공학박사학위논문

딥러닝 기반 회전기계 진단을 위한 진동신호 전처리 및 변환 연구

Investigation on Preprocessing and Transformation of Vibration Signals for Deep Learning Based Diagnosis of Rotating Machinery

2019 년 2 월

서울대학교 대학원

기계항공공학부

정 준 하

Abstract

Investigation on Preprocessing and Transformation of Vibration Signals for Deep Learning Based Diagnosis of Rotating Machinery

Joon Ha Jung
Department of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

Large-scale rotating machinery requires a reliable diagnosis method that accurately predicts health state, since these systems are frequently operated in safety-related and mission-critical systems (e.g., turbines in power plants). Among various methods for rotor-system diagnosis, the data-driven approach has received considerable interest from industry and academia. Specifically, the number of research papers on deep learning based rotor system diagnosis has risen steeply in the past few years. Interest is

driven, in part, by the fact that deep learning algorithms are applicable to complex systems without the need for a complete comprehension of the physics of the system. However, powerful performance of these diagnosis methods can only be achieved with the use of optimal preprocessing techniques for each target system. Thus, this dissertation focuses on developing preprocessing and transformation steps for a deep learning based diagnosis system for rotating machinery. This work specifically focuses on fluid-film bearing rotor systems.

The dissertation investigates three thrusts of preprocessing and transformation of vibration signals: 1) study of the optimal vibration image size, considering filter size, 2) research into a label-based, mini-batch gradient descent method with filter sensitivity analysis, and 3) investigation of a retraining scheme for minor classes in imbalanced data problems.

The first research thrust investigates the size of input images for convolutional neural network (CNN) based diagnosis. As a fluid-film bearing rotor system presents directional dependent health states, vibration images that consider both the temporal and the spatial correlations of omnidirectional regeneration (ODR) signals are suggested. Using the generated images, the results show that the ratio of image size to filter size affects the overall performance. Thus, the optimal range of size ratio for the vibration image is derived in this work by analyzing the performance of various ratios.

The second research thrust suggests a label-based, mini-batch gradient descent method. As the conventional random mini-batch method generates biased mini-batches in several cases, which leads to decreased overall performance, the proposed method can reduce the bias between mini-batches. In addition, various label-based, mini-batch

combinations were studied in this work and their performance deviation was analyzed by filter sensitivity analysis. The result shows that the quantity of properly sensitive filters clearly improves the overall performance of the network.

Finally, the last research thrust proposes a retraining scheme for minority class data in imbalanced data set problems. The proposed two-phase approach uses equally labeled mini-batches, proposed in the second thrust, with oversampling of the minor class samples. Furthermore, in the second phase of training, filters with high sensitivity are frozen and filters with low sensitivity are retrained to represent the minor class samples. The resulting method shows increased performance by improving the recognition of the minority class samples in several imbalanced data set problems.

Keywords: Fluid-film bearing rotor system
Convolutional neural network (CNN) based diagnosis
Optimal vibration image size
Label-based, mini-batch
Imbalanced dataset problem

Student Number: 2013-20715

Table of Contents

List of Tables

List of Figures

Nomenclature

REB	rolling element bearing
AC	alternating current
DC	direct current
CNN	convolutional neural network
y^k	k^{th} feature map of the convolutional layer
s	stride size
p_h	height of the input image
p_w	width of the input image
f_s	size of the square filter
σ	activation function
ω_{mn}	filter values at the (m, n) coordinate
$v_{(m+i\times s)(n+j\times s)}$	pixel values at of the image at ($m+i\times s$, $n+j\times s$) coordinate
b^k	bias for the k^{th} filter
ReLU	rectified linear unit
ADCNN	adaptive CNN
SVM	support vector machine
CI	class imbalanced
SMOTE	synthetic minority over-sampling technique
ML	machine learning
ELM	extreme learning machine
MFE	mean false error
MSFE	mean squared false error
SAE	stacked autoencoder
GPU	graphics processing unit
AGM	averaged gradient magnitude

HOG	histogram of oriented gradients
RSA	ratio of segmented images with AGM larger than a certain threshold
α	user defined threshold for AGM
MGD	mini-batch gradient descent
SGD	stochastic gradient descent
BGD	batch gradient descent
$\boldsymbol{\theta}_k$	weights of the network at the k^{th} iteration
η	learning rate of the gradient descent optimization
$\boldsymbol{B}_k$	k^{th} subset of the training data set
J	cost function for optimization
$\boldsymbol{x}_i$	i^{th} sample of the training data
$\lvert\boldsymbol{B}_k\rvert$	size of the k^{th} mini-batch sub-set
$\boldsymbol{s}_i$	sensitivity vector of the i^{th} filter
f	function mapping input data to the output of the network
$\boldsymbol{x}$	an instance of training data
$\boldsymbol{\theta}$	all parameters of the network
$\boldsymbol{\theta}_i$	all parameters of the network with i^{th} filter having zero values
m	number of nodes in the last layer
c	number of filters in the convolutional neural network
PSF	properly sensitive filters
$N_{\max}$	number of samples in the majority class
$N_{\min}$	number of samples in the minority class
ρ	imbalanced ratio between the minority and majority classes
GAN	generative adversarial network
VAE	variational autoencoder

Chapter 1

Introduction

1.1 Motivation

The maintenance cost of engineered systems is increasing as these systems become more complex [1]. Failure of such systems may lead to catastrophic results, including financial loss. Sudden unexpected failure can even cause casualties. Thus, to prevent the failure of engineered systems a reliable fault diagnosis system is necessary. Specifically, larger systems, such as steam turbines in power plants, are carefully monitored since a single stop causes significant economic loss and social expenses. Also, unscheduled maintenance may take a considerably long time since the repair parts may not be prepared in advance. Therefore, a reliable fault diagnosis system for monitoring safety-related and mission-critical engineered systems must be developed and implemented [2].

Fault diagnosis systems can be categorized into three approaches; model-based, data-based, and hybrid [1, 3, 4]. The model-based approach requires a profound understanding of the physics of the target engineered system. The physics-based model used in this approach can identify the type of faults and can give a clear description of the faults. However, the cost of building a physical model is getting

larger and larger because of the increased complexity of engineered systems. The hybrid approach combines the data-driven and model-based approaches; thus, this approach also requires considerable knowledge of the systems. In contrast, the data-driven technique does not required a comprehensive understanding of the physics of the system. Instead, this approach focuses on the data to discover hidden and meaningful features related to the target system. This approach has gained popularity in both industry and academia in recent years [5, 6].

The popularity of the data-driven approach has increased due to the abundance of data now available. As sensors and data storage devices have becomeme more commonly available, the data acquisition process has become easier. With the rapid growth of modern industry, a large volume of data has been generated in various industry fields. However, data processing to mine useful information from such massive data has created its own challenges. This approach is called "feature engineering" [1, 7-9]. With increased computing power and advanced deep learning research, deep learning based approaches have emerged to solve the issue [6].

Although deep learning approaches are not new, the approaches have gained in popularity since Hinton introduced the fast learning algorithm in 2006 [8, 10]. Hinton proposed a fast learning algorithm for deep belief networks, which encouraged many researchers to advance the deep learning algorithms. While it is gaining in popularity, deep learning is not a universal solution to all problems [11]. Specifically, it does not always learn patterns in a way that the researchers prefer. The training is often done in an adversarial way, which requires different approaches. Thus, data scientists have worked to accentuate both the quality and the quantity of data for such deep learning algorithms to help them act in the desired way. In addition,

data preprocessing must be used, as the performance of the method varies greatly according to the type of data preprocessing. Still, it takes significant effort and time for the algorithms to be trained. Thus, data sets need to be properly preprocessed so that they can be applied to other research areas (e.g., fault diagnosis research) with strong performance.

Correct preprocessing techniques are required in the field of prognostics and health management (PHM). Thus, in this research, preprocessing of vibration signals from fluid-film bearing rotor systems are studied to obtain robust and accurate diagnosis results. Although there is an argument about the range of preprocessing that should be used for fault diagnosis, preprocessing in this study is defined as the range until the algorithm actually starts to train. Training starts when the learning algorithms start to calculate the gradient information from the training data set to update the weights of the network. In this study, preprocessing is viewed as all steps before calculating the gradient of a data set.

1.2 Research Scope and Overview

The goal of this dissertation is to develop favorable preprocessing techniques for deep learning based fault diagnosis. Three thrusts are proposed. First, the optimal size of a vibration image with the ratio of the filter size of a convolutional neural network (CNN) is studied. Next, a label-based, mini-batch by filter sensitivity analysis method is developed. Finally, a retraining scheme for imbalanced data sets is proposed using the results of the first and the second research thrusts. The three thrusts are briefly described below.

Research Thrust 1: Optimal Ratio of Vibration Image Size and Filter Size

Research thrust 1 considers size of the vibration image with respect to that of the filter size, also known as the kernel size in a convolutional neural network (CNN). As the ability of powerful deep learning algorithms can be achieved by optimized hyper-parameters, the ratio between the input image size and the filter size must be determined to obtain high performance. First, the vibration image is generated based on the omnidirectional regeneration (ODR) signals of the gap sensor signals [12, 13]. The process of obtaining the vibration image is simplified compared to that presented in [2]. The generated images showed clear patterns for different health states; however, the prediction accuracy varied with respect to the image size. The smallest size of an image had ambiguous patterns; the results showed that this resulted in low prediction accuracy. As the size increased, the edges of the patterns in the vibration images were distinct in filter sizes, which led to higher prediction accuracy. However, as the size increased greatly, the filters failed to present sharp edges of patterns; as would be expected, this resulted in lower prediction accuracies. In short, the results showed an optimal range of ratio between the size of the image and the size of the filter.

Research Thrust 2: Label-based, Mini-batch Combination Study using Filter Sensitivity Analysis

Research thrust 2 proposes a label-based, mini-batch gradient descent approach

for a balanced data set. Various types of label-based mini-batches are studied for convolutional neural networks (CNN) to analyze the effect of label information when generating mini-batches. The equally labeled mini-batch case showed the most robust results, as compared to other mini-batches. As the performance is determined by the filters of the network, which are the weights of the CNN, sensitivity analysis of each filter is suggested. The sensitivity of a CNN filter can be computed by determining the difference between the output of the standard network and the altered network for which the value of filter has been converted to zero. Then, the total sensitivity of the CNN is determined by the three proposed criteria. The validation case study that examines the vibration signals of a fluid-film bearing rotor testbed indicates that the equally labeled mini-batch has the most sensitive filters. Also, the results show that the total sensitivity of the CNN is strongly correlated to the performance of the network.

Research Thrust 3: Retraining Minor Class Schemes for Imbalanced Data Sets

Research thrust 3 develops an efficient and robust retraining scheme for imbalanced data sets. As many engineered systems in the field run in normal conditions, the size of the available fault data set is usually much smaller than that of the data for normal conditions. To increase the performance of the imbalanced data set, a two-phase retraining scheme is proposed. In phase I, oversampling of minority classes is used to generate equally labeled mini-batches. In phase II, the least minority class is retrained by using only the insensitive filters identified in phase I. The classes are temporally turned into two classes; the least minority class

and other classes. By turning this into a two-class problem, the filters can focus on learning the general representation of the least minority class. The proposed scheme is validated on the imbalanced data set generated from the data set from the testbed. The least minority class of the imbalanced data set is fixed to an imbalanced ratio of 0.001. The results show that the phase I training can enhance the overall performance of the network. Also, the phase II training also increases the performance by elevating the accuracy of the least minority class.

1.3 Dissertation Layout

The layout of this dissertation is as follows. Chapter 2 provides a literature review of fluid-film bearing rotor system diagnosis and the deep learning based fault diagnosis. Chapter 3 describes the acquisition of data used in the following chapters. In Chapter 4, optimal vibration image generation using the gap sensor signal is suggested. Chapter 5 suggests a label-based, mini-batch gradient descent method and analyzes the results using the developed filter sensitivity analysis. Then, Chapter 6 introduces a two-phase retraining scheme for imbalanced data sets by enhancing the prediction accuracy of the minority class. Finally, Chapter 7 concludes the dissertation by summarizing the research and suggesting future research.

Chapter 2

Literature Review

This chapter reviews the literature related to convolutional neural network based diagnosis for fluid-film bearing rotor systems, specifically the review provides: (1) an overview of fluid-film bearing rotor systems, (2) an overview of convolutional neural network (CNN) based rotor system diagnosis, (3) analysis of the ratio between the input image size and the input filter size, (4) an examination of segmentation of input data for deep learning based diagnosis, and (5) a review of the strategy of deep learning based diagnosis for imbalanced data sets.

2.1 Overview of Fluid-film Bearing Rotor Systems

Fluid-film bearing rotor systems are frequently used in industrial machines that require safe and reliable operation. For example, turbines and pumps in power plants use fluid-film bearings to maintain system safety even in heavy-load and high-speed conditions. Because the fluid in the bearings supports the rotors, stable operation is possible without direct contact between the rotor and the stator. Conventionally, the diagnosis of fluid-film bearing rotor system uses the vibration signals from proximity sensors located close to the bearing. This section briefly describes the overall fluid-film bearing rotor system, and the data acquisition from the sensors in Sections 2.1.1

and 2.1.2, respectively. In Section 2.1.3, the physical characteristics of the vibration signals of major health states are described.

2.1.1 Structure of Fluid-film Bearing Rotors

The major feature of fluid-film bearings is that the lubrication oil fills the space between the rotor (shaft) and the stator (housing). The structure of a plain fluid-film bearing as presented in Figure 2-1 is simple compared to rolling element bearings (REBs). The oil between the shaft and the housing reduces the friction and increases the damping, which plays a crucial role in this system [14-16]. Thus, an extra pump that provides the oil into the bearing as well as cooling devices is required to maintain such a bearing in a healthy condition. Despite the requirement for extra devices, fluid-film bearings are widely used in large-scale machinery, such as turbines and pumps in power plants, where highly reliable operation is required. When operated properly, fluid-film bearings can keep vibration of a 500MW turbine system under a hundred micrometers.

Numerous types of fluid-film bearings are used, according to the rotor characteristics. The simplest type is the plain-sleeve fluid-film bearing, which is presented in Figure 2-1. Another type, called multi-lobe fluid-film bearings, has multiple lobes in the bearing, which preload the oil. Further, the tilting pad bearing is the most stable type, as the bearings are separated into a few tilting pads. As the outer race is divided into a few pads, the negative hydrodynamic effects can be reduced by allowing free tilting. The name ‘tilting pad’ comes from the pads that tilt via a hinged support. The number of pads varies according to the usage of the bearing.

As the structure and the dynamics are different from REB systems, the characteristics of the rotors show different behavior. Considering the fluid-film bearing features, proximity sensors are used to measure vibration of such rotor systems; this process is described in the next section.

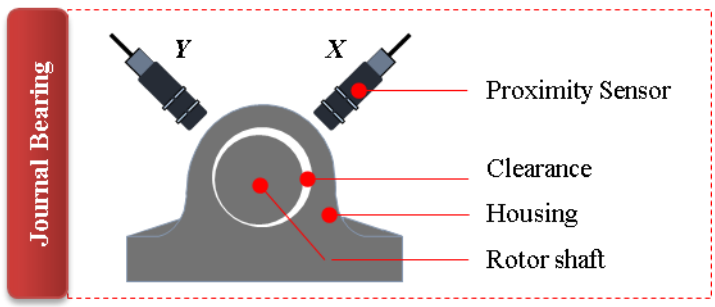


Figure 2-1 Structure of a plain-sleeve, fluid-film bearing and proximity sensors.

2.1.2 Data Acquisition of Vibration Signals from Fluid-film Bearing Rotors

Proximity sensors, also known as gap sensors, are used to measure vibration signals from fluid-film bearing rotor systems. Unlike an accelerometer, which is widely used in measuring vibration of rolling element bearing (REB) rotor systems and gear systems, the gap sensors directly measure the distance between the sensor and the rotor [17-19], as shown in Figure 2-2. In the figure, the proximity sensors measure without any contact with the rotor or housing. By measuring the change of eddy current, the voltage values are obtained through the amplifier and the data acquisition (DAQ) device. The recorded voltages, which are proportional to the gap, include AC and DC components. AC and DC components indicate the vibration of the rotors and

the average of the shaft centerline, respectively [20]. The AC components are mainly used for the diagnosis of rotor systems.

