted features. The selection of features for shallow models typically demands prior knowledge of signal processing and diagnostics. Moreover, different diagnostic tasks may necessitate distinct features, introducing inefficiencies for diverse diagnostic applications.

Recently, more and more studies have adopted deep learning (DL) models to integrate the feature extraction and selection steps with classification, achieving end-to-end fault diagnosis [34], [81]. DL models usually take raw vibration signals or their frequency spectrum as input, and they extract fault-discriminative features automatically during the learning process. Ref. [16] demonstrated the effectiveness of deep autoencoders (DAE) in diagnosing rolling element bearings and planetary gearboxes. Ref. [82] applied convolutional neural networks (CNNs) for bearing and gearboxes diagnosis. Ref. [50] proposed a modified recurrent neural network (RNN) to extract rotating speed and diagnose faults of gearboxes. Several reviews [28], [34], [40], [43],

[81] have been published on DL-based fault diagnosis.

Despite the successes in research, DL-based methods are facing two main challenges in real applications: (1) commonly used supervised training of DL models requires, impractically, a large amount of labeled data, and (2) DL models lack generalization abilities toward new fault classes and domain shift caused by changes in working conditions [49]. In other words, most existing DL models need to be trained with a large labeled dataset that covers all the fault classes and working conditions of interest. In real applications, however, collecting such a complete training dataset can be prohibitively expensive. Training data for some fault classes may only become available after the machine runs into those faulty conditions. To make DL-based fault diagnosis more practical, the following should be considered:

1. Make use of unlabeled data;
2. Identify and recognize testing data that is out-of-distribution (OOD) of the training dataset;
3. Develop an initial model then upgrade it once given new training data.

To deal with the issues related to data availability and changes in working conditions, different learning paradigms such as transfer learning (TL) [83], unsupervised domain adaptation (UDA) [55], and open-set recognition (OSR) [53] have been explored. In Section 2.2.1, a systematic discussion of different learning paradigms in fault diagnosis applications will be presented.

2.1.3 Hybrid methods

Hybrid fault diagnostic methods combine elements of both physics-based and data-driven approaches to harness the advantages of each. They may integrate expert knowledge, mathematical models, and data to achieve fault diagnosis and condition monitoring. By fusing the strengths of both approaches, hybrid methods aim to improve diagnostic accuracy, especially in situations where either pure physics-based or data-driven methods might face challenges.

One of the common ways is to use expert knowledge to guide the feature extraction process for data-driven models. For example, ref. [84] uses expert knowledge to construct a set of residual generators and a fault signature matrix for different fault types of internal combustion engines. Then, data-driven models, i.e. Support Vector Data Description (SVDD) models, are used to evaluate the residual outputs and rank possible fault modes. Similarly, refs. [65], [85] uses Bayesian Network (BN) to provide statistical reasoning based on residual outputs for different fault types. For planetary gearboxes, ref. [86] identifies the fault characteristic frequencies based on their structure. Then it uses Machine Learning (ML) algorithms to determine an optimal bandwidth to capture information near those fault characteristic frequencies, which can be used to diagnose different gear faults including tooth cracks, surface wear, and tooth missing.

Another form of the hybrid method involves using both domain knowledge and data to develop ML models. For bearing fault diagnosis, ref. [87] proposed a physics-based convolutional layer based on spectral kurtosis and envelope analysis to remove the carrier frequencies and only keep the diagnostic information. A bearing fault simulation model described in ref. [88] was used to design the kernels of the physics-based convolutional layer. Similarly, ref. [89] uses both thresholding according to sub-bands of fault characteristic frequencies and CNN models to diagnose bearing faults. Ref. [90] proposed a modal-property-dominant-generated layer and domain-conversion layer based on various signal processing methods including computed order tracking and cepstrum analysis. These studies use both domain knowledge and ML to construct the fault feature extraction part in ML models. The main purpose of applying domain knowledge is still to guide the feature extraction process.

Recently, Physics-Informed Neural Network (PINN) [91] has become a popular hybrid solution for many problems including fluid mechanics [92] and heat transfer [93]. For wind turbine fault prognosis, refs. [94], [95] use PINNs to track the fatigue damage accumulation of main bearings. PINNs are employed to solve and analyze differential equations including the standardized bearing life formula described in ISO 281 [95]. With grease data such as viscosity,

humidity penetration, and foreign particle contamination, PINNs show better prognostic results than pure data-driven methods. However, PINN has not found its applications in fault diagnosis problems as it can be difficult to define differential equations for every fault mode. Fault diagnosis studies discussed in the previous paragraph titled "physics-informed" [89], [90] are not using PINNs.

Simulation-driven machine learning [96] is another type of hybrid method for fault diagnosis, in which simulation signals from dynamic models are used to train ML models. For example, ref. [97] used a bearing signal model to generate training data for SVMs and demonstrated that the trained SVM can successfully classify signals from both the Case Western Reserve University (CWRU) bearing test rig[1] and industrial machines. Ref. [96] adopted a 3 DOF model to generate bearing vibration signals with possible outer or inner race defects. Various ML models including SVM and CNN were trained using the generated signals and then tested with two experiment datasets and data from a 2 MW industrial wind turbine. In ref. [98], a signal model is used to generate training data for a combination of hidden Markov model and gated recurrent unit. Instead of using only simulation data to train ML models, ref. [99] uses both real and simulated data, considering the case that real data for most but a few fault types are available. A bearing model is used to simulate those missing fault types to provide supplementary training data. Ref. [100] examined a hybrid TL setting where simulation and real data constitute the source and target domains respectively. Ref. [101], proposed a TL method where simulated signals with coarse fault labels are used as source domain data while real data with fine labels are used as target domain data. In this work, simulated fault types such as outer race defect and rolling element fault are coarse labels while fine fault labels may include information on defect sizes and damage distribution. It is interesting to note that all the simulation-driven works focus on bearing defects, given that they are easier to simulate compared to other components such as gears.

One of the main benefits of hybrid methods is that they can make more

[1] https://engineering.case.edu/bearingdatacenter

accurate diagnoses given limited data.This shows that domain knowledge can help ML models to learn much more efficiently. This is essential for many fault diagnosis applications as it is difficult or even impossible to obtain large training datasets with all the fault types included [43].

2.2 Deep learning

In recent years, deep learning has been extensively studied in the field of fault diagnosis, and it shows great potential for many industrial applications [28], [34], [43]. The key problems in developing DL models for fault diagnosis include selecting proper learning paradigms based on available training data, designing suitable model structures, and devising efficient training algorithms. Studies related to these three topics will be reviewed in Section 2.2.1, Section 2.2.2, and Section 2.2.3 respectively.

2.2.1 Deep learning paradigms

Supervised learning, unsupervised learning, and reinforcement learning (RL) are the three major ML paradigms [102]. RL is where the models learn to make sequential decisions based on external feedback. In either supervised or unsupervised learning, the ML models are trained to give correct answers (labels) based on input data. Supervised learning is where the ML models learn from fully labeled data, while unsupervised learning is to ask models to find patterns in unlabeled data. There are also many other learning paradigms depending on the problem and available training data. Different learning paradigms can also be blended into new ones, such as UDA, OSR, and continual learning (CL).

Supervised learning is the most studied learning paradigm for fault diagnosis. For reviews of related studies, we refer to refs. [28], [33], [34], [40], [43], [48].

RL is not commonly used to solve classification problems including fault diagnosis. It is typical applications include game playing, robotics, and recommendation systems [103]. For fault diagnosis, ref. [104] converted fault

diagnosis as a game in which the model gets rewarded when it makes a correct diagnosis. With deep RL, the model can learn to recognize bearing faults. However, the true fault labels are still required, and the proposed RL model did not show a significant advantage over supervised deep learning models. RL can also be used to aid the supervised learning of fault diagnostic models. For example, ref. [105] used RL to select training samples from imbalanced training sets. Ref. [106] used RL to search for optimal network structures for fault diagnostic tasks.

Unsupervised learning is one of the approaches to utilize unlabeled data. However, the diagnostic accuracy of unsupervised models is often challenged, especially in complex working conditions. For example, ref. [107], proposed a clustering algorithm named Weighted Euclidean Affinity Propagation (WEAP) to map the features extracted using an unsupervised deep learning network. When tested with the CWRU bearing dataset, although beating other compared unsupervised methods, its accuracy cannot reach 99% without considering changes in working conditions. Supervised learning methods such as ref. [51], however, can easily reach 99.92% with more fault classes considered. Ref. [108] used a self-organized map (SOM) to cluster test samples based on selected features. Their success, however, largely depends on the design of features with the use of physical knowledge of bearing faults. Some works, although titled 'unsupervised learning', still use fault labels for training. For example, in ref. [109] and ref. [110], although the feature extraction process is unsupervised, the classifiers still need to be trained using fault labels. The supervision of labels is the key to learning fault-discriminative features.

Many researchers have considered semi-supervised learning which makes use of both labeled and unlabeled data to develop DL models [111], [112]. Unlabeled data can not only boost the size of the training dataset but also may include information from other domains. A typical setting in fault diagnosis is to learn from both labeled data from a source machine or working condition (known as the source domain) and unlabeled data from a target machine or working condition (known as the target domain). The term UDA is used to describe such a process [113]. This also helps to address the issue of OOD

test data, meaning that DL models can be generalized toward more working conditions. Example works of UDA-based fault diagnosis include refs. [49], [114]–[118].

Dealing with new fault classes unseen during training is also a major challenge for DL models. The term OSR describes the problem setting or learning paradigm when there are more classes in the testing set than in the training set [119]. In fault diagnosis, a derivative term, Open-Set Fault Diagnosis (OSFD), has gained popularity [53]. In OSFD, the training dataset is labeled but missing one or a few fault classes. At the same time, the model needs to classify the samples of the labeled classes and recognize if a sample belongs to the unseen classes. Example works of OSFD include ref. [52], [56], [120]–[123].

CL or Lifelong Learning is to learn a sequence of tasks with different data distributions and label sets [124]. Essentially, CL is to efficiently upgrade the model towards new tasks given new training data. A core challenge in CL is mitigating catastrophic forgetting, which describes the unwanted behavior of DL models forgetting knowledge learned from old data when learning from new data [59], [60]. The model is trained in CL to perform both old and new tasks. For fault diagnostic applications, the task will change given either new working conditions, new machines, or new fault types. For turbofan engine prognosis, ref. [125] deployed DL models with elastic weight consolidation (EWC) to learn from a sequence of datasets collected under different working conditions. EWC was also used to learn from datasets of different machines [126]. To learn new fault classes, ref. [41] proposed an adaptive knowledge transfer method to help DL models diagnose bearing faults with only a few training samples. The final models of the three example studies are on all the tasks before and after working condition changes, across different machines, or additions of new fault classes, respectively.

TL as a learning paradigm is to first pre-train a model with a general dataset and then fine-tune the same model using a more specific task [61], [127]. As surveyed in refs. [43], [128], [129], TL-based fault diagnosis has been a heated topic in recent years. Ideally, the model can form a useful and robust feature extractor for the downstream task during the pre-training stage so

that fine-tuning for a related task can be successful with less training time and training data. As a more general ML term, TL transfers knowledge obtained in a source domain to a target domain [61]. Many learning paradigms including UDA, OSR, and CL can be described as some form of TL with different source and target domains. In fault diagnosis, each working condition or set of fault labels can be defined as one domain and TL is a general idea to help ML models generalize better in real fault diagnosis applications. In UDA, the target domain features a different working condition than the source domain and only contains unlabeled data. For OSR, the label set of the target domain is larger than that of the source domain. As for CL, the model undergoes multiple fine-tuning stages as the target domain continuously expands to include new fault classes and working conditions.

UDA, OSFD, and CL are the three major paradigms studied in this thesis. Many other learning paradigms, such as multitask learning, meta-learning, and curriculum learning, could also be useful for fault diagnostic applications. A short introduction to these paradigms is provided below.

Multi-task learning [130] focuses on learning multiple related tasks at one time. For example, [131] developed a multitask CNN that learns bearing fault diagnosis and localization together. Multi-task learning can be applied only when simultaneous access to training data of different tasks is available.

Meta-learning (or learning to learn) [132] was designed to learn new tasks with only a few samples available (few-shot learning). It trains a meta-model on multiple tasks so that this meta-model can be easily adapted as a specific model for a new related task. Preservation of all the knowledge of the meta-model is not considered when training those specific models.

Curriculum learning [133] is to arrange a guided learning process by using a designed order of tasks for the model. In FDI applications, the arriving order of data is determined by the task itself, and the freedom to arrange a guided training process may not be feasible.

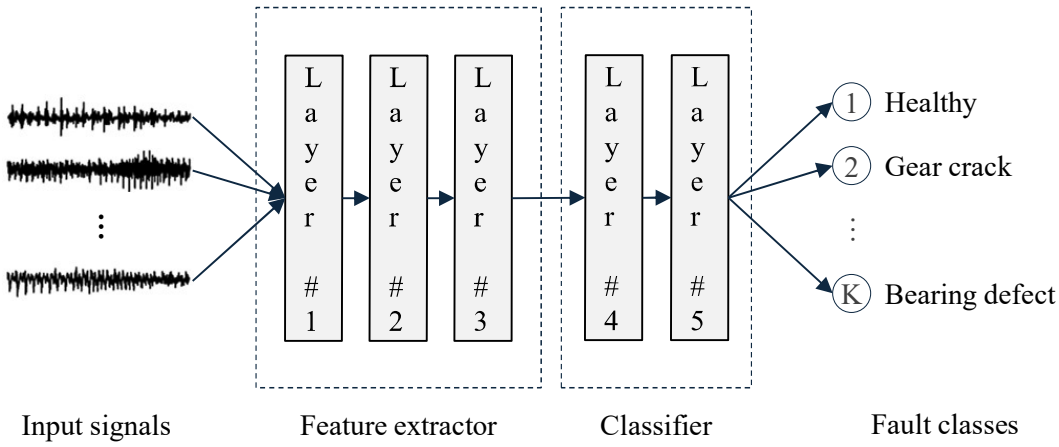


Figure 2.1: A deep learning model consists of a feature extractor and a classifier for fault diagnosis.

2.2.2 Deep learning models

DL models are essentially ANNs in which many artificial neurons with computation capabilities are organized in hierarchical structures with one layer of neurons connecting to the next layer. Technically, an ANN with 3 or more layers can be called a DNN or a DL model [45]. Simple patterns will be extracted from the input data in the lower layers and then fed into higher layers to build more complex and abstract concepts.

To map input vibration signals to target fault labels, DL models automatically learn features indicative of the health state of machines and then derive a classification result based on the learned features. In this regard, a DL model can be split into a feature extractor followed by a classifier as demonstrated in Figure 2.1. In this example, the DL model has five layers, and the outputs of the third layer are regarded as the learned features.

Different types of DL models feature different computational operations and arrangements to extract features from inputs. For example, CNNs are characterized by the convolutional operations between their convolution layers and their input, and RNNs are known for using neurons recursively connected to themselves to accommodate temporal signals. For fault diagnosis applications, determining the structure or architecture of the DL model is vital for accuracy and efficiency. In the following, four types of commonly used DL

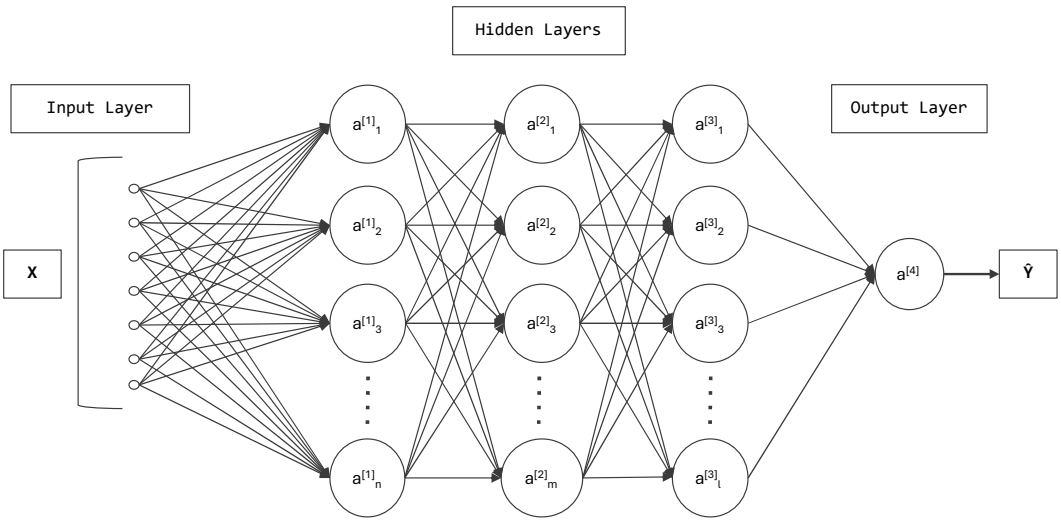


Figure 2.2: An example multilayer perceptron with 3 hidden layers.

models will be introduced and their applications in the field of fault diagnosis will be surveyed.

2.2.2.1 Multilayer perceptron

Multilayer perception (MLP) is the most fundamental ANN structure and has been widely used for fault diagnosis before the popularity of deep learning. As the example shown in Figure 2.2, an MLP may consist of one or more hidden layers (columns) of neurons in between an input layer and an output layer. MLPs are called fully connected neural networks (FCNN) as each neuron is always connected to and affected by all the neurons in its previous layer. For example, in Figure 2.2, the neuron denoted ${a^{[2]}}_1$ in the second hidden layer is connected to all the neurons of the first hidden layer denoted ${a^{[1]}}_i$, $i \in \{1 \dots n\}$.

Each hidden neuron will first execute a weighted summation based on the outputs of its previous layer and the weights connecting to it. Then a non-linear activation function will be applied to the weighted sum. This is, the output of the ith hidden neuron in the $(j+1)$th hidden layer

$$ {a^{[j+1]}}_i = f_{na}(\sum_{k=1}^{n} {a^{[j]}}_k {w^{[j]}}_{ki} + b_j) \tag{2.1} $$

where n is the number of neurons in the jth hidden layer for $j \in \{1, 2, \dots, L\}$ (L is the number of hidden layers), or the number of input features $|x|$ when

$j = 0$. ${w^{[j]}}_{ki}$ denotes the weight of the connection between the kth neuron in the jth layer and the ith neuron in the $(j+1)$th layer. b_j is the bias for the j the layer. All the weights and biases will be automatically optimized during the training process. f_{na} is a non-linear activation function and the commonly used ones are sigmoid, tanh, and ReLU which writes Eqn. 2.2, Eqn. 2.3, and Eqn. 2.4 respectively.

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.3}$$

$$ReLU(x) = \max(0, x) \tag{2.4}$$

The output layer is usually a softmax layer that executes Eqn. 2.5.

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad for\ i = 1, 2, \ldots, K \tag{2.5}$$

where K is the number of target classes. The inputs z_is are called 'logits' and the outputs can be interpreted as the probability of the input signal belonging to the ith class. The class corresponding to the highest will be regarded as the predicted one shown as $\hat{Y}$ in Figure 2.2.

MLPs can model complex non-linear relationships between input and output variables as stated by the universal approximation theorem [134]. They can scale to large datasets and complex architectures, allowing them to handle high-dimensional data and complex tasks. Generally, larger MLPs with more layers and neurons may perform better for complex tasks than smaller networks. However, large models have more parameters to tune and may cause the following problems:

1. expensive computation: large MLPs are neither memory nor time efficient, as the number of weights and calculations needed will grow exponentially as the number of neurons grows.

2. overfitting: large models prone to capture noise and random fluctuations rather than the underlying patterns of the training dataset, resulting in poor generalization to test data. The larger the model is, the more training samples it needs to avoid overfitting.

3. hard to interpret: a large MLP can be regarded as a complex high-dimensional function with numerous parameters. It is difficult to understand how each parameter contributes to the final prediction.

For the above reasons, most fault diagnosis studies do not apply MLP as an 'end-to-end' learning model. Instead of using raw vibration signals as the inputs for MLPs, many studies used designed features extracted from the signals, treating the model as other shallow models such as SVMs. For example, to detect gear tooth wear, ref. [135] used the standard deviation of 16 wavelet packet coefficients of preprocessed vibration signals as the input features for an MLP with a single hidden layer with 20 hidden neurons. To identify rotor cracks, ref. [136] used wavelet packet decomposition (WPD) and empirical mode decomposition (EMD) to extract features as the inputs of a three-layer MLP.

In these studies, the feature extraction processes, not the MLP models, play a central role in identifying faults. In recent review papers including refs. [34], [40], [43], MLP is classified as a traditional machine learning model rather than a DL model. Essentially, MLPs still heavily rely on manual feature extraction and do not provide 'end-to-end' fault diagnostic solutions that automatically map raw or lightly processed vibration signals to target fault classes.

2.2.2.2 Auto-Encoder

An Auto-Encoder (AE) is a neural network with its target output set as its input [137]. As shown in Figure 2.3, an AE contains an encoder that transforms the input into a feature vector and a decoder that reconstructs the input based on the features. The structures of the encoder and the decoder are usually symmetrical and the number of features is usually smaller than the number of dimensions of the input. AEs are trained in an unsupervised or self-supervised fashion by minimizing the difference between the original input and the reconstructed output. In this way, a low-dimensional representation of the input data can be obtained without fault labels.

Moreover, AEs can be stacked together to constitute a Stacked AE (SAE) [138] in which the output of an encoder (extracted features) is used as the input

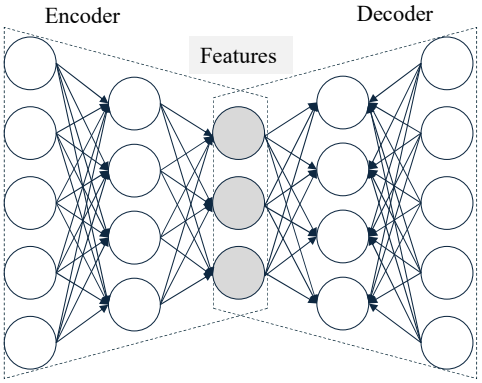


Figure 2.3: An Auto-Encoder with a 3-layer encoder and a 3-layer decoder.

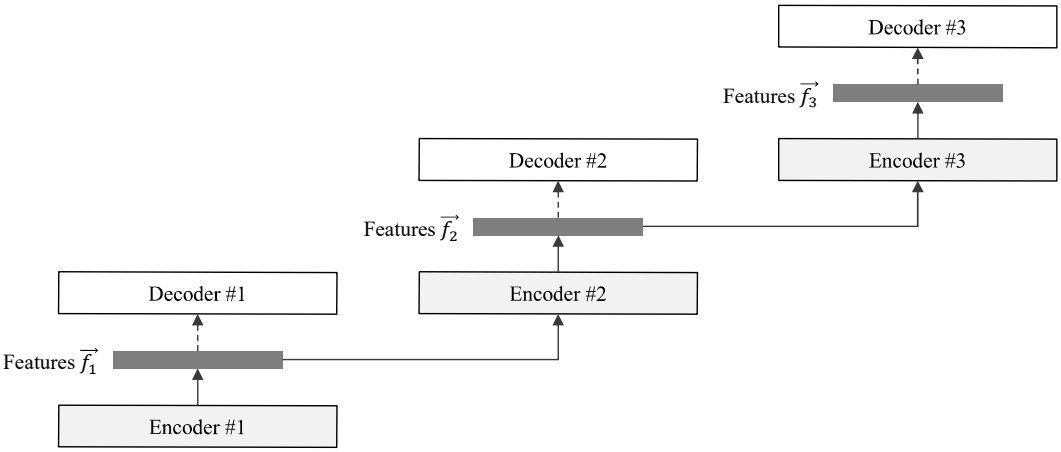


Figure 2.4: A Stacked Auto-Encoder with 3 Auto-Encoders stacked.

of another AE as shown in Figure 2.4. The training of an SAE is executed one AE at a time to mitigate overfitting. Then, after the training of AEs is completed, all the encoders can be stacked together to form a multi-layer feature extractor. Finally, an additional classification layer can be trained to map the extracted features into target class labels.

SAEs with such a layered feature extraction scheme provide the following benefits:

1. Features or representations of input can be learned without labels.

2. Training one AE at a time requires less memory than training large networks with multiple layers.

3. The dimensionality of the input data can be gradually reduced with minimal loss of information.

Many studies have been reported using AE or SAE to diagnose machine faults. Ref. [109] used AE to learn sparse filters for vibration signals. Those learned filters were demonstrated to be useful for bearing fault diagnosis and can be well interpreted in the frequency spectrum. Ref. [16] used SAE to extract features from the frequency spectrum of raw vibration signals. After fine-tuning with labeled data, SAEs are proven effective for diagnosing bearing and planetary gear faults. AEs and SAEs can also take 2-dimensional inputs. Ref. [139] built a two-layer SAE to extract features from the spectrograms of vibration signals to diagnose faults in tidal turbines.

However, AEs use fully connected layers to deal with their inputs and do not consider temporal information of the vibration signals from machines. Fully connected layers also do not support the extraction of multi-scale information which is important for interpreting time series. Converting vibration signals into frequency spectrum as did in ref. [16] or time-frequency spectrograms (ref. [139]) may help the model learn knowledge from vibration frequency, but these methods are subject to limited frequency resolutions and may fail to capture long-term dependencies.

2.2.2.3 Convolutional Neural Network

CNNs are specially designed to process structure grid data such as signals and images. A CNN consists of three main types of layers including convolutional, pooling, and fully connected layers.

Convolutional layers are designed to extract local information from inputs using convolution operations between the inputs and a set of learnable convolutional kernels (also known as filters). For fault diagnosis, the inputs are usually signal segments from one or more sensor channels. For example, signals from vibration sensors mounted for the vertical, horizontal, and axis directions. Each input x will have the dimension of $L \times C$ with L standing for the fixed length of each signal segment and C for the number of channels. Correspondingly, each convolutional kernel will have the dimension of $C \times N$ where N represents the size of the kernel. Given K different convolutional kernels, the output will have K different feature maps corresponding to different kernels.

The jth element of the ith feature map can be calculated as

$$z[i,j] = \sum_{m=1}^{C} \sum_{n=1}^{N} k_i[m,n]x[m, S(j-1)+n] + b_i \tag{2.6}$$

where $k_i \in \mathcal{R}^{K \times N}$ is the ith convolutional kernel, S stands for the stride size of the convolutional kernel, and b_i is the ith bias parameter. The length of the output L_{output} can be calculated as

$$L_{output} = \frac{L_{padded} - N}{S} + 1 \tag{2.7}$$

Note the input x may need to be padded with zeros at the beginning and the end to the size L_{padded}, which ensures L_{output} to be an integer.

Pooling layers are used to downsize feature maps. Given input feature map z_{in}, the ith element of the output of a pooling layer can be calculated as

$$z_{out}[i] = f_{pool}(z_{in}[(i-1)R : (i-1)R + F)]) \tag{2.8}$$

where F is the pooling window size, R is the pooling stride, and $z_{in}[a:b]$ stands for the segment of z_{in} from the ath to the bth element. The pooling function f_{pool} is either the max function for max pooling layers or the average function for average pooling layers. Pooling layers do not have learnable parameters.

A convolutional layer, ReLu activation, and a pooling layer are often sequentially connected to form a convolutional block. A CNN may include multiple convolutional blocks as its feature extractor and a few fully connected layers as its classifier. For example, Figure 2.5 is the first reported CNN structure named LeNet-5 [140]. There are two convolutional layers, two pooling layers, and two fully connected layers in LeNet-5 as noted using the text in the lower part of Figure 2.5.

The design of convolutional filters allows CNNs to focus on local regions and extract temporal patterns of the input. With multiple convolutional blocks stacked together, the high-layer neurons can access the whole input signal/image and learn useful features for the target task. Compared to fully connected layers, convolutional layers have much less amount of parameters. As seen in the example LeNet-5, an image sized 32×32 can be downsized to

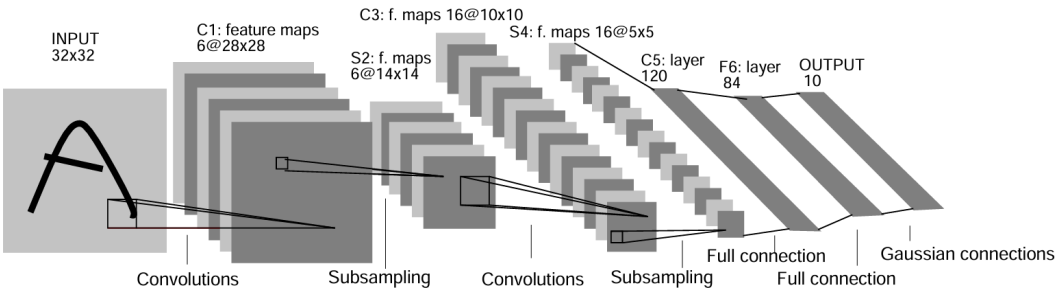


Figure 2.5: A CNN structure named LeNet-5 designed for image classification [140].

5×5 after two convolutional blocks. This will significantly reduce the number of parameters needed in the fully connected layers which not only helps prevent overfitting but also increase computation efficiency.

As CNNs are more often used and well-studied for image recognition, many fault diagnostic studies opted to convert vibration signals into images and apply 2-dimensional CNN (2DCNN). For bearing fault diagnosis, ref. [141] used wavelet package transformation to convert acoustic emission signals into 2D time-frequency representations before applying a LeNet-5 inspired 2DCNN. Ref. [142] used continuous wavelet transform scalogram (CWTS) of vibration signals to diagnose rotor faults in large rotating machines. However, these advanced signal-processing technologies may require experts to calibrate and excessive computation resources. Ref. [143] simply stacked short segments of vibration signals into a 2D matrix as the input for their CNN which reported accurate diagnostic results for rotor faults.

For 1-dimensional input including vibration signals, 1-dimensional CNN (1DCNN) with 1-dimensional convolutional kernels and pooling windows can be directly applied. For example, ref. [144] used a 1DCNN with 3 convolutional layers to detect motor faults based on down-sampled current signals. Ref. [145] used a 1DCNN to extract features from vibration signals and then an SVM to diagnose faults including rotor unbalance and bearing defects. Ref. [146] developed a 1DCNN with a wide first convolutional layer for bearing diagnosis. The wide first convolutional layer was visualized and showed abilities to extract certain frequency bands of the input vibration signals. 1DCNN may be more suitable for vibration-based fault diagnosis.

Regardless of 1D or 2D, CNNs can only take inputs with a fixed size. For high-speed rotating components, a short signal segment may include multiple cycles of their rotation, providing ample fault-related information. However, the segment size needed to capture a full revolution will be larger for low-speed mechanical components. This makes CNN challenged for diagnosis under varying speed conditions. CNNs also have limited capabilities to deal with long-term dependencies in longer vibration signals.

2.2.2.4 Recurrent Neural Network

Recurrent neural networks (RNNs) are a family of neural networks specially designed to process time-series data [147]. They store information from past data points and maintain a state to influence the processing of current data points. Figure 2.6 shows a typical RNN and how it unfolds to multiple time steps. Both the input x, the hidden state a, and the output y are multivariate time series. Symbols with superscript t, i.e. x^t, a^t, and y^t, are used to represent their values at the tth time step. The weight matrices W_{ax}, W_{ya}, and W_{aa} are learnable parameters respectively for converting input to hidden state, hidden state to output, and for updating the hidden state. In original RNNs, a^t and y^t can be calculated using Eqn. 2.9 and Eqn. 2.10 respectively.

$$a^t = tanh(W_{aa}a^{t-1} + W_{ax}x^t) \tag{2.9}$$

$$y^t = W_{ya}a^t \tag{2.10}$$

The number of time steps for input and output in an RNN-family model can be designed to adapt to different applications. For classification tasks such as fault diagnosis [148] and news article classification [149], the inputs have many time steps while the outputs only need one value, suggesting a many-to-one RNN structure. One-to-many RNNs can be useful for image captioning [150] and many-to-many structures have their found applications in machine translation [151]. Ref. [50] designed a many-to-many RNN to extract rotating speed from vibration signal to help diagnose gear and bearing faults

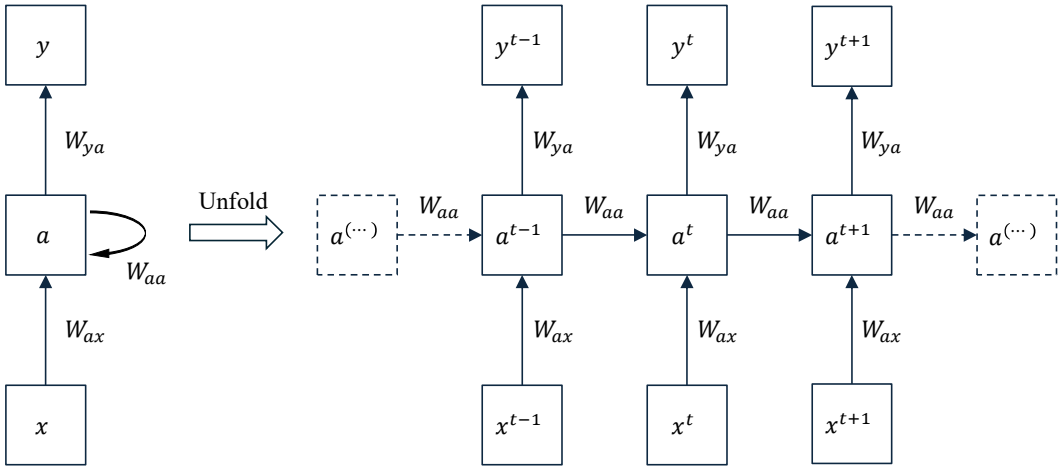


Figure 2.6: A typical recurrent neural network and how it unfolds for multiple time steps.

under varying working conditions. Note that a one-to-one RNN is effectively a traditional MLP.