Most fluid-film bearing rotors use proximity sensors in pairs. The paired sensors are installed in a right angle to acquire two independent signals, as shown in Figure 2-3 . The vibration signal acquired from one sensor can be denoted as the *x*-signal, while the signal from the other sensor can be denoted as the *y*-signal. Since *x*- and *y*-sensors are placed in a right angle, the orbit of the shaft centerline position can be determined using the two signals.

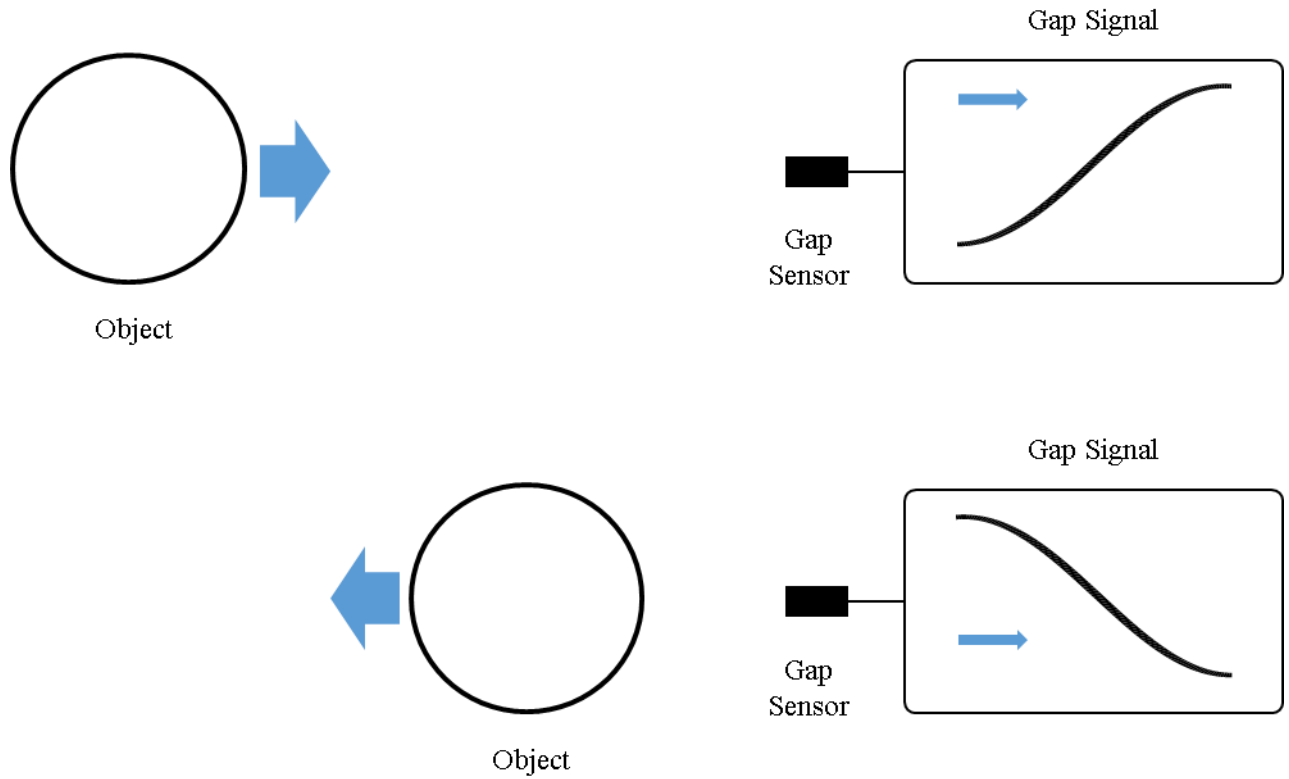


Figure 2-2 The general principle of the proximity sensor.

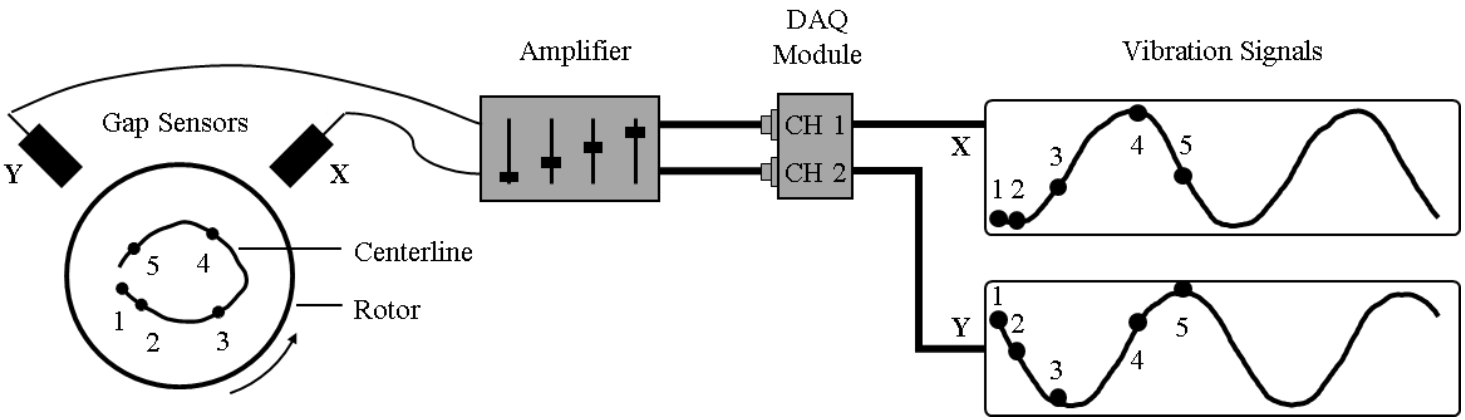


Figure 2-3 Diagram of data acquisition and proximity sensor placement.

2.1.3 Analysis of Vibration Signals for Fluid-film Bearing Rotor Systems

Vibration signals are acquired through proximity sensors located at several axial positions along the rotor. At each axial position, two sensors are placed orthogonally to measure the behavior of the rotor in all directions. The monitoring of the orbits, the trace of the centerline of the rotor, is widely used in the field as orbits present the state of the rotor system directly. By tracking centerline of the rotor, direction oriented health states (directional states) can be diagnosed accurately. In contrast, if a sensor is placed at each axial position, the vibration signals may vary according to the direction of anomaly states. The path of the rotor centerline can be expressed using the orbit plot, which can be generated from the two vibration signals, namely the *x*- and *y*-signals, from the two orthogonally placed sensors. The orbits can be obtained by illustrating the two vibration signals on each 2-dimensional Cartesian coordinate. Note that the *x*- and *y*-signals must be perfectly synced to acquire the accurate orbit plot.

Besides orbit plots, frequency response is another widely used technique for vibration signal analysis. The rotating systems generate recurring signals that can be clearly shown by the frequency response. Thus, the frequency response of each health state shows different results. However, the variation of vibration signals due to directionality affects the frequency response results, which may lead to unreliable diagnosis results. Thus, other techniques, such as full-spectrum techniques, are required for robust diagnosis [21, 22].

For the transient state of the rotors, the short-time Fourier transform may be used instead of frequency response. As the rotational frequency varies in the transient

state, the frequency response must be viewed in the time-domain. In addition, waterfall and cascade plots can be also used for analysis of vibration signals in the transient state. Also, variation of vibration amplitude and phase can be monitored using bode plots.

2.1.4 Summary and Discussion

Fluid-film bearing rotor systems show relatively clear signals, as compared to REB systems or gear systems. For example, 1x of the rotating speed frequency component is dominant in a normal state. This is because the lubrication oil supports the rotor without any mechanical components, such as rolling elements. Through the proximity sensors located in pairs at an axial position, the gap voltage signals are converted to the displacement vibration signals of the rotors. The vibration signals of four different health states indicate that directional health states exist. For the directional health states, the signals may vary with respect to the direction of the anomaly.

Some prior research efforts have tried to consider direction in rotor diagnosis by using the orbit shape and the full-spectrum of vibration signals. Yan, et al. [23] modified the orbit into seven different features to identify the state of a steam turbine generator. Wang, et al. [24] quantified the orbit information with isometric feature mapping to identify faults in rotors. Other researchers also tried to quantify the orbit shape to make more accurate diagnosis of rotors [22, 25-27]. However, in the process of quantifying the orbit shape, detailed physical interpretation of vibration signals may be diminished. In other work, the full-spectrum of vibration signals was used to

see forward and backward whirling frequency components by using *x*- and *y*- signals [18, 28-30]; however, the method could not consider vibration signals in all directions. Thus, to diagnose the directional health states robustly, a vibration image generated by the omnidirectional regeneration (ODR) technique is used in this study.

2.2 Overview of Convolutional Neural Network (CNN) based Rotor System Diagnosis

This section provides an overview of the current state-of-the-art of convolutional neural network (CNN) based rotor system diagnosis. As CNNs have shown reliable performance in vision recognition, more and more researchers are using CNN in the PHM area as well [1]. Thus, this section provides basic principles of CNN in Section 2.2.1. A summary of diagnosis research using CNN is provided in Section 2.2.2.

2.2.1 Image Recognition by Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) are widely used in various applications, including image recognition, because CNN has showed promising performance in many studies [31-34]. The CNN model was inspired by the mechanism of the visual cortex in the brain [31], which can identify patterns in images. The standard structure of a CNN contains convolutional layers, pooling layers, and fully connected layers. Combinations of the convolutional layer and the pooling layer make the deep structure of the network; the few fully connected layers come at the end.

In the convolutional layer, the input images are convoluted with weights called filters. As images contain high correlation information in a local region, the filters extract such local connectivity features by convoluting with the image. The k^{th} feature map of the convolutional layer, y_{ij}^{k}, with a grayscale (2-dimensional) image can be expressed as:

$$y_{ij}^{k} = \sigma\left(\sum_{m=1}^{f}\sum_{n=1}^{f} \omega_{mn}^{k} v_{(m+i\times s)(n+j\times s)} + b^{k}\right) \tag{2.1}$$

$$\left(0 \le i \le \frac{p_h - f_s}{s}, 0 \le j \le \frac{p_w - f_s}{s}\right)$$

where i and j denote the coordinates of the feature map, s denotes the size of stride, p_h and p_w denote the height and width of the input image, respectively, f_s denotes the size of the filter, σ denotes the activation function, ω_{mn} denotes the filter values at the (m, n) coordinate, $v_{(m+i\times s)(n+j\times s)}$ denotes the pixel values of the image at the ($m+i\times s$, $n+j\times s$) coordinate, and b^k denotes the bias for the k^{th} filter. After the convolution, activation functions, such as rectified linear unit (ReLU), are used; this allows the network to learn nonlinear features [35].

The convoluted results are then passed to the pooling layer, where one value is pooled out from a small sub-region. By applying the pooling function, similar features in sub-regions can be reduced, while maintaining the significant features. In addition, the pooling also reduces the effect of slightly modified and shifted patterns [9]. Among various pooling functions, max-pooling is the most widely used function; it returns only the maximum value. As pooling extracts one value from a few feature values, the size of the output is reduced compared to that of the input.

The fully connected (FC) layers come at the last stage of CNN. The role of the FC layers is to provide a classification using the features extracted in the convolutional and pooling layer in the previous layers. Thus, the number of nodes in the very last layer equals the number of classes. By comparing the output values of the nodes at the last layer, the data can be classified into the class that has the maximum value.

To extract high-level features from complex images, deep structures of CNN, such as ALEXNET [31], VGG [36], ResNet [37], GoogLeNet [38] have been developed. However, the work described in this paper uses shallow CNN structures for two reasons. First, images from fluid-film bearing rotor systems used in the case study are relatively simple, as compared to images used to train deep CNNs. Second, shallow structures can be used to analyze the filters that decide the performance of the network. As the structure gets deeper, the network gives higher-level features, and these features may not be interpretable. Considering these two aspects, the network used in the work outlined this paper is described in Section 5.1.

2.2.2 CNN-based Rotor System Diagnosis Based on Vibration Signals

CNN has been adopted and modified to enhance the performance of diagnosis for various rotating machinery. The application that used the most CNNs is the rolling element bearing (REB) system. As conventional bearing fault diagnosis uses the vibration from an accelerometer, the input data are transformed into 2-dimensional matrices. In [39], the vibration signals are transformed into matrices and used as the inputs to the hierarchical adaptive CNN (ADCNN). The modified CNN uses

hierarchical networks, one for fault diagnosis and the other for the identification of fault size. Each network, which includes three convolutional and pooling layers, is tested by various learning rates to obtain the best performance. In contrast, preprocessed acceleration data are used as an input to a conventional CNN [40]. The frequency response of one second long vibration signals is set as the sample. Since two accelerometers are used, the frequency responses of the two vibration signals are stacked to form a 2-dimensional input image. Both research efforts showed improved performance compared to that of the traditional approaches.

Also, CNN has been used to diagnose faults of gearboxes [41]. Vibration signals from accelerometers have been used as well. Here, the signals from multiple sensors are preprocessed and significant features are extracted. Then, two-dimensional matrices are formed and used as the inputs to the CNN-based classifier. Although 1-layer CNN was used in this study, CNN outperformed the conventional support vector machine (SVM) classifier approach.

In addition, research on motor fault detection using CNN has been conducted recently [42]. The motor fault detection is achieved by using stator current signal, which is a widely used data type to identify motor faults. Since the number of current signals is single, 1-dimensional CNN is used. The results showed improved performance without developing hand-crafted features.

2.2.3 Summary and Discussion

Convolutional neural networks (CNN) recognize patterns of images by convolution with filters, also known as kernels. The training process mainly updates the filters to

recognize the input image patterns through repeated epochs. Occasionally, the non-linearity features can be learned through the non-linear activation function, as well as the deep structured layers. Due to the high performance of CNN in vision recognition, CNNs have been widely used in fault detection and diagnosis research, as stated in the previous section. The research on fault diagnosis of rolling element bearings, motors, and gearboxes has shown increases in performance when CNN was used. In addition, pumps and general rotor systems have also been successfully diagnosed by CNN [43, 44].

However, most of the studies have not customized input data of CNN. For example, the transformation of inputs were processed without any explanation or consideration of the physical implications. As convolutions between an input image and a filter grant features of highly correlated adjacent pixels, the transformation of inputs must consider the physical interpretation. In order to properly use CNN in fault diagnosis research, insights about preprocessing of data should be considered. This need provides the motivation for the research described in this dissertation.

2.3 Strategy for Deep Learning Based Diagnosis of Class-Imbalanced Data Sets

As most engineered systems operate in normal conditions, the size of the available fault data set is often smaller than that of the normal data set. In particular, large-scale systems require an extremely high cost to acquire data for anomaly states. Thus, the class imbalanced (CI) case is common for developing the deep learning based diagnosis systems. Therefore, numerous research efforts have been conducted

regarding the imbalanced data issue. Proposed solutions can be categorized into three approaches: data-level, algorithm-level, and hybrid approaches [45, 46]. Each approach is reviewed in turn in the following sections. As the amount of available research that deals with deep learning based rotor system diagnosis is not enough, the range of literature is extended to also examine general CI problems.

2.3.1 A Data-Level Strategy for Class Imbalanced Data Set Training

The data-level approach proposed to resolve the class imbalanced (CI) training data problem has been studied since 1990s [47]. Data-level approaches developed can be used in deep learning algorithms as well. The typical data-level approach increases the data size by oversampling the minority class data or reducing the data size by undersampling the majority class data [48]. The oversampling and undersampling techniques can reduce the bias of the training data, which prevents the training process from being biased to the majority class [48]. Specifically, the sampling can be done by picking random samples or choosing directed samples [47, 49]. Thus, the combination of over-/under-sampling and choosing random/directed samples results in a few different approaches. There is no universal approach that fits in all cases; instead, the optimal approach depends on the distribution of the data set. In addition, some approaches use both oversampling and undersampling at the same time [50].

Numerous literature examples show that synthesizing the minority class samples can improve performance more effectively than oversampling the few samples [50]. The widely used synthesizing method, synthetic minority over-sampling technique (SMOTE), showed enhanced results compared to other approaches [50]. As SMOTE

randomly creates samples on a line between minority samples, the CI problem can be overcome in certain cases. However, for cases of densely distributed classes, the SMOTE may create overlapped samples, which does not help increase the overall performance. Thus, a few other modified approaches have been developed. The borderline-SMOTE and safe-level-SMOTE approaches were suggested to reduce the overlapped samples derived by SMOTE [51, 52]. Other techniques have also been developed to synthesize samples by reducing the random effects [53-56].

2.3.2 An Algorithm-Level Strategy for Class Imbalanced Data Set Training

The issue of class imbalanced (CI) data sets at the algorithm-level has been studied by numerous researchers who have used machine learning (ML) as a classifier [57]. As the algorithms are trained based on the training data set, conventional training would give a biased classifier to the majority class samples. Thus, various researchers have suggested modifying the classifier to focus more on the minority class. Specifically, kernels in support vector machine (SVM) are modified considering the distribution of the minority class [58-60]. In addition, the neurofuzzy algorithm and extreme learning machines were modified to present more general features of the minority class [61, 62].

For deep-learning algorithms, modification of the cost function for the optimization was used frequently [45, 48, 63-71]. To learn the generalized features of the training data set, the loss values of the minority class should be modified [72-76]. One study considered the cost of minority class samples by adding a customized

layer before the last layer of CNN [45]. Another study suggested mean false error (MFE) and the mean squared false error (MSFE) as the loss function when training a deep neural network [67]. In [48], two different stacked autoencoders (SAE) are used for learning different features; these showed better prediction results. Also, the loss function term that reflects the inter-cluster and inter-class margins can reduce the effect of the CI data sets [71].

2.3.3 Summary and Discussion

To tackle the class imbalanced (CI) problem in machine learning, three categories of approaches can be used: data-level, algorithm-level, and hybrid approaches. In fact, as the hybrid approaches combine data-level and algorithm-level approaches, they can be either classified into data-level or algorithm-level approaches. Through the literature review, it is clear that data-level approaches were studied primarily as machine learning techniques became popular. However, as the number of deep learning algorithms has increased since the year 2010, hybrid as well as algorithm-level approaches have been frequently used to deal CI problems. Generally, deep-learning approaches are applied to large data sets, which makes some of the complex data-level algorithms inefficient for tackling the CI problem. This issue encourages study of the hybrid or algorithm-level deep learning techniques for CI problem.

Furthermore, most of the studies that examine CI problems consider the ratio between the size majority and minority class samples of 0.01. However, the systems in the field may have a lower imbalanced ratio, such as 0.001 or less. These extreme imbalanced cases may have significant negative effects when conventional deep-

learning algorithms are used. Since training of deep learning uses gradient information of mini-batches, the composition of mini-batches may vary along with the imbalanced ratio, which directly leads to performance degradation. In addition, the various imbalanced ratio cases in a multi-class setting have not been considered in the literature. Occasionally, the size of each class in the training data set may differ in order, which may have undesirable results. Thus, a severe imbalanced ratio with various multiple cases for deep learning based rotor system diagnosis should be considered to assure the reliability of the diagnosis systems.

Chapter 3

Description of Testbed Data

This section describes the data from the testbed that is used in the research throughout the dissertation. The data was acquired from a Bently Nevada Rotor Kit (RK4), which is a testbed for a fluid-film bearing rotor system. Four different health states were considered based on the experimental setup, which is presented in Figure 3-1. Section 3.1 describes the configuration of the testbed and Section 3.2 presents an analysis of the acquired vibration signals for each health state.

3.1 Configuration of the Testbed

The vibration signals from proximity sensors of the four different health states were acquired from the testbed. The two-shaft rotor was driven by the motor at 3,600 rpm, steadily. The shorter shaft was directly driven by the motor, and the longer shaft was connected to the other end of the shorter shaft by a flexible coupling. The flexible coupling was used to reduce the signals of the motors. At the longer shaft, an 800-gram disc was fixed in the middle of the two fluid-film plain bearings supporting the shaft. For every test, a balancing process was performed to make the normal vibration amplitude consistent. Also, two channels of vibration signals from two proximity sensors at the longer shaft were acquired, along with a tacho signal. Based

on the tacho signal, the raw signals were resampled to have an equal number of samples in a vibration rotation to reduce the measurement uncertainty. For each health state, five repeated tests were conducted to avoid simplicity in the data set. Overall, five data sets, which contain four different health states for each set, were obtained and used throughout the research described in this dissertation.

3.2 Analysis of Vibration Signals for Four Health States

Four different health states were tested. First, the normal health state was defined as a system with only a small amount of unbalance. Through the balancing procedure, the root-mean-square (RMS) of the signal was set as 10 μm at 3,600 rpm. The measured signals follow a simple sinusoidal wave shape, as shown in Figure 3-2. Also, the vibration signals from actual gap sensors are similar to each other in terms of shape and amplitude. This indicates that the ***normal state*** is a non-directional health state.

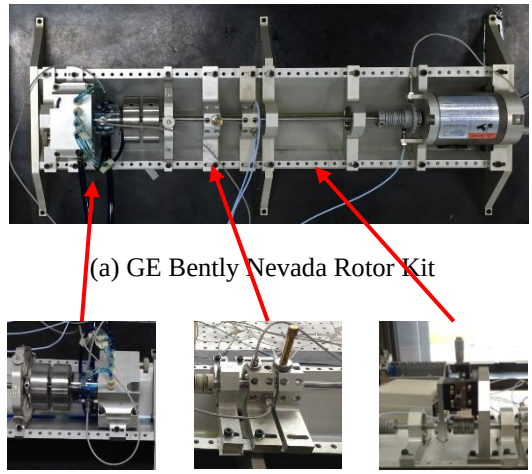
The ***oil whirl state*** is another non-directional health state, since the oil in the bearings affects the rotor system around the entire circumference of the rotor. Instability of the oil was introduced in the testbed by using the oil whirl kit of the RK4, which controls the pressure of the oil supply. At the transient states, the pressure was raised to a certain level to prevent oil whip. Only at the 3,600 rpm steady-state, was the pressure dropped and the oil whirl anomaly state created. The signals in Figure 3-3 were acquired from the gap sensors in the oil whirl kit.

The ***rubbing state***, precisely impact-rubbing, is a directional health state. The

testbed setting was exactly same as that of the normal state, but rubbing was implemented by forcing a rubbing screw to contact the shaft near the 2nd bearing. The screw partially contacts during each cycle of rotation, which can be viewed as an impact. The two measured signals show differences in shape and amplitude, and the rubbing effect is reflected as the cut at the peak, as presented in Figure 3-4. The rubbing test was also done after the balancing procedure. The rubbing was then implemented at the steady state of 3,600 rpm.

The ***misalignment state*** was tested by shifting the shorter shaft downwards using a customized jig. Since the shaft was shifted in one direction, this state can be grouped as another directional health state. The measured signals shown in Figure 3-5 also indicate that shaft rotation is affected by the direction of the shift. The two signals are very different in terms of the shape and amplitude. Each test was performed after a balancing procedure, and was run at a 3,600 rpm steady-state.

Overall, the test configurations define the normal and the oil whirl states as non-directional health states, while rubbing and misalignment states were defined as directional health states. This can be confirmed using a simple analytical example, which is provided in Chapter 4.



(a) GE Bently Nevada Rotor Kit

(b) Oil whirl (c) Rubbing (d) Misalignment

Figure 3-1 Experimental setup of rotor kit RK4.

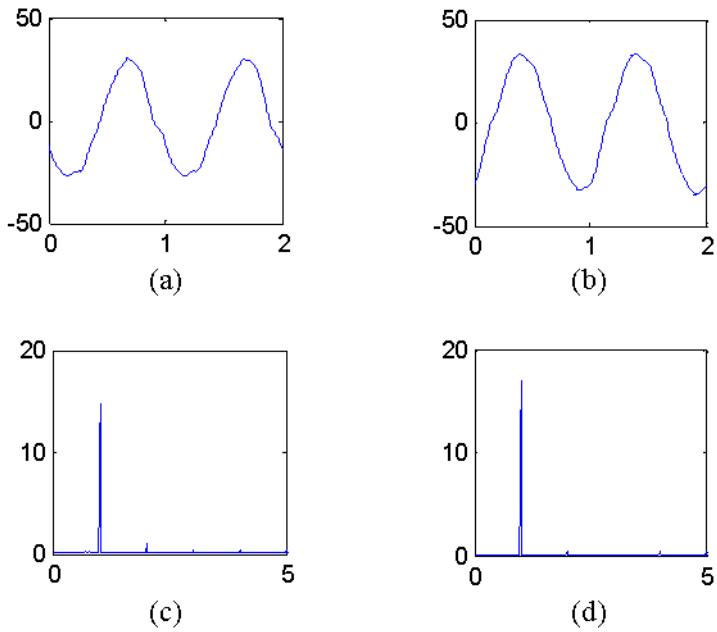


Figure 3-2 Normal state time signals from (a) *x* and (b) *y* gap sensors; and frequency responses from (c) *x* and (ds) *y* gap sensors.

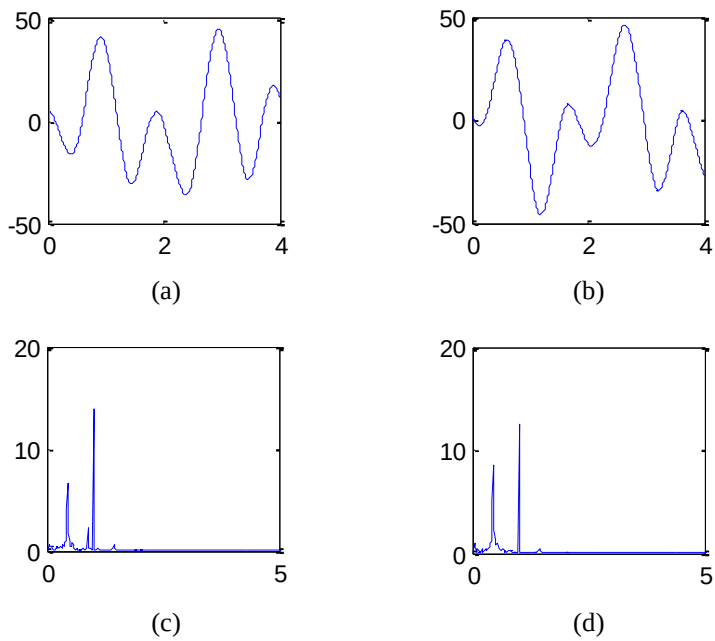


Figure 3-3 Oil whirl state time signals from (a) *x* and (b) *y* gap sensors; and frequency responses from (c) *x* and (d) *y* gap sensors.

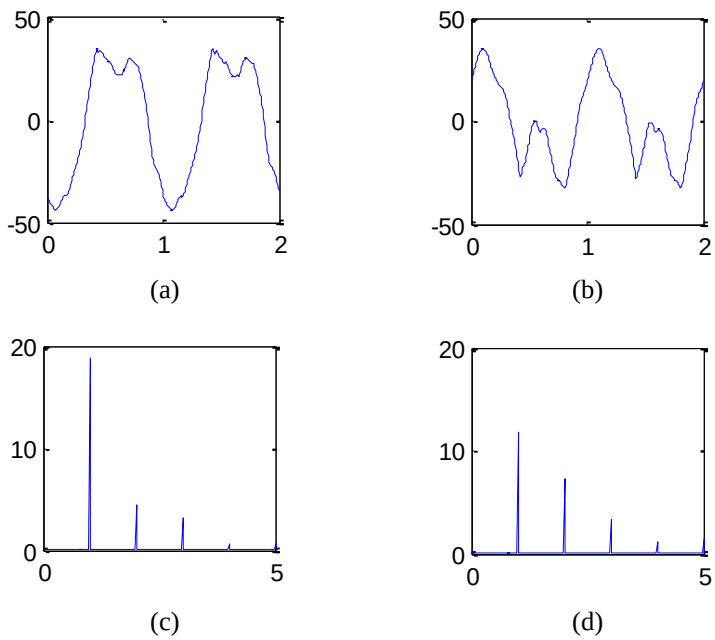


Figure 3-4 Rubbing state time signals from (a) *x* and (b) *y* gap sensors; and frequency responses from (c) *x* and (d) *y* gap sensors.

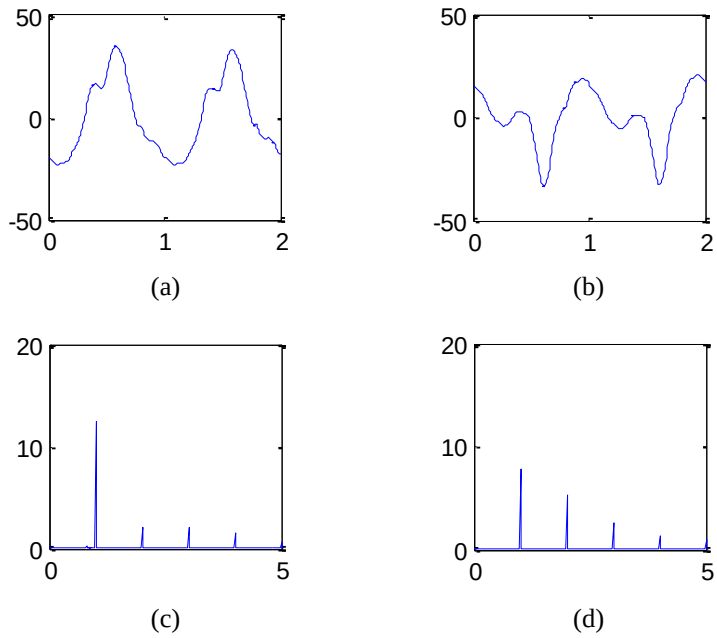


Figure 3-5 Misalignment state time signals from (a) *x* and (b) *y* gap sensors; and frequency responses from (c) *x* and (d) *y* gap sensors.

Chapter 4

Determining an Optimal Size of Vibration Images Considering Filter Size

The size of input images for convolutional neural network (CNN) based diagnosis need to be determined with respect to the filter size. The size ratio between the input images and the filters in CNN can be viewed as one hyper-parameter that is decided heuristically. Specifically, the image pattern that separates one class from another class does vary according to the type of images. For example, recognition of a handwritten digit is different from recognition of traffic lights in a vehicle [77]. The digits can be distinguished by a few particular edges of each digit; thus, a filter size that includes the edges is adequate. In contrast, traffic lights may be detected as a whole traffic light, among other objects in an image; this requires the filter size to be equivalent or bigger than the traffic light. Thus, the size ratio between the input and the filters must be defined for each type of image.

To use CNN-based diagnosis for fluid-film bearing rotor systems, the first preprocessing step is to generate the input images from the vibration signals. As few anomalies of fluid-film bearing rotor systems are directionally dependent of anomaly position, the vibration signals from all round directions should be acquired for robust

diagnosis [12, 78, 79]. Using these vibration signals, a vibration image can be generated, which represents the directionality of the anomaly or anomolies. Then, different sizes of vibration images with a fixed-filter size CNN can be studied to find the most suitable vibration image size.

This chapter is divided into two parts. In Section 4.1, the vibration image generation process is described. Section 4.2 shows the results of the different image sizes with respect to fixed-size filters.

4.1 Vibration Image Generation by Omnidirectional Regeneration (ODR)

This section describes the vibration image generation steps that were used through the research described in this dissertation. The motivation for generating images from the gap sensor signals of a fluid-film bearing rotor system is to robustly diagnose the anomaly health states. As some anomaly states present different vibration signals through fixed gap sensors, vibration signals around the circumference of the rotor are required to obtain robust prediction results. However, fluid-film bearing rotor systems only use two gap sensors, located orthogonally in an axial location of the rotor. Since the location of gap sensors is limited, a novel method is required to acquire vibration signals. Thus, the omnidirectional regeneration (ODR) technique is used to generate any vibration signals from the two acquired signals. Based on the ODR signals, a vibration image is generated that shows distinct patterns for each health state.