Original RNNs are challenged to extract long-term relationships from time series and are often bugged by vanishing or exploding gradient problems during training. Long short-term memory (LSTM) units were later introduced to allow RNNs to learn an additional forget gate to provide short-term memories that last thousands of time steps [152]. LSTM also partially solved the gradient vanishing problem. In recent years, ref. [153] proposed a gradient truncation scheme to tackle the gradient exploding problem, and ref. [154] introduced Gated Recurrent Unit (GRU) which is similar to LSTM but uses fewer parameters to reduce overfitting. To learn a hierarchy of time-scales and allow RNNs to model more complex time series, ref. [155] designed a deep RNN structure that contains multiple hidden layers. Bi-directional RNNs were proposed in ref. [156] to incorporate information from future time steps for current predictions.

However, RNN-family models including LSTM, GRU, and Bi-directional RNNs still have difficulties capturing very long-term dependencies [157]. They are also prone to overfitting and require massive amounts of training data. Even provided with a large training dataset, the sequential nature of RNN-family models makes them computationally intensive and incompatible with

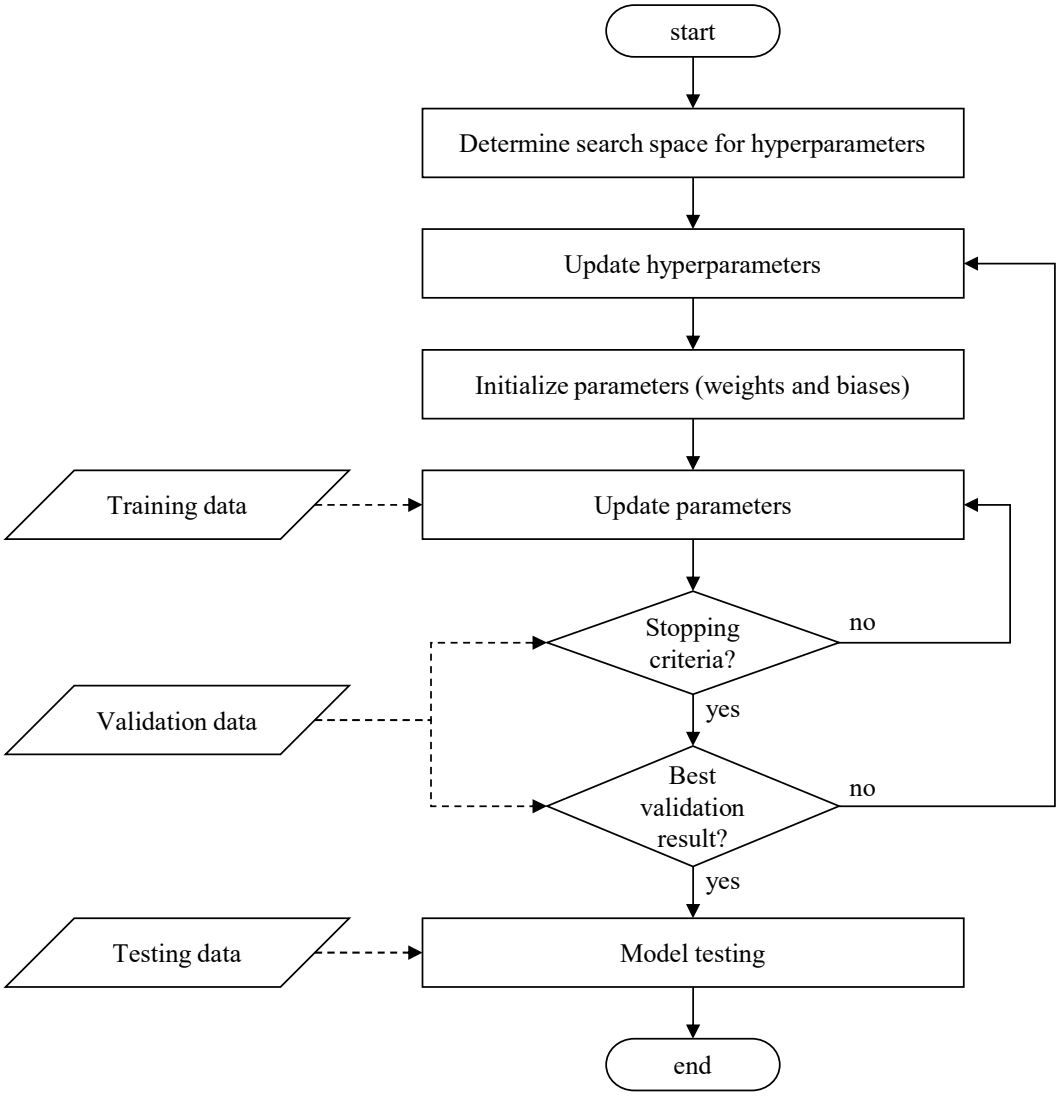


Figure 2.7: The flowchart for DL model training and hyperparameter tuning.

modern advanced parallel computing techniques.

2.2.3 Deep learning algorithms

Training a DL model often involves optimizing two sets of parameters: 1) hyperparameters such as the number of kernels in a convolutional layer and learning rate, and 2) the weights and biases in that DL model. Figure 2.7 shows the general flowchart of how a DL model is trained, validated, and tested.

2.2.3.1 Loss functions

A loss function measures the disparity between the predicted output of a DL model and the actual labels in the training data. Minimizing the loss function is the goal of updating parameters (weights and biases) in DL models. For regression tasks, common loss functions for DL models include Mean Absolute Error (MAE) and Mean Squared Error (MSE) write Eqn. 2.11 and Eqn. 2.12 respectively.

$$L_{MAE} = \frac{1}{N}\sum_{i=1}^{N}|x_i - y_i| \tag{2.11}$$

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{2.12}$$

where x and y are both N-dimensional vectors and x_i denotes the value on the ith dimension of x. For classification tasks, Cross Entropy (CE) loss writes Eqn. 2.13 is often used.

$$L_{CE} = -\sum_{c=1}^{K} y_{o,c}\log(p_{o,c}) \tag{2.13}$$

where K is the number of classes, $y_{o,c}$ is a binary indicator (0 or 1) if class label c is the correct classification for observation o, and $p_{o,c}$ is the predicted probability observation o is of class c. Softmax function (Eqn. 2.5) is often used to calculate the predicted probabilities.

Apart from minimizing the discrepancies between the model predictions and the ground truth, regularization or penalty terms are often used to enforce certain desirable characteristics of the model. Besides, two or more loss functions may be imposed on a model, representing multiple objectives of that model. L1 or Lasso regularization (Eqn. 2.14) can be used to reduce overfitting and promote sparsity in models. Similarly, L2 or Ridge regularization (Eqn. 2.15) helps prevent large weights from dominating the learning process to boost generalization ability.

$$L1 = \sum_{i}|w_i| \tag{2.14}$$

$$L2 = \sum_i w_i^2 \tag{2.15}$$

where w_is are learnable weights in the model.

Maximum Mean Discrepancy (MMD) as written in Eqn. 2.16 is another popular loss or penalty term in DL, especially TL [158], [159]. It computes the difference in mean embeddings of two distributions in a reproducing kernel Hilbert space (RKHS) and can guide the alignment of feature distributions across source and target domains.

$$\text{MMD}^2[\mathcal{F}, \mathcal{G}] = \mathbb{E}_{x,x' \sim \mathcal{F}}[k(x, x')] + \mathbb{E}_{y,y' \sim \mathcal{G}}[k(y, y')] - 2\mathbb{E}_{x \sim \mathcal{F}, y \sim \mathcal{G}}[k(x, y)] \tag{2.16}$$

where $\mathcal{F}$ and $\mathcal{G}$ are probability distributions, $\mathbb{E}$ denotes the expectation operator, and k is a selected kernel function such as polynomial kernel and radial basis function kernel.

Kullback–Leibler divergence (KL-divergence) write Eqn. 2.17 is another popular term for measuring the difference between two probability distributions. It has found application in training various types of DL models including variational autoencoders [160] and Bayesian neural networks [161].

$$D_{KL}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{Q(x)}{P(x)} \tag{2.17}$$

where $P(x)$ and $Q(x)$ represent the two probability distributions and $\mathcal{X}$ denotes the sample space where x lives in.

Finally, the loss function or the training objective of a DL model can be written as a weighted summation of loss terms. The weighting coefficients are often regarded as hyperparameters to be selected and optimized as shown in Figure 2.7. Notably, all the terms in the loss function should be differentiable to suit the use of gradient-based optimization methods (see Section 2.2.3.3).

2.2.3.2 Evaluation metrics

To evaluate the performance of DL models, different metrics can be used. For Regression problems, the MAE and MSE loss terms (see Eqn. 2.11 and Eqn. 2.12) can also be used as evaluation metrics. For classification tasks, however, the commonly used Cross Entropy loss term does not have a clear physical

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Table 2.1: Confusion matrix for binary classification.

meaning, and other metrics are often used. Accuracy is the most commonly used metric for classification tasks and it is simply the proportion of correctly classified samples out of the total tested samples as writes Eqn. 2.18.

$$\text{Accuracy} = \frac{\text{number of correct prediction}}{\text{number of tested samples}} \tag{2.18}$$

For a binary classification problem, a prediction can be either true positive (TP), false positive (FP), false negative (FN), or true negative (TN) as shown in Table 2.1. When dealing with imbalanced datasets with different numbers of positive and negative samples, precision, recall, and F1 score write Eqns. 2.19, 2.20, and 2.21 are often used. Precision is the proportion of true positive predictions among all positive predictions, indicating the model's ability to avoid false positives. Recall or sensitivity is the proportion of true positive predictions among all actual positive instances, indicating the model's ability to capture all positive instances. F1 score is the harmonic mean of precision and recall, providing a balanced view of the two metrics.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.19}$$

$$\text{Precision} = \frac{TP}{TP + FN} \tag{2.20}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.21}$$

2.2.3.3 Optimization methods and stopping criteria

Gradient-based methods are usually used to optimize the parameters (weights and biases) in DL models. Essentially, these algorithms (also known as optimizers) iteratively update the parameters in the direction of the negative gradient of the loss function with respect to the parameters being optimized. This will guide the model towards parameters that produce lower and lower loss values.

The most fundamental gradient-based optimizer is stochastic gradient descent (SGD) and its parameter updating scheme can be written as Eqn. 2.22.

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta; x_i, y_i) \tag{2.22}$$

where θ_t is the parameter at iteration t, η is the learning rate, and $\nabla_\theta L(\theta; x_i, y_i)$ is the gradient of the loss function L with respect to the parameter θ evaluated on a mini-batch of training examples (x_i, y_i). However, DL models often present complex optimization landscapes with many local minima, plateaus, or saddle points, challenging the performance of the vanilla SGD method.

To accelerate the convergence of SGD, a momentum term considering the previous update step can be used as shown in Eqn. 2.23.

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t) + \mu \Delta\theta_{t-1} \tag{2.23}$$

where μ is the momentum coefficient and $\Delta\theta_{t-1}$ is the update vector from the previous time step.

Another reasonable and more popular optimizer is Adam which utilizes both momentum, adaptive learning rate, and scaling [44]. Its calculation can be described using Eqn. 2.24

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{2.24}$$

with

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t, \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \\ \hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, \\ \hat{v}_t &= \frac{v_t}{1 - \beta_2^t} \end{aligned}$$

where η is the learning rate, ϵ is a small constant added to prevent division by zero, m_t and v_t are exponentially decaying moving averages of the first moment (the mean) and the second moment (the uncentered variance) of the gradients, respectively. g_t is the gradient at time step t. $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected estimates of the first and second moments, respectively. β_1 and β_2

are the decay rates for the moving averages of the gradients and the squared gradients, respectively.

For a new DL model, its parameters are often randomly initialized before training. Proper initial parameters can prevent vanishing or exploding gradients. Among many initialization methods, Kaiming initialization [38] has become a widely accepted default option. In Kaiming initialization, weights are randomly sampled out of a Gaussian distribution with a mean of zero and a standard deviation of $\sqrt{2/d_{in}}$, where d_{in} is the dimensionality of the input, and biases are initialized as zeros.

However, training a new model from scratch is rare in many applications including image recognition and natural language processing. Instead of initializing a model with random numbers, the parameters of a pre-trained model can be copied. It is also common to freeze a part of the parameters (typically the ones in lower layers) and only allow the rest to be updated for the target task. For example, ref. [162] pre-trained CNN models with the CWRU bearing dataset, freeze a portion of parameters, and then fine-tune the rest with data from their target testbed. This will allow transfer learning from one dataset to the target one. Using pre-trained models as start points and then fine-tuning them for downstream tasks is more efficient and may lead to better accuracy [163]. For example, the well-known AI chatbots OpenAI ChatGPT[2] based on generative pre-trained transformers (GPT) [164].

The stop or end of the parameter updating can be decided using one or a few stopping criteria. Applying thresholds on validation loss or validation accuracy are two typical stopping methods. A maximum number of epochs or training time can also be enforced. Before the thresholds above, training can also be stopped when the training or validation loss fails to decrease for a specified number of consecutive epochs, suggesting that the model has converged. This is called early stopping in the literature and it can effectively reduce overfitting [45]. As demonstrated in Figure 2.8, lower training error can be achieved using longer training time but this does not improve the model's performance on the validation set [165]. This indicates that the model may

[2]https://openai.com/index/chatgpt/

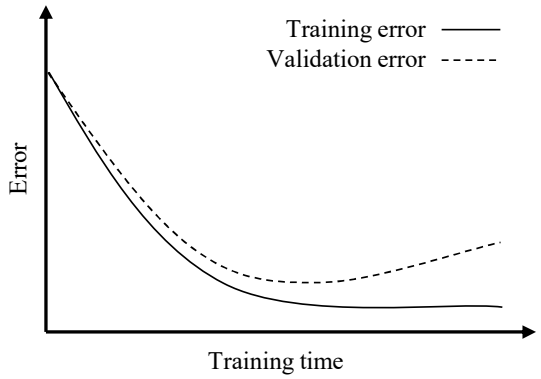


Figure 2.8: Idealized training and validation error curves [165].

be overfitting the training data and early stopping techniques can resolve this issue.

Hyperparameter selection plays a vital role in training successful DL models. As shown in Figure 2.7, the goodness of hyperparameters is determined based on the model's performance on the validation set. The goal is to ensure the trained model generalizes well for testing data. If not, the process in Figure 2.7 should be iterated with refined hyperparameter search space. Numerous variables including the number of kernels in a convolutional layer, the learning rate for the optimizers, and the coefficients of different loss terms can be regarded as hyperparameters. It takes expert knowledge and experience to design a thorough but efficient search space. Usually, the complexity of the model, the learning rate, the batch size, and the coefficients of different loss terms are among the top priorities during searching. Advanced searching methods such as Bayesian optimization [166] and Hyperband [167] can also be considered.

2.2.3.4 Other techniques

The success story of DL is not complete without mentioning some other techniques.

Batch normalization (BatchNorm) is a technique commonly used for different types of DL models [168]. A BatchNorm layer introduces two learnable parameters (scale and shift) to normalize its inputs across a data mini-batch. This will reduce the internal covariate shift problem by ensuring consistent in-

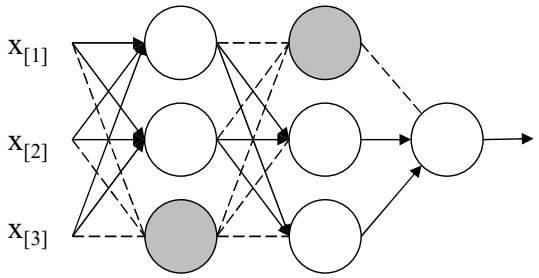


Figure 2.9: An MLP with dropout neurons.

puts to its subsequent layers, leading to improved training stability and faster convergence.

Dropout is a technique to reduce overfitting [169]. It is simply to randomly set the output values of a fraction of neurons to zero during training. Figure 2.9 demonstrates how dropout works in an MLP, where the dropout neurons are marked grey to indicate their zero output value. The edges (weights) connected to the dropout neurons are marked dash lines as they become idle. This will encourage the model to develop a more diverse set of features that generalize better to new data. It also forces neurons to be more independent and reduces the chances of overfitting.

Data augmentation is to generate additional training data by applying transformations to existing data. This is essential in combating overfitting. For images, scaling, flipping, and so on can be applied [170]. For signals, segmentation with overlapped windows can create more training samples [146].

Learning rate is one of the most important hyperparameters in DL. Instead of using a fixed learning rate throughout the whole learning process, a schedule to change the learning rate can be applied. In practice, it is common to gradually decrease the learning rate over time to allow faster convergence [45].

Gradient clipping is to limit the magnitude of gradients during training so that the optimization does not diverge and oscillate. It can be useful for training RNNs or models with complicated regularization terms such as MMD [158].

Chapter 3

Weighted domain adaptation networks for machinery fault diagnosis

3.1 Introduction

Fault diagnosis of rotating machines is of vital importance in modern industry. A reliable early fault diagnosis system is key to reduce maintenance costs and can help preventing major failures [171]. Nowadays, intelligent fault diagnosis has received much research interests as it can work for real-time diagnostic applications and its performance grows with data volume [43], [172]. Three intelligent fault diagnostic frameworks are often studied: conventional, deep learning (DL) based, and transfer learning framework [115].

Conventional intelligent fault diagnosis mainly involves signal processing, feature extraction, feature selection, and fault classification [34], [40]. This conventional framework heavily relies on expert knowledge as the above technologies need to be tailored for different machines and different working conditions (different combinations of load and rotating speed) [29], [173]. DL based diagnosis framework incorporates signal processing, feature extraction, and fault classification into one step [34], [81]. DL framework has been successfully applied for the diagnosis of various mechanical components, such as

bearings [16], [174], [175], gears [49], [173], and rotors [176], [177]. However, neither the conventional nor the DL based framework is intelligent enough [178] as they may not perform well under the environments of (1) lack of labeled data and (2) variable working conditions.

The above two frameworks are based on supervised learning, which will learn only from labeled data. In many fault diagnosis cases, labeled data are expensive or even prohibitive. In addition, the two frameworks have limitations in coping with the influence of working condition changes. That is, a trained fault classifier may not perform well under different rotating speeds or load levels [49], [179], [180]. The key is to extract fault-discriminative but working-condition-invariant features from raw data.

To this end, a recent developed technique, domain adaptation [113], [179], [181], can support a more intelligent transfer learning framework [61] for fault diagnosis. In a fault diagnosis context, a working condition can constitute a domain. The domains with labeled data available are called source domains while those with unlabeled data are called target domains. Domain adaptation is to train a model using both source-labeled data and target-unlabeled data, so that the trained model can perform well in the target working condition. In mechanical fault diagnosis, domain adaptation has been developed and applied to combat rotating speed changes [49], [115], [116], load level changes [49], [114], [117], and cross-machine diagnosis [55]. However, these existing studies are limited to single source domain adaptation. In practice, fault data may be collected from two or more working conditions, constituting a multiple source domain adaptation problem. Methods for multiple source domain adaptation have not been adequately investigated for machinery fault diagnosis.

For natural language processing and computer vision applications, many multiple source domain adaptation methods have been developed. The key is to efficiently learn and combine knowledge from multiple sources. Refs. [182]–

[185] construct different source-specific predictors for each source domain and then combine, with possibly different weights, their predictions on the target data. The idea of learning from multiple sources is to be utilized in this study for machinery fault diagnosis.

For machinery fault diagnosis, the number of source working conditions can be arbitrarily large and constructing that many source-specific predictors is computationally inefficient. Ref. [63] train a single target domain predictor utilizing multiple source domains. This is a good idea for fault diagnosis which can largely reduce computational costs. However, when training that target predictor, ref. [63] uses equal weighting on different sources. Ref. [186] demonstrated that non-uniform weighting on different source could give better results but did not discuss how to determine the weights. The weighting schemes in [183]–[185] for natural language processing and computer vision applications are all based on multiple source-specific predictors and cannot be applied to single predictor methods [63]. The necessity of weighing the source domains differently, we believe, depends on where the differences of the domains are from. In natural language processing studies, the differences are likely to come from novel samples of one domain to the others [187]. In such applications, treating each domain equally is a good choice. However, for fault diagnosis, we can have clear physical meanings of domains differences and we have prior knowledge that the samples within each domain should be similar. With the above considerations, we believe that it is important to propose a good domain weighting scheme for fault diagnosis applications.

Apart from its positive effects, domain adaptation has the risk of causing negative transfer [188], in which training with both the source and the target data leads to a worse performance than only with the source data. To alleviate such risk, refs. [189], [190] reported to use thresholds to filter out negative data during adaptation, ref. [188] studied when to cease the adaptation. Es-

sentially, we need to know where not to adapt. If the risk of negative transfer is high, switching back to traditional supervised training with source data only is needed.

In this study, we are going to develop a multiple source domain adaptation method with different weights applied to different source domains (working conditions) for machinery fault classification. Our motivations are (1) Multiple source domains characterized by different working conditions can help the diagnosis on a different target working condition and (2) The contributions of different sources should be different given that their working condition deviations from the target working condition are different. We follow ref. [63] to construct a single target domain classifier for all the domains, but we assign non-uniform weights to different sources. Different schemes of weight assigning will be investigated. To avoid negative transfer for good classification accuracy, we propose to determine whether to use domain adaptation or traditional supervised learning first before commencing the training. Domain adaptation might not help when the source and the target working conditions are too close. Two case studies on two different experiment test rigs are performed, and both speed change and load level change will be studied. The effectiveness of our proposed multiple source domain adaptation will be demonstrated, and the soundness of our weighting scheme will be examined. In summary, the main novelty of this paper are:

1. We treat each working condition as a domain and learn from multiple source-labeled and one target-unlabeled domains;

2. We weigh different source domains differently to scale the contributions of different source working conditions;

3. We assess the necessity of domain adaptation first before training to avoid negative transfer.

In the following parts, Section 3.2 discusses the preliminary knowledge, Section 3.3 explain our proposed method and the reported methods to be compared with, Section 3.4 presents two case studies to demonstrate the proposed method, and then Section 3.5 concludes this paper.

3.2 Preliminaries

3.2.1 Domain adaptation

When the training and testing data are drawn from different distributions, traditional machine learning algorithms do not perform well. Domain adaptation become useful in this situation [61].

A *domain* consists of a data space $\mathcal{X}$ and a probability distribution $P(X)$ on its samples $X \in \mathcal{X}$. Domain adaptation means to adapt useful knowledge from source-labeled domain S to be applied to target-unlabeled domain T. Specifically, we are given a source-labeled dataset $(X_S, Y_S) = \{(x_S^1, y_S^1), (x_S^2, y_S^2), \ldots, (x_S^m, y_S^m)\}$ and a target-unlabeled dataset $X_T = \{x_T^1, x_T^2, \ldots, x_T^n\}$ for model training. The trained model is expected to have good classification or regression performance on unseen target domain samples.

The following two assumptions are made for domain adaptation in ref. [61]. 1) The data distributions of the source and the target domains should be different but similar, i.e. $P_S(X_S) \neq P_T(X_T)$ but $P_S(X_S) \approx P_T(X_T)$, so that the knowledge learned from the source can be adapted to the target. 2) The data space and label space $\mathcal{Y}$ should be the same across the source and the target domains, i.e. $\mathcal{X}_S = \mathcal{X}_T$ and $\mathcal{Y}_S = \mathcal{Y}_T$, so that neither new format nor new class of data will come in when testing. In this study, we follow these definitions and focus on scenarios meeting these assumptions.

3.2.2 Domain adversarial training

In domain adversarial training, a neural network can be viewed as consisting of three parts [113]: *feature extractor* F, *label classifier* C, and *domain discriminator* D. The feature extractor transforms the input data x into a feature vector f, i.e. $f = F(x; \theta_f)$. With f, the label classifier C calculates a vector of class scores $c = C(f; \theta_c)$ for each class while D produces a vector of domain scores $d = D(f; \theta_d)$ for source or target domain. That is, the label classifier predicts the label of an input and the domain discriminator tries to tell if the input is from the source or the target domain. The indices of the highest score are regarded as the class/domain predictions.

The key of domain adversarial training is to put C and D against each other as two players in a minimax game. The label classifier C plays to minimize its labeling error while D is set to maximize "*domain confusion*" [62]. A Nash equilibrium can be achieved if the feature extractor F is trained to produce label-discriminative yet domain-invariant features. More formally, considering the function $\widetilde{E}(X, Y; \theta_f, \theta_c, \theta_d) = L_C(X, Y; \theta_f, \theta_c) - L_D(X; \theta_f, \theta_d)$, it is to search for parameters $\left(\hat{\theta}_f, \hat{\theta}_c, \hat{\theta}_d\right) = \min_{\theta_f, \theta_c} \max_{\theta_d} \widetilde{E}$ that deliver a saddle point at maximum domain confusion L_D yet minimum labeling error L_C [113].

Further, by inserting a Gradient Reversal Layer (GRL) between F and D, this parameter searching can be implemented as a simple minimization via backpropagation [113]:

$$\min_{\theta_f, \theta_c, \theta_d} E = L_C(X, Y) + L_D(X). \tag{3.1}$$

The GRL simply copies f into D during the forward feeding, while reverses the sign of the gradient that backpropagates from it. This reversed gradient will move F towards the negative direction of minimizing the L_D term, so that the domains can be maximally confused. For the loss functions L_C and L_D,

standard Cross Entropy Loss [45] will be used our study:

$$L_C(X,Y) = -\frac{1}{m}\sum_{(x,y)\in(X_S,Y_S)} \log\frac{\exp(c[y])}{\sum_i \exp(c[i])}, \tag{3.2}$$

$$L_D(X) = -\frac{1}{m}\sum_{x\in X_S} \log\frac{\exp(d[1])}{\sum_i \exp(d[i])} - \frac{1}{n}\sum_{x\in X_T} \log\frac{\exp(d[2])}{\sum_i \exp(d[i])}, \tag{3.3}$$

where (X_S, Y_S) is the source-labeled dataset with m samples, X_T is the target-unlabeled dataset with n samples, x denotes a data sample, y is its corresponding label, and $c[i]$ and $d[i]$ are respectively the ith element of the class score vector and the domain score vector which were explained in the first paragraph of Section 3.2.2.

3.2.3 Maximum Mean Discrepancy

Maximum Mean Discrepancy (MMD) measures the difference between two distributions on the basis of samples drawn from the two distributions [191]. Given samples $X = \{x_1, x_2, \ldots, x_m\}$ and $Z = \{z_1, z_2, \ldots, z_n\}$, using a kernel $\mathcal{K}(\cdot,\cdot)$, it can be estimated using Eq. 3.4:

$$MMD(X,Z) = \big[\frac{1}{m^2}\sum_{i,j=1}^{m}\mathcal{K}(x_i,x_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n}\mathcal{K}(x_i,z_i) + \frac{1}{n^2}\sum_{i,j=1}^{n}\mathcal{K}(z_i,z_j)\big]^{\frac{1}{2}}. \tag{3.4}$$

The MMD value is expected to be a small quantity if the distributions of X and Z are similar. The choice of kernel is critical to the power of this discrepancy measurement. A typical choice is Radial Basis Function (RBF) kernel, i.e. $\mathcal{K}(x,z) = \exp(-\|x-z\|^2/b)$, where b is the bandwidth of the RBF kernel. Ref. [192] further discussed that a linear combination of multiple kernels can render a good kernel. MMD using a combined kernel is often called Multi-Kernel MMD, or MK-MMD [117], [193]. In addition, squared formulation of MMD, or MMD^2, is used and aliased with MMD in refs. [117], [193]. For simplicity, we will use MMD to denote the squared MK-MMD in the following parts of this paper.

Most MMD based domain adaptation methods will not apply MMD on the input data. Instead, MMDs of extracted features are usually measured [49], [117], [172], [193]. This fits the goal of learning domain-invariant features. In practise, considering computational constraints and efficiency, the inputs of Eq. 3.4 are mostly mini batches (small partitions of a whole dataset).

3.3 The proposed method

As discussed in Section 3.1, we study domain adaptation to combat the impact of working condition changes on machinery fault diagnosis. To efficiently learn knowledge from multiple source working conditions, a good weighting scheme is needed. In addition, to avoid negative transfer, a criterion of determining whether to perform domain adaptation is to be adopted.

In this paper, a Weighted Domain Adaptation neural Network (WDAN) is proposed. Unlike ref. [179] which merges multiple labeled working conditions as one source domain and build only one domain discriminator (see Section 3.2.2), we treat each working condition as a separate domain, and follow ref. [63], which has been successfully applied to natural language processing and computer vision, to build multiple source-specific domain discriminators. On top of ref. [63], we insert a weighting block so that different weights can be assigned on different source domains. The weights will be assigned based on the MMD measure explained in Section 3.2.3. The measured MMD values will also be used to determine whether to perform domain adaptation or traditional supervised learning, so that negative transfer can be avoided.

In the following part of this section, Section 3.3.1 presents the architecture of the proposed WDAN, and Section 3.3.2 explains the training procedure of WDAN. All the compared methods will be summarized in Section 3.3.3.

3.3.1 WDAN architecture

As shown in Figure 3.1, WDAN has a feature extractor F, a classifier C, multiple domain discriminators Ds, and a domain weighting block. Each D has the same structure but is associated with a specific source domain, so that multiple source working conditions can be accommodated. Note that ref. [179] has only one domain discriminator and ref. [63] dose not have the domain weighting block which will be explained in Section 3.3.2.

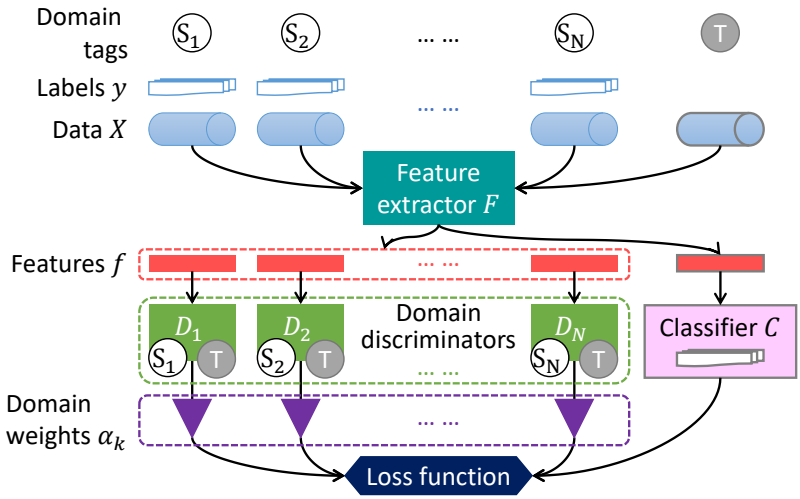


Figure 3.1: Schematic diagram of WDAN.

The structures of F, C, and D are respectively shown in the upper, left lower, and right lower boxes of Figure 3.2. The 3-d blocks annotated with Conv, FC, and BN in Figure 3.2 are 1-d convolution layers, fully connected layers, and batch normalization layers, respectively. The three comma separated digits listed under a Conv layer in Figure 3.2 are its number of input channels, number of output channels, and kernel size; the numbers pointed by an arrow under the FC layers are their output dimensions. All three types of layers have learnable parameters that will be updated during training. The empty V-shaped arrows are layers (or operations) without learnable parameters. As annotated at the top of Figure 3.2, they are (max) pooling layers with a down sampling ratio of 0.25, ReLU activation layers, dropout layers with a

drop rate of 0.5, a flatten layer to reshape high order tensors into the 1st order vectors, and a GRL that was explained in Section 3.2.2.