This section includes the directionality of some health states in Section 4.1.1. Then, the process of ODR is described briefly in Section 4.1.2. Lastly, the process for modified image generation from vibration signals is explained in Section 4.1.3.

4.1.1 Directional Health States in Fluid-film Bearing Rotor Systems

Health states, such as rubbing and misalignment, present inconsistent vibration signal patterns due to the directional nature of rubbing and misalignment. For these direction-oriented health states, the two fixed sensors in perpendicular position (as presented in Figure 4-1) may not accurately represent the state of the rotor system because an anomaly can happen in a direction that is not in line with the sensors. Thus, this section shows the variance of the directional health states with respect to the direction of the anomalies by using a simple mathematical model and the experimental data.

A simple lumped rotor model is used to verify the directionality of rubbing and misalignment states. The model consists of a shaft with a disc supported by two fluid-film bearings, as shown in Figure 4-2. To make the model simple, it is assumed that the damping and stiffness only exist at the bearing. This is presented with a black triangle. In addition, the mass of the shaft, as well as the gyroscopic effect of the disc, are ignored. The model of a normal state that has a bit of unbalance can be presented as follows:

$$M\ddot{x} + C\dot{x} + Kx = mr\omega^2\cos(\omega t + \delta) \tag{4.1}$$

$$M\ddot{y} + C\dot{y} + Ky = mr\omega^2\sin(\omega t + \delta) \tag{4.2}$$

where M, C, K are mass, damping, and stiffness, respectively, and m, r, and δ are unbalance mass, radius, and angular orientation, respectively, ω is rotational speed, x and y are lateral displacements of the disk in orthogonal directions as a function of time t [20].

Based on the model of a normal state, contact with the rubbing screw is enforced, which creates friction between the rotor and the screw. The friction can be modeled by modifying the normal state model and can be expressed as:

$$M\ddot{x} + C\dot{x} + Kx = mr\omega^2\cos(\omega t + \delta) - F_N(\cos\theta - \mu\sin\theta) \tag{4.3}$$

$$M\ddot{y} + C\dot{y} + Ky = mr\omega^2\sin(\omega t + \delta) - F_N(\mu\cos\theta + \sin\theta) \tag{4.4}$$

where M, C, K, m, r, δ, ω, x, y have identical meaning as in equations (4.1) and (4.2), F_N indicates the normal force, μ is the friction coefficient, and θ is the direction angle of the rubbing part [20].

Similar physical characteristic can be observed in the basic misalignment model. An extra external force is applied in the direction of the misalignment, independent of the rotor speed. This non-harmonic force creates nonlinear equations and can be described as:

$$M\ddot{x} + C\dot{x} + Kx + K_x x^2 = mr\omega^2\cos(\omega t + \delta) + P\cos\gamma \tag{4.5}$$

$$M\ddot{y} + C\dot{y} + Ky + K_y y^2 = mr\omega^2\sin(\omega t + \delta) + P\sin\gamma \tag{4.6}$$

where M, C, K, m, r, δ, ω, x, y have identical meaning as in equations (4.1) and (4.2), K_x and K_y are nonlinear stiffness coefficients, P is a radial force due to a misaligned rotor, and γ is a misaligned direction angle [20]. Using the example system data from [80], the responses of normal, rubbing, and misalignment states are obtained as

shown in Figure 4-3, Figure 4-4, and Figure 4-5.

The responses of the normal rotor model, x and y in Figure 4-3, display the same vibration signals with a $\pi/2$ phase difference, which indicates that the vibration signals are not direction-oriented. The responses of the rubbing model in Figure 4-4 are affected by the direction of rubbing (θ). The misalignment case in Figure 4-5 shows different behavior upon the direction of misalignment (γ) as well.

The above-stated mathematical models indicate that the external forces of rubbing and misalignment have directional nature, θ and γ. Thus, the responses of the two health states depend upon the direction of external forces. Overall, the directional nature in some health states hinders one from correct diagnosis using two gap signals; this leads to the significance of the ODR signals for robust diagnosis of fluid-film bearing rotor systems.

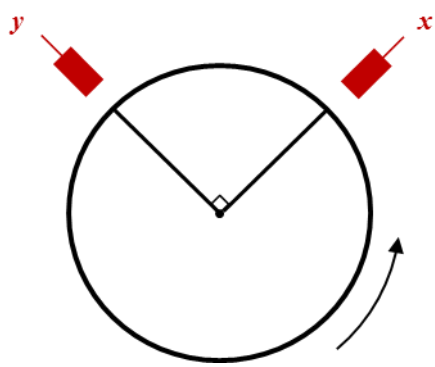


Figure 4-1 Configuration of gap sensor signals for a fluid-film bearing rotor system.

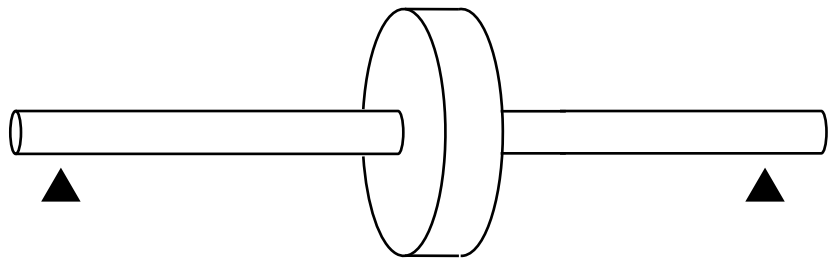


Figure 4-2 A simple analytical model for validation of directionality for some anomaly health states.

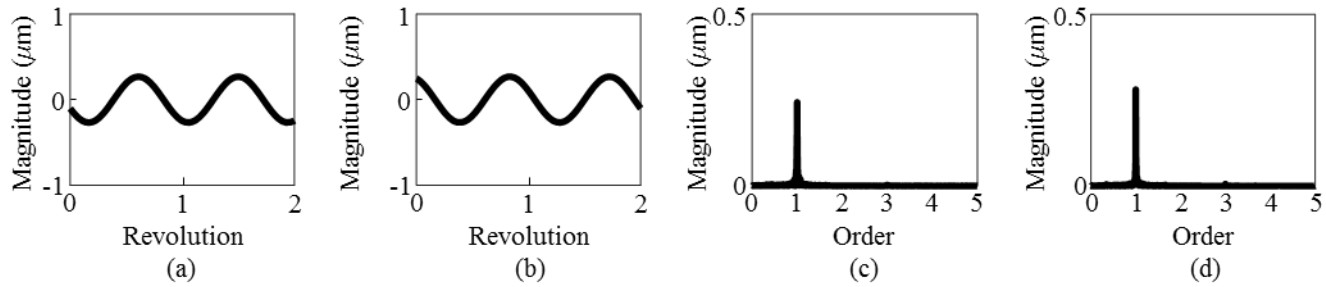


Figure 4-3 Time plots of signal (a) *x* and (b) *y*, and frequency response plots of signal (c) *x* and (d) *y* from response of a normal-state model.

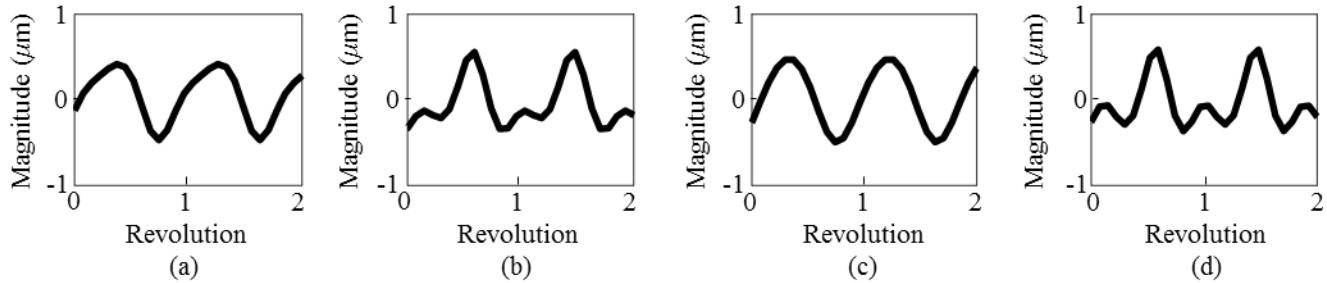


Figure 4-4 Time plots of signal (a) *x* and (b) *y* at θ=0°, and (c) *x* and (d) *y* at θ=45° from response of rubbing-state model.

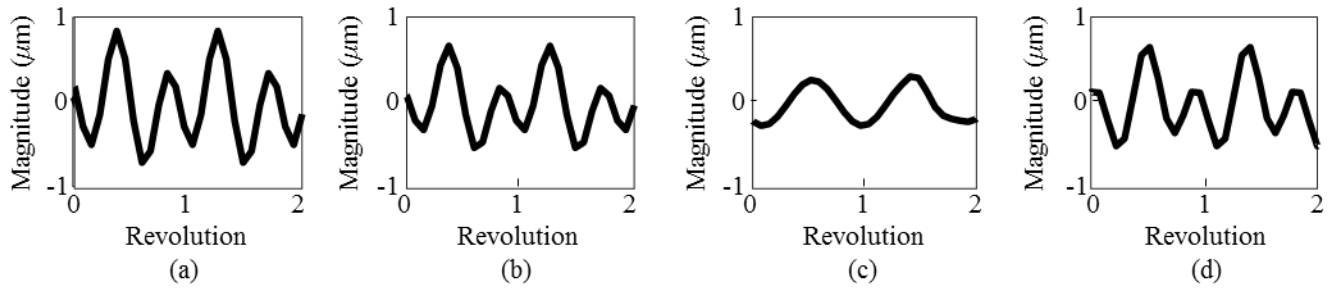


Figure 4-5 Time plots of signal (a) *x* and (b) *y* at γ =0°, and (c) *x* and (d) *y* at γ =45° from response of misalignment-state model.

The directionality of rubbing and misalignment states can also be validated by experimental data. The two health states were tested on the RK4 rotor kit testbed, as presented in Figure 3-1. Two shafts of 10 mm diameter are supported by three plain sleeve fluid-film bearings. The longer shaft is supported by two bearings with an 800-gram disc attached in the middle of the shaft. For the normal state, the amplitude of the vibration signals was set at a 10 μm root-mean squares (rms) level, which was determined from ISO 7919-2 [81]. The impact rubbing state was obtained by placing a rubbing screw into the normal state of the rotor generating harmonic external force. To test the misalignment state on the testbed, a customized jig was attached to shift the long shaft horizontally or vertically. A detailed description of the testbed configuration is provided in [12, 79].

The time plots and frequency response plots of vibration signals are shown in Figure 4-7 to Figure 4-12. The vibration signals are acquired through sensors placed in four different directions, as presented in Figure 4-6. Signals from x_0, $x_{N/4}$, $x_{N/2}$ and $x_{3N/4}$ sensors are presented in sub-figures (a), (b), (c), and (d), respectively. The normal state in Figure 4-7 shows a slight difference between the signals with a phase shift. The FFT plots in Figure 4-8 also show that the frequency responses are similar to each other, with 1x of rotating frequency (60Hz) being dominant. In contrast, the time plots of the rubbing state in Figure 4-9 indicate that the magnitude and the shape of the signals change greatly. This is also reflected in the frequency response in Figure 4-10, where 1x magnitude varies significantly with respect to the sensors. Other harmonic components change as well, which clearly indicates the directional dependency. Likewise, the misalignment signals also show a large variance over the directions, as presented in Figure 4-11 and Figure 4-12.

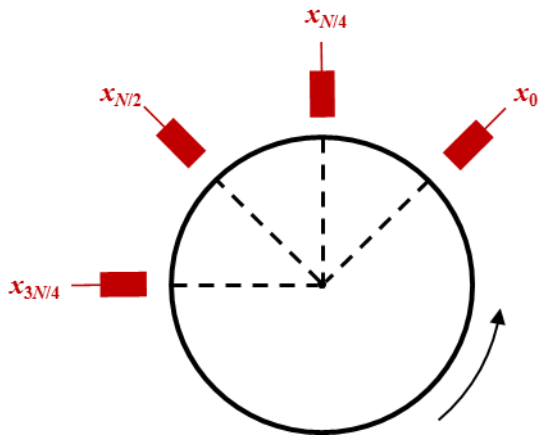


Figure 4-6 Sensor directions for vibration data acquisition.

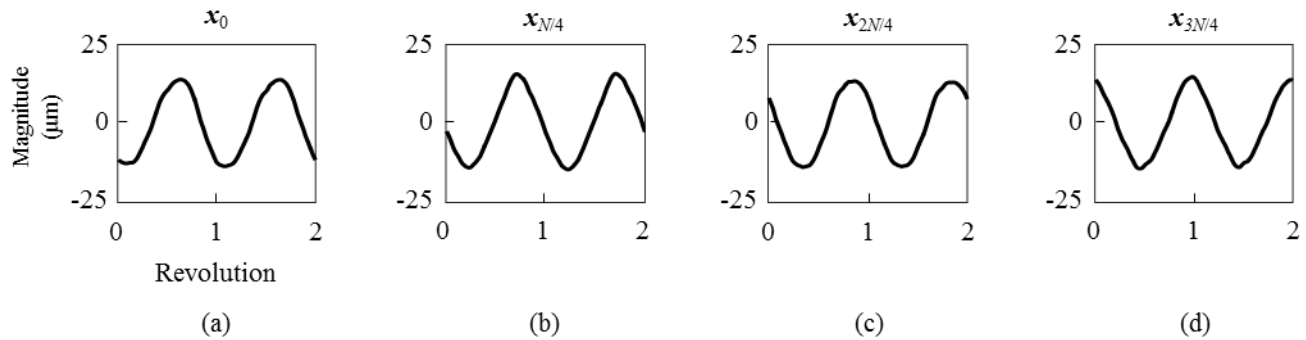


Figure 4-7 Time plots of testbed signal (a) x_0, (b) $x_{N/4}$, (c) $x_{N/2}$, and (d) $x_{3N/4}$ of a normal state.

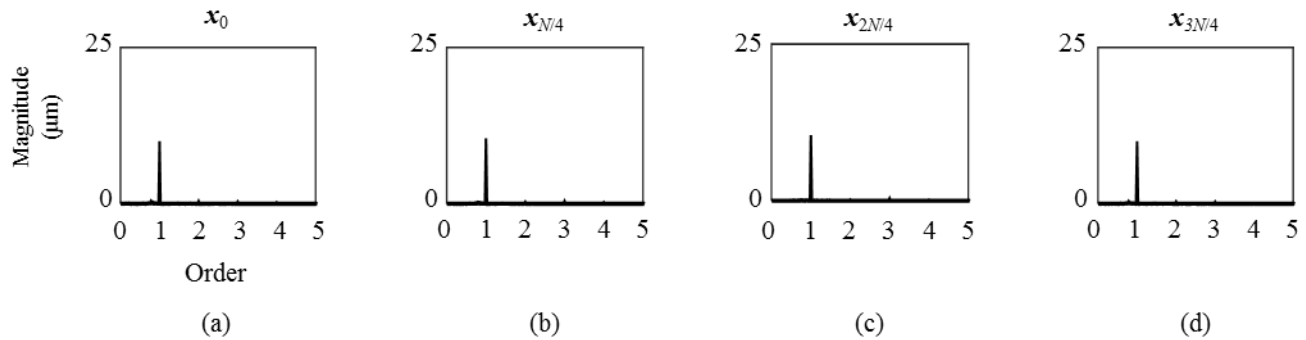


Figure 4-8 Frequency response plots of testbed signal (a) x_0, (b) $x_{N/4}$, (c) $x_{N/2}$, and (d) $x_{3N/4}$ of a normal state.

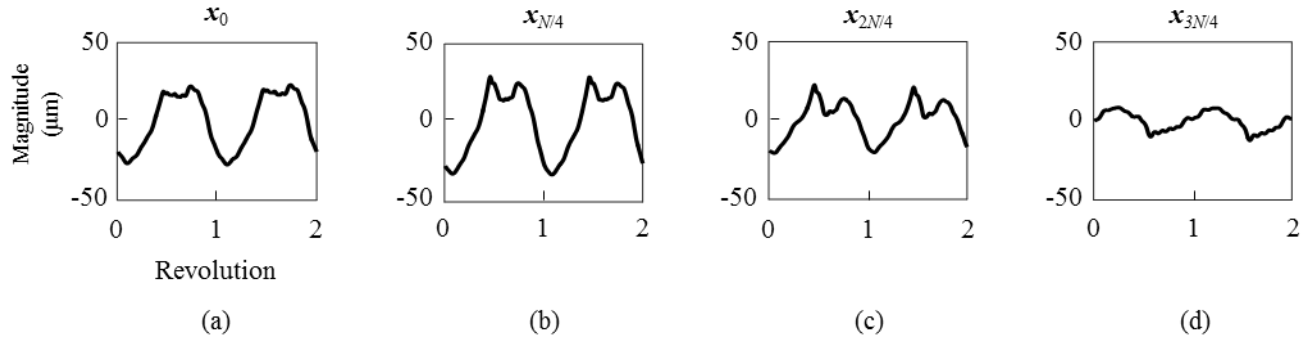


Figure 4-9 Time plots of testbed signal (a) x_0, (b) $x_{N/4}$, (c) $x_{N/2}$, and (d) $x_{3N/4}$ of a rubbing state.

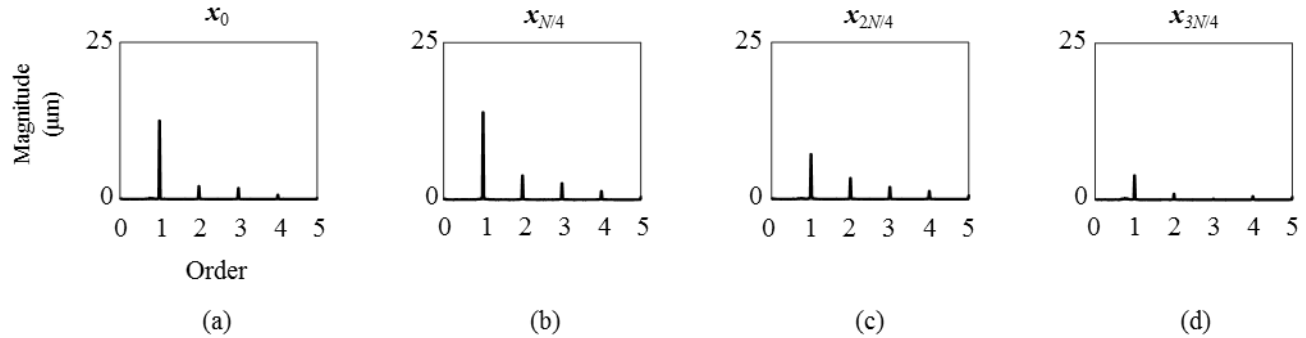


Figure 4-10 Frequency response plots of testbed signal (a) x_0, (b) $x_{N/4}$, (c) $x_{N/2}$, and (d) $x_{3N/4}$ of a rubbing state.

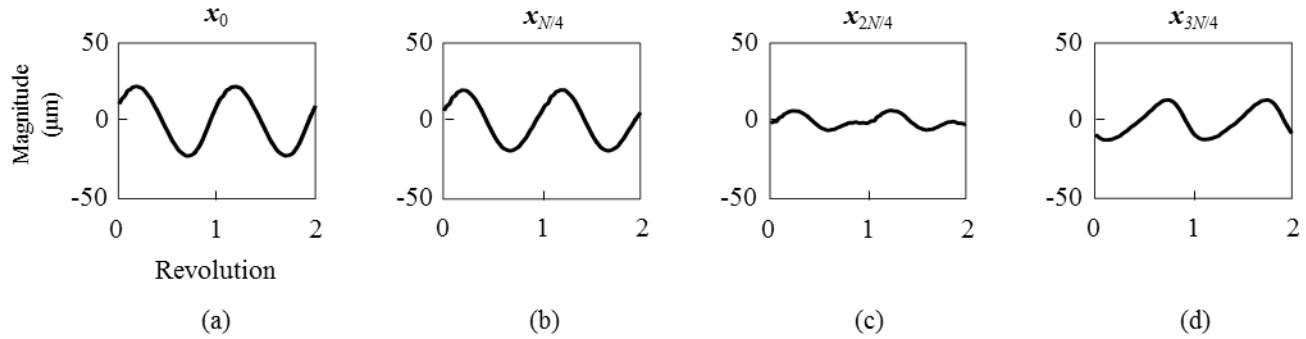


Figure 4-11 Time plots of testbed signal (a) x_0, (b) $x_{N/4}$, (c) $x_{N/2}$, and (d) $x_{3N/4}$ of a misalignment state.

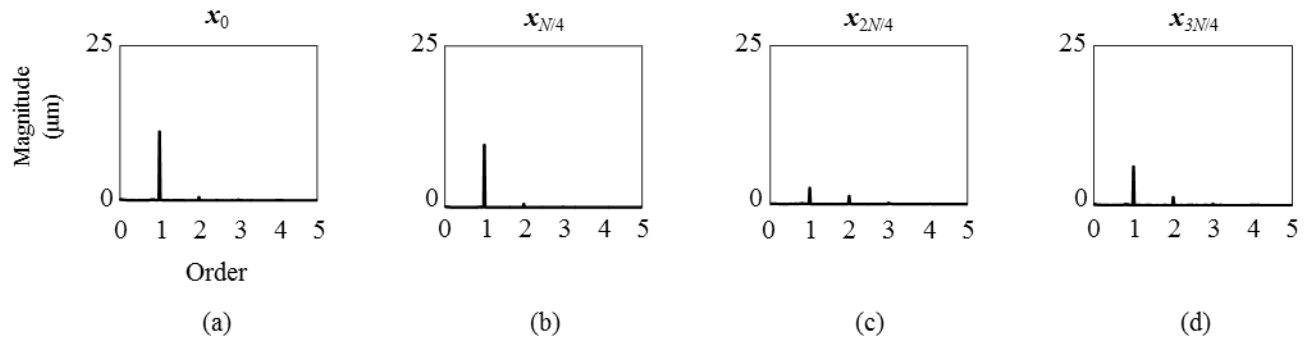


Figure 4-12 Frequency response plots of testbed signal (a) x_0, (b) $x_{N/4}$, (c) $x_{N/2}$, and (d) $x_{3N/4}$ of a misalignment state.

4.1.2 Omnidirectional Regeneration of Vibration Signals

In fluid-film bearing rotor systems, two perpendicular fixed sensors are implemented, as presented in Figure 4-1. However, the two acquired signals from the two fixed sensors may not accurately represent the state of the rotor system because an anomaly can happen in a direction that is not line with the sensors. The directional dependency of a few health states has been confirmed in Section 4.1.1. To obtain reliable and accurate diagnosis results, vibration signals from other directions should also be acquired. However, adding extra sensors requires the structure of the system to be modified, and may not always be possible. Thus, the omnidirectional regeneration (ODR) technique has been developed [12, 79].

The ODR technique can be referred to as placing virtual sensors in any direction and acquiring signals from those virtual sensors. The ODR signals from the i^{th} virtual sensors, x_i and y_i, are defined (i=1, 2, …, N) as:

$$x_i = [\cos(i \times \Delta\theta)]x_0 - [\sin(i \times \Delta\theta)]y_0 \tag{4.7}$$

$$y_i = [\sin(i \times \Delta\theta)]x_0 + [\cos(i \times \Delta\theta)]y_0 \tag{4.8}$$

where x_0 and y_0 are the vectors that consist of vibration signals from actual x- and y-axis proximity sensors, respectively; N is the maximum number of the ODR signals to be generated; and $\Delta\theta$ is the increment of the rotation angle.

The ODR technique can produce vibration signals from any arbitrary direction in addition to the acquired vibration signals (x_0 and y_0). It is worth noting that (1) $\Delta\theta$ should be determined by considering the tradeoff between computing power and diagnostic robustness, (2) ODR signals are within the range of a π circumference angle due to the characteristic of symmetry, and (3) x_i covers all y_i if the ODR covers

more than half of a circumference.

The virtual sensors can be placed at any orientation without any physical change to the structure of the system. These virtual sensors can then be used to obtain omnidirectional regeneration (ODR) signals. Based on the ODR signals, a vibration image can be generated; this is described in next section.

4.1.3 Vibration Image Generation

As stated in Section 4.1.1, ODR signals around the circumference of the rotor are required to diagnose directional health states, such as rubbing and misalignment. Based on these ODR signals, vibration images can be generated that contain spatial information as well as temporal information [2, 79]. A few cycles of single channel vibration signal represent temporal information of the rotor, while ODR signals represent directional information that is the spatial information of the rotor. Thus, stacking multiple ODR signals of few cycles can generate a vibration image containing both temporal and spatial information, as shown in Figure 4-14. The vibration images used in this dissertation modified the process used in [2, 79], as shown in Figure 4-13.

The process of generating a vibration image involves three steps, as presented in Figure 4-16. First, the ODR signals are generated around the circumference of the rotor using the signals from the two proximity sensors. The ODR signals can be regarded as vibration signals from virtual sensors. Then, the ODR signals of two revolutions are stacked vertically. The two rotor revolutions are defined due to the characteristics of the sub-harmonic dominant health state, i.e. oil whirl state. Finally,

each pixel value of the image is normalized using the biggest pixel value, which is equivalent to the biggest peak value among the ODR signals. The final step of normalization generates the grayscale image, which shows white if the pixel value is 1 and black if pixel value is 0. In addition, the size of the images were changed to 28-by-28 pixels. Although it is preferable to use images without size conversion, the patterns are maintained even after the size is reduced. Therefore, the reduced images were used for the work described in this dissertation to accelerate the computation, while maintaining the classification performance.. Examples of standard and modified vibration images are shown in Figure 4-15 (a) and (b), respectively.

A few features have been modified from the original vibration image generation steps because a convolutional neural network was used in this dissertation research. The basic convolution of filters in CNN reduces the second and third steps that find the reference ODR signals and the synchronization of the phase. The pattern of white and black gradients can be captured independent of the location in the image. Thus, the two steps are not required when using CNN. In addition, to make the patterns appear as a whole, the rotation angle for the ODR signal generation has been increased from π to 2π. The white and black gradient patterns in the standard vibration images can be disconnected if the π angle of the ODR signals is used.

The generated vibration images of four health states – normal, rubbing, misalignment, and oil whirl – are shown in Figure 4-17. The normal images present a diagonal stripe pattern, which indicates the ODR signals have 1x of rotor speed frequency as a main component. However, other images present complex patterns. Specifically, the rubbing and misalignment images have different shapes among each ODR vibration signal due to the directionality. In addition, the vibration image of

the oil whirl state indicates a weak 1x frequency pattern with an overlapped region between the white stripes.

The resized vibration images in Figure 4-17 can be distinguishable by human vision recognition. The CNN algorithm can also classify the images by the correct labels, if the features (or patterns) of images can be recognized properly. Thus, a proper filter size, which convolutes with images with respect to the size of images, must be defined to obtain great performance.

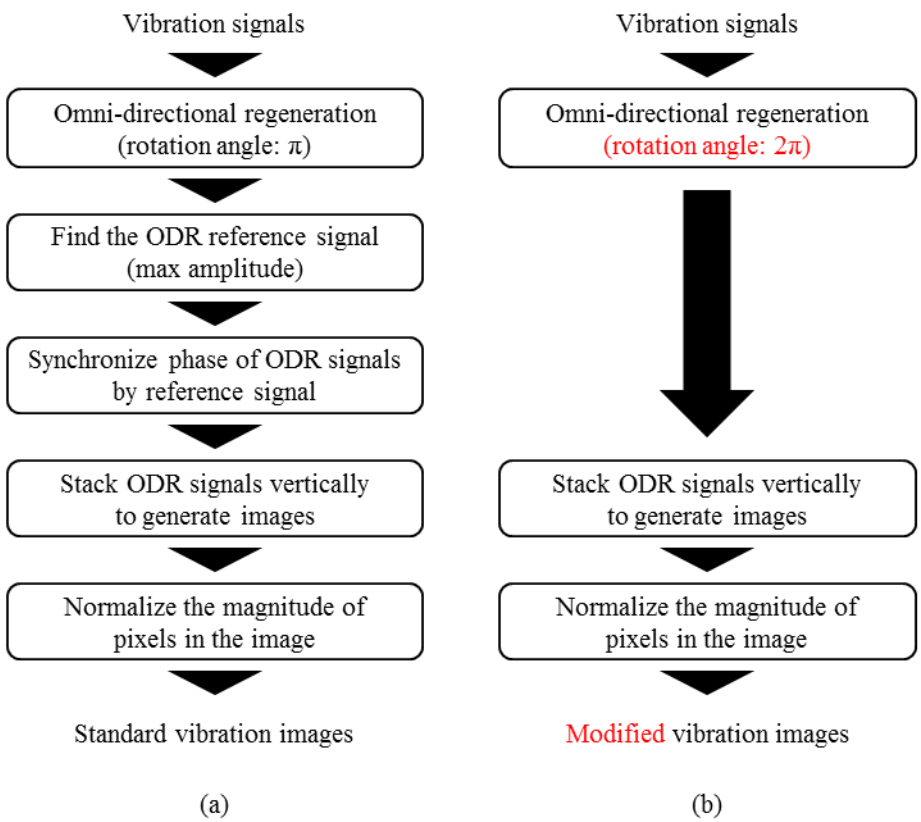


Figure 4-13 Vibration image generation steps: (a) standard vibration images and (b)

modified vibration images.

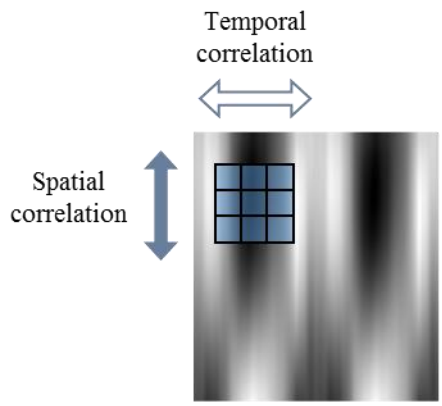
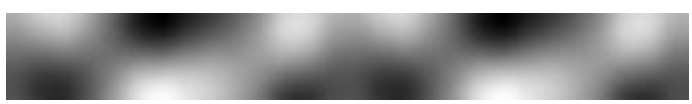


Figure 4-14 Temporal and spatial information direction of a vibration image.



(a)

(b)

Figure 4-15 Example of (a) a standard vibration image and (b) a modified vibration image.

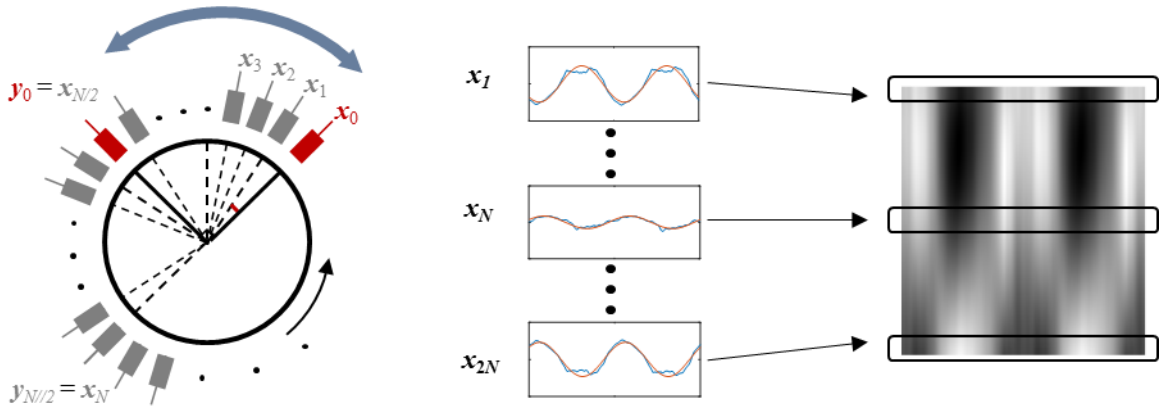


Figure 4-16 Process of vibration image generation.

(a) (b) (c) (d)

Figure 4-17 Examples of vibration image: (a) normal, (b) rubbing, (c) misalignment, and (d) oil whirl.