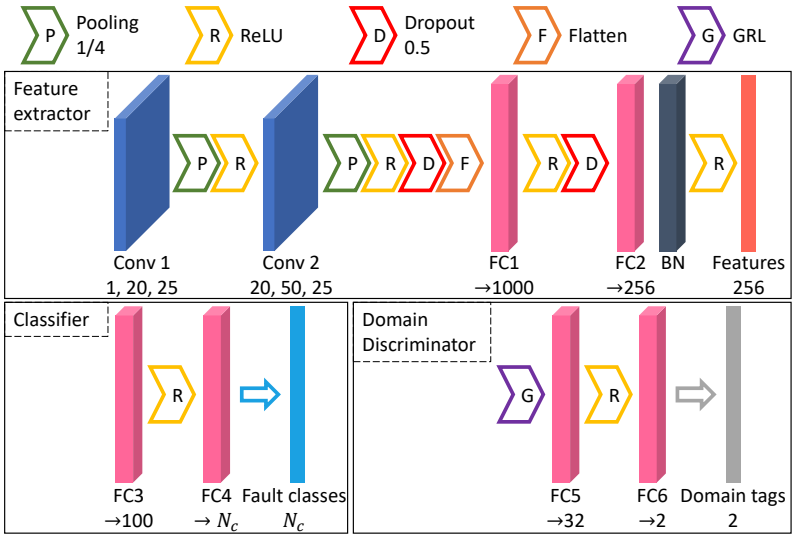


Figure 3.2: Network structures used in this paper.

From Figure 3.2, we can see that the design of the feature extractor is a one-dimensional version of the very first reported convolutional neural network [140]. The kernel size of the Conv layers and the downsampling ratio of the pooling layers follow the original design. The use of ReLU activation [194] is standard for deep neural networks. We select the output dimension of FC1 from {500, 1000, 2000}, the output dimension of FC2 from {128, 256, 512}, the use of dropout operation, and the use of BN layer based on the average of testing accuracies on the source datasets of ST-1 in Table 3.2 (to be explained later) of source-only method (to be explained in Section 3.3.3). The classifier and domain discriminators are both standard two-layer perceptrons [45]. The size of FC4 equals the number of machine's fault classes while FC6 has only 2 neurons respectively for the source and the target. The output dimension of FC3 was pre-determined. We select the output dimension of FC5 from {16, 32, 64} based on the testing accuracy on the target dataset of ST-1 of equal-weighting method (to be explained in Section 3.3.3).

After the training is completed, only the F and C will be saved for testing while the Ds will be discarded. The saved F and C are expected to have good performance on the target domain. We use a held-out labeled dataset from the target domain to test the classification accuracy of the trained networks. All the input data samples are raw vibration signal segments, and their corresponding labels are machine's fault classes. Before being sent into WDAN, a normalization step is completed in which a data sample will be subtracted and divided respectively by the element-wise mean and the standard deviation of its original dataset. The held-out test set will use its own mean and standard deviation.

3.3.2 Training procedure

We train the WDAN with two objectives: (1). minimal classification error on each source domain; (2). maximum "domain confusion" between each source and the target. If there is only a single source domain, vanilla domain adversarial training (Eq. 3.1 in Section 3.2.2) can be applied. When there are multiple source domains, we need to combine their loss terms properly. A log-sum-exp operation is used in ref. [63] but every source is equally weighted. We propose to assign different weights to different sources and formulate the training of WDAN as:

$$\min \log \sum_{k=1}^{N} \exp N\alpha_k \left(L_{Ck} + L_{D_k}\right), \tag{3.5}$$

where N is the number of source domains, $L_{Ck} = L_C\left(X_{S_k}, Y_{S_k}\right)$ is the labeling error term on the kth source dataset, $L_{D_k} = L_{D_k}\left(\{X_{S_k}, X_T\}\right)$ is the domain confusion term of the kth domain discriminator, and they are calculated using Eq. 3.2 and Eq. 3.3 in Section 3.2.2, respectively, and α_k is the weight for domain k with two constraints $\sum_k \alpha_k = 1$ and $\alpha_k \geq 0$, which is proposed to replace the original uniform weighting of sources in ref. [63].

Generally, higher weights should be assigned to sources more similar to the target. Refs. [183], [184] measure distributional similarities between the sources and the target, and then divide the measured similarities by their sum. In such a way, all the sources were assigned with different and non-zero weights. In this paper, we propose to use the distribution metric MMD explained in Section 3.2.3 as the (negative) domain similarity measure. Then, we apply softmax function to the measured MMD values so that the two constraints listed at the end of the previous paragraph on weights can be met. Formally, for a source domain S_k, we assign

$$\alpha_k = \frac{\exp\left(-\beta v_k\right)}{\sum_i \exp\left(-\beta v_i\right)}, \tag{3.6}$$

where v_k is the measured MMD between the S_k and the target T. Following [49], [117], [172], [193], the MMD are measured based on Eq. 3.4 with the extracted features f of mini-batch data as its input. Average MMD of all the mini-batches will be used. The coefficient $\beta \geq 0$ controls the "hardness" of weight assigning. Smaller β is softer as it gives close weight values for all sources. When $\beta = 0$, our method is relaxed to ref. [63] with equal weights; when $\beta \to \infty$, our weighting scheme becomes "hard max" as only the best source(s) will be assigned with non-zero weight(s). We believe that generally speaking, allowing a positive β which gives non-uniform weights to multiple sources is the best practice.

Apart from weight assigning, the measured MMD values will also be used to assess the risk of negative transfer. If the MMD values are all below a certain threshold, the source and the target domains are already well matched. In such a case, rather than risking testing performance with target-unlabeled data, we perform traditional supervised learning with source-labeled data only. That is, we merge all the source domains as one ($N = 1$) and deactivate the domain discriminators ($L_{D_k} = 0$). The pseudo code of our proposed WDAN training

procedure is described in Algorithm 1.

Algorithm 1 Training procedure of WDAN.

Input:

- Datasets $\{(X_{S_k}, Y_{S_k})\}_{k=1}^{N} = \left\{\left\{\left(x_{S_k}^i, y_{S_k}^i\right)\right\}_{i=1}^{m_k}\right\}_{k=1}^{N}$ and $X_T = \{x_T^i\}_{i=1}^{n}$,
- Initial WDAN $\{F, C, D\}$,
- Hardness parameter β,
- Mini-batch size p,
- MMD kernel $\mathcal{K}$,
- Transfer threshold τ.

Output: trained WDAN $\{F, C\}$

```
# calculate average MMDs: v_k s (initial v_k = 0)
for k from 1 to N do
    # randomly split into mini-batches of p samples
    {(X_{S_k}^i, Y_{S_k}^i)}_{i=1}^{⌊m_k/p⌋} ← split (X_{S_k}, Y_{S_k})
    {X_T^i}_{i=1}^{⌊n/p⌋} ← split (X_T)
    for j from 1 to ⌊n/p⌋ do
        if j > size{(X_{S_k}^i, Y_{S_k}^i)} then
            {(X_{S_k}^i, Y_{S_k}^i)} ← duplicate ({(X_{S_k}^i, Y_{S_k}^i)}) # use S_k repeatedly
        end if
        v_k ← ν_k + MMD (F (X_{S_k}^j), F (X_T^j); K) / ⌊n/p⌋
    end for
end for
# apply threshold to avoid negative transfer
if ∀v_k < τ then
    min Σ_{k=1}^{N} L_{C_k} # source-only training
else
    for k from 1 to N do
        α_k ← exp(−βv_k) / Σ_i exp(−βv_i) # assign weights using Eq. 3.6
    end for
    min log Σ_{k=1}^{N} exp Nα_k (L_{Ck} + L_{D_k}) # adaptive training (Eq. 3.5)
end if
```

3.3.3 Compared methods

We compare our proposed WDAN with the following methods: **source-only** using Eq. 3.5 with $N = 1$ and $L_{D_k} = 0$; **best-single** which applies single source domain adaptation [113] on every single source domain ($N = 1$) and then report the best possible result; **merge-as-one** [179] and then apply single source domain adaptation with $N = 1$; and ref. [63] that apply

equal-weighting on sources by setting $\beta = 0$. We use **WDAN-β** to denote our proposed method with a "hardness" coefficient of β. Note that WDAN-0 is identical to ref. [63]. We are going to explore the impact of different real positive values of β.

To put all the above methods on equal footing, all will use the same network structures in Figure 3.2. We use standard mini-batch Stochastic Gradient Descend (SGD) with momentum [195] to solve their corresponding training objective functions. Note that refs. [179] and [113] both used SGD with momentum. Refs. [196] and [197] also support that SGD generalizes better than adaptive gradient methods including Adam. Empirically, the momentum is set to be 0.9, the mini-batch size is set to be 100, and the best learning rate for each method will be grid-searched from {0.1, 0.01, 0.001}. For efficiency and ease of implementation, the mini-batch size for calculating the average MMD values is also set to be 100.

3.4 Experiments

We present two case studies on two different test rigs located at Tsinghua University (THU) and University of Alberta (UofA). Computational experiments are run on a computer with a single Intel i7-6700 CPU and a single Nvidia GTX-1060 GPU. All the tested methods are implemented using Pytorch[1].

3.4.1 Case study I

3.4.1.1 THU planetary gearbox test rig

The HS-200 single-stage planetary gearbox[2], located at Tsinghua University, was used to conduct experiments and collect data in year 2019 by one of the co-authors. During the physical experiment, the gearbox was driven by a motor

[1] `https://pytorch.org/`

[2] `http://www.sh-wxjd.net/product/qdq/49.html`

and the output rotating speed of the motor was set at 26 different constant levels, ranging from 15Hz to 40Hz with an interval of 1Hz. No external load was applied by the load motor. Two accelerometers were installed in horizontal and vertical directions the of gearbox casing. By seeding artificial damages in the sun gear or one of the planetary gears, 9 different fault classes are created and tested. The 9 tested fault classes are described in Table 3.1. Figure 3.3 shows the structure of the planetary gearbox and 4 example damaged gears. The seeded gear tooth cracks span the whole width of space of the tooth and the cut is 0.1 mm wide.

Table 3.1: Tested fault classes of the THU planetary gearbox.

Classes	Types	Levels
H	Health	-
SC1	Sun tooth crack	Crack depth: 1/8 dedendum
SC2	Sun tooth crack	Crack depth: 1/4 dedendum
SC3	Sun tooth crack	Crack depth: 1/2 dedendum
SB	Sun tooth broken	Break position: 1/3 tooth depth
PC1	Planet tooth crack	Crack depth: 1/8 dedendum
PC2	Planet tooth crack	Crack depth: 1/4 dedendum
PC3	Planet tooth crack	Crack depth: 1/2 dedendum
PB	Planet tooth broken	Break position: 1/3 tooth depth

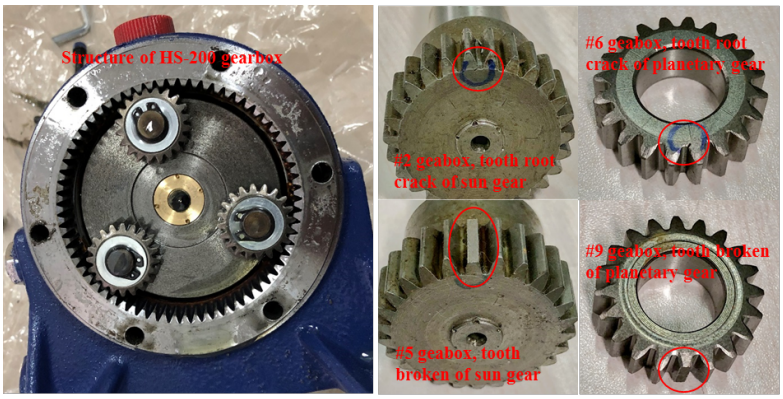


Figure 3.3: Structure of the HS-200 planetary gearbox and 4 example damaged gears.

Table 3.2: Adaptation tasks on THU dataset.

Adaptation task	Source domains	Target domain
ST-1	40Hz, 34Hz, 28Hz	16Hz
ST-2	16Hz, 40Hz	28Hz
ST-3	16Hz, 22Hz, 28Hz	40Hz

3.4.1.2 Data description

We regard each rotating speed as a domain. Three different source-target adaptation tasks listed in Table 3.2 are studied. The physical implications of the three tasks are ST-1: adapt to a lower speed; ST-2: adapt to a speed in the middle; ST-3: adapt to a higher speed.

For each rotating speed and each health condition, the vibration data measured in the horizontal direction at a sampling frequency of 20kHz, for a consecutive 256 seconds are used. The first 204.8s are used for training and the rest 51.2s are held out for testing. We slice the vibration signals into 2048 sized input samples for our neural networks. That is, there will be 2000 and 500 samples per domain per class respectively for training and testing.

3.4.1.3 Results and discussions

We apply all the five methods discussed in Section 3.3.3 and compare their performances (average over 10 repeat runs) on ST-1 to ST-3. For all the methods, the number of training epochs is 10. For the choice of β, we tested 4 different values, 5, 10, 25, and 100 on ST-1 and found that $\beta = 10$ provided the best accuracy on ST-1 (see Figure 3.4). In this study, $\beta = 10$ is used as the default choice for other tasks. We follow ref. [117] to calculate MMD values with 5 RBF kernels with bandwidths of 1, 2, 4, 8, and 16.

We first test the source-only method on the source domain data. The source test accuracies are 99.993%, 99.966%, and 99.993% respectively for ST-1, ST-2, and ST-3. These accuracies show that, when the testing and training

data are from the same set of domains, our designed feature extractor and classifier (Section 3.3.1) can have very good performances. Then, we test all 5 methods on the target domains and show the target test accuracies in Table 3.3.

Table 3.3: Target domain test accuracies and training time costs on THU dataset.

Method	ST-1 (%)	ST-2 (%)	ST-3 (%)	Training time (seconds)	
				ST-1 or ST-3	ST-2
source-only	79.489±2.082	94.382±3.276	69.836±4.396	85.16±0.35	57.83±0.14
merge-as-one [179]	83.476±4.942	97.816±1.506	85.647±11.18	58.94±0.16	59.27±0.25
best-single [113]	89.962±3.365	94.924±4.909	90.102±7.154	171.06±0.31	115.13±0.38
eqaul-weigthing [63]	85.322±4.006	99.042±0.469	90.382±6.279	107.94±0.24	83.86±0.33
WDAN-10 (proposed)	**91.102**±1.369	**99.158**±0.253	**94.380**±4.063	113.69±0.27	88.22±0.33

From the left-hand side of Table 3.3, we can see that, when tested under a different working condition, the accuracies of source-only models drop 20.504%, 5.584%, and 30.457%, respectively for the three tasks, comparing to their source test accuracies. These numbers show how much the rotating speed gaps affect the traditional source-only learning method. Our proposed WDAN reduced the three previous listed accuracy drops to 8.891%, 0.808%, and 5.613%, by gaining 11.613%, 4.776%, 24.544% of accuracies comparing to source-only. It achieves the best performance among all the compared methods, which demonstrates the effectiveness of multiple domain discriminators and our proposed weighting scheme. Other three domain adaptation methods, i.e. merge-as-one, best-single, and equal-weighting, can also gain accuracies on top of the baseline source-only approach. With no negative transfer observed, domain adaptation should be applied for all the three tasks in this case study. The criterion for switching back to source-only will be investigated with the next case study in Section 3.4.2.

It is also observed that the weights assigned using MMDs well agrees the physical meanings of the working conditions. A source with smaller speed

gap to the target will be assigned with higher weight. For the ST-2 where the two sources have the same speed gap to the target, the source of higher speed conditions will be assigned with higher weight. The ranking of sources reported by the best-single method also agrees with the ranking of weights assigned by MMDs. The best sources are both 28Hz for ST-1 and ST-3, and the best-single source for ST-2 is 40Hz. Although, all the best-single's testing accuracies are lower than the WDAN's. It can be said that adapting from multiple sources can be better than from only from one.

Among the three tasks, the task of adapting to a middle speed (ST-2) is the easiest as its accuracy of source-only drops the least across domains. The adaptation gains of applying domain adaptation methods are also less significant comparing to the other two tasks. For example, adaptation gains of the best-single method are 10.47% and 20.27% respectively for ST-1 and ST-3, but only 0.54% for ST-2. That said, domain adaptation is more useful when the target speed condition is out of the range covered by the known speeds.

The influence of the "hardness" coefficient β is shown in Figure 3.4. Task ST-1 with a target domain of 16Hz is used for demonstration. In Figure 3.4, α_1, α_2 and α_3 are the weights assigned to the three source domains 40Hz, 34Hz, and 28Hz respectively. The solid line with markers shows the target test accuracies for each chosen β values. It shows equal-weighting ($\beta = 0$) gives the lowest accuracy and WDAN-10 is the best for ST-1. Gradient Norms (GNs) of the FC5 layers of D_1, D_2, and D_3 (see Section 3.3.1) are shown as the three dash lines. GN means the 2nd norm of the back-propagated gradients and here we use average GN across all learning steps. These gradients will be reverse by the GRL and backpropagate into the feature extractor F. The average GN from a D_k can describe how fast and how much the F changes in order to confuse its corresponded S_k and the T. We can see (GN1 and GN2

are overlapped) that GNs are positively correlated with the assigned weights. This proves that the sources assigned with higher weights are playing more important roles during learning.

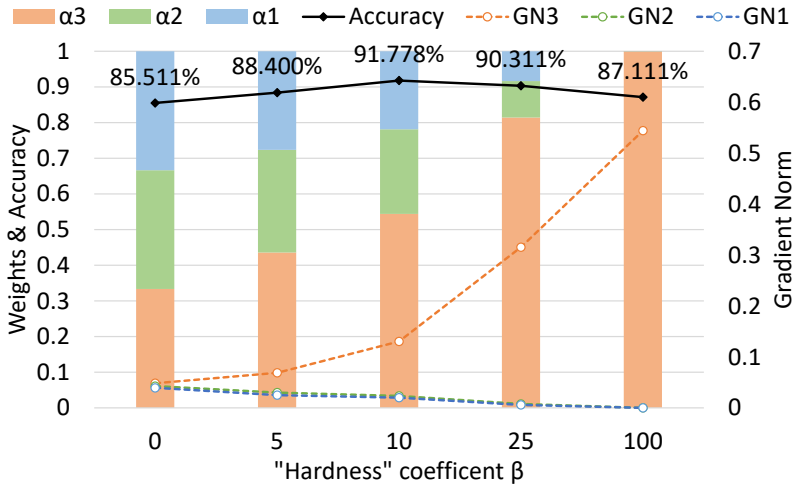


Figure 3.4: Influence of β on ST-1. Rotating speeds of S_1: 40Hz, S_2: 34Hz, S_3: 28Hz, T: 16Hz.

For the case of $\beta = 100$, only the domain of 28Hz is assigned with non-zero weight, making our WDAN approaches the best-single method. However, WDAN has more parameters to learn as it has three domain discriminators while best-single only has one. This makes the WDAN-100 more likely to overfit and achieve lower accuracy. A "soft" β will put WDAN in between pay equal attentions to every source (equal-weighting) and learn only from the best-single source.

To visualize the classification performance for each different category, Figure 3.5 shows the confusion matrices of the source-only method and our proposed WDAN on ST-3. The notations of the fault classes are explained in Table 3.1. We can see that the source-only method performs poorly on classifying certain fault classes. For example, the accuracy is 0% for SC3 and 9% for PC3. This implies that, due to rotating speed difference, the target samples of a certain class may be wrongly aligned with the source samples of

other classes. In contrast, domain adaptation (WDAN) can better align the source and target samples of each classes and produces much better classification performance. Its accuracies are 84% and 91% higher, for SC3 and PC3 respectively, than those of source-only.

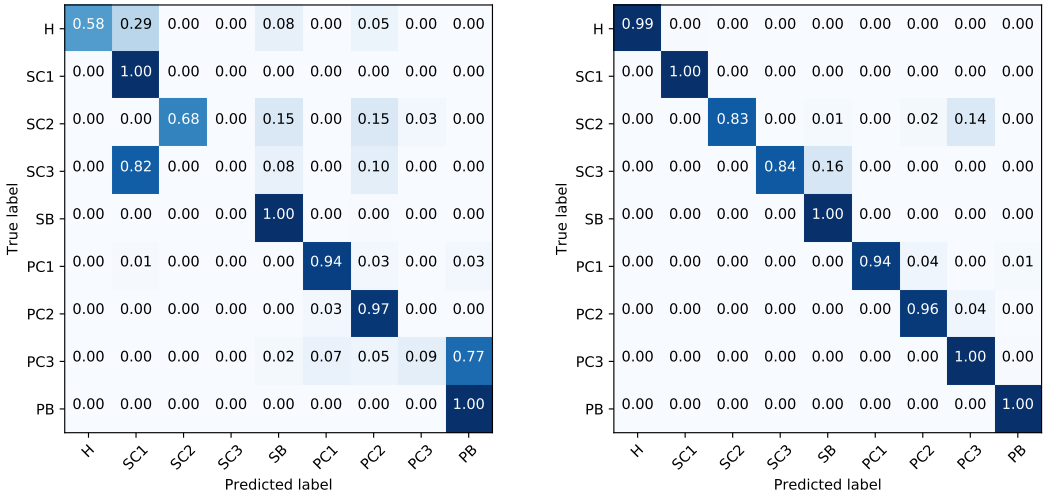


Figure 3.5: Confusion matrices on ST-3. Left: source-only; Right: WDAN-10 (proposed).

The trade-offs behind adaptation gains include extra computational time and memory for training. As shown in the right-hand side of Table 3.3, for a 3-source-1-target task (ST-1 or ST-3), WDAN spends about 5.75 seconds to compute the weights. In addition, source-only needs about 85.16s for 10 epochs of training while WDAN takes about 113.69s. Memory cost is also higher for WDAN as more data and more learnable parameters will be used than it is in source-only. Nonetheless, all the five listed methods have the same speed and memory usage during testing stage.

3.4.2 case study II

For the 2nd case study, a gearbox test rig at the University of Alberta [50], [198] was used to collect the data in year 2018. The data is used directly in this case study. Six fault classes are considered including five different levels

of gear tooth crack and one healthy state.

3.4.2.1 Data description

During the physical experiment, the rotating speed was set at 10Hz and the machine was run at three different load levels: 3% (low), 8% (middle), and 13% (high) of the load motor's rated capacity (100klb-in). We regard the three load levels as domains and enumerate all three possible 2-source-1-target adaptation tasks. They are ST-4: the 8% and 13% loadings are sources while the 3% loading is the target; ST-5: the 3% and 13% loadings are sources while the 8% loading is the target; and ST-6: the 3% and 8% loadings are sources while the 13% loading is the target.

The vibration data provided by the accelerometer on its bearing cap (Sensor #2 in ref. [198]), with a sampling frequency of 25.6kHz is to be used in this case study. For each load level and each fault class, four repeats of 30-second runs were conducted during the experiments. The first three runs are used for training and the last one is held out for testing. We slice the vibration signals into 2048 sized input samples for our neural networks. That is, there will be 1125 and 375 samples per domain per class respectively for training and testing.

3.4.2.2 Results and discussions

Following the first case study, we show the average performances over 10 repeat runs on ST-4 to ST-6. For WDAN, we keep $\beta = 10$ and the MMD kernels are also the same as in case study I. For all the methods, the number of training epochs is increased to 25 (from 10 in case study I). Under such setting, the number of training steps will be similar for the two case studies, given that the THU dataset is 2.667 times larger.

When tested by the source domain data, source-only models give accura-

Table 3.4: Target domain test accuracies and training time costs on UofA dataset.

Method	ST-4 (%)	ST-5 (%)	ST-6 (%)	Training time (seconds)
source-only	83.689±4.643	**91.138**±2.460	83.929±6.583	50.58±0.21
merge-as-one [179]	81.831±5.708	85.084±9.803	78.031±9.031	52.63±0.22
best-single [113]	83.360±6.127	88.196±2.038	87.244±3.104	103.82±0.59
eqaul-weigthing [63]	83.996±3.649	89.964±2.867	86.476±4.296	79.32±0.26
WDAN-10 (proposed*)	**85.591**±2.696	90.382±3.475	**87.316**±6.837	81.64±0.23

* Applied regardless of the MMD-based threshold for avoiding negative transfer

cies of 96.667%, 95.482%, and 94.169% respectively for ST-4, ST-5, and ST-6. This is a solid performance for crack level classification. The target test accuracies are shown in the left-hand side of Table 3.4. Same as in case study I, WDAN shows adaptation gains, over source-only, of 1.902% and 3.387% respectively on tasks ST-4 and ST-6, achieving the best accuracies among all the compared methods. However, negative transfer [199] is observed in this case study. Utilizing both the source and target data may result lower accuracy than source-only. Under such a case, our proposed MMD-based criterion comes into use.

Based on our observation, the measured MMDs in this case study is significantly lower than those in case study I. With the distributions of the sources and the target are already similar, performing domain adaptation become less beneficial. While the risk of false alignment of different categories becomes prominent. This explains why negative transfer occurs in this case study. Using our proposed MMD-based criterion, WDAN's negative transfer on ST-5 could be avoid. A critical issue is to select a proper threshold. If we use a threshold value of 0.07, source-only method will be executed only for ST-5, eliminating the negative transfer for our WDAN method. For other datasets, the optimal choice of the threshold value may be different.

We select ST-6 to study the influence of β in this case study and plot the weights, accuracies and GNs in Figure 3.6. We can see that the best β is

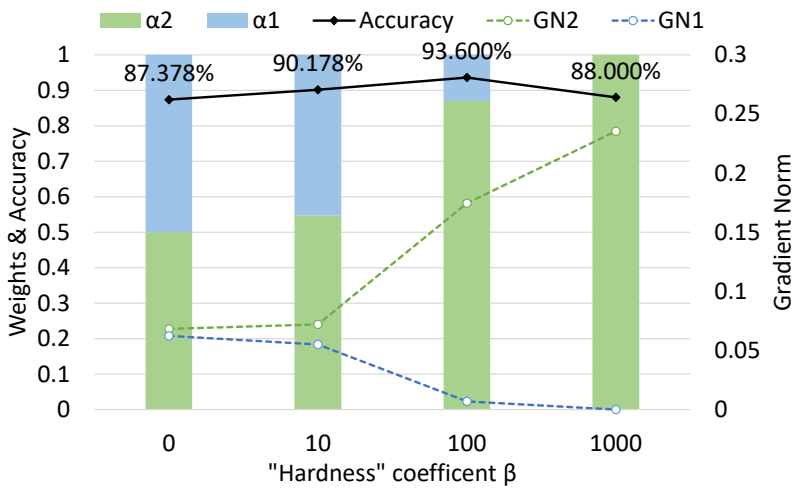


Figure 3.6: Influence of β on ST-6. Load levels of S_1: 3%, S_2: 8%, T: 13%.

100, among 0, 10, 100, and 1000. This is different to ST-1 where WDAN-10 produces the highest accuracy and WDAN-100 may assign zero weight values. For ST-6, the difference between domains in terms of their MMDs is smaller than it for ST-1. This explains why domain adaptation can gain less accuracies on the UofA dataset than on the THU dataset.

Confusion matrices of the source-only method and WDAN on ST-4 are shown in Figure 3.7. The class labels "H" stands for healthy and "C1" to "C5" denote crack level 1 to level 5. Comparing the two matrices, we can see that the efficacy of domain adaption (WDAN) is divided for different classes. WDAN performs better than source-only on "H", "C1", and "C4" while loses accuracy on "C2" and "C5". This indicates that WDAN may correctly or falsely align the source and target samples for different classes. In ST-4, the benefit of applying WDAN outweighs the risk of false alignment as a higher overall accuracy can be obtained.

As shown in the right-hand side of Table 3.4, for the three tasks in this case study, WDAN spends about 2.32 seconds more than equal-weighting does to compute the weights. In addition, source-only needs about 50.58s for 25 epochs of training while WDAN takes about 81.64s.

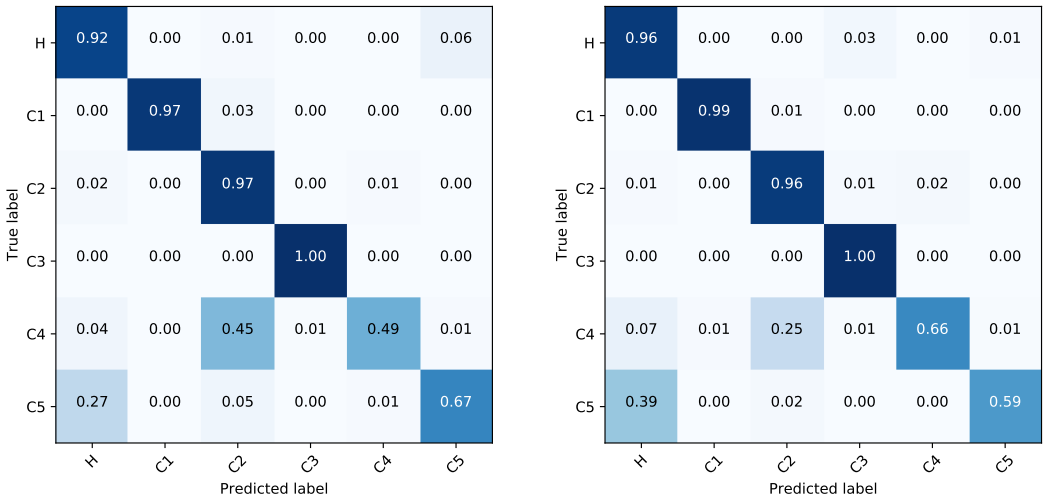


Figure 3.7: Confusion matrices on ST-4. Left: source-only; Right: WDAN-10 (proposed).

3.5 Summary and Conclusion

The main contributions of this work are summarized as follows: 1) A multiple source domain adaptation method is presented for mechanical fault diagnosis tasks. Compared with other related domain adaptation methods, the proposed is more suitable and powerful for industrial applications; 2) Different weights on different source domains are assigned during model training. A balance weighting between uniform weighting and emphasizing on a single source is optimal. The assigned weights by MMD are demonstrated to be in accordance with physical meanings (speed and load level) of the domains; 3) MMD values can also help us avoid negative transfer.

Beyond combating working condition changes, a wider scope of applications can be considered, such as adaptation across different machines. To successfully avoid negative transfer, we need a good method to set proper threshold for our proposed MMD-based criterion. On the probability metric, this study limits to MMD, while other metric such as KL-divergence and K-S statistic, can be considered. Better and task-specific neural network structures may be

searched to further boost fault classification accuracy.

Chapter 4

Open-set fault diagnosis for industrial rotating machines based on trustworthy deep learning

4.1 Introduction

In the era of industrial 4.0, Artificial Intelligence (AI) powered Prognostics and Health Management (PHM) is crucial to ensure the reliability and resilience of Industrial Cyber-Physical Systems (ICPS) such as wind turbines, airplanes, and manufacturing machines [200]. Real-time fault detection and diagnosis of the rotating mechanical components such as gears, bearings, and rotors are crucial to ensure safe operation, avoid downtime, and reduce maintenance costs [8], [201]. Using Deep Learning (DL) models and vibration sensors, infant mechanical faults, such as gear cracks [30] and bearing defects [202], can be diagnosed within a matter of seconds. DL-based fault diagnosis is gaining increasing interest for real-time diagnosis and health management of ICPS [203], [204].

Most existing DL-based diagnostic methods are developed to solve close-set recognition problems, where the target fault classes and working conditions (e.g., rotating speed and load level) are known and fixed. The testing data

in a close-set setting are said to be in-distribution (ID) as they share the same sample and label distribution with the training data [205]. In real-world applications, however, constructing a labeled training dataset and covering all target fault classes and working conditions of interest can be cost-prohibitive. Out-of-distribution (OOD) testing samples must be considered and adapted given new fault classes and new working conditions. For this reason, fault diagnosis of complex ICPS should be regarded as an Open-Set Fault Diagnosis (OSFD) [53] (as known as Open-set recognition (OSR) [206]) problem. In other words, the model needs to (1) classify those 'known' fault classes included in the training set, (2) detect whether an input is OOD or belongs to an 'unknown' class, and (3) report how uncertain it is when making a prediction. Solving these problems can make fault diagnosis rely less on historical data and are more robust to constant changes in various industrial applications. Models with OSFD capabilities are trustworthy for ICPS as they "know when it doesn't know" and can report their uncertainties to request possible human intervention [123], [205].

Early OSR approaches often relied on shallow models, such as Support Vector Machines [52], [207] and Gaussian Mixture Models [120]. However, crafting class-discriminative features for shallow models is challenging, and the presence of potential 'unknown' classes further complicates matters. DL models like Convolutional Neural Networks (CNNs) offer the advantage of automatically extracting discriminative features, but they still face challenges in generalizing to OOD samples [208]. To compound this issue, conventional deep learning models based on the softmax function often exhibit 'over-confidence' by assigning high probabilities to incorrect classes or adversarial samples [56]. To address this issue, a better Uncertainty Quantification (UQ) metric [209] for DL models is to be developed. Many works on UQ-based fault diagnosis methods can be found in the literature [120], [123], [210]–[215] and Section

4.2.1 presents a deep discussion on existing UQ methods.