4.2 Determining the Optimal Ratio Between Vibration Image Size and Filter Size

The size of a standard vibration image is defined by the resampling rate and the number of ODR signals. The width of a standard vibration image is the double of the resampling rate as the two revolutions form an image. The height of a standard vibration image is the number of ODR signals around the full circumference of the rotor. Thus, the resampling rate of 128 samples per revolution and the 32 ODR signals make a 32-by-256 pixel sized image, which adds up to 8,192 pixels. Although the standard image prevents losing information when the image is resized, the original size is too large to be processed using a typical graphics processing unit (GPU)[1] that is used for deep learning on a desktop.

The effect of the modified vibration signals allows the data being trained and tested by a GPU, which is considerably faster than using the central processing unit (CPU) only. Also, the step of phase sync is neglected, which gives much more available data for deep learning based diagnosis. Obviously, the performance was maintained even with the modified images. Thus, the modified vibration images are used throughout the research described in this dissertation. The modified images will be simply called 'vibration images' in the rest of the sections.

In fact, the optimal size of the vibration image with respect to the filter size should

[1] NVIDIA GeForce GTX TITAN X, NVIDIA TITAN Xp, NVIDIA GeForce GTX 1080 Ti

be determined. The ratio of the two sizes can be regarded as one of the hyper-parameters for the convolutional neural network (CNN) that needs to be tuned based on the vibration images. As too large an image size with respect to the filter size may not capture any meaningful features, the size of the vibration image may have an upper bound that maintains a certain performance level. There may be a lower limit as well. Thus, this section studies the optimal size of images with a fixed filter size.

The prediction accuracies are obtained by using *k*-fold validation of five different sets [82, 83]. Each set includes four different health states: normal, rubbing, misalignment, and oil whirl.

4.2.1 Vibration Image Size with Respect to Filter Size

The ratio of vibration image size and the CNN filter size should be determined. As this dissertation focuses on preprocessing, the CNN filter size was fixed as 5-by-5 pixels and the vibration image size was altered. The ratio can be defined as the number of the width or height of pixels, which considers only square-sized images and filters. The ratio can be presented as:

$$\text{Image to filter ratio} = \frac{\text{width or height of an input image}}{\text{width or height of a CNN filter}}$$

For example, if the vibration image is set as 25-by-25, the ratio is 25/5 = 5.

The minimum ratio is 1, which indicates an image of the same size as the filter. The maximum ratio is not limited, as the size of the image can be increased infinitely when the filter size is fixed. The vibration images are resized from the original size

of 128-by-256 using a bi-cubic interpolation method [84]. An example of an original size image is shown in Figure 4-18. The resized images are shown in Figure 4-19 (a) to (d). If the filter size is 5-by-5, each sub-figure has a ratio of 1, 5, 10, 20, respectively. The gradient patterns are preserved as resized, but the image of ratio 1 is hardly recognizable by the human eye. In addition, an image size of ratio 20 may not be recognizable because the gradient patterns may not appear clearly in the size of the filters. Thus, the optimal size of images with respect to the size of filters needs to be studied.

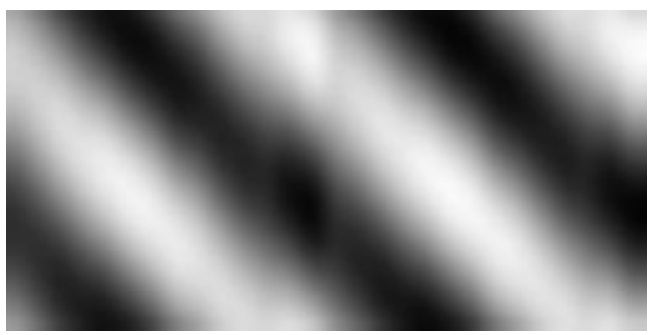


Figure 4-18 A vibration image of original size 128-by-256.

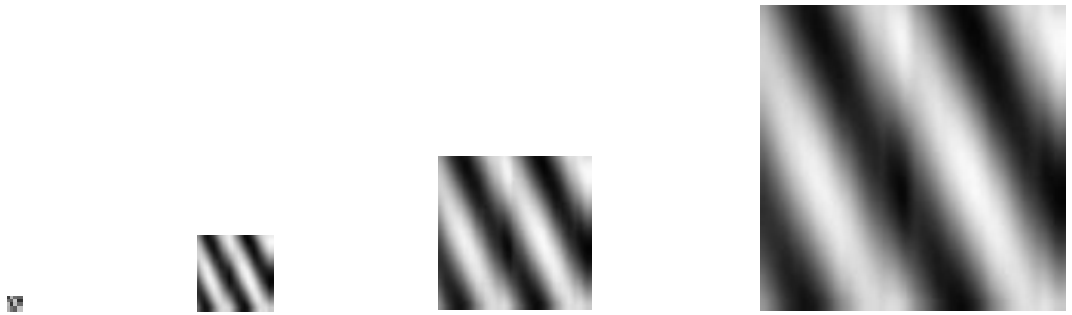


Figure 4-19 Resized vibration images of size (a) 5-by-5, (b) 25-by-25, (c) 50-by-50, and (d) 100-by-100.

4.2.2 Gradient of Vibration Image

In a convolutional neural network (CNN), filters that have a less size than the images convolute through the images [85]. Through sequences of convolutions between the filters and the images, representations of features are obtained as the layers go through. The best feature representations can be achieved by finding optimal hyper-parameters, which include the sizes of filters and images. However, there is no rule of thumb that decides the optimal ratio, since it varies according to the type of images and the structure of the CNN. Thus, this section suggests a quantified criteria for the vibration image for fluid-film bearing rotor systems.

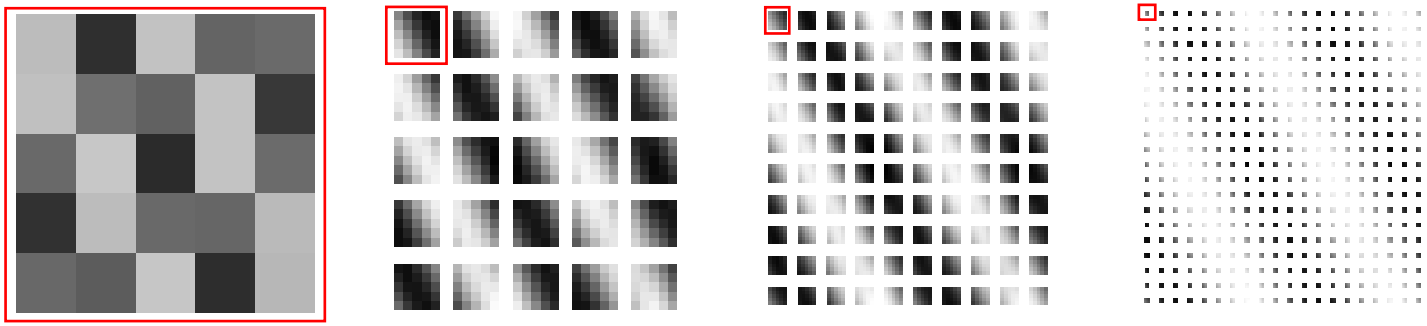
As stated above, the overall performance of a CNN is closely related to the filters' capability to capture the feature patterns. Figure 4-20 presents the segmented vibration images using a filter size of 5-by-5 with different image sizes. Each sub-figure is scaled to the same size. In Figure 4-21, the first segmented images in Figure 4-20 are displayed. The segmented figures indicate that the larger gradient can be seen for the image size of 25-by-25 in Figure 4-21 (b). Likewise, if the gradient patterns of each health state image can be clearly presented in the size of the filter, it may have a higher chance of showing better performance. In contrast, if the ratio is too big to represent gradient patterns in a filter, such as in Figure 4-20 (d), the training may not succeed in learning the representative features, which may result in lower performance. Thus, as the size ratio grows bigger, the number of segmented images with weaker gradients increases, which may decrease the performance for diagnosis of fluid-film bearing rotor systems.

To quantify the degree of gradient in segmented images, the image gradient is used. The gradient of segmented images can be used as a criteria for determining the

optimal ratio of the image size and the filter size. The image gradient of the segmented images can be calculated by the following steps, as presented in Figure 4-22. First, the images are segmented by the size of the filters, as presented in Figure 4-21. Then, the gradient within each segment can be derived using a gradient operator for image processing [86-89]. Between the magnitude and the direction of the image gradient, the magnitude component is averaged within the segment. The final step is to count the number of segments that has an averaged magnitude larger than a certain threshold. As the number of segments that have a strong gradient pattern increases, the overall performance may also increase because the segments present strong patterns.

The threshold value for separating a strong or weak gradient of segmented images should be defined by the user, considering the patterns of the images. For the gradient-based images, examples of various image gradient are shown in Figure 4-23. The figures presents average magnitude of 0.1, 0.2, 0.3, and 0.4 of 5-by-5 pixel segmented images, respectively. The gradients (c) and (d) are clearly visible, while (a) and (b) show weak gradients.

It should be noted that the suggested gradient criteria can be applied to the images of any gradient type. If the image size is too large compared to the size of the filter, the number of weakly gradient segmented images will increase, which eventually will decrease the performance. The number of strong gradient segmented images will be compared to the performance in the next section.



(a) (b) (c) (d)

Figure 4-20 Vibration images segmented by the 5-by-5 size of (a) 5-by-5, (b) 25-by-25, (c) 50-by-50, and (d) 100-by-100 sized images.

(a) (b) (c) (d)

Figure 4-21 Segmented images of (a) 5-by-5, (b) 25-by-25, (c) 50-by-50, and (d) 100-by-100 sized images in Figure 4-20.

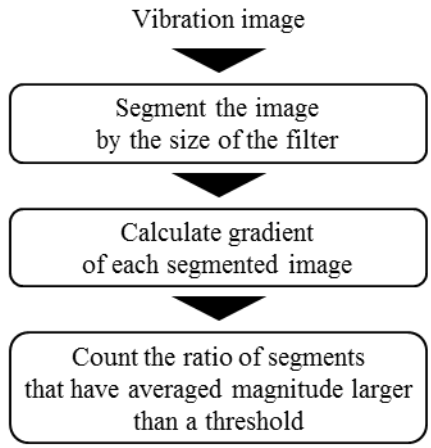
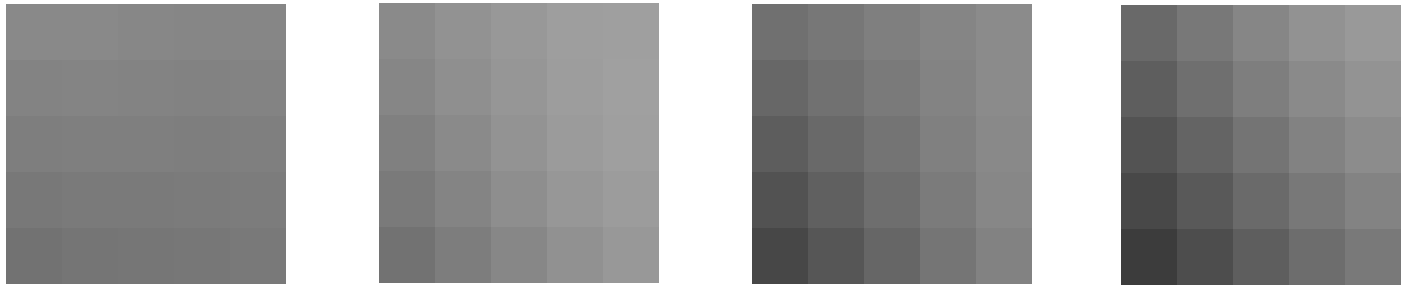


Figure 4-22 Steps to determine the criteria for a strong gradient pattern.



(a) (b) (c) (d)

Figure 4-23 Example of segmented images of gradient magnitude (a) 0.1, (b) 0.2, (c) 0.3, and (d) 0.4.

4.2.3 Study of the Optimal Ratio of Vibration Image Size to Filter Size

The ratio of the image size to the filter size (the size ratio), which is a hyper-parameter of convolutional neural network (CNN), clearly affects the overall performance. As the size ratio increases, the gradients of the vibration images are unlikely to be stored in the size of the filters. Thus, the image gradient of each segment is suggested to provide a reference for finding the optimal size ratio. The threshold for a strong gradient of the segmented images has been set as the averaged gradient magnitude (AGM) greater than a threshold (α). Note that the magnitude of the image gradient is used in this study, while the histogram of oriented gradients (HOG) uses the direction component of the image gradients [90].

To see how many segments with AGM of larger than a certain threshold (α) take up from all segments, the following ratio is defined as follows:

$$\text{RSG}= \frac{\text{Number of segmented images (AGM larger than } \alpha)}{\text{Total number of segmented images}}$$

$$\left(\alpha = \frac{1}{\text{filter size}}\right)$$

where RSG stands for the ratio of strong gradient segments and α is the user-defined threshold. For the vibration images, α is defined as 1 over the filter size, which gives 0.2 for 5-by-5 sized filters. If the filter size is increased to 10-by-10, then α would be reduced to 0.1. This inversely proportional relationship is due to the decrease in gradient magnitude as the images are stretched. Based on this relation, RSG is defined by ratio of the number of segmented images that have gradients larger than α to the total number of segmented images.

The RSG values of each size ratio are shown in Figure 4-24 and Figure 4-25.

Figure 4-24 presents the RSG of 5-by-5 segmented images, and Figure 4-25 presents the RSG of 10-by-10 segmented images. Each sub-figure corresponds to the normal, rubbing, misalignment, and oil whirl vibration images from the fluid-film bearing rotor system. A figure presents the RSG values with respect to the ratio of image size to filter size (size ratio) for the vibration images of a health state. Note that the RSG starts to decrease at the size ratio of 5 for the oil whirl state vibration image in Figure 4-24(d), whereas, in the normal case it starts to drop at 7. The difference is due to the distinct patterns of each health state. As shown in Figure 4-17 (a) and (d), the oil whirl image has less gradient pattern than the normal image. This difference can be enlarged as the size ratio increases.

The averages of the RSG for all vibration image data sets are shown in Figure 4-26. As expected, the RSG value decreases as the size ratio increases. In other words, if the image size gets considerably bigger than the filter size, the segmented images show weaker gradients. In addition, the 10-by-10 segmented image cases show a similar trend to that of the 5-by-5 cases.

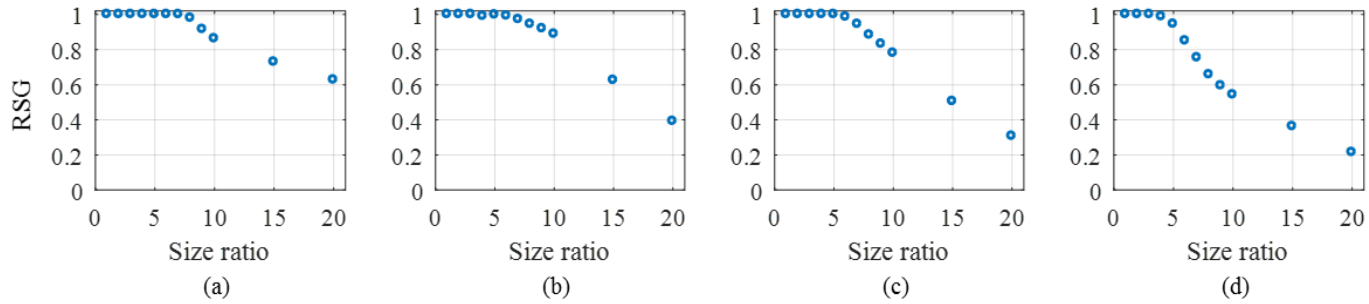


Figure 4-24 Average of the RSG values for 5-by-5 segmented image gradients of set 1: (a) normal, (b) rubbing, (c) misalignment, and (d) rubbing.