Apart from UQ, Abstaining Classifier (AC) [216] is another effective approach to identify 'unknown' samples. Besides all the known classes, an AC learns an additional abstention option to reject 'unknown' samples. While UQ matrices are often developed in unsupervised fashions, ACs utilize the supervision of labels from each known class thus being more prudent. The key challenge is to construct an auxiliary set that represents the abstention class and does not lead to confusion with the known classes. For vibration-based fault diagnosis, the physical meanings and characteristics of mechanical vibrations can be utilized. Ref. [201] used simulation models to generate auxiliary training data for bearing fault diagnosis. This requires expert knowledge of the machine of interest and extra steps to adapt the model for both real and simulated signals. Refs. [121] and [122] generate auxiliary training data based on signals from known classes. They used Autoencoders to extract features from existing vibration signals, subsequently applying statistical perturbations to these features. However, such feature-level perturbations may not fully capture the unique characteristics of faulty signals. In this chapter, we consider signal-level perturbations that align with the physical characteristics of faulty mechanical signals. In this way, the auxiliary training set can be constructed without involving detailed knowledge of the target machine and capture much information from the known training sample.

In this chapter, we propose two signal perturbation operations, namely superposition and noise injection, to generate auxiliary training samples based on known training data. These auxiliary training samples represent possible OOD data and help DL models recognize unknown classes. Additionally, we leverage Evidential Deep Learning (EDL) [217] to obtain better fault-discriminative features and enhance the UQ capabilities of DL models. Ultimately, we integrate AC with UQ and introduce the Evidential Abstaining Classifier (EAC)

for OSFD of rotating machines. The overall goal of EAC is to accurately recognize both 'known' fault classes with labeled training data and 'unknown' faults without training data under variable working conditions. This will reduce the reliance of data-driven diagnostics on historical data and the developed UQ metric can provide better transparency to the diagnostic procedure. The innovations in this chapter can be summarized as follows:

1. Auxiliary training: We generate auxiliary samples based on physics and existing training data to represent the abstention class, thereby enhancing the system's capability to recognize previously unseen fault classes.

2. Enhanced UQ: We improve existing EDL by introducing an L1 (also known as Lasso) regularization term, which improves the distinguishability of uncertainty measurements between 'known' and 'unknown' samples.

3. AC and UQ Integration: We combine AC with thresholding on UQ metrics to facilitate trustworthy OSFD and demonstrate its efficacy in diagnosing gear and bearing faults.

This chapter is structured as follows: Section 5.2 provides a review of relevant literature on UQ and AC. In Section 5.3, we introduce our proposed EAC for fault classification. Section 4.4 contains the experimental setups, results, and in-depth analysis. Finally, in Section 5.6, we conclude this chapter and summarize the key findings.

4.2 Related studies

The proposed EAC leverages both UQ and AC to achieve OSFD. In the following, studies related to UQ and AC will be reviewed in Section 4.2.1 and Section 4.2.2, respectively.

4.2.1 Uncertainty quantification

Conventionally, neural networks infer their classification results by ranking the softmax probabilities for each class using Eqn. 4.1.

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad for\ i = 1, \ldots, K \tag{4.1}$$

where p_i is the softmax probability, $z_i = f_\theta^i(x)$ is the output (logit) of the network for class i, and K is the total number of fault classes. $f_\theta(\cdot)$ is the network's function, θ denotes the learned parameters, and x is the input sample. Traditionally, a high p_i value means that the model has high confidence in assigning the input as class i. However, softmax probability provides only a point estimate for the class probabilities of a sample and thus may not be a good uncertainty index. In fact, softmax probabilities are often over-confident when dealing with OOD samples [123], [217].

Alternatively, uncertainties in a sample can be quantified based on its distance or novelty concerning the ID training samples. For instance, in ref. [212], a novelty score was proposed using the Euclidean distance between transformed training and testing samples. Similarly, refs. [210] and [213] calculated their metrics based on Kullback-Leibler (KL) divergence and Mahalanobis distance, respectively. However, selecting the proper distance metric is challenging, and its effectiveness for unknown samples is often uncertain. Some other approaches involve constructing multiple one-class models for each fault class, as seen in refs. [120] and [211], and using statistical fitness to indicate if the input is novel compared to all the known classes. Yet, finding a probabilistic model that fits all fault types, especially the unknown ones, can be difficult.

DL models can also be utilized or modified to produce uncertainty measurements. In ref. [208], an ensemble of CNNs was used, and the entropy of their potentially different predictions on a single input was employed as a UQ metric. However, ensemble models are known to be computationally expensive

and prone to overfitting [218]. Bayesian neural networks, where parameters are treated as random variables rather than deterministic values, have been studied for fault diagnosis with UQ [123]. These models often entail long training and testing times due to the need for multiple forward passes. Importantly, many of these methods cannot distinguish between different types of uncertainties, such as model uncertainty, data uncertainty, and distributional uncertainty [219].

Recently, Evidential Deep Learning (EDL) [217] has gained considerable research attention for its unique ability to estimate all three different types of uncertainties. EDL requires minimal modifications to common neural network structures and operates efficiently in a single-pass manner. Notably, researchers like Zhou et al. [214], [215] have successfully applied EDL in bearing fault diagnosis. In this chapter, we extend the application of EDL to gearbox fault diagnosis. Additionally, we modify the loss function of EDL by incorporating an L1 regularization term. This modification promotes sparse model outputs and results in improved UQ outcomes.

4.2.2 Abstaining classification

AC is known as 'classification with rejection' [207] or 'extended classifiers' [121] in the literature. In existing studies, various approaches have been employed to represent the abstention class, including the use of different yet related datasets [205], forged samples [220], perturbed samples [221], and more. However, these studies often deal with general input types such as images and natural language.

In fault diagnosis literature, the implementation of an abstention option can vary. For instance, one study [120] constructs multiple one-versus-set binary classifiers, abstaining if all binary classifiers report a negative prediction on an input sample. However, this approach, similar to model ensemble, can be

cumbersome and slow. In contrast, another study [222] develops a standalone binary classifier and trains it together with the original classification module. Ref. [223] combines both classifier extension and standalone classifiers. Such standalone modules introduce additional complexity to the model and necessitate well-coordinated training efforts. Notably, both refs. [222] and [223] utilize samples from unknown classes in training, which deviates from the OSFD setting. For single-pass models, a different approach [121] simply extends the original classifier to incorporate the abstention option. This minimal change renders it compatible with a wide range of network structures, including CNNs.

With only samples of known classes, training ACs to learn discriminative features and effective decision boundaries for unseen classes is challenging. It is essential to construct auxiliary samples that contrast with seen samples and represent potential unseen classes, ensuring the supervised learning of ACs. The creation of auxiliary samples for fault diagnosis can be approached in various ways. Options include sourcing samples from a different but related dataset [205], generating them via adversarial learning [220], or introducing alterations to ID training samples through perturbations [121], [122]. In fault diagnosis applications, obtaining another related dataset may not always be feasible. In the meantime, interpreting signals obtained through adversarial generation can be challenging, particularly in the context of vibration-based fault diagnosis. For bearing fault diagnosis, ref. [201] created a digital twin of the bearing system and used simulated data to assist training. However, such a simulation model needs expert knowledge of the bearing system and does not adapt well to general machine systems. In this chapter, we explore the option of generating auxiliaries by applying interpretable and general perturbations on the ID vibration signal from the training set.

In a prior study on fault diagnosis [121], a statistical perturbation named

Soft Brownian Offset was employed on ID samples to generate OOD samples as auxiliaries. However, these generated samples lack physically meaningfulness. Simpler yet interpretable perturbations like noise injection and input mix-up [224] should be considered as potential alternatives.

4.3 Proposed method

In this chapter, we propose a novel OSFD method based on vibration signals named EAC. It combines uncertainty measurements with abstaining options to determine whether an input sample falls into the OOD category.

4.3.1 EAC framework

The proposed EAC method consists of the following steps:

1. Data acquisition: Vibration signals are collected to construct a training and a testing dataset. The training dataset is collected when the machine is healthy and under some known faulty conditions, while the testing dataset will include one or more novel fault conditions unseen in the training dataset. The testing dataset may be collected under a different working condition than that of the training data.

2. Model training: Auxiliary training samples are generated based on the training dataset, and then the proposed EAC model (shown in Figure 4.1) is trained using both the training dataset and the auxiliary dataset.

3. Model inference: the trained EAC model is used to classify and report uncertainties for all the testing data.

4. Diagnosis: A final diagnosis is made based on the corresponding classification result and reported uncertainty for each test sample. A sample

will be recognized as 'unknown' if the reported uncertainty exceeds a pre-set threshold.

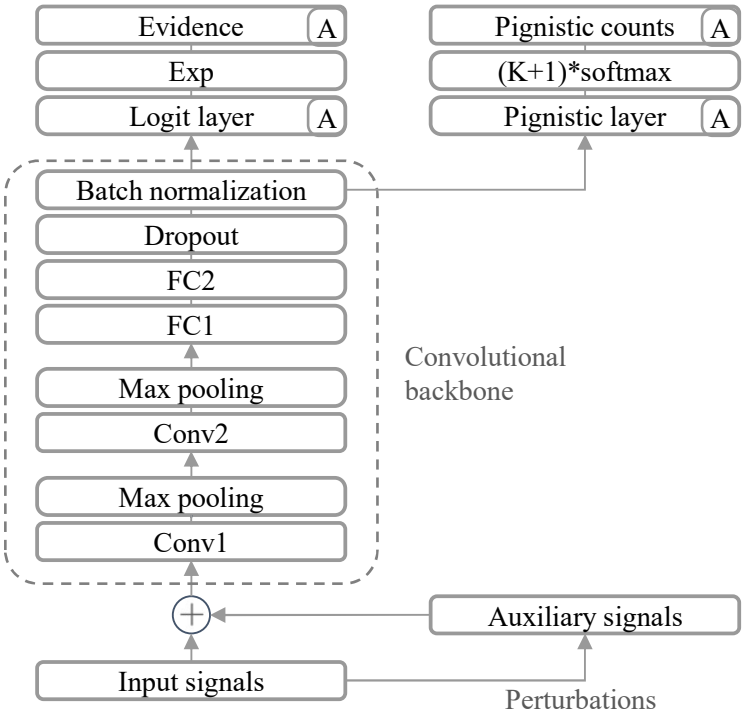


Figure 4.1: Structure of the proposed EAC with a two-layer convolutional backbone.

The structure of the proposed EAC model is illustrated in Figure 4.1, where the input vibration signals enter from the bottom and the output evidence and pignistic counts are displayed on the top. The convolution backbone, circled in dash lines, follows the structure of a standard LeNet-like CNN [140]. It has two convolutional layers (Conv1 and Conv2) and two fully connected (FC) layers to extract useful features from the input signals. The logit layer and the pignistic layer are both fully connected layers and they are designed to convert the features into evidence and pignistic counts (see Section 4.3.2), respectively. The letter 'A' attached on the right-hand side of the logit and pignistic layers indicates that these layers have an additional neuron to host the abstention option for unknown classes. Note that the steps to generate auxiliary signals

and pignistic counts are only used in model training but omitted during model inference.

Details about the perturbation operations for generating auxiliary signals will be presented in Section 4.3.3. The discussion of hyperparameters, including the kernel sizes of the convolutional layers, FC layer sizes, dropout rate, and a comparison with other methods, will be covered in Section 4.3.4.

4.3.2 EDL and L1 regularization

In the context of EDL, the neural network outputs (logits) serve to represent the weights of evidence that a sample carries for various categories [217]. These evidence values are subsequently employed as density parameters to create a Dirichlet distribution across categorical probabilities. In other words, rather than conventional models providing single-point estimates for the probability of each class, EDL offers a distribution encompassing a range of potential outcomes.

Following ref. [217], given K possible categories, the weight of evidence for the ith category is calculated as $c_i = e^{z_i}$, where z_i is the ith logit. A Dirichlet distribution $D(p|\alpha)$ with parameters $\alpha = [c_1 + 1, c_2 + 2, \ldots, c_K + 1]$ will be the output of the EDL model. The probability of the input sample belonging to category i can be estimated as Eqn. 4.2 and the uncertainty of an EDL model can be quantified using the entropy of the predicted categorical probabilities as shown in Eqn. 4.3.

$$p_i = \frac{\alpha_i}{\sum_{j=1}^{K} \alpha_j} \quad for\ i = 1, \ldots, K \tag{4.2}$$

$$H[y|p] = -\sum_{i=1}^{K} p_i \log p_i \tag{4.3}$$

To train EDL models, ref. [217] proposed to minimize the Mean Square Error (MSE) writes Eqn. 4.4 and the KL divergence writes Eqn. 4.5.

$$L_{MSE} = \sum_{i=1}^{K}(y_i - \hat{p}_i)^2 + \frac{\hat{p}_i(1-\hat{p}_i)}{1+\sum_{j=0}^{K}\alpha_j} \tag{4.4}$$

$$L_{KL} = KL\{D(p|\widetilde{\alpha}) \,||\, D(p|[1,1,\ldots,1])\} \tag{4.5}$$

where $\hat{p}_i$ is the estimated probability on class i, y_i is the ith element of the one-hot label vector y, and $\widetilde{\alpha} = y + (1-y) \odot \alpha$ only include evidence for the wrong classes. Ref. [225] later introduced an additional pignistic risk term:

$$L_{risk} = \sum_{i=1}^{K} R_{yi}(c_i + d_i) \tag{4.6}$$

where R is a $K \times K$ risk matrix with element R_{yi} indicating the risk of misclassifying a sample of class y to the class i, and $d = K * softmax\,(s)$ with s standing for the output of an additional pignistic layer in parallel to the logit layer. By default, all diagonal elements R_{ii} are 0 and others are 1.

Ideally, the evidence produced by the logit layer should be sparse and concentrated in one or a few categories. In the original EDL [225], although the KL-divergence term suppresses evidence on the wrong classes and forces evidence to be concentrated on the correct class, it may shrink the total evidence values. This may increase the overall loss and make the entropy-based UQ less sensitive. To address this, we propose to add L1 regularization (see Eqn. 4.7) to the logit layer, facilitating the achievement of such sparsity.

$$L_1 = \sum |\theta_{logit}| \tag{4.7}$$

Unlike conventional L1 regularizations that are applied throughout the entire network, we specifically penalize the weights for the logit layer, denoted as θ_{logit}. The pignistic layer is free from the constraints of both KL-divergence and L1 penalty to better focus on the misclassification risks. With L1 regularization and KL divergence included in the loss function, both concentration

and sparsity of evidence can be enforced. Better and more reliable estimates for evidence and uncertainty values can then be achieved. The final loss functional of our modified EDL becomes:

$$L_{total} = L_{MSE} + \lambda_t L_{KL} + \beta L_{risk} + \gamma L_1 \tag{4.8}$$

where λ_t, β, γ are coefficients for the three regularization terms respectively. $\lambda_t = \min\left(1, \frac{t}{10}\right)$ is an annealing coefficient based on the number of epochs trained t, while β and γ are preset.

It is important to select proper β and γ to ensure that the pignistic risk term and the L1 regularization term have similar or smaller magnitudes compared with the MSE term. Otherwise, they will over-power the other terms and lead to a high classification error. The KL divergence term will automatically approach zero with its coefficient λ_t decreases after each training epoch.

4.3.3 AC and auxiliary samples

Integrating the concept of AC into EDL provides a dual-screening mechanism to recognize OOD samples. In the case that an OOD sample was falsely reported with low uncertainty, an AC can still recognize and separate it from other 'known' classes. More importantly, gathering and allocating evidence to the abstention class allows the model to learn more effective fault-related features. Note that the pignistic layer is also equipped with the abstention option, enabling the fine-tuning of the trade-off between abstaining and the possibility of misclassification.

To train ACs effectively, it is crucial to design an auxiliary dataset that accurately represents the abstention class in contrast to the known classes. Due to the complex mechanical structure and varying working conditions of rotation machines, their vibration signals are often noisy, and non-stationary [30]. It takes expert knowledge and advanced signal processing to analyze and

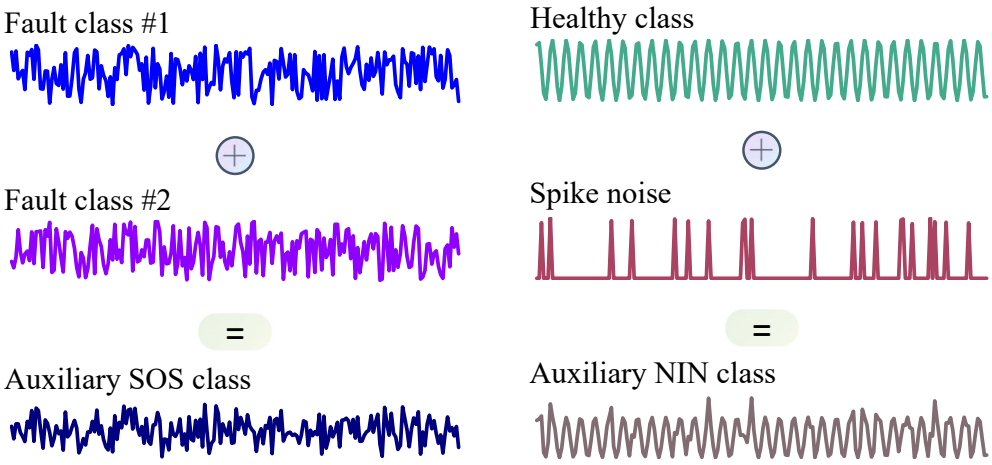
simulate signals with a certain fault condition. However, in OSFD settings, we can utilize both the physical characteristics and the 'known' training data. Training samples collected under a healthy condition can be used as a background to inject fault-indicative signal components, while samples of faulty classes can be used to provide fault-related information. In this chapter, given a training dataset with signals of the healthy class and two or more faulty classes, we propose two input-level operations for creating auxiliary signals:

1. Superposition of Signals (SOS): The SOS operation, as demonstrated in Figure 4.2a, involves combining two signals from different known fault classes to create a new auxiliary signal. This combination is performed via element-wise addition (denoted as $\oplus$ in Figure 4.2). This approach is inspired by the image mix-up method [224], which generates augmented samples lying between two classes. By applying SOS, we create a pool of signals that fall in between two fault classes, and these are labeled as the abstention class.

2. Noise Injection (NIN): The NIN operation is designed to produce auxiliary signals that are distinct from all known classes. It includes the following steps:

    - Generate a random series where each element is drawn from a uniform distribution in the range [0, 1].
    - Set the elements of the generated random series with values between 0 and 0.9 to 0.
    - Add the normalized random series to a healthy sample element-wise.

Figure 4.2b illustrates the NIN operation, providing an example of a noise signal generated through the three aforementioned steps, along with its im-



(a) SOS operation. (b) NIN operation.

Figure 4.2: Demonstration of the two proposed auxiliary signal generation operations.

pact on the original signal. The idea of injecting impulse into healthy signals is inspired by other simulation-driven methods including ref. [201]. The injected noise is characterized by random spikes, effectively emulating the impulsive patterns typically found in vibration signals caused by faults [226]. This approach enables the efficient generation of auxiliary samples that accurately represent the 'not healthy' state without the need for complex and time-consuming simulation models.

All the samples in the training dataset will be used to generate auxiliary training samples and a fixed number of auxiliary samples will be randomly selected for training. For simplicity, in this chapter, the number of samples from each 'known' class and auxiliary samples used are the same. The auxiliary dataset will be half SOS samples and half NIN samples. Before merging, each type of auxiliary sample is normalized to have a zero mean and a unit standard deviation. This normalization ensures consistency and comparability in the dataset.

Although simple to operate, the SOS and NIN operations can generate effective auxiliary training samples, equipping ACs with the ability to make

informed decisions when confronted with unknown fault classes.

4.3.4 Compared methods and hyperparameters

The proposed EAC method is primarily based on studies of EDL and AC, and these should be compared to demonstrate the effectiveness of our proposed innovations. To start, a baseline using a standard CNN with a softmax layer is established. Then, the EDL algorithm described in Section 4.3.2 is implemented following ref. [225]. An EDL variant with L1 regularization on the logit layer, denoted as EDL-L1, is also included in the comparison. In addition, to demonstrate the effectiveness of our designed auxiliary data, these data were incorporated into the training of the three methods mentioned above. The resulting augmented methods are denoted as CNN+, EDL+, and EAC. Lastly, the effectiveness of our designed auxiliary data is examined in comparison to real data from another dataset, as presented in ref. [205]. By using healthy signals from another machine as the auxiliaries, EAC is mutated into EAC-R, completing the 7-way comparison. All the models are extended with the abstention option to suit OSFD tasks, and the entropy of output probabilities serves as their UQ metric.

To ensure that the compared methods are on equal footing, we employ the same CNN backbone described earlier in Section 5.3 for all methods. The choice of a small network for the backbone is intentional to prevent overfitting and ensure that it does not overpower the studied learning algorithms. The Adam [44] algorithm is used to optimize the learnable parameters in the neural network. The shared hyperparameters for the backbone structure and the optimizer are detailed in Table 4.1. For the coefficients of the loss terms in Eqn. 4.8, we use $\beta = 0.001$ and $\gamma = 0.0001$ based on our trial and error with the later-presented experiment data to make sure that all the loss terms in Eqn. 4.8 are at a similar magnitude.

Table 4.1: Selected hyperparameters for all the compared methods.

Hyperparameter names	Value
Length of input signals	2048
Conv1 & Conv2 kernel size	25
No. channels (Conv1)	64
No. channels (Conv2)	128
Max pooling kernel size	4
No. output neurons (FC1)	128
No. output neurons (FC1)	16
Dropout rate	0.5
Learning rate	0.0001
Batch size	50
No. training epochs	100

In all the methods, a threshold on their UQ metric (entropy values calculated using Eqn. 4.3) must be determined. Samples with entropy values lower than this threshold will be classified into known classes, while those exceeding the threshold will be abstained. Typically, this threshold is determined based on the statistical quantiles of an entire testing set, as in ref. [215]. However, for practicality in this chapter, the authors opt to avoid using additional testing samples and instead set the threshold at 80% of the maximum possible entropy, which is $\ln(K+1)$. For ease of interpretation, entropy values are normalized by dividing them by $\ln(K+1)$, ensuring that uncertainty values fall within the range [0,1], and the threshold becomes 0.8.

4.4 Experiment

In this study, vibration data collected from two distinct test rigs are utilized to validate the proposed methods. Various fault diagnostic tasks are examined, encompassing the challenges of diagnosing both gear and bearing faults, recognizing both known and unknown fault classes, and conducting diagnosis under varying machine rotating speeds and load levels. The implementation of all methods is carried out using PyTorch and executed on a computer equipped

with an AMD Ryzen 7 5700G CPU and an Nvidia RTX 3060 GPU.

4.4.1 Datasets and tasks

The THU-EPE dataset, collected in 2019 at the Department of Energy and Power Engineering, Tsinghua University [118], comprises five distinct fault classes: healthy (H1), sun gear tooth crack (SC), sun gear tooth broken (SB), planet gear tooth crack (PC), and sun gear tooth crack (PB). Each class consists of 2000 training samples and 500 testing samples. On the other hand, the THU-ME dataset, collected in 2021 at the Department of Mechanical Engineering, Tsinghua University [227], focuses on bearing faults in a planetary gearbox. It includes IB-IRF, IB-ORF, PB-IRF, and PB-ORF, where IB and PB represent input shaft and planet bearing, and ORF and IRF stand for outer and inner race faults, respectively. The healthy class in the THU-ME dataset is denoted as H2. In this dataset, each class comprises 1764 training samples and 588 testing samples. For both datasets, vibration signals are measured using accelerometers mounted vertically on the top of the gearbox cases. Each sample consists of 2048 sample points, and the sampling rate is 20,000 Hz.

Table 4.2: Description of tested tasks.

Task	Training classes	Unseen class	Speed/Load (RPM/Nm)	
			Training	Testing
T1	H1, SB, PC, PB	SC	1500/0	1500/0
T2	H1, SC, PC, PB	SB		
T3	H1, SC, SB, PB	PC		
T4	H1, SC, SB, PC	PB		
T5	H2, IB-ORF, PB-IRF, PB-ORF	IB-IRF	1200/-0.61	1200/-0.61
T6	H2, IB-IRF, PB-IRF, PB-ORF	IB-ORF		
T7	H2, IB-IRF, IB-ORF, PB-ORF	PB-IRF		
T8	H2, IB-IRF, IB-ORF, PB-IRF	PB-ORF		
T9	H1, SB, PC, PB	SC	1500/0	**1200**/0
T10	H2, IB-IRF, IB-ORF, PB-IRF	PB-ORF	1200/-0.61	1200/**-1.65**

The study encompasses a total of 10 testing tasks, each outlined in Table

Table 4.3: Test classification accuracies on each task by compared methods.

Task	CNN	CNN+	EDL	EDL+	EDL-L1	EAC-R	EAC
T1	0.8516	0.8084	0.8856	0.9079	0.8535	0.8541	**0.9402**
T2	0.9206	0.9208	0.8949	**0.9548**	0.8935	0.8297	0.9241
T3	0.8400	0.9138	0.6223	0.8982	0.6863	0.7548	**0.9656**
T4	0.8570	**0.9012**	0.7363	0.8967	0.7918	0.8488	0.8942
T5	0.8085	0.8083	0.7349	0.8801	0.7651	0.8862	**0.8874**
T6	**0.5764**	0.5143	0.4731	0.4848	0.4609	0.4752	0.5317
T7	0.8803	0.9181	0.7803	0.8717	0.8171	0.8298	**0.9511**
T8	0.8041	0.7990	0.8396	0.8799	0.8342	0.8343	**0.8848**
T9	0.7463	0.7994	0.7380	0.7904	0.7234	0.7833	**0.8743**
T10	0.8091	0.7979	0.7946	0.8064	0.7816	0.7994	**0.8325**
Ave.	0.8094	0.8181	0.7500	0.8371	0.7607	0.7896	**0.8686**

4.2. In tasks T1 to T4, the four fault classes in the THU-EPE dataset are alternately designated as the unseen class during testing. Similarly, in T5 to T8, this approach is employed using the THU-ME dataset. For T1 to T8, both the training and testing data are acquired under identical working conditions. The remaining two tasks introduce changes in working conditions between the training and testing data. T9 involves adaptations in two different rotating speeds, while T10 transitions from one load level to another. In the case of the EAC-R method, auxiliary data for T1-T4 and T9 are derived from class H2, collected under 1500 RPM and 0 load from the load motor. On the other hand, auxiliary data for T5-T8 and T10 are obtained from class H1, collected under 1200 RPM and 0 load.

4.4.2 Results and analysis

Classification accuracies for different methods are presented in Table 4.3, with each column representing a different method, and each row corresponding to a tested task. The reported accuracy values for T1 to T10 are the averages of 10 repeated runs, while the 'Ave.' row provides the average accuracy across all tasks.

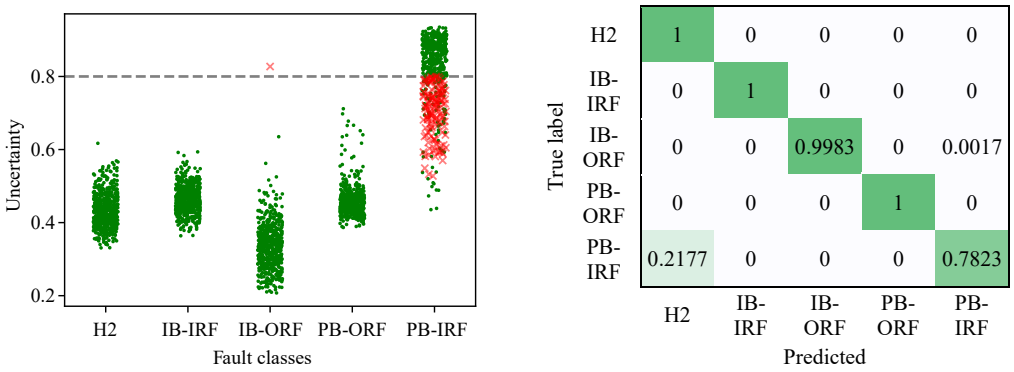
In Table 4.3, it is shown that the average accuracy of EAC surpasses the

baseline CNN by 5.92% and the state-of-the-art EDL model by 11.86%. Notably, the introduction of L1 regularization on the logit layer in the EDL-L1 model results in an improvement of 1.08% when compared to the EDL model. Furthermore, when comparing EDL+ and EAC, the addition of L1 regularization enhances the accuracy by 3.15%, highlighting the effectiveness of L1 regularization.

The advantages of utilizing designed auxiliary data are also evident in Table 4.3, leading to improvements of 0.87%, 8.71%, and 10.79% in the case of the CNN, EDL, and EDL-L1 models, respectively. EAC-R, which employs real signals from another dataset, achieves a 2.89% higher accuracy than EDL-L1. Nonetheless, EAC, which utilizes synthetic signals, consistently outperforms EAC-R across all tasks. EAC excels in accuracy in 7 out of 10 tested tasks, including T9 and T10, where the working conditions differ between training and testing data. Notably, in T2 and T4, although EAC does not report the best accuracy, the top-performing methods both utilized our designed auxiliary signals. T6 stands as an anomaly where none of the tested methods achieve a classification accuracy of 60%. Overall, the proposed auxiliary training data allows our model to outperform the original EDL model [225] and model trained using other datasets as auxiliary as reported in ref. [205].

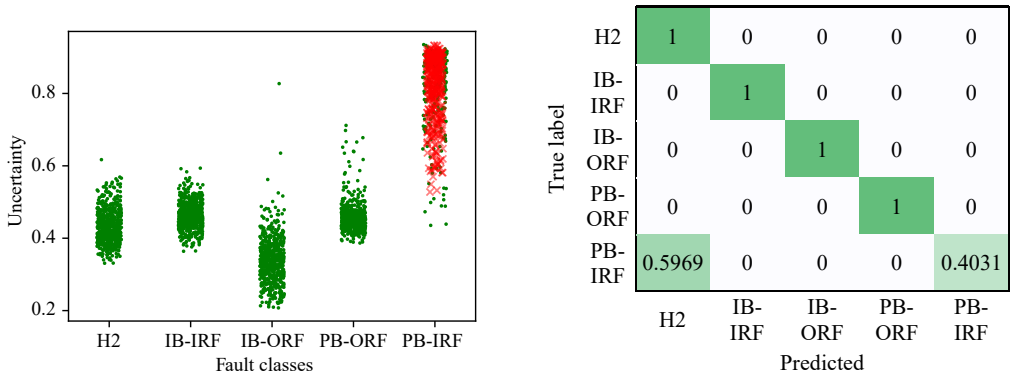
To illustrate the functionality of the UQ metric and threshold, Figure 4.3 displays the classification outcomes and uncertainty values for task T7 as reported by the proposed EAC methods. In Figure 4.3a, a strip plot is used to represent all test samples, with their true fault classes on the horizontal axis and uncertainty values on the vertical axis. Samples that were classified correctly are denoted by dots, while those with false predictions are marked with 'x'. The plot reveals that samples from the training classes exhibit significantly lower uncertainty values compared to the unknown class PB-IRF. By applying the uncertainty threshold of 0.8, samples with higher uncertainty

values can be accurately identified as belonging to the unseen fault class. Figure 4.4 presents the results on T7 by EAC without applying the threshold. When comparing the two confusion matrices in Figure 4.3b and Figure 4.4b, it becomes evident that applying the threshold leads to a 37.92% increase in the number of correctly classified PB-IRF samples, resulting in an overall classification accuracy improvement of 7.55%.



(a) Strip plot. (b) Confusion matrix.

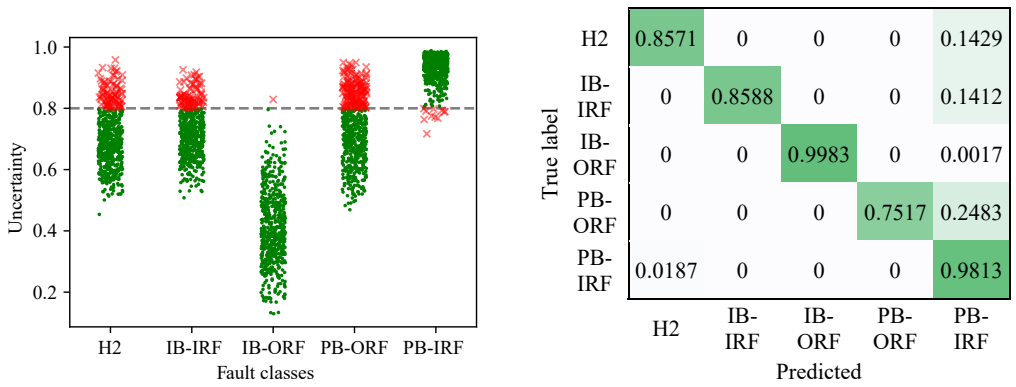
Figure 4.3: fault classification and uncertainty quantification for T7 by EAC.



(a) Strip plot. (b) Confusion matrix

Figure 4.4: fault classification and uncertainty quantification for T7 by EAC without thresholding.

To highlight the effectiveness of our proposed L1 regularization, Figure 4.5 displays the strip plot and confusion matrix for task T7 as obtained by the EDL+ model. As shown in Figure 4.5a, the EDL+ model [225] may assign



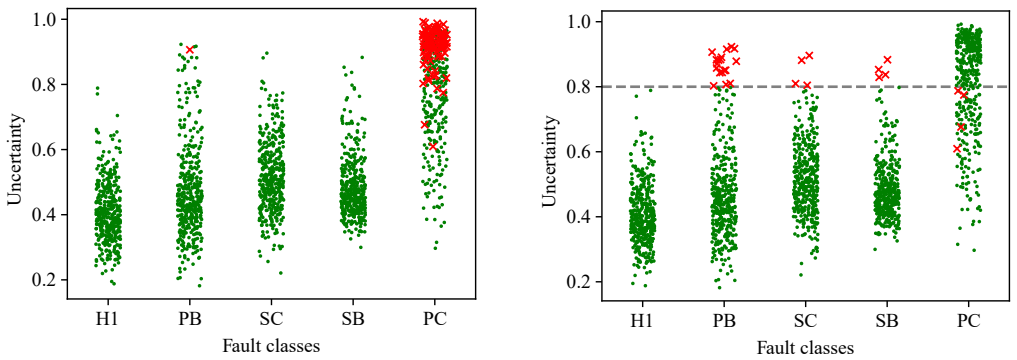
(a) Strip plot. (b) Confusion matrix.

Figure 4.5: fault classification and uncertainty quantification for T7 by EDL+.

high uncertainty values for samples from both known and unknown training classes. This makes it hard to recognize the unseen PB-IRF class based on uncertainty. On the other hand, as shown in Figure 4.3a, the proposed EAC model is capable of generating well-concentrated and low uncertainties for the known classes, while assigning higher uncertainties for the unseen class PB-IRF. As depicted in Figure 4.5b, even though the EDL+ model correctly recognizes 98.13% of the PB-IRF samples, the presence of false classifications within the training classes reduces the overall accuracy to 88.95%, which is 6.66% lower than the accuracy achieved with L1 regularization.

The EAC method's abstention classifier itself contributes significantly to recognizing unseen fault classes. As demonstrated in Figure 4.6a, EAC can already identify 74.75% of the testing samples from the unseen class PC without applying the uncertainty threshold. With the threshold in place, only 4 misclassifications are observed out of the 400 test samples from the unseen class, as shown in Figure 4.6b.

To analyze the mechanism behind the successful recognition of the unseen class states, t-SNE visualization [228] of features formed by CNN, EDL-L1, and EAC are compared in Figure 4.7. Each t-SNE plot also displays decision boundaries generated using the k-NN algorithm [47]. The features used for



(a) Classification w/o threshold. (b) Uncertainty threshold at 0.8.

Figure 4.6: visualizations of uncertainty values and thresholding for test samples in T3.

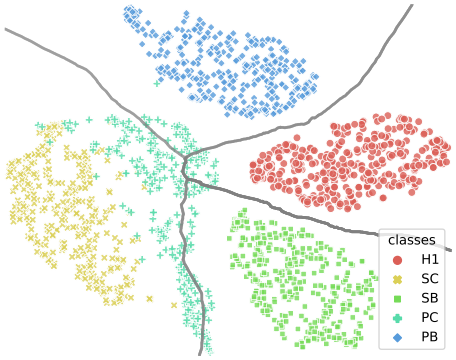
visualization are based on the output of the convolutional backbone (as seen in Figure 4.1).

Upon examining the four t-SNE plots, a few key observations can be made:

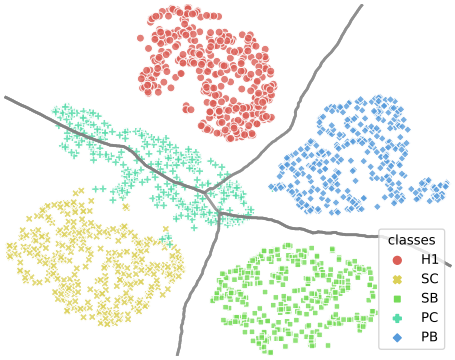
1. Comparing CNN in Figure 4.7a to the other three methods in Figures 4.7b, 4.7c, and 4.7d, the latter three plots reveal clearer separations between the seen class SC ('x' markers) and the unseen class PC ('+' markers). This indicates that better fault-discriminative features are formed using evidential learning, and these features are also applicable to the unseen fault type PC.

2. The inclusion of designed auxiliary samples enables EAC to create more sophisticated decision boundaries in Figure 4.7d, effectively grouping many of the PC samples with them. This is how the EAC model correctly recognizes these samples from the unseen class.

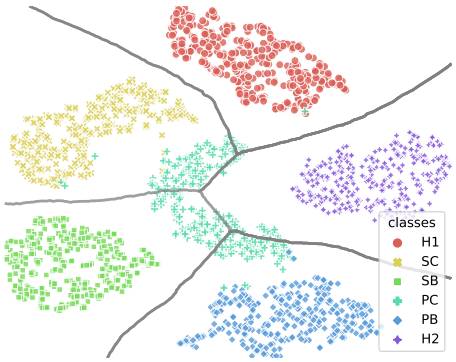
3. In Figure 4.7c, the features and decision boundaries are formed using class H2 as an auxiliary. However, the H2 samples and the PC samples are well separated, making it impossible to recognize them as one OOD class together.



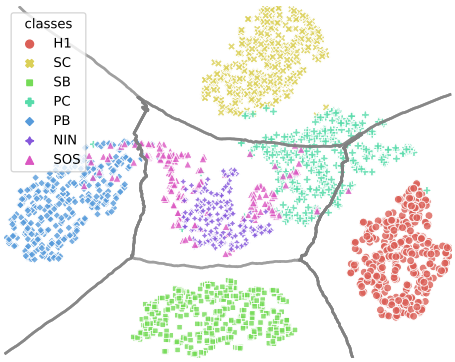
(a) CNN (no auxiliary, softmax learning).



(b) EDL-L1 (no auxiliary, evidential learning).



(c) EAC-R (real auxiliary, evidential learning).



(d) EAC (synthetic auxiliary, evidential learning).

Figure 4.7: t-SNE plots of features formed in T3 by CNN, EDL-L1, and EAC (best view in color).

These observations highlight the advantages of our devised auxiliary training. It enables the development of better fault-discriminative features and decision boundaries through the use of auxiliary samples, which, in turn, aids in the effective recognition of unseen fault classes.

Setting a proper uncertainty threshold is crucial for the accuracy of the EAC models. Figure 4.8 shows how the diagnostic accuracies for tasks T3 and T7 change given different pre-set uncertainty thresholds. As shown by the two call-out boxes, the highest test accuracies are obtained using thresholds of 0.58 and 0.86, achieving accuracies of 99.18% and 98.7% respectively for T3 and T7. That is, different datasets and tasks may have different optimal

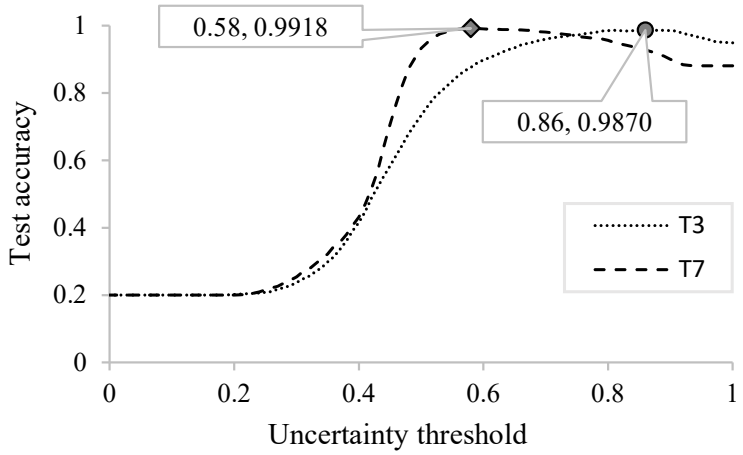


Figure 4.8: test accuracy vs. uncertainty threshold by EAC for T3 and T7.

thresholds. For T7, as shown in Figure 4.3a, a lower threshold may help to recognize more samples of unseen classes. However, successful classifications of the samples from the known classes might be wrongly overturned. For T3, as shown in Figure 4.6b, a higher threshold may induce fewer misclassifications for the PB, SC, and SB classes. If testing samples from different classes are available, good thresholds can be selected using existing methods such as ref. [215]. With a threshold of 0.8 set in this chapter, EAC achieves accuracies of 98.5% and 95.61% for T3 (Figure 4.6b) and T7 (Figure 4.3a), respectively. Although not optimal, these test accuracies are higher than those not using thresholds by 3.6% and 7.55%.

4.5 Conclusion

In this study, we present a novel OSFD method named EAC for diagnosing bearing and gear faults. Our approach, including its innovations, has proven effective in recognizing previously unseen fault classes and accurately quantifying classification uncertainties. The devised auxiliary training samples are readily obtainable and significantly contribute to enabling deep learning models to establish fault-discriminative features and efficient decision boundaries

for OSFD tasks. Additionally, the proposed L1 regularization enhances the model's capability to estimate classification uncertainty. The collaborative integration of AC and uncertainty thresholding demonstrates its effectiveness in achieving accurate OSFD results.

In future works, the applicability of EAC can be extended to other types of sensor data beyond vibration signals, such as acoustic and eddy current signals. More sophisticated auxiliary signals can be designed to aid the training of EAC models. Better approaches to selecting uncertainty thresholds are needed to ensure the effectiveness of UQ-based OSFD methods.

Chapter 5

Continual learning for fault diagnosis considering variable working conditions

5.1 Introduction

Machine learning techniques, especially Neural Networks (NNs), have found extensive application in vibration-based machine fault detection and isolation (FDI) [34]. With a short segment of measured vibration signal, a well-trained NN can identify machine faults such as gear tooth crack [50] and bearing defects [229], [230] in a matter of milliseconds. This automation can reduce the need for human labor in FDI, enabling condition-based maintenance [8] on a larger scale.

When developing NNs for FDI problems, it is critical to consider both the training data and the training method. However, existing training methods are typically developed with two assumptions that may not hold in real-world FDI applications [54], [115]: (1) they assume that training data for all classes of interest are available, and (2) they assume that training and testing data are drawn from the same distribution. In real-world scenarios, machines such as gearboxes may develop different types of faults and go through variable working conditions, making these assumptions unreliable.

To overcome these challenges, Continual Learning (CL) [58], [231] can enable us to build FDI models with a few classes and data distributions and then continuously improve the model with more data. This approach allows us to incorporate new fault classes and adapt to changes in working conditions, making it a more effective solution for real-world FDI applications.

For assumption (1), new classes of training data should be considered once an unseen fault type occurs. Typically, only one or a few types of faults can be detected within one maintenance cycle [232]. In a series of maintenance cycles, different types of faults may occur. This kind of CL setting is termed class-incremental learning (Class-IL) [233]. NNs need to adapt to new classes and remember all the old classes. A trivial solution is to use both old and new data to re-train a model from scratch for every newly discovered fault type. However, in many FDI applications, this is not possible due to privacy, legal, and technical reasons [126], [234]. Re-training is also computation and storage intensive which make it not suitable for real-time FDI applications [235]. A more viable and efficient approach is to first train a single model using currently available data (pre-train) and then fine-tune [236] this model when a new class of data is collected. Fine-tuning a pre-trained model can provide two advantages: (1) it is faster than training from scratch, and (2) less storage of training data is needed. Existing works including [237]–[239] have studied Class-IL for FDI applications. However, these works did not discuss the impact of variable working conditions.

For assumption (2), shifts of data distributions should be considered in most FDI applications. Essentially, the distributions of testing data will be different than that of the training data due to working conditions changes, including rotating speed change [86], [240] and load level change [15]. In CL, this is called a domain-incremental learning (Domain-IL) setting [233]. Traditional machine learning based FDI methods, for instance, ref. [82] did not consider

distribution differences between the training and the testing data, thus may perform poorly under variable working conditions. Specialized signal processing methods such as [241], [242] may be applied to reduce the impacts of working condition changes, but they usually require calibrations for different patterns of working condition changes. To avoid such expert-dependent and laborious calibrations, domain adaptation (DA) methods [61] may be developed to train NNs that are robust to working condition changes. By treating different working conditions as different distribution domains and applying DA methods, NNs can adapt to different rotating speeds [115], different load levels [10], or even different machines [55]. However, current works on DA-aided FDI methods did not consider the Class-IL problem discussed earlier. They assume all the fault types are readily available for training and no new fault types will show up.

To relax the two assumptions mentioned earlier and address the challenges posed by both the Class-IL and Domain-IL problems in many FDI applications, it is important to consider the Task-IL setting [233] in CL. In this work, using the Task-IL approach, we build an NN-based FDI system that can deal with both fault class increments and working condition changes. The key challenge is to simultaneously maintain the model's knowledge obtained from old data while being able to quickly adapt to new data and new tasks. This challenge is known as the stability-plasticity dilemma for both artificial and biological NNs [59]. In CL, stability enables the model to remember what has been learned while plasticity allows the model to learn new tasks. In existing FDI research works, using the plasticity of NNs, Xing et al. [41] showed that CL can enable the diagnosis of a new type of bearing fault using only a few new training samples. Zhang and Gao [234] developed a CL algorithm that can quickly recognize new faults in a wind turbine system. However, they did not investigate the stability of NNs or the Catastrophic Forgetting (CF) problem

[60]. That is, NNs may perform poorly on historical fault types upon training on new ones. Maschler et al. [125] studied the CF problem for remaining useful life prediction. However, only a single type of fault (wear) is considered in this work. Li et al. [54] proposed a DCTL-DWA framework to address the Class-IL and domain and domain adaptation problem. Their approach is to solve the Class-IL problem given a single fixed source and a fixed target distribution to adapt. In real applications, however, the changes in working conditions may occur multiple times.

This chapter presents a novel CL-based FDI method that incorporates DA within the CL paradigm to tackle both the problems of fault class incrementation (Task-IL) and changes in machine working (Domain-IL). A Task-Balanced Sampling (TBS) method is designed to help CL models to better remember diagnostic tasks with different fault classes. A multi-way DA is introduced to adapt different working conditions among tasks using only healthy data. We name the proposed method TBS-DA to highlight our novelties. The method is suitable for use with various types of machines, and we test it on experiment datasets for both gear faults and bearing faults. We examine changes in fault location, fault type, rotating speed, and load level to evaluate the effectiveness of the method. The contributions of this chapter can be summarized as follows:

1. We propose a novel CL-based FDI method named TBS-DA that addresses both fault class incrementation and changes in machine working conditions.

2. We introduce a task-balanced replay mechanism for training data to aid the model in remembering all diagnostic tasks.

3. We apply multi-way DA to healthy data from different tasks to improve diagnostic accuracy under variable working conditions.

The remaining parts of this chapter are structured as: Section 5.2 reviews existing CL methods; Section 5.3 describes baseline and proposed methods; Sections 5.4 and 5.5 present two experimental case studies; and Section 5.6 concludes this chapter.

5.2 Review on CL methods

CL is also referred to as Lifelong Learning in the literature [58]. In CL, a sequence of learning tasks is deployed for the model to learn. The goodness of the model will be judged based on its performance on all the learned tasks. This is aligned with most FDI applications - different mechanical faults are likely to come in a sequential manner and the order of learning must go with the order of fault occurrences.

There are mainly three categories of CL methods [231], including regularization-based, parameter isolation, and replay methods. Regularization-based methods [243], [244] use penalty functions to constrain the change of models' parameters when learning new tasks. Parameter [245], [246] isolation is to allocate parts of the NN or subsets of the model parameters to specific tasks. These two types of methods enforce a strict ban on accessing training data of prior tasks. However, they may require storage for prior or task-specific model parameters and extra computation to get parameter importance. In this chapter, we focus on replay methods that utilize training data from prior tasks. They are simple, efficient, and can be used together with the other two types of methods.

Considering possible storage constraints, replay methods either store only a subset of representative samples for each task in a memory buffer (rehearsal) [247] or maintain generative models to produce pseudo samples that mimic past data (pseudo-rehearsal) [248]. Those stored or generated representative samples are often termed as exemplars in the literature. Rehearsal methods

are typically more straightforward compared to pseudo-rehearsal. Maintaining generative models for pseudo-rehearsal methods in a continual fashion can be cumbersome [249]. Rehearsal methods that directly replay stored data or representations of data may be more efficient. Experience Replay (ER) [247], [250] is the plainest version of all rehearsal methods, which adds a batch of stored data of past tasks to each batch of new data from the current task. In this way, the model is jointly trained by both the past tasks and the current tasks to have good global performance. Since the stored past data are constrained to be a small size, it is crucial to tackle overfitting problems in replay methods. To address overfitting, ref. [251] proposed to select samples from the memory buffer based on gradient, and ref. [252] is based on samples' difficulty, as opposed to random sampling in refs. [247]. Besides, strategies on how to choose samples to store are also studied [253]. Last but not least, efficient utilization of saved samples is key to successful CL. Refs. [254] use gradients of stored samples to constraint the learning of the model and avoid CF. Ref. [255] classifies samples into their nearest-mean-of-exemplars so that the classifier is robust against changes in the feature representation. These above replay methods are all developed for image recognition, text classification, and reinforcement learning.

In this study, we focus on replay methods and propose to enhance traditional ER methods to improve fault classification performance. Note that replay methods can be used together with the other two types of methods.

5.3 Methodology

5.3.1 Notations

To formalize the problems and the CL settings in this study, we adopt notations from ref. [61]. A domain $\mathcal{D} = \{\mathcal{X}, P_X\}$ in which $\mathcal{X}$ is a feature space and P_X

is a marginal probability distribution of dataset $X = \{x_1,\ x_2,\ \ldots,\ x_n\}$ with $X \in \mathcal{X}$. A task $\mathcal{T} = \{\mathcal{D}, \mathcal{Y}, f(\cdot)\}\, where \mathcal{Y}$ is a label space and $f(\cdot)$ is an objective predictive function. This function $f(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$ is not observed but to be learned from training data triplets (x_i, y_i, t_i), where $x_i \in X$, $y_i \in \mathcal{Y}$, and t_i denotes which task does this triplet belongs to.

In Task-IL, the goal is to learn a sequence of tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_L$ throughout N_t learning phases, and a task $\mathcal{T}_l$ requires a predictive function that maps its unique $\mathcal{X}_l$ in domain t to $\mathcal{Y}_l$. We denote the learned predictive function after training phase l as f_l and its corresponding parameters as θ_l. Given a sequence of datasets $X_1, X_2, \ldots, X_{N_t}$, which are respectively drawn from domains $D_1, D_2, \ldots, D_{N_t}$, f_l should only be trained using datasets from X_1 to X_l, but not data from X_{l+1} to X_{N_t} . Furthermore, during the lth training phase, access to previous datasets (from X_1 to X_{l-1}) is very limited in CL settings.

For FDI, a task $\mathcal{T}_l$ is to learn a f_l that can identify fault classes in label space $\mathcal{Y}_l$ under a certain working condition characterized by a domain $\mathcal{D}_l$. For this domain $\mathcal{D}_l$, a training dataset $X_l = \{x_1,\ x_2,\ \ldots,\ x_{n_l}\}$ which contains n_l vibration signal segments will be collected from the machine of interest. With fault class label y_i and task ID t_i corresponding to each data sample x_i, models can learn an objective f under the CL paradigm.

5.3.2 Neural network structure

In this work, one dimensional Convolutional Neural Networks (1DCNNs) [144] will be used across all different compared CL methods. CNN models are used in this chapter considering their strong feature learning abilities. They are suitable for signal processing and can form meaningful signal filters to extract fault discriminative features for fault diagnosis purposes as demonstrated in refs. [16], [144], [238]. For the structure of the 1DCNN, we refer to the very first

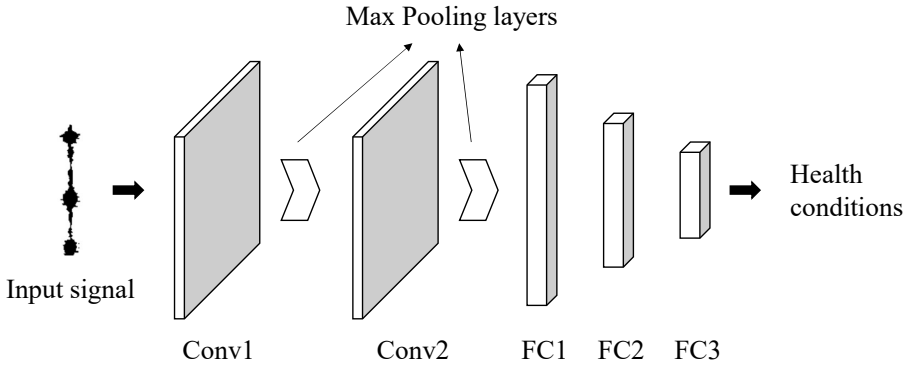


Figure 5.1: General structure of the used 1DCNN.

designed CNN [140] and use 2 convolutional (Conv) layers, 2 (max) pooling layers, and 3 fully connected (FC) layers. Such a simple structure can help us better focus on the learning algorithms. The general structure of the used 1DCNN is shown in Figure 5.1. Standard Rectified Linear Units (ReLUs) [194] are appended to each pooling layer, FC1, and FC2 to induce non-linearity in the network. The output from the FC3 layer will either be used to calculate Cross Entropy (CE) Loss [256] in the training stage or to generate predictions in the testing stage. Note that the input data will flow from the left-hand side of Figure 5.1 to the right-hand side.

There are several structural parameters to select in 1DCNN. For a Conv layer, a kernel size and the number of input and output channels are needed. For a max pooling layer, its kernel size and stride should be set reasonably to reduce the number of features by a certain ratio. For an FC layer, the numbers of input and output are required. Note that the number of input channels of Conv2 will be equal to the number of output channels of Conv1, and the input of an FC layer should be equal to the output of its previous layer. The number of input channels is determined by the number of input signal channels and the output of FC3 is determined by the total number of fault classes of interest. Parameter selection for this chapter will be presented in Section 5.4.2.

5.3.3 Baseline methods

Solving classification problems in machine learning is to train a model that can classify samples x to one class y out of others. This is usually done via minimizing a loss term:

$$Loss = \frac{1}{n} \sum_{x \in X} L(y, f_\theta(x)) \tag{5.1}$$

where n is the number of samples in dataset X, $L(\cdot)$ stands for a loss function such as the Cross-Entropy Loss [256], $f_\theta(\cdot)$ is the predictive function for the model to learn, and θ stands for the parameters of f.

The plainest CL approach is to train the model to fine-tune the model for each task. Given a sequence of N_t tasks and datasets $X_1, X_2, \ldots, X_{N_t}$, fine-tune is to minimize Eqn. 5.1 for N_t times for each task. When fine-tuning for the lth task, only the new dataset X_l will be used. This is prone to cause CF of the previous tasks and the diagnostic accuracy of some learned fault classes may become very low.

ER leverages limited access to $X_1, X_2, \ldots, X_{l-1}$ during the lth training phase. A set of exemplars X_e are randomly drawn from $X_1, X_2, \ldots, X_{l-1}$ and saved for replay training, and the corresponding minimization term becomes:

$$Loss_{ER} = \frac{1}{n_l} \sum_{x \in X_l} L(y, f_\theta(x)) + \frac{1}{n_e} \sum_{x \in X_e} L(y, f_\theta(x)) \tag{5.2}$$

where n_e is the number of stored exemplars. The exemplar set X_e (or buffer set) has an upper limit for the number of stored exemplars (buffer size B). A larger buffer size will lead to higher memory costs but will be more helpful in combating CF. Different buffer sizes will be tested in the experiment study (see Section 5.4 and 5.5). It is also critical to establish a reasonable mechanism to update the exemplar set. This will be discussed in Section 5.3.4.1. In this chapter, Reservoir sampling [257] and Balanced Reservoir Sampling (BRS) [252] will be compared as two baselines.

Oracle method is possible only when we can fully store and access all the datasets $X_1, X_2, \ldots, X_{N_t}$. It re-trains a brand-new model once a task comes in. Formally, it is to run Eqn. 5.1 with $X = \{X_1,\ X_2, \ldots,\ X_{N_t}\}$. Note that the Oracle method operates beyond the constraints of CL. It mainly provides an upper bound for other CL methods in this study.

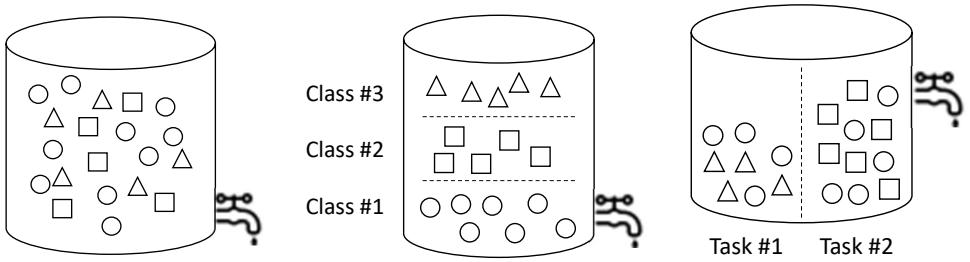
To measure the performance of all the tested methods, after the model is trained over all the N_t tasks, we calculate Average Accuracy (ACC) [258] over all the learned tasks. Testing accuracies on individual tasks at different training phases will also be examined. A high ACC or testing accuracy from a model indicates that the model can give accurate fault classification. Model training time and memory usage of these methods will also be compared.

5.3.4 Proposed improvements

Given a limited buffer size, it is critical to (1) choose a set of exemplars that can properly represent all the training samples from past tasks and (2) make efficient use of the stored exemplars. Considering these two points, we proposed two improvements to the existing ER method respectively.

5.3.4.1 Task balanced reservoir

Standard ER uses Reservoir sampling [257] (see Figure 5.2a) to randomly replace exemplars when the exemplar set is full. With Reservoir sampling, as shown in Figure 5.2a, any of the exemplars may be dropped out through the valve. This may cause class imbalance or even leave some classes out of the exemplar set. CF of those minority classes will damage the model's accuracy. BRS [252] tracks which class has the greatest number of exemplars, and only the exemplars of that class will be replaced by the new ones (see Figure 5.2b). However, in fault diagnosis, every task will have its own healthy-class signals that may be collected under different working conditions. In fact,



(a) Reservoir sampling replaces exemplars randomly.

(b) BRS replaces exemplars from the largest class.

(c) TBS replaces from tasks larger than average.

Figure 5.2: Comparison of Reservoir sampling [257], BRS [252], and the proposed TBS.

in limited training steps, an exemplar set maintained by the BRS scheme will have more healthy-class exemplars. The BRS will constantly replace healthy-class exemplars with samples from new tasks, making the exemplar set both task-imbalanced and class-imbalanced. This is detrimental to the model's performance.

In this chapter, for FDI applications, we propose a TBS scheme to select exemplars for ER-based CL models. A graphic illustration on the differences between Reservoir sampling [257], BRS [252], and the proposed TBS is shown in Figure 5.2. Different shapes of markers stand for exemplars of different fault classes, and they may come from different tasks. Instead of balancing on different classes, TBS balances on different tasks assuming each task characterizes a unique working condition. As shown in Figure 5.2c, we track which task does an exemplar is from and the number of exemplars from each task. When updating the exemplar set, only those from the tasks that contain more exemplars than average will be dropped. The detailed flowchart of how TBS updates the exemplar set is shown in Figure 5.3.

The procedure shown in Figure 5.3 is run every time a sample is used to train the model. In each run, it will first determine whether to update the exemplar set or not. If the exemplar set is not full, the input sample goes

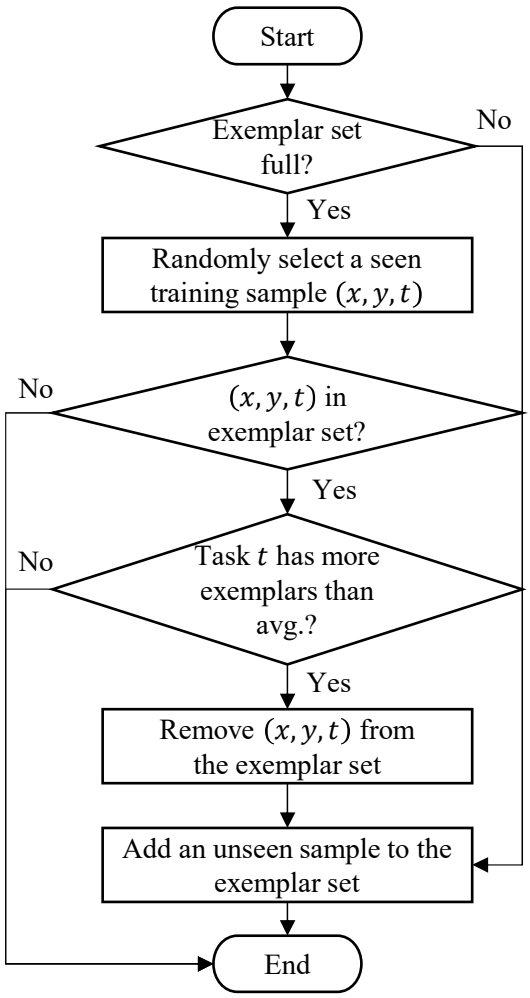


Figure 5.3: Flowchart of the proposed TBS updates its exemplar set.

straight in until it is filled. After the exemplar set is filled, one existing exemplar must be removed to accept a new exemplar. A sample will be randomly chosen from all the seen training samples, and it will be replaced by the input sample under two conditions: (1) the chosen sample is in the exemplar set, and (2) the chosen sample is of a class that has more exemplar than the average of all seen classes. As more training samples are seen, the probability and frequency of updates become lower. By tracking the task label and only replacing exemplars of larger tasks, TBS enforces that every task will have about the same number of exemplars. After training on N_t tasks, every task will have about B/N_t stored exemplars. This would not only help preserve all fault classes from different tasks but also ensure that all the seen working conditions of different tasks are included in the exemplar set. Although TBS does not produce class-balanced exemplar sets, further treatments such as cosine normalization [54] may be applied on top of TBS. Note that BRS may not be class-balanced as discussed above. In this chapter, the proposed TBS focuses on balancing different tasks to help the model better adapt to variable working conditions and achieve higher fault classification accuracy.

5.3.4.2 Working condition adaptation

Considering different working conditions, the healthy data from different tasks will have different distributions. If the model can learn from such differences, it may adapt to different working conditions better.

In this chapter, we propose to conduct DA during each fine-tuning stage of CL models. A DA loss term is added to the standard ER loss (Eqn. 5.2). The standard ER loss will only maintain fault-discriminative abilities, while the additional DA loss can constrain the distribution discrepancy of learned features between health data of different tasks. With DA, the model may learn to match the distributions of healthy data collected under different working

conditions. Formally, the proposed new learning algorithm is to minimize the following loss function:

$$Loss_l = \frac{1}{n_l} \sum_{x \in X_l} L(y, f_\theta(x)) + \frac{1}{n_e} \sum_{x \in X_e} L(y, f_\theta(x)) + \alpha L_{DA} \tag{5.3}$$

where L_{DA} is the DA loss and α is the weight of the DA loss. Note that the DA loss term will be calculated using only healthy data from all the seen tasks. In ref. [54] adversarial loss is applied. However, this is more suitable for one-way DA with a single source and a single target. In this work, we consider multiple source and target domains and need to conduct multi-way DA. We use Maximum Mean Discrepancy (MMD) [259] to measure the distribution discrepancy between each domain and combine all the discrepancies as the multi-way DA loss term. That is,

$$L_{DA} = \frac{1}{l-1} \sum_{t=0}^{l-1} MMD\left(x_l^{y=0}, x_t^{y=0}\right) \tag{5.4}$$

where l is the number of seen tasks (or the index of the task that the model is currently being trained on), and x with a superscript $y = 0$ stands for healthy data.

The proposed multi-way DA is different from existing DA methods such as refs. [10], [55] mainly on these two aspects:

1. Traditional DA only applies to two different working conditions, while the proposed is a multi-way adaptation among the current task and multiple previous tasks.

2. Traditional DA usually operates on domains with the same classes, while the proposed is more general as it only based on healthy samples.

Using the proposed Eqn. 5.3, the CL model is set to learn to deal with variable working conditions and give more accurate fault classification.

5.4 Case study I

To validate the effectiveness of the proposed method, we conducted two case studies using two different experiment datasets. The first case study uses a dataset collected at the Department of Energy and Power Engineering, Tsinghua University (THU-EPE) in 2019. All computational experiments were carried out on a computer with an Intel i7-6700 CPU and an Nvidia GTX-1060 GPU. All the tested methods are implemented using Pytorch on the Windows 10 operation system.