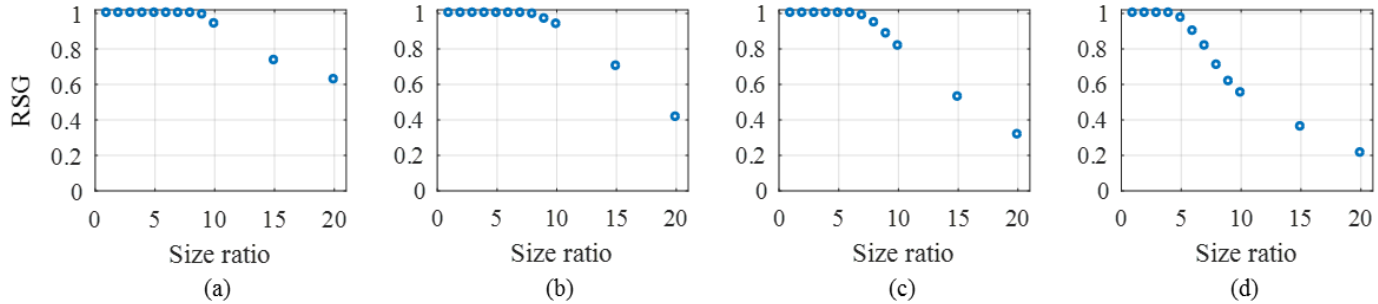
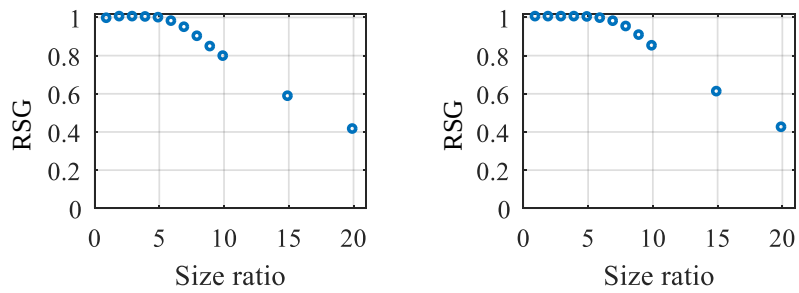


Figure 4-25 Average of the RSG values for 10-by-10 segmented image gradients of set 1: (a) normal, (b) rubbing, (c) misalignment, and (d) rubbing.



(a) (b)

Figure 4-26 Average of the RSG values for all health states: (a) 5-by-5 and (b) 10-by-10 segmented images.

Prediction accuracies of various size ratios by *k*-fold validation are presented in Figure 4-27 and Figure 4-28 using 5-by-5 and 10-by-10 sized filters, respectively. All combinations of three sets out of five sets were trained and evaluated by the other two sets, which gives 10 combinations. The last 10 epochs of all 10 combinations were averaged to see the overall performance. In addition, four other CNN structures were used to get the generalized results, which are shown as four sub-figures in Figure 4-27 and Figure 4-28. Specifically, cases (a) and (b) show the result of 1-layer CNN with a mini-batch size of 12 and 100, respectively. Similarly, cases (c) and (d) represent the results of 2-layer CNN with a mini-batch size of 12 and 100, respectively.

First, when the filter size was fixed at 5-by-5, four different CNN structure cases maintained a prediction accuracy higher than 0.95 until the size ratio reached 10, which denotes an image size of 50-by-50. In detail, size ratios from 6 to 10 showed the best results: greater than 0.97. In contrast, the prediction accuracy at the size

ratios of 15 and 20 presented lower accuracies. The two cases had the lowest and the second-lowest RSG values, as shown in Figure 4-26, which implies difficulty in learning the features of vibration image patterns. Furthermore, the filter size 10-by-10 showed similar results as well, which is given in Figure 4-28. The optimal range of the size ratio lies between 5 to 8, which is a little less than the 5-by-5 size filter case.

The results of the two filter size cases denote that size ratios of 10 or lower give prediction accuracy greater than 0.9. A size ratio greater than 10 presents lower accuracy, as the filters are not big enough to recognize the pattern of the vibration images. The decrease of the RSG values infers a decrease of the overall performance. Hence, optimal performance for diagnosing the health state of a fluid-film bearing rotor system can be achieved by using a size ratio that has an RSG value greater than 0.8. Note that this value will vary according to various parameters, such as the number of classes and the type of images.

In the rest of this dissertation, an image size of 28-by-28 pixels and a filter size of 10-by-10 pixels are used; this gives a size ratio of 2.8. The size ratio of 2.8 resides in the optimal range that was derived in this thrust. The image size of 28-by-28 pixels was chosen from the referenced MNIST data set sizes [77]. The larger filter size, as compared to the widely used CNN method, was selected to analyze the effect of the filter analysis in the next research thrust. Such sizes were fixed to reduce the number of hyper-parameter combinations.

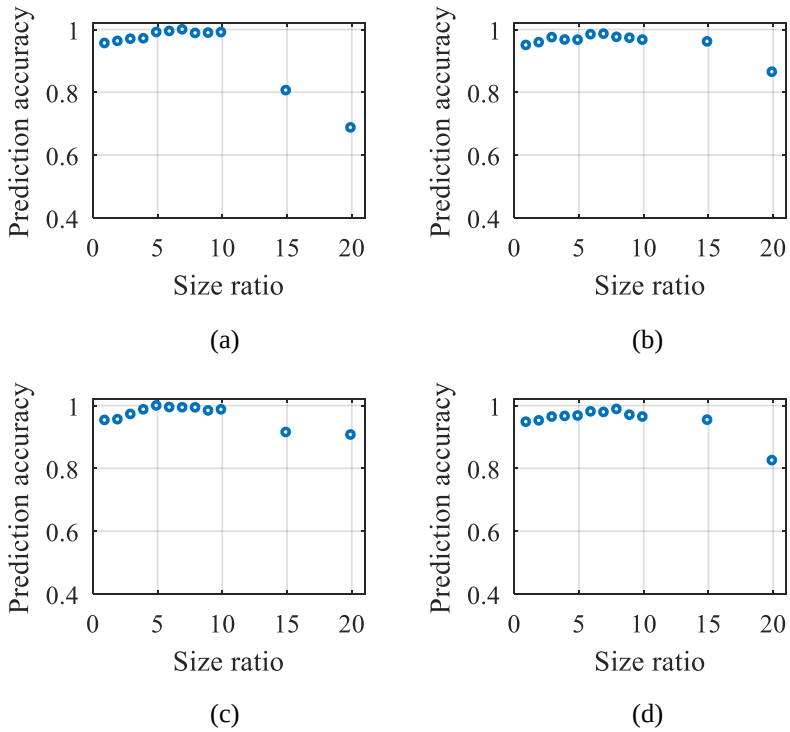


Figure 4-27 Prediction accuracy of a 5-by-5 filter size with respect to the size ratio of four different CNN structures: (a) 1-layer, batch size 12, (b) 1-layer, batch size 100, (c) 2-layer, batch size 12, and (d) 2-layer, batch size 100.

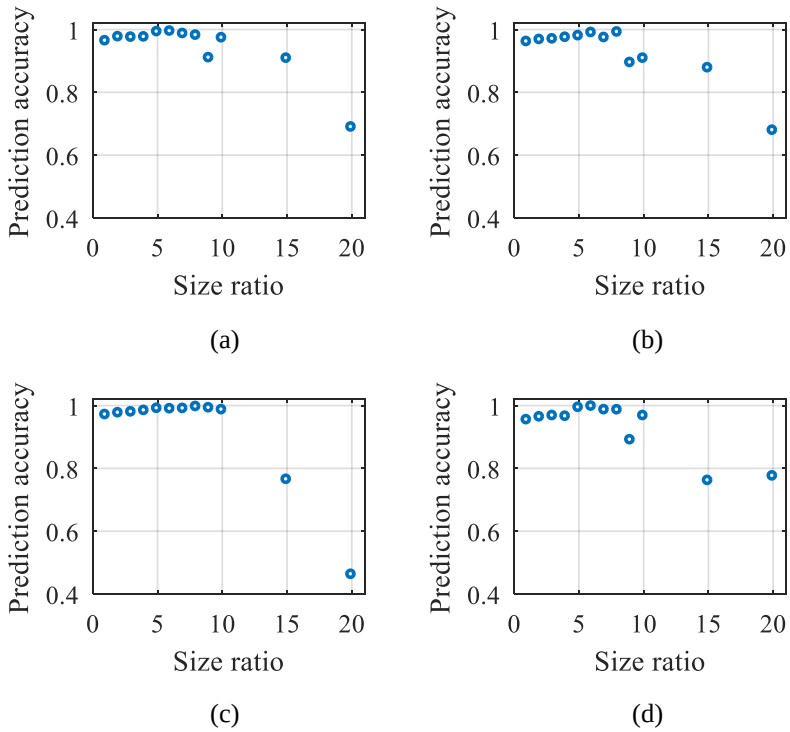


Figure 4-28 Prediction accuracy of a 10-by-10 filter size with respect to the size ratio of four different CNN structures: (a) 1-layer, batch size 12, (b) 1-layer, batch size 100, (c) 2-layer, batch size 12, and (d) 2-layer, batch size 100.

Sections of this chapter have been published as the following journal articles:

1) **Joon Ha Jung**, Byung Chul Jeon, Byeng D. Youn, Myungyon Kim, Donghwan Kim, and Yeonwhan Kim, "Omnidirectional Regeneration (ODR) of Proximity Sensor Signals for Robust Diagnosis of Journal Bearing Systems," *Mechanical Systems and Signal Processing*, Vol. 90, pp. 189-207, 2017.

Chapter 5

Label-based, Mini-batch Combinations Study by Filter Sensitivity Analysis

The convolutional neural network (CNN), one widely used deep learning algorithm, was proposed more than a decade ago [91]; however, CNNs have been used in only limited applications due to a lack of computational power and the immaturity of big data management strategies. Recently, the use of graphic processing units (GPU) in computation, and advances in big data analytics, have facilitated the use of CNNs [9]. CNNs are currently used in various types of applications. Correspondingly, the number of CNN-based fault diagnosis studies has increased significantly in the past few years [39-41, 92-94]. Most of the studies focus primarily on feature extraction [40-42, 95-97], which affects the overall performance of the data-driven approach. Since the deep structures of the network can learn representations from a large and complex dataset without human intervention, these deep networks are used to extract health features of engineered systems, a process that usually requires a considerable amount of effort and time. The health features extracted via CNNs have achieved robust diagnosis results, as compared to conventional data-driven approaches [40-

42, 96]. To extract reliable health features from such big data, several techniques that accelerate the learning process are required.

Most CNN-based fault diagnosis studies have adopted the random mini-batch gradient descent optimization method to accelerate the training process and to obtain a more stable solution [98-100]. Since each mini-batch is sampled randomly for every iteration, the variance of mini-batches is generated, which leads to the variance of the gradient information. The large variance between mini-batches normally slows down the learning process of the network; thus, computer science researchers have developed several techniques to reduce the variance. In [101, 102], sample sizes were dynamically changed during the training to improve the convergence of optimization. Other researches proposed a sampling technique for formulating mini-batches to reduce the variance [103, 104]. However, previous studies have not considered label information when generating mini-batches. Since fault diagnosis of mechanical engineering systems considers label information of data as a critical factor, label information in mini-batches may have a significant effect on overall performance. For example, sequential ordering of a biased-label mini-batch may force the network to learn a certain health state of a label.

Thus, to validate the effect of labels in generating mini-batches for CNN-based fault diagnosis, we propose here a label-based, mini-batch gradient descent method. The proposed method uses label information when sampling mini-batches from the training data set; this is different from the process used for conventional random sampling. In our method, mini-batch compositions are formulated based on the label information, which indicates the health states of the mechanical system. Since the label portions of each mini-batch can be different from each other, the order in which

the mini-batches are used to update the gradient information also affects the performance. As a result, some label-based compositions and orders may present excellent performance, while others may show poor performance. The results are analyzed through the proposed filter sensitivity process, as well as through conventional performance measures. The label-based, mini-batch gradient descent method is validated using experimental data from a fluid-film bearing rotor testbed using a CNN.

This chapter is organized as follows. In Section 5.1, the concepts of CNNs and mini-batch gradient descent are explained briefly. Section 5.2 describes the proposed label-based, mini-batch gradient descent method. Section 5.3 presents the description of the case study data used in this paper. The performance of the proposed method is analyzed in Section 0, including the filter sensitivity analysis.

5.1 Mini-batch Gradient Descent in Convolutional Neural Network

This section briefly describes the conventional convolutional neural network (CNN). Generally, CNNs are updated by the gradient descent method from mini-batches.

5.1.1 Overview of Convolutional Neural Networks (CNN)

In this study, two layers of CNNs are used; 1-layer and 2-layer CNN. The basic structure of the two CNNs are shown in Figure 5-1 and Figure 5-2, respectively. In the figures, each convolutional layer has 16 kernels (filters), which are called weights

in a neural network, generally [105]. As stated in Section 2.2, the filters are convoluted through the input image, which can be viewed as weight sharing. This significantly reduces the number of parameters to train, while emphasizing the local connectivity within the images [9]. The hyper-parameters of the two example CNNs are presented in Table 5-1. Two fully connected (FC) layers were attached at the end of the network. The size of the filter was set as 10-by-10 because the images showed large sized patterns. The dimension of a filter for the first and the second convolutional layers are two and three, respectively, as the first convolutional layer adds a dimension to the input feature map.

In addition, the rectified linear unit (ReLU) method was used for the activation function at each layer and a max-pooling function was applied to the output of the convolutional layers. The size of the image was preserved by using zero-padding at the convolutional layers; however, the image size was reduced by half at the pooling layers. Thus, the number of weights that connect the convolutional layer and the first FC layer is different for the two CNN structures. For example, if the 16 filters are used in each convolutional layer, the number of the parameters are 3,136 and 784, respectively. An adaptive moment estimation (Adam) algorithm was used for training the filters with a learning rate of 0.001 [106] and with the size of the mini-batch fixed to 12. To avoid the network being overfitted to the training data set, a dropout ratio of 0.3 was used for each layer.

Table 5-1 Example of hyper-parameters of the convolutional neural network.

Number of conv. layers	1	2
Number of filters	Layer 1: 16	Layer 1: 16 Layer 2: 16
Batch size	12	12 100
Fully connected (FC) layer	FC layer 1: 16 nodes FC layer 2: 4 nodes	
Learning rate	0.001	
Activation function	ReLU	
Pooling function	Max pooling	
Dropout ratio	0.3	
Optimizer	Adam	

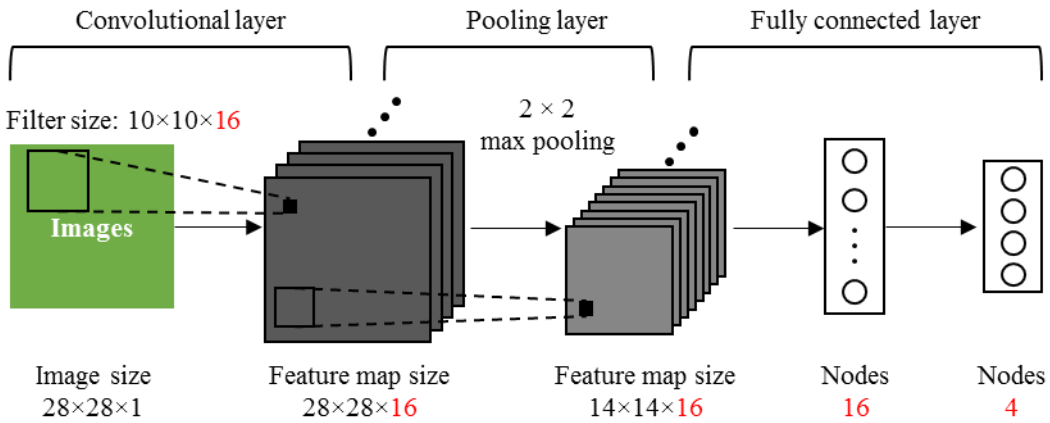


Figure 5-1 Structure of 1-layer convolutional neural network (CNN).