5.4.1 Data description

The dataset collected at the THU-EPE is studied here. It has also been used in ref. [118]. Four different types of gear faults in a planetary gearbox and four different rotating speeds are considered in this case study. Four different tasks listed in Table 5.1 are considered and they will be sequentially used to train the proposed CL model. In each task, for each machine health condition, there are 2000 training samples and 500 testing samples. Both the training and testing samples are collected with a sampling frequency of 20 kHz and each sample has 2048 sample points. Example signals of each task and machine health condition are shown in Figure 5.4. The text shown on the left-hand side of each subplot is the task ID and the corresponding health condition of that signal. The raw signals vary upon changes in health conditions or working conditions. It is critical that the model can be both class-discriminative and adaptive to different working conditions.

5.4.2 Hyper-parameters

For the structural parameters of 1DCNN discussed in Section 5.3.3, we use a fixed kernel size of 25 for the two Conv layers and the number of input channels for Conv1 is 1. The max pooling layers are with a kernel size of 4 and a stride of

Table 5.1: Four different tasks in the THU-EPE gearbox case study.

Task IDs	Machine health conditions	Rotating speeds
$\mathcal{T}_1$	Healthy (H), sun gear crack (SC)	30Hz
$\mathcal{T}_2$	H, sun gear broken (SB)	35Hz
$\mathcal{T}_3$	H, planet gear crack (PC)	25Hz
$\mathcal{T}_4$	H, planet gear broken (PB)	20Hz

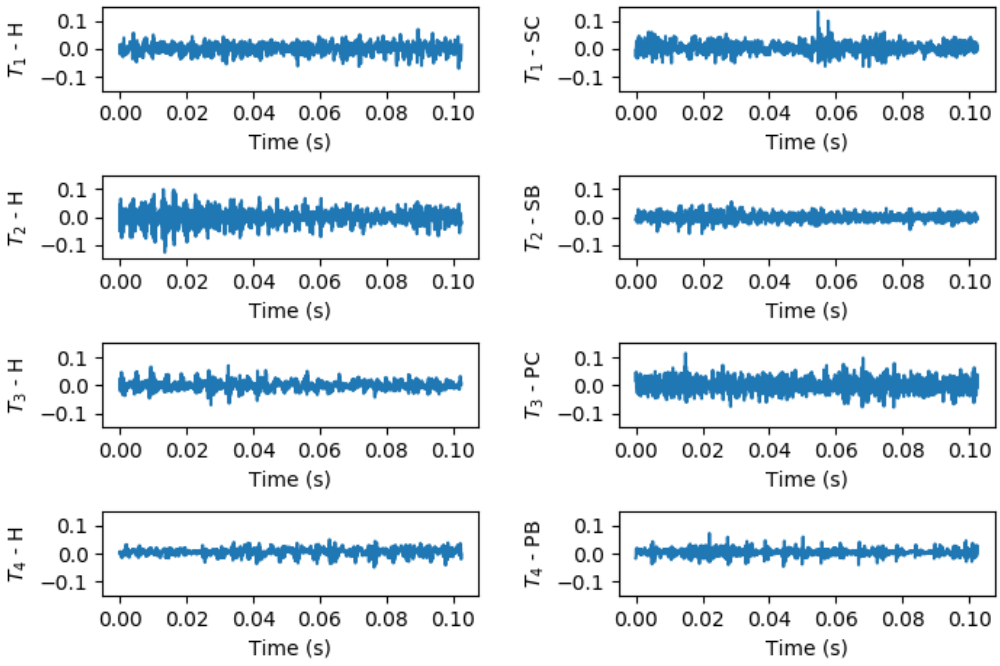


Figure 5.4: Example raw signals from the THU-EPE dataset.

Table 5.2: Candidate 'Out' values of 1DCNN.

Parameter name	Candidate values
Conv1 - No. of out channels	10, 20, **30**
Conv2 - No. of out channels	30, **50**, 70
FC1 - No. of outputs	**100**, 300, 900
FC2 - No. of outputs	10, **30**, 90

4 to reduce the number of features to 1/4 each time. A grid search among the candidate values listed in Table 5.2 is run using the experiment data presented in Case Study I. The learning rate and the batch size are also grid-searched among [0.1, 0.01, 0.001] and [50, 100, 200], respectively.

The standard ER with Reservoir sampling method is applied and the Stochastic Gradient Descent (SGD) method is used as the optimizer to train each task for 30 epochs. The hyper-parameters that give the highest ACC are regarded as the best. The best hyper-parameters will be used throughout the two case studies in Section 5.4 and 5.5.

For the THU-EPE dataset, with a learning rate of 0.01, a batch size of 50, and the best network structure configurations listed and marked bold in Table 5.2, the standard ER with Reservoir sampling method produced its highest ACC of 93.051%.

5.4.3 Performance comparison

With the network structure fixed, we implement 6 methods, i.e., fine-tune, ER, BRS, TBS, Oracle, and our proposed TBS-DA method discussed in Section 5.3.3. Their performances are shown in Table 5.3. For each method, 10 repeat runs were executed and the mean ACC and standard deviation are shown in the ACC column. Training data size and total training time are measured with a buffer size of 100 samples, a batch size of 50, and the number of training epochs is 30. For the TBS-DA method, the weight of DA loss α is fixed to be 0.1.

Table 5.3: Comparison of different methods on the THU-EPE gearbox dataset.

Method	Average accuracy	Training memory (Mega bytes)	Training time (s)
Fine-tune	62.490%	32.800	81.733
ER	93.051%	33.620	179.975
BRS [252]	89.761%	33.620	181.297
TBS	94.571%	33.621	179.884
TBS-DA	97.971%	33.621	263.326
Oracle	99.998%	131.072	60.108

From Table 5.3, we can see that the fine-tune method which does not include CL capability performs the worst with an ACC of 62.49%. In fact, the model achieved almost 100% accuracy for the final task but only about 50% accuracy for the previously learned three tasks. This is the most servere level of CF displayed. The Oracle method, which uses all the data from all the training phases, can give an almost perfect accuracy (99.998%) for all four tasks. No forgetting behavior is shown by the Oracle method. Utilizing an exemplar set, ER with Reservoir sampling achieved an ACC of 93.051% over the four tasks, 30.561% higher than the fine-tune method. TBS without domain adaptation performed 1.52% better than Reservoir, showing the benefit of maintaining a task-balanced exemplar set. The proposed TBS-DA method further improved the ACC to 97.971%, beating the TBS method by 3.4%. This shows that conducting DA can further reduce CF and boost classification accuracy. TBS-DA shows only 2.207% lower accuracy than the Oracle method while using only 25.65% of the memory size used by Oracle during training. More importantly, the TBS-DA model can be deployed right after every training phase while the Oracle model will only be available after the final training phase. The total training time needed by TBS-DA is higher as additional computations are needed for (1) updating the exemplar set (98.151 seconds), and (2) MMD-based working condition adaptation (94.911 seconds) compared to the fine-tune method.

The BRS method showed the lowest ACC (89.761%) among all the compared CL methods. The main reason is that it failed to enforce a class-balanced exemplar in this FDI setting. In a typical run of BRS, after all the training phases, the number of exemplars of the 5 classes is 50, 7, 5, 13, and 25 for H, SC, SB, PC, and PB respectively. It also gives a task-imbalanced exemplar set as tasks $\mathcal{T}_1$ to $\mathcal{T}_4$ have 52, 13, 16, and 19 exemplars respectively. Using TBS, however, the numbers of exemplars for the 4 tasks are balanced (26, 25, 25, and 24) respectively. Although TBS is still class-imbalanced, it covers all the seen working conditions better. A 4.81% ACC gain is achieved compared to BRS.

5.4.4 Analysis of key parameters

The performances of ER (with Reservoir sampling) and TBS-DA are largely dependent on the size of the exemplar set. Figure 5.5 shows how the size of the exemplar set affects the performances of the ER and the TBS-DA methods. The bars in two different colors indicate the mean accuracies of the two methods respectively, and the error bars on top of each colored bar show the standard deviation of the corresponding mean accuracies out of 10 repeated runs.

From Figure 5.5 we can see that the performance of both ER and TBS-DA ($\alpha = 0.1$) methods increases as the set buffer size increases. When the buffer size is set to 200, ER and TBS-DA give the same mean accuracies of 98.60%. When the buffer size is 50, ER is down to 84.62%, 11.69% lower than the proposed TBS-DA. This implies that the BRS-managed exemplar set failed to preserve enough samples of different tasks and caused forgetting of seen working conditions. The proposed TBS balances samples of all the tasks to avoid forgetting. Together with DA, TBS-DA showed an advantage under variable working conditions. Besides, the standard deviations (0.043%,

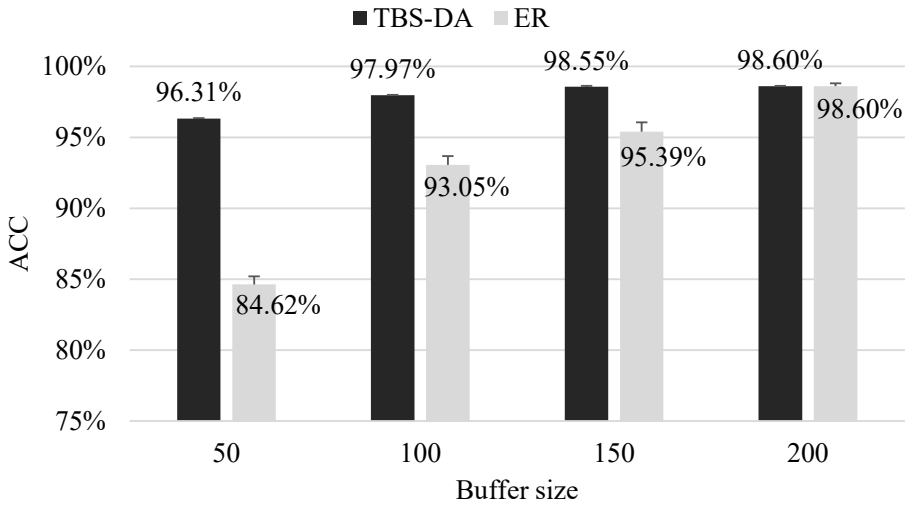


Figure 5.5: Mean accuracies of ER and TBS-DA with different buffer sizes on the THU-EPE dataset.

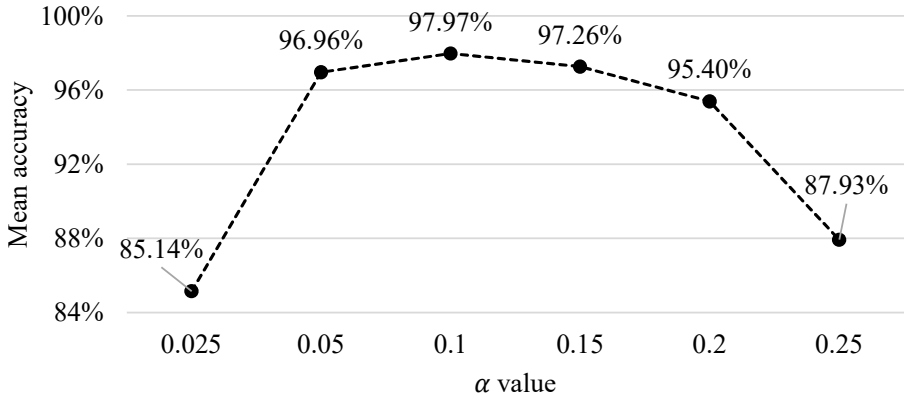


Figure 5.6: Mean accuracies of TBS-DA versus different α values on the THU-EPE dataset.

0.023%, 0.069%, 0.019% for TBS-DA and 0.570%, 0.616%, 0.657%, 0.202% for ER given buffer sizes of 50, 100, 150, and 200 respectively) in Figure 5.5 shows that the TBS-DA method is more stable than standard ER.

The value of α in Eqn. 5.3 determines the weight of the proposed DA term (Eqn. 5.4) in the loss function and should be selected properly so that both classification error and domain discrepancy can be jointly minimized. Figure 5.6 shows different mean accuracies of the TBS-DA method when using α values of 0.025, 0.05, 0.1, 0.15, and 0.2. Note that the buffer size is fixed as 100 to generate Figure 5.6.

Table 5.4: Four different tasks in the THU-ME gearbox case study.

Task IDs	Machine health conditions	Rotating speeds	Load levels
$\mathcal{T}_1$	H, PB-ORF	10Hz	0.13Nm
$\mathcal{T}_2$	H, IB-IRF	25Hz	-1.7Nm
$\mathcal{T}_3$	H, IB-ORF	20Hz	-1.65Nm
$\mathcal{T}_4$	H, PB-IRF	25hz	-0.657Nm

Figure 5.6 shows that the choice of α values is critical for the performance of the TBS-DA method. When α is set between 0.05 and 0.2, the performance of TBS-DA is much better than TBS without DA, and $\alpha = 0.1$ gives the best ACC among the 5 values shown in Figure 5.6. When α is 0.025, the performance of TBS-DA is lower than TBS but still much higher than the Fine-tune method which does not have CL capability.

5.5 Case study II

5.5.1 Data description

Case study II is about bearing faults in planetary gearboxes. In addition, both the rotating speed and the load level may change. We use the test rig presented in ref. [227], [260] and the dataset was collected at the Department of Mechanical Engineering, Tsinghua University (THU-ME), in 2021. Four different tasks listed in Table 5.4 are considered. Note that in the Machine health conditions column, H stands for healthy, PB for planet bearing, IB for input bearing, ORF for outer race fault, and IRF for inner race fault.

In each task, samples for each machine health condition, there are 1764 training samples and 588 testing samples. The sample frequency and the length of segments are the same as in case study I (20 kHz and 2048 respectively). Example signals of each task and each machine health condition are shown in Figure 5.7.

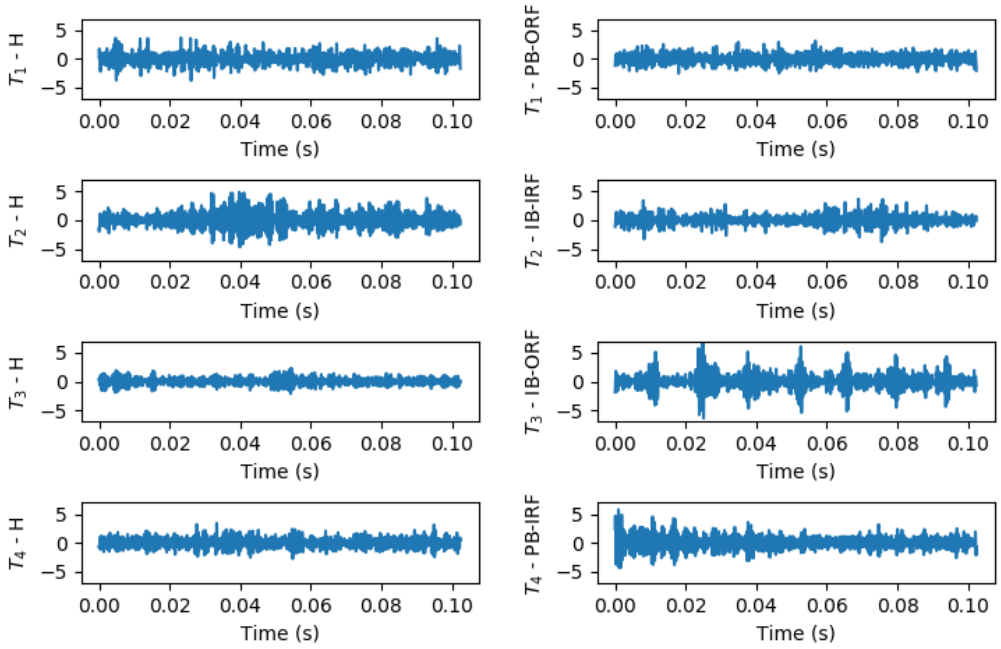
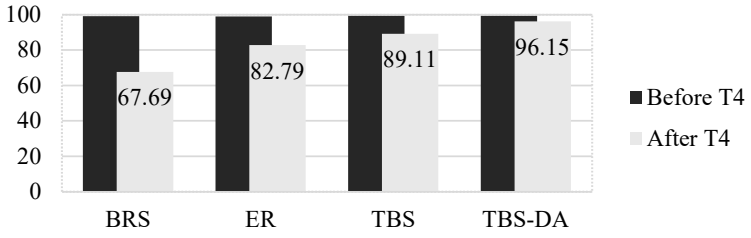


Figure 5.7: Example raw signals from the THU-ME dataset.

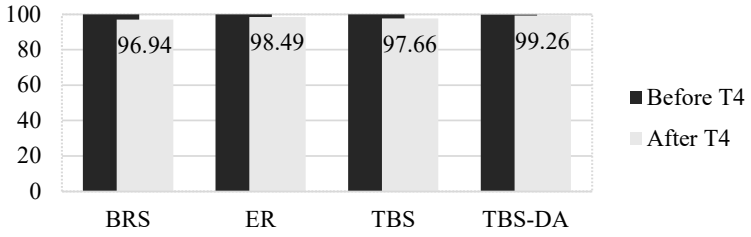
5.5.2 Performance comparison

In case study II, we continue to use the best network structure (shown in Table 5.2), learning rate (0.01), and batch size (50) selected using the THU-EPE dataset. With such a setting, the model trained using the Fine-tune method only gets an ACC of 50.48% on the 4 tasks in the THU-ME dataset. The accuracies on the 4 tasks are 6.26% ,45.91%, 49.97%, and 99.77% respectively. This shows that the model forgot $\mathcal{T}_2$ and $\mathcal{T}_3$ and completely failed on $\mathcal{T}_4$ during the process of learning other tasks. The Oracle method, using all the data from all 4 tasks, not surprisingly, gives an almost perfect ACC of 99.63%.

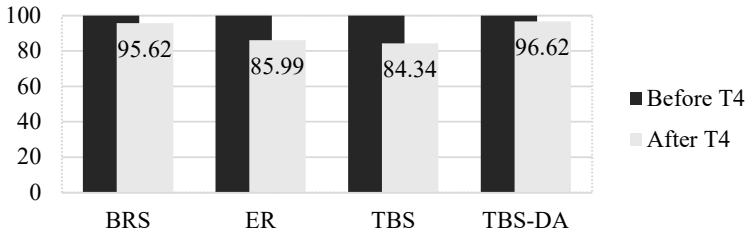
To demonstrate the efficacies of both the proposed TBS and multi-way DA, the 4 studied CL methods i.e., ER with Reservoir, BRS, TBS, and the proposed TBS-DA, are compared. In Figure 5.8, testing accuracies for each task before and after the training phase $\mathcal{T}_4$ are shown. Each subfigure shows testing accuracies on a task reported by the 4 CL methods. Each pair of overlapped bars shows the difference in testing accuracy before and after training



(a) Accuracy on $\mathcal{T}_1$ (%).



(b) Accuracy on $\mathcal{T}_2$ (%).



(c) Accuracy on $\mathcal{T}_3$ (%).

Figure 5.8: Comparison of accuracies on different tasks before and after training on $\mathcal{T}_4$.

the model on a new task $\mathcal{T}_4$. The shown results are average numbers across 10 runs with different random seeds.

Different degrees of forgetting phenomenon shown by all the 4 methods are observed in Figure 5.8. These degrees of forgetting can be measured by the drops of accuracies before and after training on $\mathcal{T}_4$. Our proposed TBS-DA reported the least degrees of forgetting, dropping only 3.2%, 0.58%, and 3.38% of accuracies on $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$ respectively. The worst degrees of forgetting on the 3 tasks are reported by BRS (31.38%), BRS (3%), and TBS (15.66%) respectively. Averaging across the 3 tasks, BRS, ER, TBS, and TBS-DA dropped 12.92%, 10.56%, 9.37%, and 2.39% of accuracies respectively. These numbers prove that the proposed TBS-DA has the best performance in preventing CF among the 4 studied CL methods.

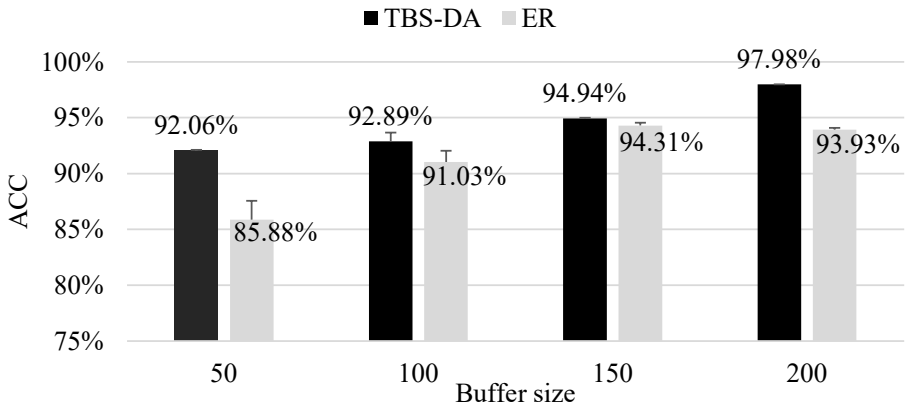


Figure 5.9: Mean accuracies of ER and TBS-DA with different buffer sizes on the THU-ME dataset.

5.5.3 Analysis of key parameters

Figure 5.9 shows a comparison of results between the ER and the proposed TBS-DA (α=0.1) methods given different buffer sizes. It is seen that TBS-DA beats the ER method by 6.18%, 2.9%, 0.63%, and 4.05% at buffer sizes of 50, 100, 150, and 200 respectively. TBS-DA is also consistently increasing its accuracy from 92.06% to 97.98% as we enlarge the buffer size from 50 to 200. The ER method, however, does not give higher accuracy when the buffer size increases from 150 to 200. The proposed DA term in the TBS-DA method was able to better utilize those additional data and beat ER by 4.05%. When the buffer size is only 50, the ER is only 85.88% accurate (6.18% lower than TBS-DA) and unstable (standard deviation is 1.686%). This indicates that BRS brought unstable coverage of all the working conditions. While TBS-DA with TBS is balanced over working conditions and showed better performance.

Figure 5.10 shows the impact of α value (see Eqn. 5.3) on the ACC of the TBS-DA method on the THU-ME dataset. Note that the buffer size is fixed as 100 to generate Figure 5.10. It is seen that all the tested α values can give higher mean accuracies than the ER method (α=0). The highest accuracy (95.07%) on the THU-ME dataset is achieved when α is 0.15. The best α for

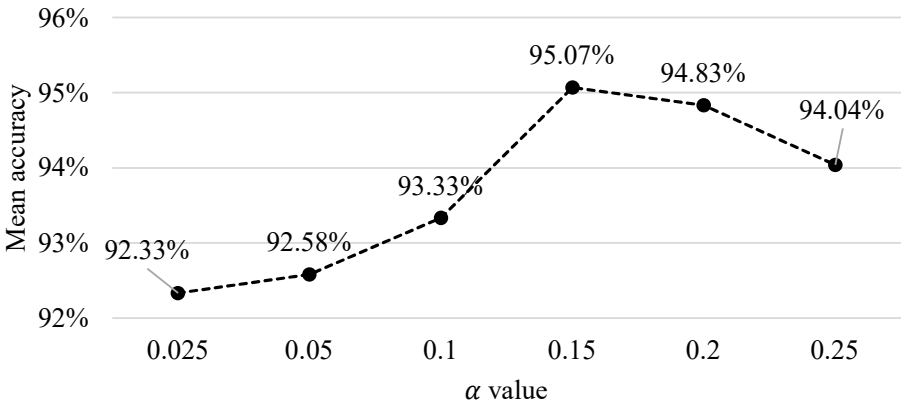


Figure 5.10: Mean accuracies of TBS-DA versus different α values on the THU-ME dataset.

the THU-EPE dataset (0.1) can also give solid results (93.33% accuracy) on the THU-ME dataset. Meanwhile, the best α for the THU-ME dataset (0.15) also works well for the THU-EPE dataset, achieving an ACC of 97.97%. Any α between (and including) 0.05 and 0.2 can make TBS-DA outperform the ER method on both the THU-EPE and the THU-ME datasets.

5.6 Conclusion

This chapter presents a novel CL-based fault classification method, TBS-DA, which considers both fault class incrementation and changes in machine working conditions. Firstly, the exemplar set, managed by the TBS replaying mechanism, preserves samples from all tasks to facilitate learning. Secondly, the proposed DA loss term enables the model to better adapt to varying working conditions. As a result, the TBS-DA method mitigates forgetting of both fault classes and working conditions, resulting in higher diagnostic accuracies. Although only tested for gearboxes, TBS-DA has the potential to be a valuable FDI method for many modern industries such as wind farms and manufacturing factories.

Future studies in this field could explore the application of CL for remaining useful life prediction, where the boundaries between tasks are ambiguous.

Additionally, it is important to first build a fault detection model using only healthy data before constructing and refining classification models with faulty data. Lastly, other advanced models and feature extraction methods can be explored to amend the limitations of 1DCNN not being able to quantify prediction uncertainty and to deal with low frequency signals.

Chapter 6

Summary and future directions

This chapter summarizes the work of this thesis and suggests future research directions.

6.1 Summary

This thesis developed three new deep transfer learning algorithms for fault diagnosis of wind turbine gearboxes including gear faults and bearing faults. Considering the complexity and low signal-to-noise ratio of the vibration signals of wind turbine gearboxes, deep learning models (mainly convolutional neural networks) are use to automatically extract fault-discriminative features from raw vibration signals collected using accelerometers mounted on the casing of the gearboxes. Instead of developing deep models using traditional supervised learning, different deep learning paradigms including domain adaptation, open-set recognition (or open-set fault diagnosis), and continual learning are explored to suit for different diagnostic tasks and available training datasets. Knowledge transfer or transfer learning across different diagnosis tasks with unlabeled training dataset and variable fault label sets, and adaptation across different working conditions (rotating speed and load levels) are the main novelties throughout the three presented research topics. Major works of this thesis are summarized as follows.

1) Weighted domain adaptation networks for machinery fault diagnosis: in supervised learning, the training and testing data are drawn from the same domain, and the training data are labeled. In this work, we treat different working condition as distinct domains and proposed a weighted source domain adaptation method to utilize labeled data collected under other working conditions to help the diagnosis under a working condition with only unlabeled data. While a convolutional neural network was used to automatically extract features from raw vibration signals, multiple fully connected neural networks were used as domain discriminators to force the model learn features that apply to both the labeled source domains and the unlabeled target domain. Besides, distributional differences measured using maximum mean discrepancy were used to weigh the importance of different source domains. Finally, the trained model can recognize faults under the target working condition without labeled data. The proposed method was tested in two case studies using data from two different experiment test rigs. The results show that compare to traditional supervised learning which does not learn from unlabeled target data, the proposed weighted source domain adaptation can boost the diagnostic accuracy up to 30.457%. It also outperforms other single source domain adaptation methods and equally weighted multiple source domain adaption. It is also found that distributional differences between the target and the source domains can also be used to determine if domain adaptation can boost diagnostic accuracy or will cause negative transfer.

2) Open-set fault diagnosis for industrial rotating machines based on trustworthy deep learning: in real applications, it is common to encounter new fault classes that are not included in the training dataset. Conventional deep learning models, however, do not have the abilities to deal with unseen classes and will report over-confident predictions for samples from unseen classes or unseen working conditions. In this work, we extended conventional models with

an abstention option to contain unseen classes. An auxiliary dataset consist of superposition of faulty signals and noised injected healthy signals is used to represent the 'abstention' class. Evidential deep learning, which formulates learning as an evidence acquisition process, is adopted and enhanced to learn a sparse evidential distribution from training data, providing a better modeling of classification probabilities. We name the proposed method as evidential abstaining classifier and tested it with two datasets collected from two different experiment test beds. One for gear faults and another for bearing faults. The results show that our designed auxiliary training samples can help deep learning models to establish fault-discriminative features and efficient decision boundaries for unseen fault classes. Our enhanced evidential deep learning with L1 regularization can assign high classification uncertainties for samples from unseen classes and low for classes included in the training set with higher contrast compared to the original evidential deep learning. Our proposed evidential abstaining classifier are trustworthy as it has the abilities to report classification uncertainty and recognize new fault classes.

3) Continual learning for fault diagnosis considering variable working conditions: Traditional methods only focus on developing a new model from scratch using a complete training dataset including all the fault classes. However, the training dataset may never be complete in fault diagnosis applications and deep learning models need to be continuously upgraded in considering changes in working conditions and occurrence of new faults. In this work, we proposed a multi-staged continual learning algorithm that learns a sequence of diagnostic tasks featuring different fault classes and working conditions. Using the parameters pre-trained on previous tasks, the model can learn quicker compared to randomly initialized models. A multi-way domain adaptation is also conducted to adapt the model for all the different working conditions featured in different tasks. To overcome the catastrophic forgetting behavior of neural

networks without storing a cumbersome meta dataset, a small exemplar set including samples from all previous tasks is maintained for the use of the next training stage. A task-balance sampling scheme is proposed to make sure each task is well represented in the exemplar set so that the model can remember all the fault classes and adapt to all the previous working conditions. The same two datasets used in our open-set fault diagnosis work is used again to verify the efficacy of the proposed continual learning method. The results show that the proposed method can achieve an average accuracy of 97.97% across all the learned tasks with limited access to historical training data. This shows that catastrophic forgetting has been successfully mitigated.

6.2 Future directions

There are still many challenges in building artificial intelligence for fault diagnostic applications. Here are three possible research topics for the near future.

1) Holistic diagnosis based on multi-modal inputs. Mechanical vibration is the sole input for deep learning models used in this thesis. The monitoring system of modern wind turbines may provide other information such as pitch angle, historical weather, and thermal imagery, presenting multi-modal inputs including scalars, 1-dimensional signals, 2-dimensional images etc. Advanced deep learning models can be trained to deal with these multi-modal inputs. For example, automatic image captioning can be achieved by using convolutional neural networks to deal with images and recurrent neural networks to learn and produce text. With an organized training dataset, these two parts can be synchronized and jointly trained. Multi-modal inputs in fault diagnosis may also provide information on different time-scales that are useful for diagnostic decisions. For example, vibration and thermal imagery can be jointly used to locate faults. If a component shows irregular vibration patterns and had a recent temperature surge, there might be a fault has occurred in this

component. This will require the model to be able to process short vibration patterns lasting about seconds and monitor the temperature changes within a few minutes.

2) Edge computing and federated learning. Wind turbine and wind farms are spread across the world and operated by different owner/operators. Edge computing is to process data and update models locally on edge devices rather than relying solely on centralized cloud servers. This is vital for real-time fault diagnosis. On the other hand, federated learning enables collaborative model training across distributed edge devices without accessing to private data. By combining edge computing with federated learning, wind farm owner/operators can build stronger fault diagnosis systems that leverage real-time data analytics while preserving data privacy and security. This will also allow deep learning models to learn knowledge from diverse wind turbines deployed across different locations and environments, leading to more adaptive and robust fault diagnosis results.

3) Explainable artificial intelligence (XAI) for physical understanding and maintenance decision making. As discussed in Chapter 4, transparency of deep learning models is vital for decision makers to trust the models' results. XAI can also help us understand the mechanism of mechanical degradation and faults. Existing interpretability methods such as saliency maps [261] can highlight the features that contribute most to the model's predictions. For complex deep models, Local Interpretable Model-agnostic Explanations (LIME) [262] can provide explanations by approximating the model's behavior using simpler and more interpretable models. These techniques may facilitate physical understanding of deep models by revealing relationships between learned features and the predicted fault. The underlying causes of failures might also be better understood to help owner/operators make better operation and maintenance decisions.

References

[1] B. Parhami, "Defect, Fault, Error,..., or Failure?" *IEEE Transactions on Reliability*, vol. 46, no. 4, pp. 450–451, Dec. 1997, ISSN: 1558-1721. (visited on 11/01/2023).

[2] AIBN, "Report on the air accident near Turøy, øygarden municipality, Hordaland county, Norway 29 April 2016 with Airbus Helicopters EC 225 LP, LN-OJF, operated by CHC Helikopter Service AS — nsia," Accident Investigation Board Norway, Turøy, Hordaland, Norway, Tech. Rep. SL 2018/04, Jul. 2018. (visited on 11/02/2023).

[3] E. Zio, "Prognostics and Health Management (PHM): Where are we and where do we (need to) go in theory and practice," *Reliability Engineering & System Safety*, vol. 218, p. 108 119, Feb. 2022, ISSN: 0951-8320. (visited on 11/09/2023).

[4] I. R. Society, "IEEE Reliability Society Annual Technical Report 2007," *IEEE Transactions on Reliability*, vol. 57, no. 3, pp. 398–425, Sep. 2008, ISSN: 1558-1721. (visited on 11/09/2023).

[5] J. Koochaki, J. Bokhorst, H. Wortmann, and W. Klingenberg, "Evaluating condition based maintenance effectiveness for two processes in series," *Journal of Quality in Maintenance Engineering*, vol. 17, no. 4, pp. 398–414, Jan. 2011, ISSN: 1355-2511. (visited on 11/02/2023).

[6] L. Biggio and I. Kastanis, "Prognostics and Health Management of Industrial Assets: Current Progress and Road Ahead," *Frontiers in Artificial Intelligence*, vol. 3, 2020, ISSN: 2624-8212. (visited on 11/09/2023).

[7] K. Martin, "A review by discussion of condition monitoring and fault diagnosis in machine tools," en, *International Journal of Machine Tools and Manufacture*, vol. 34, no. 4, pp. 527–551, May 1994, ISSN: 08906955. DOI: 10.1016/0890-6955(94)90083-3. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0890695594900833 (visited on 05/30/2024).

[8] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, Oct. 2006, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2005.09.

012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327005001512 (visited on 11/06/2023).

[9] R. B. Randall, *Vibration-based condition monitoring: industrial, automotive and aerospace applications*, en. John Wiley & Sons, Dec. 2010, tex.copyright: 2011 John Wiley & Sons, Ltd, ISBN: 978-0-470-97766-8. (visited on 11/06/2023).

[10] X. Wang, T. Wang, A. Ming, W. Zhang, A. Li, and F. Chu, "Cross-operating condition degradation knowledge learning for remaining useful life estimation of bearings," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2021, ISSN: 1557-9662.

[11] P. Henriquez, J. B. Alonso, M. A. Ferrer, and C. M. Travieso, "Review of Automatic Fault Diagnosis Systems Using Audio and Vibration Signals," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 5, pp. 642–652, May 2014, ISSN: 2168-2232.

[12] T. H. Mohamad, F. Nazari, and C. Nataraj, "A Review of Phase Space Topology Methods for Vibration-Based Fault Diagnostics in Nonlinear Systems," en, *Journal of Vibration Engineering & Technologies*, vol. 8, no. 3, pp. 393–401, Jun. 2020, ISSN: 2523-3939. DOI: 10.1007/s42417-019-00157-6. [Online]. Available: https://doi.org/10.1007/s42417-019-00157-6 (visited on 11/06/2023).

[13] M. Tiboni, C. Remino, R. Bussola, and C. Amici, "A Review on Vibration-Based Condition Monitoring of Rotating Machinery," en, *Applied Sciences*, vol. 12, no. 3, p. 972, Jan. 2022, Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2076-3417. DOI: 10.3390/app12030972. [Online]. Available: https://www.mdpi.com/2076-3417/12/3/972 (visited on 11/06/2023).

[14] P. Tchakoua, R. Wamkeue, M. Ouhrouche, F. Slaoui-Hasnaoui, T. A. Tameghe, and G. Ekemb, "Wind Turbine Condition Monitoring: State-of-the-Art Review, New Trends, and Future Challenges," en, *Energies*, vol. 7, no. 4, pp. 2595–2630, Apr. 2014. (visited on 10/11/2017).

[15] W. A. Smith and R. B. Randall, "Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study," en, *Mechanical Systems and Signal Processing*, vol. 64-65, pp. 100–131, Dec. 2015, ISSN: 0888-3270.

[16] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," en, *Mechanical Systems and Signal Processing*, vol. 72-73, pp. 303–315, May 2016, ISSN: 0888-3270.

[17] X. Liang, M. J. Zuo, and Z. Feng, "Dynamic modeling of gearbox faults: A review," *Mechanical Systems and Signal Processing*, vol. 98, pp. 852–876, Jan. 2018, ISSN: 0888-3270. (visited on 03/14/2018).

[18] U. o. M. Center for Sustainable Systems, "Wind Energy Factsheet," en, Center for Sustainable Systems, University of Michigan, Tech. Rep. CSS07-09, 2023. (visited on 11/15/2023).

[19] C. R. E. Association, *By the Numbers – Canadian Renewable Energy Association*, en-US. (visited on 11/15/2023).

[20] C. E. R. Government of Canada, *CER – Canada's Energy Future 2023: Energy Supply and Demand Projections to 2050*, eng, Oct. 2023. (visited on 11/15/2023).

[21] P. Qian, X. Ma, and D. Zhang, "Estimating Health Condition of the Wind Turbine Drivetrain System," en, *Energies*, vol. 10, no. 10, p. 1583, Oct. 2017, ISSN: 1996-1073. (visited on 11/16/2023).

[22] T. Wang, Q. Han, F. Chu, and Z. Feng, "Vibration based condition monitoring and fault diagnosis of wind turbine planetary gearbox: A review," *Mechanical Systems and Signal Processing*, vol. 126, pp. 662–685, Jul. 2019, ISSN: 0888-3270. (visited on 11/18/2023).

[23] A. Ragheb and M. Ragheb, "Wind turbine gearbox technologies," in *2010 1st International Nuclear & Renewable Energy Conference (INREC)*, Mar. 2010. (visited on 11/15/2023).

[24] H. Gu, W. Y. Liu, Q. W. Gao, and Y. Zhang, "A review on wind turbines gearbox fault diagnosis methods," en, *Journal of Vibroengineering*, vol. 23, no. 1, pp. 26–43, Feb. 2021, Number: 1 Publisher: JVE International Ltd., ISSN: 1392-8716, 2538-8460. DOI: 10.21595/jve.2020.20178. [Online]. Available: https://www.extrica.com/article/20178 (visited on 11/15/2023).

[25] S. Sheng, *Gearbox Typical Failure Modes, Detection, and Mitigation Methods (Presentation): NREL (National Renewable Energy Laboratory)*, American English, San Diego, California, Jan. 2014. [Online]. Available: https://research-hub.nrel.gov/en/publications/gearbox-typical-failure-modes-detection-and-mitigation-methods-pr (visited on 05/31/2024).

[26] S. Sheng, "Wind Turbine Gearbox Condition Monitoring Round Robin Study - Vibration Analysis," English, National Renewable Energy Lab. (NREL), Golden, CO (United States), TECHNICAL REPORT NREL/TP-5000-54530, Jul. 2012. DOI: 10.2172/1048981. [Online]. Available: https://www.osti.gov/biblio/1048981 (visited on 07/09/2024).

[27] H. Ma, J. Zeng, R. Feng, X. Pang, Q. Wang, and B. Wen, "Review on dynamics of cracked gear systems," *Engineering Failure Analysis*, vol. 55, pp. 224–245, Sep. 2015, ISSN: 1350-6307. DOI: 10.1016/j.engfailanal.2015.06.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1350630715001879 (visited on 01/10/2024).

[28] S. Zhang, S. Zhang, B. Wang, and T. G. Habetler, "Deep learning algorithms for bearing fault Diagnostics-A comprehensive review," *IEEE access : practical innovations, open solutions*, vol. 8, pp. 29 857–29 881, 2020, ISSN: 2169-3536.

[29] Y. Chen, X. Liang, and M. J. Zuo, "Time series modeling of vibration signals from a gearbox under varying speed and load condition," in *2018 IEEE International Conference on Prognostics and Health Management (ICPHM)*, Jun. 2018.

[30] M. Zhang, K. Wang, D. Wei, and M. J. Zuo, "Amplitudes of characteristic frequencies for fault diagnosis of planetary gearbox," *Journal of Sound and Vibration*, vol. 432, pp. 119–132, 2018.

[31] Z. Feng and M. J. Zuo, "Vibration signal models for fault diagnosis of planetary gearboxes," *Journal of Sound and Vibration*, vol. 331, no. 22, pp. 4919–4939, Oct. 2012, ISSN: 0022-460X. (visited on 03/14/2018).

[32] H. Yang, J. Mathew, and L. Ma, "Vibration feature extraction techniques for fault diagnosis of rotating machinery : A literature survey," en, in *Asia-Pacific Vibration Conference*, tex.copyright: free_to_read, Australia, 2003, pp. 801–807. (visited on 11/18/2023).

[33] A. Stetco, F. Dinmohammadi, X. Zhao, *et al.*, "Machine learning methods for wind turbine condition monitoring: A review," en, *Renewable Energy*, vol. 133, pp. 620–635, Apr. 2019, ISSN: 0960-1481. (visited on 07/30/2020).

[34] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," en, *Mechanical Systems and Signal Processing*, vol. 115, pp. 213–237, Jan. 2019, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2018.05.050.

[35] A. M. TURING, "I.—COMPUTING MACHINERY AND INTELLIGENCE," *Mind*, vol. LIX, no. 236, pp. 433–460, Oct. 1950, ISSN: 0026-4423. DOI: 10.1093/mind/LIX.236.433. [Online]. Available: https://doi.org/10.1093/mind/LIX.236.433 (visited on 11/30/2023).

[36] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," en, *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, Number: 7553 Publisher: Nature Publishing Group, ISSN: 1476-4687. DOI: 10.1038/nature14539. [Online]. Available: https://www.nature.com/articles/nature14539 (visited on 11/29/2023).

[37] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," en, *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020, ISSN: 1556-4967. DOI: 10.1002/rob.21918. [Online]. Available: https://doi.org/10.1002/rob.21918 (visited on 11/30/2023).

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015.

[39] OpenAI, "GPT-4 technical report," Mar. 2023.

[40] R. Liu, B. Yang, E. Zio, and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: A review," en, *Mechanical Systems and Signal Processing*, vol. 108, pp. 33–47, Aug. 2018, ISSN: 0888-3270.

[41] S. Xing, Y. Lei, B. Yang, and N. Lu, "Adaptive knowledge transfer by continual weighted updating of filter kernels for few-shot fault diagnosis of machines," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2021, ISSN: 1557-9948. DOI: 10.1109/TIE.2021.3063975.

[42] M. Krishnamurthi and D. T. Phillips, "An expert system framework for machine fault diagnosis," *Computers & Industrial Engineering*, vol. 22, no. 1, pp. 67–84, Jan. 1992, ISSN: 0360-8352. DOI: 10.1016/0360-8352(92)90034-H. [Online]. Available: https://www.sciencedirect.com/science/article/pii/036083529290034H (visited on 11/30/2023).

[43] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," en, *Mechanical Systems and Signal Processing*, vol. 138, p. 106 587, Apr. 2020, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2019.106587.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980 [cs]*, Dec. 2014. [Online]. Available: http://arxiv.org/abs/1412.6980.

[45] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[46] M. Feurer and F. Hutter, "Hyperparameter Optimization," en, in *Automated Machine Learning: Methods, Systems, Challenges*, ser. The Springer Series on Challenges in Machine Learning, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., Cham: Springer International Publishing, 2019, pp. 3–33, ISBN: 978-3-030-05318-5. DOI: 10.1007/978-3-030-05318-5_1. [Online]. Available: https://doi.org/10.1007/978-3-030-05318-5_1 (visited on 12/07/2023).

[47] J. Laaksonen and E. Oja, "Classification with learning k-nearest neighbors," ser. Proceedings of International Conference on Neural Networks (ICNN'96), vol. 3, 1996, 1480–1483 vol.3.

[48] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 21, no. 6, pp. 2560–2574, Aug. 2007, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2006.12.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327007000027 (visited on 12/01/2023).

[49] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "Deep Model Based Domain Adaptation for Fault Diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2296–2305, Mar. 2017, ISSN: 1557-9948.

[50] M. Rao, Q. Li, D. Wei, and M. J. Zuo, "A deep bi-directional long short-term memory model for automatic rotating speed extraction from raw vibration signals," en, *Measurement*, vol. 158, p. 107 719, Jul. 2020, ISSN: 0263-2241. DOI: 10.1016/j.measurement.2020.107719.

[51] F. Jia, Y. Lei, N. Lu, and S. Xing, "Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization," *Mechanical Systems and Signal Processing*, vol. 110, pp. 349–367, Sep. 2018, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2018.03.025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327018301444 (visited on 12/05/2023).

[52] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1757–1772, 2013, ISSN: 1939-3539.

[53] A. Ur Rehman, W. Jiao, J. Sun, H. Pan, and T. Yan, "Open Set Recognition Methods for Fault Diagnosis: A Review," in *2023 15th International Conference on Advanced Computational Intelligence (ICACI)*, May 2023. (visited on 11/16/2023).

[54] J. Li, R. Huang, Z. Chen, G. He, K. C. Gryllias, and W. Li, "Deep continual transfer learning with dynamic weight aggregation for fault diagnosis of industrial streaming data under varying working conditions," en, *Advanced Engineering Informatics*, vol. 55, p. 101 883, Jan. 2023, ISSN: 1474-0346. DOI: 10.1016/j.aei.2023.101883.

[55] L. Guo, Y. Lei, S. Xing, T. Yan, and N. Li, "Deep Convolutional Transfer Learning Network: A New Method for Intelligent Fault Diagnosis of Machines With Unlabeled Data," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 9, pp. 7316–7325, Sep. 2019, ISSN: 1557-9948. DOI: 10.1109/TIE.2018.2877090.

[56] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," 2015, pp. 427–436. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Nguyen_Deep_Neural_Networks_2015_CVPR_paper.html.

[57] S. Lu, J. Lu, K. An, X. Wang, and Q. He, "Edge Computing on IoT for Machine Signal Processing and Fault Diagnosis: A Review," *IEEE Internet of Things Journal*, vol. 10, no. 13, pp. 11 093–11 116, Jul. 2023, Conference Name: IEEE Internet of Things Journal, ISSN: 2327-4662.

DOI: 10.1109/JIOT.2023.3239944. [Online]. Available: https://ieeexplore.ieee.org/document/10026418 (visited on 12/06/2023).

[58] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," en, *Neural Networks*, vol. 113, pp. 54–71, May 2019, ISSN: 0893-6080.

[59] M. Mermillod, A. Bugaiska, and P. BONIN, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," *Frontiers in Psychology*, vol. 4, 2013, ISSN: 1664-1078.

[60] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," en, in *Psychology of learning and motivation*, G. H. Bower, Ed., vol. 24, Academic Press, Jan. 1989, pp. 109–165.

[61] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, ISSN: 1558-2191. DOI: 10.1109/TKDE.2009.191.

[62] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous Deep Transfer Across Domains and Tasks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4068–4076.

[63] H. Zhao, S. Zhang, G. Wu, J. M. F. Moura, J. P. Costeira, and G. J. Gordon, "Adversarial multiple source domain adaptation," in *Advances in neural information processing systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., Curran Associates, Inc., 2018, pp. 8559–8570.

[64] L. Liao and F. Köttig, "A hybrid framework combining data-driven and model-based methods for system remaining useful life prediction," *Applied Soft Computing*, vol. 44, pp. 191–199, Jul. 2016, ISSN: 1568-4946. DOI: 10.1016/j.asoc.2016.03.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494616301223 (visited on 01/24/2024).

[65] M. A. Atoui and A. Cohen, "Coupling data-driven and model-based methods to improve fault diagnosis," *Computers in Industry*, vol. 128, p. 103 401, Jun. 2021, ISSN: 0166-3615. DOI: 10.1016/j.compind.2021.103401. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361521000087 (visited on 01/24/2024).

[66] G. Litak and M. I. Friswell, "Dynamics of a Gear System with Faults in Meshing Stiffness," en, *Nonlinear Dynamics*, vol. 41, no. 4, pp. 415–421, Sep. 2005, ISSN: 1573-269X. DOI: 10.1007/s11071-005-1398-y. [Online]. Available: https://doi.org/10.1007/s11071-005-1398-y (visited on 01/10/2024).

[67] Z. Tian, M. J. Zuo, and S. Wu, "Crack propagation assessment for spur gears using model-based analysis and simulation," en, *Journal of Intelligent Manufacturing*, vol. 23, no. 2, pp. 239–253, Apr. 2012, ISSN: 1572-8145. DOI: 10.1007/s10845-009-0357-8. [Online]. Available: https://doi.org/10.1007/s10845-009-0357-8 (visited on 01/10/2024).

[68] A. Parey, M. El Badaoui, F. Guillet, and N. Tandon, "Dynamic modelling of spur gear pair and application of empirical mode decomposition-based statistical analysis for early detection of localized tooth defect," *Journal of Sound and Vibration*, vol. 294, no. 3, pp. 547–561, Jun. 2006, ISSN: 0022-460X. DOI: 10.1016/j.jsv.2005.11.021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022460X05007352 (visited on 01/10/2024).

[69] S. Jia and I. Howard, "Comparison of localised spalling and crack damage from dynamic modelling of spur gear vibrations," *Mechanical Systems and Signal Processing*, vol. 20, no. 2, pp. 332–349, Feb. 2006, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2005.02.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327005000300 (visited on 01/10/2024).

[70] Y. Shao and Z. Chen, "Dynamic features of planetary gear set with tooth plastic inclination deformation due to tooth root crack," en, *Nonlinear Dynamics*, vol. 74, no. 4, pp. 1253–1266, Dec. 2013, ISSN: 1573-269X. DOI: 10.1007/s11071-013-1038-x. [Online]. Available: https://doi.org/10.1007/s11071-013-1038-x (visited on 01/10/2024).

[71] R. G. Parker, S. M. Vijayakar, and T. Imajo, "NON-LINEAR DYNAMIC RESPONSE OF A SPUR GEAR PAIR: MODELLING AND EXPERIMENTAL COMPARISONS," *Journal of Sound and Vibration*, vol. 237, no. 3, pp. 435–455, Oct. 2000, ISSN: 0022-460X. DOI: 10.1006/jsvi.2000.3067. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022460X00930670 (visited on 01/11/2024).

[72] R. G. Parker, V. Agashe, and S. M. Vijayakar, "Dynamic Response of a Planetary Gear System Using a Finite Element/Contact Mechanics Model," *Journal of Mechanical Design*, vol. 122, no. 3, pp. 304–310, May 1999, ISSN: 1050-0472. DOI: 10.1115/1.1286189. [Online]. Available: https://doi.org/10.1115/1.1286189 (visited on 01/11/2024).

[73] D. G. Lewicki, R. F. Handschuh, L. E. Spievak, P. A. Wawrzynek, and A. R. Ingraffea, "Consideration of Moving Tooth Load in Gear Crack Propagation Predictions," *Journal of Mechanical Design*, vol. 123, no. 1, pp. 118–124, Oct. 2000, ISSN: 1050-0472. DOI: 10.1115/1.1338118. [Online]. Available: https://doi.org/10.1115/1.1338118 (visited on 01/11/2024).

[74] R. B. Randall, "A New Method of Modeling Gear Faults," *Journal of Mechanical Design*, vol. 104, no. 2, pp. 259–267, Apr. 1982, ISSN: 0161-8458. DOI: 10.1115/1.3256334. [Online]. Available: https://doi.org/10.1115/1.3256334 (visited on 01/12/2024).

[75] P. D. McFadden and J. D. Smith, "An Explanation for the Asymmetry of the Modulation Sidebands about the Tooth Meshing Frequency in Epicyclic Gear Vibration," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 199, no. 1, pp. 65–70, Jan. 1985, Publisher: IMECHE, ISSN: 0954-4062. DOI: 10.1243/PIME_PROC_1985_199_092_02. [Online]. Available: https://doi.org/10.1243/PIME_PROC_1985_199_092_02 (visited on 01/12/2024).

[76] J. Lin and M. Zhao, "A review and strategy for the diagnosis of speed-varying machinery," in *2014 International Conference on Prognostics and Health Management*, Jun. 2014, pp. 1–9. DOI: 10.1109/ICPHM.2014.7036368. [Online]. Available: https://ieeexplore.ieee.org/document/7036368 (visited on 01/14/2024).

[77] N. Ahamed, Y. Pandya, and A. Parey, "Spur gear tooth root crack detection using time synchronous averaging under fluctuating speed," *Measurement*, vol. 52, pp. 1–11, Jun. 2014, ISSN: 0263-2241. DOI: 10.1016/j.measurement.2014.02.029. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224114000888 (visited on 01/12/2024).

[78] J. Rafiee and P. W. Tse, "Use of autocorrelation of wavelet coefficients for fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 23, no. 5, pp. 1554–1572, Jul. 2009, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2009.02.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327009000685 (visited on 01/16/2024).

[79] D. Wang, "K-nearest neighbors based methods for identification of different gear crack levels under different motor speeds and loads: Revisited," *Mechanical Systems and Signal Processing*, vol. 70-71, pp. 201–208, Mar. 2016, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2015.10.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327015004677 (visited on 01/12/2024).

[80] B. Samanta, "Artificial neural networks and genetic algorithms for gear fault detection," *Mechanical Systems and Signal Processing*, vol. 18, no. 5, pp. 1273–1282, Sep. 2004, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2003.11.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327003001468 (visited on 01/16/2024).

[81] S. Khan and T. Yairi, "A review on the application of deep learning in system health management," en, *Mechanical Systems and Signal Processing*, vol. 107, pp. 241–265, Jul. 2018, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2017.11.024.

[82] M. Xia, T. Li, L. Xu, L. Liu, and C. W. de Silva, "Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 101–110, 2018, ISSN: 1941-014X.

[83] M. Xia, H. Shao, D. Williams, S. Lu, L. Shu, and C. W. de Silva, "Intelligent fault diagnosis of machinery using digital twin-assisted deep transfer learning," *Reliability Engineering & System Safety*, vol. 215, p. 107 938, Nov. 2021, ISSN: 0951-8320. DOI: 10.1016/j.ress.2021.107938. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832021004531 (visited on 01/24/2024).

[84] D. Jung, K. Y. Ng, E. Frisk, and M. Krysander, "Combining model-based diagnosis and data-driven anomaly classifiers for fault isolation," *Control Engineering Practice*, vol. 80, pp. 146–156, Nov. 2018, ISSN: 0967-0661. DOI: 10.1016/j.conengprac.2018.08.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066118304404 (visited on 03/06/2024).

[85] J. Wang, Z. Wang, V. Stetsyuk, X. Ma, F. Gu, and W. Li, "Exploiting Bayesian networks for fault isolation: A diagnostic case study of diesel fuel injection system," *ISA Transactions*, vol. 86, pp. 276–286, Mar. 2019, ISSN: 0019-0578. DOI: 10.1016/j.isatra.2018.10.044. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019057818304233 (visited on 03/18/2024).

[86] M. Zhang, D. Li, K. Wang, *et al.*, "An adaptive order-band energy ratio method for the fault diagnosis of planetary gearboxes," en, *Mechanical Systems and Signal Processing*, vol. 165, p. 108 336, Feb. 2022, ISSN: 0888-3270.

[87] M. Sadoughi and C. Hu, "Physics-Based Convolutional Neural Network for Fault Diagnosis of Rolling Element Bearings," *IEEE Sensors Journal*, vol. 19, no. 11, pp. 4181–4192, Jun. 2019, Conference Name: IEEE Sensors Journal, ISSN: 1558-1748. DOI: 10.1109/JSEN.2019.2898634. [Online]. Available: https://ieeexplore.ieee.org/document/8638772 (visited on 03/18/2024).

[88] P. D. McFadden and J. D. Smith, "Model for the vibration produced by a single point defect in a rolling element bearing," *Journal of Sound and Vibration*, vol. 96, no. 1, pp. 69–82, Sep. 1984, ISSN: 0022-460X. DOI: 10.1016/0022-460X(84)90595-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022460X84905959 (visited on 03/18/2024).

[89] S. Shen, H. Lu, M. Sadoughi, *et al.*, "A physics-informed deep learning approach for bearing fault detection," *Engineering Applications of Artificial Intelligence*, vol. 103, p. 104 295, Aug. 2021, ISSN: 0952-1976. DOI: 10.1016/j.engappai.2021.104295. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197621001421 (visited on 03/18/2024).

[90] Q. Ni, J. C. Ji, B. Halkon, K. Feng, and A. K. Nandi, "Physics-Informed Residual Network (PIResNet) for rolling element bearing fault diagnostics," *Mechanical Systems and Signal Processing*, vol. 200, p. 110 544, Oct. 2023, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2023.110544. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327023004521 (visited on 03/18/2024).

[91] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, Feb. 2019, ISSN: 0021-9991. DOI: 10.1016/j.jcp.2018.10.045. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999118307125 (visited on 03/18/2024).

[92] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for fluid mechanics: A review," en, *Acta Mechanica Sinica*, vol. 37, no. 12, pp. 1727–1738, Dec. 2021, ISSN: 1614-3116. DOI: 10.1007/s10409-021-01148-1. [Online]. Available: https://doi.org/10.1007/s10409-021-01148-1 (visited on 03/18/2024).

[93] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, "Physics-Informed Neural Networks for Heat Transfer Problems," *Journal of Heat Transfer*, vol. 143, no. 060801, Apr. 2021, ISSN: 0022-1481. DOI: 10.1115/1.4050542. [Online]. Available: https://doi.org/10.1115/1.4050542 (visited on 03/18/2024).

[94] Y. A. Yucesan and F. A. C. Viana, "Hybrid physics-informed neural networks for main bearing fatigue prognosis with visual grease inspection," *Computers in Industry*, vol. 125, p. 103 386, Feb. 2021, ISSN: 0166-3615. DOI: 10.1016/j.compind.2020.103386. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361520306205 (visited on 03/18/2024).

[95] Y. A. Yucesan and F. A. C. Viana, "A hybrid physics-informed neural network for main bearing fatigue prognosis under grease quality variation," *Mechanical Systems and Signal Processing*, vol. 171, p. 108 875, May 2022, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2022.108875. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S088832702200070X (visited on 03/18/2024).

[96] C. Sobie, C. Freitas, and M. Nicolai, "Simulation-driven machine learning: Bearing fault classification," *Mechanical Systems and Signal Processing*, vol. 99, pp. 403–419, Jan. 2018, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2017.06.025. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888327017303357 (visited on 08/20/2017).

[97] K. C. Gryllias and I. A. Antoniadis, "A Support Vector Machine approach based on physical model training for rolling element bearing fault detection in industrial environments," *Engineering Applications of Artificial Intelligence*, Special Section: Local Search Algorithms for Real-World Scheduling and Planning, vol. 25, no. 2, pp. 326–344, Mar. 2012, ISSN: 0952-1976. DOI: 10.1016/j.engappai.2011.09.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197611001631 (visited on 03/15/2024).

[98] T. Ai, Z. Liu, J. Zhang, H. Liu, Y. Jin, and M. Zuo, "Fully Simulated-Data-Driven Transfer-Learning Method for Rolling-Bearing-Fault Diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–11, 2023, Conference Name: IEEE Transactions on Instrumentation and Measurement, ISSN: 1557-9662. DOI: 10.1109/TIM.2023.3301901. [Online]. Available: https://ieeexplore.ieee.org/document/10208207 (visited on 03/15/2024).

[99] J. Liu, H. Cao, S. Su, and X. Chen, "Simulation-Driven Subdomain Adaptation Network for bearing fault diagnosis with missing samples," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106 201, Aug. 2023, ISSN: 0952-1976. DOI: 10.1016/j.engappai.2023.106201. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197623003858 (visited on 03/15/2024).

[100] X. Li, S. Li, D. Wei, L. Si, K. Yu, and K. Yan, "Dynamics simulation-driven fault diagnosis of rolling bearings using security transfer support matrix machine," *Reliability Engineering & System Safety*, vol. 243, p. 109 882, Mar. 2024, ISSN: 0951-8320. DOI: 10.1016/j.ress.2023.109882. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832023007962 (visited on 03/15/2024).

[101] C. Liu and K. Gryllias, "Simulation-Driven Domain Adaptation for Rolling Element Bearing Fault Diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 5760–5770, Sep. 2022, Conference Name: IEEE Transactions on Industrial Informatics, ISSN: 1941-0050. DOI: 10.1109/TII.2021.3103412. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9512445 (visited on 03/15/2024).

[102] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," en, *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 2015, ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aaa8415.

[Online]. Available: https://www.science.org/doi/10.1126/science.aaa8415 (visited on 03/21/2024).

[103] R. S. Sutton and A. G. Barto, *Reinforcement Learning, second edition: An Introduction*, en. MIT Press, Nov. 2018, Google-Books-ID: uWV0DwAAQBAJ, ISBN: 978-0-262-35270-3.

[104] Y. Ding, L. Ma, J. Ma, *et al.*, "Intelligent fault diagnosis for rotating machinery using deep Q-network based health state classification: A deep reinforcement learning approach," *Advanced Engineering Informatics*, vol. 42, p. 100 977, Oct. 2019, ISSN: 1474-0346. DOI: 10.1016/j.aei.2019.100977. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034619305506 (visited on 03/21/2024).

[105] S. Fan, X. Zhang, and Z. Song, "Imbalanced Sample Selection With Deep Reinforcement Learning for Fault Diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2518–2527, Apr. 2022, Conference Name: IEEE Transactions on Industrial Informatics, ISSN: 1941-0050. DOI: 10.1109/TII.2021.3100284. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9497679 (visited on 03/21/2024).

[106] J. Zhou, L. Zheng, Y. Wang, C. Wang, and R. X. Gao, "Automated Model Generation for Machinery Fault Diagnosis Based on Reinforcement Learning and Neural Architecture Search," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022, Conference Name: IEEE Transactions on Instrumentation and Measurement, ISSN: 1557-9662. DOI: 10.1109/TIM.2022.3141166. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9673794 (visited on 03/21/2024).

[107] X. Zhao and M. Jia, "A novel unsupervised deep learning network for intelligent fault diagnosis of rotating machinery," en, *Structural Health Monitoring*, vol. 19, no. 6, pp. 1745–1763, Nov. 2020, Publisher: SAGE Publications, ISSN: 1475-9217. DOI: 10.1177/1475921719897317. [Online]. Available: https://doi.org/10.1177/1475921719897317 (visited on 01/23/2024).

[108] Y. Yang, Y. Liao, G. Meng, and J. Lee, "A hybrid feature selection scheme for unsupervised learning and its application in bearing fault diagnosis," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11 311–11 320, Sep. 2011, ISSN: 0957-4174. DOI: 10.1016/j.eswa.2011.02.181. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417411004052 (visited on 01/23/2024).

[109] Y. Lei, F. Jia, J. Lin, S. Xing, and S. X. Ding, "An Intelligent Fault Diagnosis Method Using Unsupervised Feature Learning Towards Mechanical Big Data," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 5, pp. 3137–3147, May 2016, Conference Name: IEEE Transactions on Industrial Electronics, ISSN: 1557-9948. DOI: 10.1109/TIE.

2016.2519325. [Online]. Available: https://ieeexplore.ieee.org/document/7386639 (visited on 01/23/2024).

[110] Z. Zhang, S. Li, J. Wang, Y. Xin, and Z. An, "General normalized sparse filtering: A novel unsupervised learning method for rotating machinery fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 124, pp. 596–612, Jun. 2019, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2019.02.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327019300895 (visited on 01/23/2024).

[111] J. M. Ramírez-Sanz, J.-A. Maestro-Prieto, Á. Arnaiz-González, and A. Bustillo, "Semi-supervised learning for industrial fault detection and diagnosis: A systemic review," *ISA Transactions*, Sep. 2023, ISSN: 0019-0578. DOI: 10.1016/j.isatra.2023.09.027. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019057823004342 (visited on 12/06/2023).

[112] Z. Zhao, Q. Zhang, X. Yu, *et al.*, "Applications of Unsupervised Deep Transfer Learning to Intelligent Fault Diagnosis: A Survey and Comparative Study," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–28, 2021, Conference Name: IEEE Transactions on Instrumentation and Measurement, ISSN: 1557-9662. DOI: 10.1109/TIM.2021.3116309. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9552620 (visited on 01/23/2024).

[113] Y. Ganin and V. Lempitsky, "Unsupervised Domain Adaptation by Backpropagation," en, in *International Conference on Machine Learning*, Jun. 2015, pp. 1180–1189.

[114] X. Li, W. Zhang, and Q. Ding, "Cross-Domain Fault Diagnosis of Rolling Element Bearings Using Deep Generative Neural Networks," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 7, pp. 5525–5534, Jul. 2019, ISSN: 1557-9948.

[115] T. Han, C. Liu, W. Yang, and D. Jiang, "Deep transfer network with joint distribution adaptation: A new intelligent fault diagnosis framework for industry application," en, *ISA Transactions*, vol. 97, pp. 269–281, Feb. 2020, ISSN: 0019-0578. DOI: 10.1016/j.isatra.2019.08.012.

[116] B. Zhang, W. Li, X.-L. Li, and S.-K. Ng, "Intelligent Fault Diagnosis Under Varying Working Conditions Based on Domain Adaptive Convolutional Neural Networks," *IEEE access : practical innovations, open solutions*, vol. 6, pp. 66 367–66 384, 2018, ISSN: 2169-3536.

[117] X. Li, W. Zhang, Q. Ding, and J.-Q. Sun, "Multi-Layer domain adaptation method for rolling bearing fault diagnosis," en, *Signal Processing*, vol. 157, pp. 180–197, Apr. 2019, ISSN: 0165-1684.

[118] D. Wei, T. Han, F. Chu, and M. J. Zuo, "Weighted domain adaptation networks for machinery fault diagnosis," en, *Mechanical Systems and Signal Processing*, vol. 158, p. 107 744, Sep. 2021, ISSN: 0888-3270.

[119] F. Li and H. Wechsler, "Open set face recognition using transduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1686–1697, Nov. 2005, ISSN: 1939-3539.

[120] S. Schmidt and P. S. Heyns, "An open set recognition methodology utilising discrepancy analysis for gear diagnostics under varying operating conditions," *Mechanical Systems and Signal Processing*, vol. 119, Mar. 2019, ISSN: 0888-3270.

[121] P. Peng, J. Lu, T. Xie, S. Tao, H. Wang, and H. Zhang, "Open-set fault diagnosis via supervised contrastive learning with negative out-of-distribution data augmentation," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 3, pp. 2463–2473, 2023, ISSN: 1941-0050.

[122] F. Moller, D. Botache, D. Huseljic, F. Heidecker, M. Bieshaar, and B. Sick, "Out-of-distribution detection and generation using soft brownian offset sampling and autoencoders," ser. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 46–55.

[123] T. Zhou, T. Han, and E. L. Droguett, "Towards trustworthy machine fault diagnosis: A probabilistic Bayesian deep learning framework," en, *Reliability Engineering & System Safety*, vol. 224, p. 108 525, Aug. 2022, ISSN: 0951-8320.

[124] Z. Chen and B. Liu, "Lifelong Machine Learning, Second Edition," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 12, no. 3, pp. 1–207, Aug. 2018, Publisher: Morgan & Claypool Publishers, ISSN: 1939-4608. DOI: 10.2200/S00832ED1V01Y201802AIM037. [Online]. Available: https://www.morganclaypool.com/doi/10.2200/S00832ED1V01Y201802AIM037 (visited on 05/30/2020).

[125] B. Maschler, H. Vietz, N. Jazdi, and M. Weyrich, "Continual learning of fault prediction for turbofan engines using deep learning with elastic weight consolidation," ser. 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), vol. 1, 2020, pp. 959–966.

[126] B. Maschler, T. T. Huong Pham, and M. Weyrich, "Regularization-based Continual Learning for Anomaly Detection in Discrete Manufacturing," *Procedia CIRP*, 54th CIRP CMS 2021 - Towards Digitalized Manufacturing 4.0, vol. 104, pp. 452–457, Jan. 2021, ISSN: 2212-8271. DOI: 10.1016/j.procir.2021.11.076. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827121009744 (visited on 03/22/2024).

[127] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A Survey on Deep Transfer Learning," en, in *Artificial Neural Networks and Machine Learning – ICANN 2018*, V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 270–279, ISBN: 978-3-030-01424-7. DOI: 10.1007/978-3-030-01424-7_27.

[128] W. Li, R. Huang, J. Li, *et al.*, "A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges," *Mechanical Systems and Signal Processing*, vol. 167, p. 108 487, Mar. 2022, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2021.108487. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S088832702100830X (visited on 03/23/2024).

[129] C. Li, S. Zhang, Y. Qin, and E. Estupinan, "A systematic review of deep transfer learning for machinery fault diagnosis," *Neurocomputing*, vol. 407, pp. 121–135, Sep. 2020, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2020.04.045. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220306123 (visited on 03/23/2024).

[130] R. Caruana, "Multitask Learning," en, *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul. 1997, ISSN: 1573-0565. DOI: 10.1023/A:1007379606734. [Online]. Available: https://doi.org/10.1023/A:1007379606734 (visited on 05/30/2020).

[131] S. Guo, B. Zhang, T. Yang, D. Lyu, and W. Gao, "Multitask Convolutional Neural Network With Information Fusion for Bearing Fault Diagnosis and Localization," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 9, pp. 8005–8015, Sep. 2020, Conference Name: IEEE Transactions on Industrial Electronics, ISSN: 1557-9948. DOI: 10.1109/TIE.2019.2942548.

[132] R. Vilalta and Y. Drissi, "A Perspective View and Survey of Meta-Learning," en, *Artificial Intelligence Review*, vol. 18, no. 2, pp. 77–95, Jun. 2002, ISSN: 1573-7462. DOI: 10.1023/A:1019956318069. [Online]. Available: https://doi.org/10.1023/A:1019956318069 (visited on 03/23/2024).

[133] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09, New York, NY, USA: Association for Computing Machinery, Jun. 2009, pp. 41–48, ISBN: 978-1-60558-516-1. DOI: 10.1145/1553374.1553380. [Online]. Available: https://doi.org/10.1145/1553374.1553380 (visited on 03/22/2024).

[134] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Networks*, vol. 3, no. 5, pp. 551–560, Jan. 1990,

ISSN: 0893-6080. DOI: 10.1016/0893-6080(90)90005-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0893608090900056 (visited on 04/29/2024).

[135] J. Rafiee, F. Arvani, A. Harifi, and M. H. Sadeghi, "Intelligent condition monitoring of a gearbox using artificial neural network," *Mechanical Systems and Signal Processing*, vol. 21, no. 4, pp. 1746–1754, May 2007, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2006.08.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327006001750 (visited on 04/30/2024).

[136] G. F. Bin, J. J. Gao, X. J. Li, and B. S. Dhillon, "Early fault diagnosis of rotating machinery based on wavelet packets—Empirical mode decomposition feature extraction and neural network," *Mechanical Systems and Signal Processing*, vol. 27, pp. 696–711, Feb. 2012, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2011.08.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327011003207 (visited on 04/30/2024).

[137] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, Publisher: American Association for the Advancement of Science. DOI: 10.1126/science.1127647. [Online]. Available: https://www.science.org/doi/10.1126/science.1127647 (visited on 04/30/2024).

[138] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research*, vol. 11, no. 110, pp. 3371–3408, 2010, ISSN: 1533-7928. [Online]. Available: http://jmlr.org/papers/v11/vincent10a.html (visited on 04/30/2024).

[139] G. S. Galloway, V. M. Catterson, T. Fay, A. Robb, and C. Love, "Diagnosis of tidal turbine vibration data through deep neural networks: Third European Conference of the Prognostics and Health Management Society 2016," *Proceedings of the Third European Conference of the Prognostics and Health Management Society 2016*, I. Eballard and A. Bregon, Eds., pp. 172–180, Jul. 2016, ISSN: 9781936263219. [Online]. Available: https://www.phmsociety.org/ (visited on 04/30/2024).

[140] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, ISSN: 1558-2256.

[141] M. M. M. Islam and J.-M. Kim, "Automated bearing fault diagnosis scheme using 2D representation of wavelet packet transform and deep convolutional neural network," *Computers in Industry*, vol. 106, pp. 142–153, Apr. 2019, ISSN: 0166-3615. DOI: 10.1016/j.compind.

2019.01.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361518302689 (visited on 05/03/2024).

[142] S. Guo, T. Yang, W. Gao, and C. Zhang, "A Novel Fault Diagnosis Method for Rotating Machinery Based on a Convolutional Neural Network," en, *Sensors*, vol. 18, no. 5, p. 1429, May 2018, Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 1424-8220. DOI: 10.3390/s18051429. [Online]. Available: https://www.mdpi.com/1424-8220/18/5/1429 (visited on 05/03/2024).

[143] R. Liu, G. Meng, B. Yang, C. Sun, and X. Chen, "Dislocated Time Series Convolutional Neural Architecture: An Intelligent Fault Diagnosis Approach for Electric Machine," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1310–1320, Jun. 2017, Conference Name: IEEE Transactions on Industrial Informatics, ISSN: 1941-0050. DOI: 10.1109/TII.2016.2645238. [Online]. Available: https://ieeexplore.ieee.org/document/7797508 (visited on 05/03/2024).

[144] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-D convolutional neural networks," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, Nov. 2016, ISSN: 1557-9948.

[145] W. Sun, R. Zhao, R. Yan, S. Shao, and X. Chen, "Convolutional Discriminative Feature Learning for Induction Motor Fault Diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1350–1359, Jun. 2017, Conference Name: IEEE Transactions on Industrial Informatics, ISSN: 1941-0050. DOI: 10.1109/TII.2017.2672988. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7862893?casa_token=A7Sl09hVePAAAAAA:-GEtqNmssY5ismSBVjq-4t9_ktirvjCeKp2UEG6MQyvPeejOykvSvAs6WPA3QfTvA-xN1ejp (visited on 05/02/2024).

[146] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A New Deep Learning Model for Fault Diagnosis with Good Anti-Noise and Domain Adaptation Ability on Raw Vibration Signals," en, *Sensors*, vol. 17, no. 2, p. 425, Feb. 2017, Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 1424-8220. DOI: 10.3390/s17020425. [Online]. Available: https://www.mdpi.com/1424-8220/17/2/425 (visited on 05/02/2024).

[147] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," en, *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, Publisher: Nature Publishing Group, ISSN: 1476-4687. DOI: 10.1038/323533a0. [Online]. Available: https://www.nature.com/articles/323533a0 (visited on 05/04/2024).

[148] J. Zhu, Q. Jiang, Y. Shen, C. Qian, F. Xu, and Q. Zhu, "Application of recurrent neural network to mechanical fault diagnosis: A review," en, *Journal of Mechanical Science and Technology*, vol. 36, no. 2, pp. 527–542, Feb. 2022, ISSN: 1976-3824. DOI: 10.1007/s12206-022-0102-1. [Online]. Available: https://doi.org/10.1007/s12206-022-0102-1 (visited on 05/03/2024).

[149] R. Saputra, A. Waworuntu, and A. Rusli, "Classification of Indonesian News using LSTM-RNN Method," in *2021 6th International Conference on New Media Studies (CONMEDIA)*, Oct. 2021, pp. 72–77. DOI: 10.1109/CONMEDIA53104.2021.9617187. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9617187 (visited on 05/03/2024).

[150] T. Ghandi, H. Pourreza, and H. Mahyar, "Deep Learning Approaches on Image Captioning: A Review," *ACM Computing Surveys*, vol. 56, no. 3, 62:1–62:39, Oct. 2023, ISSN: 0360-0300. DOI: 10.1145/3617592. [Online]. Available: https://doi.org/10.1145/3617592 (visited on 05/03/2024).

[151] S. A. Mohamed, A. A. Elsayed, Y. F. Hassan, and M. A. Abdou, "Neural machine translation: Past, present, and future," en, *Neural Computing and Applications*, vol. 33, no. 23, pp. 15 919–15 931, Dec. 2021, ISSN: 1433-3058. DOI: 10.1007/s00521-021-06268-0. [Online]. Available: https://doi.org/10.1007/s00521-021-06268-0 (visited on 05/03/2024).

[152] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735 (visited on 05/04/2024).

[153] J. Langford, L. Li, and T. Zhang, "Sparse Online Learning via Truncated Gradient," *Journal of Machine Learning Research*, vol. 10, no. 28, pp. 777–801, 2009, ISSN: 1533-7928. [Online]. Available: http://jmlr.org/papers/v10/langford09a.html (visited on 05/31/2024).

[154] K. Cho, B. van Merriënboer, C. Gulcehre, *et al.*, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179. [Online]. Available: https://aclanthology.org/D14-1179 (visited on 05/04/2024).

[155] M. Hermans and B. Schrauwen, "Training and Analysing Deep Recurrent Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 26, Curran Associates, Inc., 2013. [Online]. Avail-

able: https://papers.nips.cc/paper_files/paper/2013/hash/1ff8a7b5dc7a7d1f0ed65aaa29c04b1e-Abstract.html (visited on 05/04/2024).

[156] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997, Conference Name: IEEE Transactions on Signal Processing, ISSN: 1941-0476. DOI: 10.1109/78.650093. [Online]. Available: https://ieeexplore.ieee.org/document/650093 (visited on 05/04/2024).

[157] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (visited on 05/04/2024).

[158] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A Kernel Two-Sample Test," *Journal of Machine Learning Research*, vol. 13, no. 25, pp. 723–773, 2012, ISSN: 1533-7928. [Online]. Available: http://jmlr.org/papers/v13/gretton12a.html (visited on 05/06/2024).

[159] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep Transfer Learning with Joint Adaptation Networks," en, in *Proceedings of the 34th International Conference on Machine Learning*, ISSN: 2640-3498, PMLR, Jul. 2017, pp. 2208–2217. [Online]. Available: https://proceedings.mlr.press/v70/long17a.html (visited on 05/06/2024).

[160] D. P. Kingma and M. Welling, *Auto-Encoding Variational Bayes*, arXiv:1312.6114 [cs, stat], Dec. 2022. DOI: 10.48550/arXiv.1312.6114. [Online]. Available: http://arxiv.org/abs/1312.6114 (visited on 05/06/2024).

[161] S. Sun, C. Chen, and L. Carin, "Learning Structured Weight Uncertainty in Bayesian Neural Networks," en, in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ISSN: 2640-3498, PMLR, Apr. 2017, pp. 1283–1292. [Online]. Available: https://proceedings.mlr.press/v54/sun17b.html (visited on 05/06/2024).

[162] T. Han, T. Zhou, Y. Xiang, and D. Jiang, "Cross-machine intelligent fault diagnosis of gearbox based on deep learning and parameter transfer," en, *Structural Control and Health Monitoring*, vol. 29, no. 3, e2898, 2022, ISSN: 1545-2263. DOI: 10.1002/stc.2898.

[163] X. Han, Z. Zhang, N. Ding, *et al.*, "Pre-trained models: Past, present and future," *AI Open*, vol. 2, pp. 225–250, Jan. 2021, ISSN: 2666-6510. DOI: 10.1016/j.aiopen.2021.08.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666651021000231 (visited on 05/09/2024).

[164] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018, Publisher: OpenAI. [Online]. Available: https://www.mikecaptain.com/resources/pdf/GPT-1.pdf (visited on 05/09/2024).

[165] L. Prechelt, "Early Stopping - But When?" en, in *Neural Networks: Tricks of the Trade*, G. B. Orr and K.-R. Müller, Eds., Berlin, Heidelberg: Springer, 1998, pp. 55–69, ISBN: 978-3-540-49430-0. DOI: 10.1007/3-540-49430-8_3. [Online]. Available: https://doi.org/10.1007/3-540-49430-8_3 (visited on 05/07/2024).

[166] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in *Advances in Neural Information Processing Systems*, vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html (visited on 05/07/2024).

[167] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6765–6816, Jan. 2017, ISSN: 1532-4435.

[168] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," en, in *Proceedings of the 32nd International Conference on Machine Learning*, ISSN: 1938-7228, PMLR, Jun. 2015, pp. 448–456. [Online]. Available: https://proceedings.mlr.press/v37/ioffe15.html (visited on 05/07/2024).

[169] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014, ISSN: 1533-7928. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html (visited on 05/07/2024).

[170] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, ISSN: 0001-0782. DOI: 10.1145/3065386. [Online]. Available: https://dl.acm.org/doi/10.1145/3065386 (visited on 05/07/2024).

[171] E. P. Carden and P. Fanning, "Vibration Based Condition Monitoring: A Review:" en, *Structural Health Monitoring*, Aug. 2016.

[172] B. Yang, Y. Lei, F. Jia, and S. Xing, "An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings," en, *Mechanical Systems and Signal Processing*, vol. 122, pp. 692–706, May 2019, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2018.12.051.

[173] M. Rao and M. J. Zuo, "A New Strategy for Rotating Machinery Fault Diagnosis Under Varying Speed Conditions Based on Deep Neural Networks and Order Tracking," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2018, pp. 1214–1218. DOI: 10.1109/ICMLA.2018.00197.

[174] O. Janssens, V. Slavkovikj, B. Vervisch, *et al.*, "Convolutional Neural Network Based Fault Detection for Rotating Machinery," en, *Journal of Sound and Vibration*, vol. 377, pp. 331–345, Sep. 2016, ISSN: 0022-460X.

[175] L. Li, M. Zhang, and K. Wang, "A Fault Diagnostic Scheme Based on Capsule Network for Rolling Bearing under Different Rotational Speeds," *Sensors (Basel, Switzerland)*, vol. 20, no. 7, Mar. 2020, ISSN: 1424-8220. DOI: 10.3390/s20071841.

[176] S. Ma and F. Chu, "Ensemble deep learning-based fault diagnosis of rotor bearing systems," en, *Computers in Industry*, vol. 105, pp. 143–152, Feb. 2019, ISSN: 0166-3615.

[177] D. Wei, K. Wang, S. Heyns, and M. J. Zuo, "Convolutional Neural Networks for Fault Diagnosis Using Rotating Speed Normalized Vibration," en, in *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, A. Fernandez Del Rincon, F. Viadero Rueda, F. Chaari, R. Zimroz, and M. Haddar, Eds., ser. Applied Condition Monitoring, Cham: Springer International Publishing, 2019, pp. 67–76, ISBN: 978-3-030-11220-2.

[178] S. Legg and M. Hutter, "A Collection of Definitions of Intelligence," in *Proceedings of the 2007 conference on Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms: Proceedings of the AGI Workshop 2006*, NLD: IOS Press, Jun. 2007, pp. 17–24, ISBN: 978-1-58603-758-1.

[179] T. Han, C. Liu, W. Yang, and D. Jiang, "A novel adversarial learning framework in deep convolutional neural network for intelligent diagnosis of mechanical faults," en, *Knowledge-Based Systems*, vol. 165, pp. 474–487, Feb. 2019, ISSN: 0950-7051. DOI: 10.1016/j.knosys.2018.12.019.

[180] W. Zhang, C. Li, G. Peng, Y. Chen, and Z. Zhang, "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load," en, *Mechanical Systems and Signal Processing*, vol. 100, pp. 439–453, Feb. 2018, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2017.06.022.

[181] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised Domain Adaptation with Residual Transfer Networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 136–144.

[182] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," in *Advances in neural information processing systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., 2009, pp. 1041–1048.

[183] R. Chattopadhyay, Q. Sun, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, "Multisource domain adaptation and its application to early detection of fatigue," *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 4, Dec. 2012, ISSN: 1556-4681.

[184] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin, "Deep Cocktail Network: Multi-Source Unsupervised Domain Adaptation With Category Shift," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3964–3973.

[185] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment Matching for Multi-Source Domain Adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1406–1415.

[186] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," in *Advances in neural information processing systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds., 2008, pp. 129–136.

[187] J. Jiang and C. Zhai, "Instance Weighting for Domain Adaptation in NLP," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 264–271.

[188] L. Gui, R. Xu, Q. Lu, J. Du, and Y. Zhou, "Negative transfer detection in transductive transfer learning," en, *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 2, pp. 185–197, Feb. 2018, ISSN: 1868-808X.

[189] H. Rhee and N. I. Cho, "Efficient and Robust Pseudo-Labeling for Unsupervised Domain Adaptation," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Nov. 2019, pp. 980–985.

[190] Y. Gao, A. J. Ma, Y. Gao, J. Wang, and Y. Pan, "Adversarial open set domain adaptation via progressive selection of transferable target samples," en, *Neurocomputing*, vol. 410, pp. 174–184, Oct. 2020, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2020.05.032.

[191] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A Kernel Method for the Two-Sample-Problem," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds., MIT Press, 2007, pp. 513–520.

[192] A. Gretton, D. Sejdinovic, H. Strathmann, *et al.*, "Optimal kernel choice for large-scale two-sample tests," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., 2012, pp. 1205–1213.

[193] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning Transferable Features with Deep Adaptation Networks," en, in *International Conference on Machine Learning*, Jun. 2015, pp. 97–105.

[194] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," en, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Jun. 2011, pp. 315–323.

[195] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," en, in *International Conference on Machine Learning*, Feb. 2013, pp. 1139–1147.

[196] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Advances in neural information processing systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, 2017, pp. 4148–4158.

[197] P. Zhou, J. Feng, C. Ma, C. Xiong, S. C. H. Hoi, and W. E, "Towards Theoretically Understanding Why Sgd Generalizes Better Than Adam in Deep Learning," en, in *34th International Conference on Neural Information Processing Systems*, Dec. 2020, ISBN: 978-1-71382-954-6.

[198] Y. Chen, X. Liang, and M. J. Zuo, "An improved singular value decomposition-based method for gear tooth crack detection and severity assessment," en, *Journal of Sound and Vibration*, vol. 468, p. 115 068, Mar. 2020, ISSN: 0022-460X.

[199] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-Adversarial Domain Adaptation," en, in *Thirty-Second AAAI Conference on Artificial Intelligence*, Apr. 2018.

[200] J. Chae, S. Lee, J. Jang, S. Hong, and K.-J. Park, "A Survey and Perspective on Industrial Cyber-Physical Systems (ICPS): From ICPS to AI-Augmented ICPS," *IEEE Transactions on Industrial Cyber-Physical Systems*, vol. 1, pp. 257–272, 2023, ISSN: 2832-7004. DOI: 10.1109/TICPS.2023.3323600.

[201] K. Feng, Y. Xu, Y. Wang, *et al.*, "Digital Twin Enabled Domain Adversarial Graph Networks for Bearing Fault Diagnosis," *IEEE Transactions on Industrial Cyber-Physical Systems*, vol. 1, pp. 113–122, 2023, ISSN: 2832-7004. DOI: 10.1109/TICPS.2023.3298879.

[202] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen, "A sparse auto-encoder-based deep neural network approach for induction motor faults classification," *Measurement*, vol. 89, pp. 171–178, Jul. 2016, ISSN: 0263-2241. DOI: 10.1016/j.measurement.2016.04.007.

[203] P. Mucchielli, B. Bhowmik, B. Ghosh, and V. Pakrashi, "Real-time accurate detection of wind turbine downtime - An Irish perspective," *Renewable Energy*, vol. 179, pp. 1969–1989, Dec. 2021, ISSN: 0960-1481. DOI: 10.1016/j.renene.2021.07.139.

[204] W. Yan, J. Wang, S. Lu, M. Zhou, and X. Peng, "A Review of Real-Time Fault Diagnosis Methods for Industrial Smart Manufacturing," *Processes*, vol. 11, no. 2, p. 369, Feb. 2023, Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2227-9717. DOI: 10.3390/pr11020369.

[205] S. Thulasidasan, S. Thapa, S. Dhaubhadel, G. Chennupati, T. Bhattacharya, and J. Bilmes, "An Effective Baseline for Robustness to Distributional Shift," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2021, pp. 278–285. (visited on 11/16/2023).

[206] C. Geng, S.-J. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3614–3631, Oct. 2021, ISSN: 1939-3539.

[207] M. D. Scherreik and B. D. Rigling, "Open set recognition for automatic target classification with rejection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 2, pp. 632–642, 2016, ISSN: 1557-9603.

[208] T. Han and Y.-F. Li, "Out-of-distribution detection-assisted trustworthy machinery fault diagnosis approach with uncertainty-aware deep ensembles," en, *Reliability Engineering & System Safety*, vol. 226, p. 108 648, Oct. 2022, ISSN: 0951-8320.

[209] M. Abdar, F. Pourpanah, S. Hussain, *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," en, *Information Fusion*, vol. 76, pp. 243–297, Dec. 2021, ISSN: 1566-2535. DOI: 10.1016/j.inffus.2021.05.008.

[210] A. Lundgren and D. Jung, "Data-driven fault diagnosis analysis and open-set classification of time-series data," *Control Engineering Practice*, vol. 121, p. 105 006, Apr. 2022, ISSN: 0967-0661.

[211] X. Yu, Z. Zhao, X. Zhang, *et al.*, "Deep-learning-based open set fault diagnosis by extreme value theory," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 185–196, 2022, ISSN: 1941-0050.

[212] Y. Tian, Z. Wang, L. Zhang, C. Lu, and J. Ma, "A subspace learning-based feature fusion and open-set fault diagnosis approach for machinery components," *Advanced Engineering Informatics*, vol. 36, pp. 194–206, Apr. 2018, ISSN: 1474-0346. DOI: 10.1016/j.aei.2018.04.006.

[213] J. Xu, M. Kovatsch, and S. Lucia, "Open set recognition for machinery fault diagnosis," ser. 2021 IEEE 19th International Conference on Industrial Informatics (INDIN), 2021.

[214] H. Zhou, W. Chen, L. Cheng, D. Williams, C. W. De Silva, and M. Xia, "Reliable and intelligent fault diagnosis with evidential VGG neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, 2023, ISSN: 1557-9662.

[215] H. Zhou, W. Chen, L. Cheng, J. Liu, and M. Xia, "Trustworthy fault diagnosis with uncertainty estimation through evidential convolutional neural networks," *IEEE Transactions on Industrial Informatics*, 2023, ISSN: 1941-0050.

[216] Y. J. Choe, "Comparing forecasters and abstaining classifiers," en, phd, Carnegie Mellon University, Jun. 2023.

[217] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," ser. Advances in Neural Information Processing Systems, vol. 31, 2018.

[218] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review," *Engineering Applications of Artificial Intelligence*, vol. 115, p. 105 151, Oct. 2022, ISSN: 0952-1976.

[219] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," ser. Advances in Neural Information Processing Systems, vol. 31, 2018.

[220] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li, "Open Set Learning with Counterfactual Images," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 613–628.

[221] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," en, in *Sixth international conference on learning representations*, Feb. 2018.

[222] W. Zhang, X. Li, H. Ma, Z. Luo, and X. Li, "Open-set domain adaptation in machinery fault diagnostics using instance-level weighted adversarial learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7445–7455, Nov. 2021, ISSN: 1941-0050.

[223] C. Zhao and W. Shen, "Dual adversarial network for cross-domain open set fault diagnosis," *Reliability Engineering & System Safety*, vol. 221, p. 108 358, May 2022, ISSN: 0951-8320.

[224] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," en, Feb. 2018.

[225] M. Sensoy, M. Saleki, S. Julier, R. Aydogan, and J. Reid, "Misclassification risk and uncertainty quantification in deep classifiers," ser. 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), 2021, pp. 2483–2491.

[226] P. Zhou, S. Chen, Q. He, D. Wang, and Z. Peng, "Rotating machinery fault-induced vibration signal modulation effects: A review with mechanisms, extraction methods and applications for diagnosis," *Mechanical Systems and Signal Processing*, vol. 200, p. 110 489, Oct. 2023, ISSN: 0888-3270.

[227] Y. Kong, Q. Han, and F. Chu, "Sparsity assisted intelligent recognition method for vibration-based machinery health diagnostics," *Journal of Vibration and Control*, p. 10 775 463 221 113 732, Jul. 2022, ISSN: 1077-5463.

[228] v. d. L. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008, ISSN: ISSN 1533-7928.

[229] Y. Kong, F. Chu, Z. Qin, and Q. Han, "Sparse learning based classification framework for planetary bearing health diagnostics," en, *Mechanism and Machine Theory*, vol. 173, p. 104 852, Jul. 2022, ISSN: 0094-114X.

[230] X. Wang, T. Wang, A. Ming, W. Zhang, A. Li, and F. Chu, "Generalized cross-severity fault diagnosis of bearings via a hierarchical cross-category inference framework," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 7240–7251, Oct. 2022, ISSN: 1941-0050.

[231] M. De Lange, R. Aljundi, M. Masana, *et al.*, "A Continual Learning Survey: Defying Forgetting in Classification Tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, Jul. 2022, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2021.3057446. [Online]. Available: https://ieeexplore.ieee.org/document/9349197 (visited on 11/16/2023).

[232] R. J. Yardley, J. G. Kallimani, J. F. Schank, and C. A. Grammich, *Increasing aircraft carrier forward presence: Changing the length of the maintenance cycle*, en. Rand Corporation, Apr. 2008, ISBN: 978-0-8330-4595-9.

[233] E. Belouadah, A. Popescu, and I. Kanellos, "A comprehensive study of class incremental learning algorithms for visual tasks," en, *Neural Networks*, vol. 135, pp. 38–54, Mar. 2021, ISSN: 0893-6080. DOI: 10.1016/j.neunet.2020.12.003.

[234] D. Zhang and Z. Gao, "An ensemble approach for fault diagnosis via continuous learning," ser. 2021 IEEE 19th International Conference on Industrial Informatics (INDIN), 2021.

[235] D. Li, S. Liu, F. Gao, and X. Sun, "Continual learning classification method and its application to equipment fault diagnosis," en, *Applied Intelligence*, vol. 52, no. 1, pp. 858–874, Jan. 2022, ISSN: 1573-7497. DOI: 10.1007/s10489-021-02455-7.

[236] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" In *Advances in neural information processing systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 3320–3328.

[237] M. Jiménez-Guarneros, J. Grande-Barreto, and J. d. J. Rangel-Magdaleno, "Multiclass incremental learning for fault diagnosis in induction motors using fine-tuning with a memory of exemplars and nearest centroid classifier," en, *Shock and Vibration*, vol. 2021, e6627740, Oct. 2021, ISSN: 1070-9622. DOI: 10.1155/2021/6627740.

[238] W. Yu and C. Zhao, "Broad convolutional neural network based industrial process fault diagnosis with incremental learning capability," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 6, pp. 5081–5091, 2020, ISSN: 1557-9948.

[239] S. Liu, J. Huang, J. Ma, and J. Luo, "Class-incremental continual learning model for plunger pump faults based on weight space meta-representation," en, *Mechanical Systems and Signal Processing*, vol. 196, p. 110 309, Aug. 2023, ISSN: 0888-3270.

[240] X. Yang, P. Zhou, M. J. Zuo, and Z. Tian, "Normalization of gearbox vibration signal for tooth crack diagnosis under variable speed conditions," en, *Quality and Reliability Engineering International*, vol. 38, no. 6, pp. 3072–3098, Dec. 2021, ISSN: 1099-1638.

[241] K. Feng, K. Wang, Q. Ni, M. J. Zuo, and D. Wei, "A phase angle based diagnostic scheme to planetary gear faults diagnostics under non-stationary operational conditions," *Journal of Sound and Vibration*, vol. 408, pp. 190–209, 2017.

[242] X. Yang, M. J. Zuo, and Z. Tian, "Development of crack induced impulse-based condition indicators for early tooth crack severity assessment," en, *Mechanical Systems and Signal Processing*, vol. 165, p. 108 327, Feb. 2022, ISSN: 0888-3270.

[243] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018, ISSN: 1939-3539.

[244] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, p. 3521, Mar. 2017.

[245] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," en, ser. Proceedings of the 35th International Conference on Machine Learning, PMLR, Jul. 2018, pp. 4548–4557.

[246] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, *et al.*, "Progressive neural networks," *arXiv:1606.04671 [cs]*, Sep. 2016.

[247] A. Chaudhry, M. Rohrbach, M. Elhoseiny, *et al.*, "On tiny episodic memories in continual learning," *arXiv:1902.10486 [cs, stat]*, Jun. 2019.

[248] J. Ramapuram, M. Gregorova, and A. Kalousis, "Lifelong generative modeling," en, *Neurocomputing*, vol. 404, pp. 381–400, Sep. 2020, ISSN: 0925-2312.

[249] E. Verwimp, M. De Lange, and T. Tuytelaars, "Rehearsal revealed: The limits and merits of revisiting samples in continual learning," *arXiv:2104.07446 [cs]*, Apr. 2021.

[250] R. Ratcliff, "Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions," *Psychological Review*, vol. 97, no. 2, pp. 285–308, 1990, ISSN: 1939-1471.

[251] R. Aljundi, E. Belilovsky, T. Tuytelaars, *et al.*, "Online continual learning with maximal interfered retrieval," H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Eds., ser. Advances in neural information processing systems, vol. 32, 2019.

[252] P. Buzzega, M. Boschini, A. Porrello, and S. Calderara, "Rethinking experience replay: A bag of tricks for continual learning," English, IEEE Computer Society, Jan. 2021, pp. 2180–2187, ISBN: 978-1-72818-808-9.

[253] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Eds., ser. Advances in neural information processing systems, vol. 32, 2019.

[254] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-GEM," en, Sep. 2018.

[255] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," 2017, pp. 2001–2010.

[256] J. Schmidhuber, "Deep learning in neural networks: An overview," en, *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015, ISSN: 0893-6080.

[257] J. S. Vitter, "Random sampling with a reservoir," en, *ACM Transactions on Mathematical Software*, vol. 11, no. 1, pp. 37–57, 1985, ISSN: 0098-3500, 1557-7295.

[258] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., ser. Advances in neural information processing systems, vol. 30, 2017.

[259] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel method for the two-sample-problem," ser. Advances in Neural Information Processing Systems, vol. 19, MIT Press, 2006. [Online]. Available: `https://proceedings.neurips.cc/paper/2006/hash/e9fb2eda3d9c55a0d89c98d6c54b5b3e-Abstract.html`.

[260] Q. Han, Z. Jiang, Y. Kong, T. Wang, and F. Chu, "Motor current model for detecting localized defects in planet rolling bearings," *IEEE Transactions on Industrial Electronics*, 2022, ISSN: 1557-9948.

[261] K. Simonyan, A. Vedaldi, and A. Zisserman, *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*, arXiv:1312.6034 [cs], Apr. 2014. DOI: 10.48550/arXiv.1312.6034. [Online]. Available: http://arxiv.org/abs/1312.6034 (visited on 05/10/2024).

[262] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 1135–1144, ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939778. [Online]. Available: https://dl.acm.org/doi/10.1145/2939672.2939778 (visited on 05/10/2024).