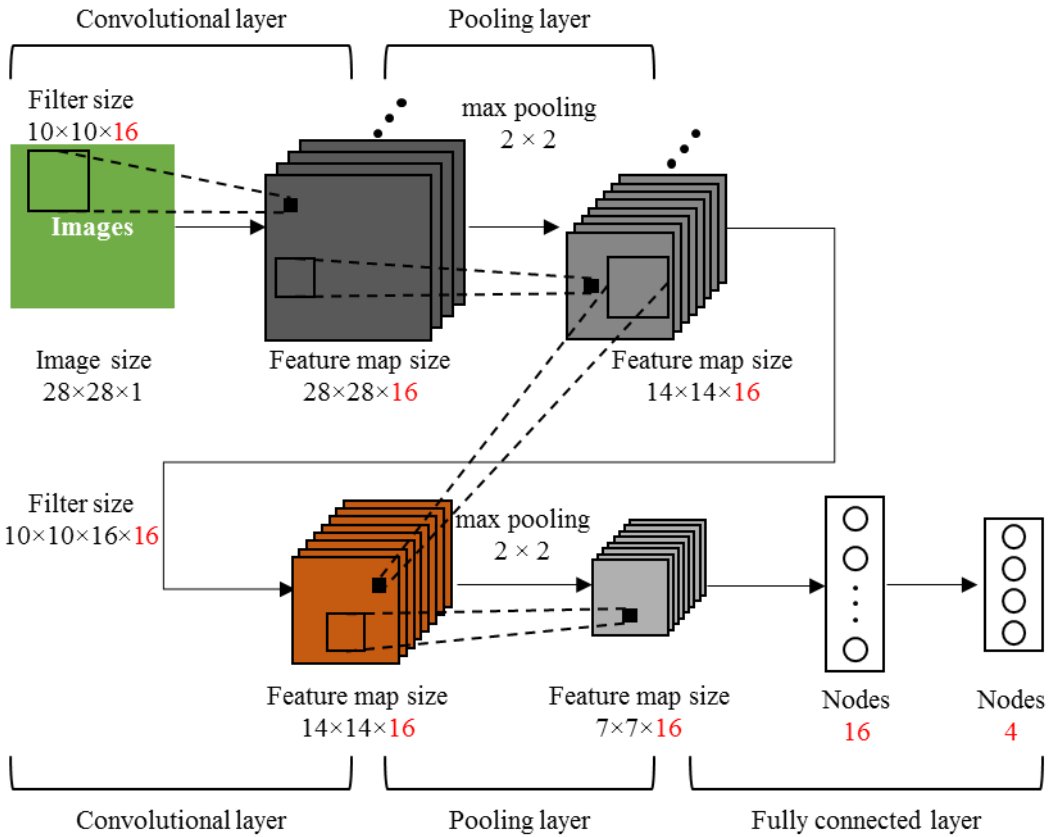


Figure 5-2 Structure of 2-layer convolutional neural network (CNN).

5.1.2 Mini-batch Gradient Descent

The mini-batch gradient descent (MGD) method is a technique that is widely used to accelerate the training process, while improving the generalization ability of the network. As is denoted by the name, the mini-batch method splits training data into subsets, of which cardinality is larger than one and smaller than the size of the training data. For every mini-batch, the average gradient is used to update the weight parameters of the network, which can be stated as:

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k - \frac{\eta}{|\boldsymbol{B}_k|} \sum_{i \in \boldsymbol{B}_k} \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_k; \boldsymbol{x}_i) \tag{5.1}$$

where $\boldsymbol{\theta}_k$ is the weight of the network at the k^{th} iteration, η is the learning rate, $\boldsymbol{B}_k$ is the k^{th} subset of the training data, J is the cost function for the optimization problem, and $\boldsymbol{x}_i$ is the i^{th} training data.

As stated in the previous paragraph, most deep learning algorithms adopt MGD learning instead of stochastic gradient descent (SGD) or batch gradient descent (BGD). SGD updates the weight parameters for all training data, as shown in equation (5.2), and the training process is slower than MGD [107]. Also, the variance between each iteration is smaller in MGD than in SGD; thus, MGD leads to more stable convergence. In addition, BGD updates the weight parameters using the entire training data set, as stated in equation (5.3); this requires a large computational memory [108]. For large training data sets, the BGD updates may be very slow and may even converge into local minimums, because the gradients are computed from the same training data set for every epoch.

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k - \eta \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \tag{5.2}$$

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k - \eta \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}; \boldsymbol{x}_i; \boldsymbol{y}_i) \tag{5.3}$$

Thus, the advantages of MGD over other learning methods are apparent. However, there are a few issues regarding the generation of mini-batches that must be addressed. First, the size of the mini-batch must be defined empirically, considering the computational cost and the convergence speed. As stated in the existing research literature, large-sized mini-batches invoke a generalization gap problem, whereas, small sized mini-batches may cause long training times [108]. To resolve such issues, a criterion was suggested based on the variances [101, 102]. Another issue is the variance among the mini batches that arises with their generation. This variance often leads to a fluctuating gradient, which makes it difficult to converge to an optimal point [109]. To reduce such variance, several approaches have been developed in prior works [110, 111]. MGD is also widely used in deep learning based fault diagnosis studies [2, 112-115]. The prior works have mainly focused on diagnosis performance; thus, the issues outlined above have not been explored thoroughly. Thus, our research studies mini-batches for deep learning based fault diagnosis, focusing on mini-batch generation.

5.2 Label-based, Mini-batch Gradient Descent Study

This section proposes a label-based, mini-batch gradient descent method to update

filters in the learning process. Normally, mini-batches are sampled randomly from the training data set, which produces the biased mini-batches. For some problems, the biased mini-batches cause inaccurate and inconsistent results. In contrast, the proposed method samples mini-batches based on label information, which makes the bias of mini-batches controllable. To analyze the effect of the proposed method, a novel filter sensitivity analysis method is suggested. Since the performance relies on the ability of filters to capture patterns, the sensitivity of filters is considered in the analysis. To see the correlation between the performance and the sensitivity, the quantity of properly sensitive filters is defined by the proposed criteria. Section 5.2.1 describes the label-based mini-batches, and Section 5.2.2 illustrates the proposed filter sensitivity analysis method. Lastly, Section 5.2.3 states the three criteria to identify properly sensitive filters.

5.2.1 Label-based, Mini-batch Generation

Unlike conventional random mini-batch gradient descent, the proposed label-based mini-batch process samples training data based on label information. Using the labels, various types of label-based mini-batches can be defined by the two norms: the composition of the mini-batch and the order of the iteration sequence. At first, the composition is decided by a single label or multiple labels. A single-label composition indicates that only one label is in a mini-batch, while a multiple-label composition specifies data with more than one label in a mini-batch. Then, the order of the iteration sequence is decided. Since the compositions of the mini-batches are different from each other, the order of mini-batches used to update the gradient information can be varied. In this research, we proposed two orders: sequential and

mixed orders. A sequential order means that mini-batches of the same composition are used continuously in terms of the filter update sequence. In contrast, a mixed order means that mini-batches of different compositions are used successively in terms of the filter update sequence. The combination of the two compositions and the two orders results in four combinations, as shown in Table 5-2. Although there are many more combinations, for simplicity, we focused on the double-labeled case in this paper for the multiple labeled case. In addition to the four combinations, the equal mini-batches correspond to the equal number of each class of data in a mini-batch. Since the composition of all equal mini-batches is the same, there is no sequence order difference. In short, five combinations of mini-batches are proposed in this paper; each is analyzed using the criteria explained in the next section.

For example, assume that the training data consists of 20 data points for each of the four classes and that the batch size is fixed to 10; this generates 8 mini-batches. The mini-batches can be in various combinations, according to the label-based mini-batches stated above; this is shown in Table 5-3. In the figure, $\boldsymbol{y}_s^t$ represents a label vector of a mini-batch ($\boldsymbol{X}_s^t$) of iteration sequence *s* at epoch *t*, and the numbers in the set indicate labels of each class. As shown in Table 5-3, single-labeled mini-batches have only one label per mini-batch, *e.g.* $\boldsymbol{y}_s^t \in \{1\}$, while double-labeled mini-batches have two labels per mini-batch, *e.g.* $\boldsymbol{y}_s^t \in \{1, 2\}$. Also, the sequential order option arranges the same composition mini-batches continuously, while the mixed order choice arranges different composition mini-batches successively. For every epoch, the mini-batches are sampled again, independently from the previous ones.

Table 5-2 Combinations of label-based mini-batches.

Compositions \ Orders	Sequential	Mixed
Single label	Single-sequential	Single-mixed
Multi labels — **Double**	Double-sequential	Double-mixed
Multi labels — **Equal**	Equal	

Table 5-3 Example of label group of each label-based mini-batch.

Health states	y_1^t	y_2^t	y_3^t	y_4^t	y_5^t	y_6^t	y_7^t	y_8^t
Single-sequential	1	1	2	2	3	3	4	4
Double-sequential	1,2	1,2	1,2	1,2	3,4	3,4	3,4	3,4
Single-mixed	1	2	3	4	1	2	3	4
Double-mixed	1,2	1,2	3,4	3,4	1,2	1,2	3,4	3,4
Equal	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4

5.2.2 Filter Sensitivity Analysis

The label-based, mini-batch gradient descent method shows different performance results, as compared to the conventional random, mini-batch gradient descent method. The discrepancy between the two methods comes from the difference in trained weights; the trained weights catch patterns of the training data set. Specifically, in a convolutional neural network (CNN), where filters are 2-dimensional filters, filters that capture the core pattern of the images will present better performance than filters that catch no patterns. For example, a part of an image is shown (with filters A and B) in Figure 5-3. If the image is convoluted with filters A and B using the pixel values in the figure, the convolution result of filter A gives a much higher value than that of filter B. Just by comparing filters A and B with the image, the result is predictable. Nevertheless, it is impractical to visually inspect all of the filters, because deep networks have a large number of filters. Thus, filter sensitivity analysis is suggested to quantitatively evaluate the effect of the filters.

The sensitivity of a filter is defined as the change of output when one of the filters is inactivated in the testing process; this is analogous to "leave-one-out" cross validation. The definition can be stated as:

$$\boldsymbol{s}_i = f(\boldsymbol{x}; \boldsymbol{\theta}) - f(\boldsymbol{x}; \boldsymbol{\theta}_i) \tag{5.4}$$

where $\boldsymbol{s}_i$ is a sensitivity vector of the i^{th} filter being perturbed; f is the function that maps input data to output values of the network; $\boldsymbol{x}$ is one instance of testing data; $\boldsymbol{\theta}$ is a parameter of the network, including filters; and $\boldsymbol{\theta}_i$ is a parameter of the network of which pixels of the i^{th} filter have zero values. If the pixels of a filter that extracts significant representation are manually modified to have zero values, the sensitivity value will have a large value. In contrast, if the pixels of a poor filter (one

that does not catch any meaningful patterns) are set to have zero values, the sensitivity values will be small.

For the output values of networks, $f(\boldsymbol{x};\boldsymbol{\theta})$ and $f(\boldsymbol{x};\boldsymbol{\theta}_i)$ in Figure 5-4, the softmax function has been applied to each testing data set to make the values comparable. The size of the sensitivity vector equals the number of nodes in the last layer, of which each output represents the possibility of each class. Thus, each scalar value in the sensitivity vector indicates how sensitive the filter is to each class prediction. For example, if a testing data set presents the largest sensitivity value in the j^{th} node when the pixels of the i^{th} filter have been set to zero, it can be concluded that the i^{th} filter is most sensitive to the j^{th} class for the testing data. By using the sensitivity criterion, we can evaluate how the label-based, mini-batch gradient descent method affects the performance not by visually inspecting the filters, but instead by using a quantitative metric.

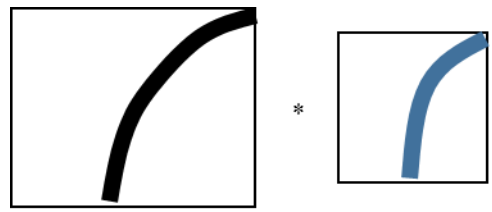


Image Filter A

(a)

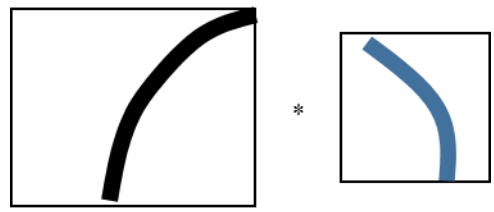


Image Filter B

(b)

Figure 5-3 Examples of convolution of filters with an image.

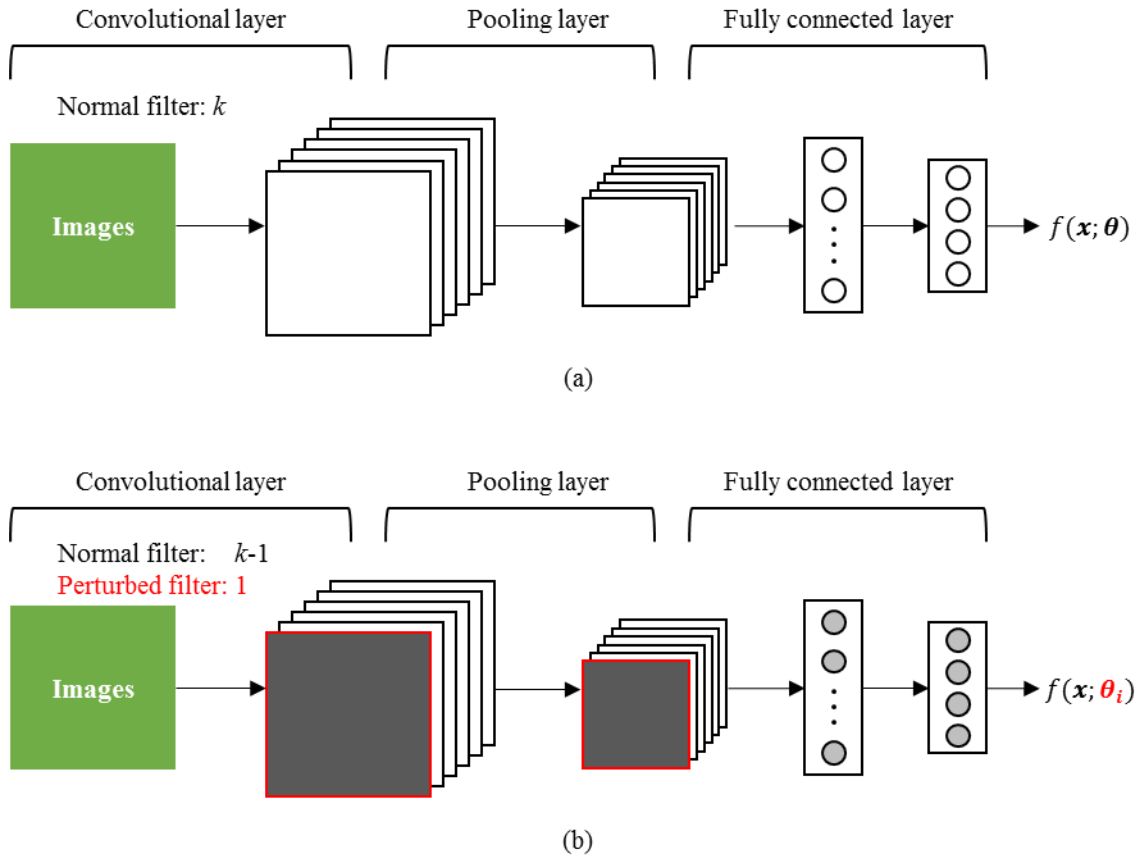


Figure 5-4 Networks with (a) normal filters and (b) a perturbed filter.

5.2.3 Criteria of Properly Sensitive Filters

The sensitivity of filters defined in Section 5.2.2 can represent the effect of filters on the performance of the network. However, some filters may be properly sensitive, while other filters may be incorrectly sensitive. To see the effect of the properly sensitive filters on the performance, the criteria for properly sensitive filters (PSF) are defined as the following criteria.

1) At a node, the health state of the largest sensitivity value among different health states should match the health state that the node it is representing.

2) For a health state of the testing data, the sensitivity value should be the largest at the corresponding node.

3) The sensitivity value should be larger than zero.

These criteria are used in each node of a filter to assess the suitability of filter sensitivity, which adds up to a maximum of m (number of nodes in the last layer) per filter. Since c number of filters are used in the CNN, the measured quantity of PSF in a network can have a maximum value of $m \times c$. This quantity is normalized by the maximum value to consider different structures of a network.

5.3 Description of Data Set

The amount of vibration image data generated from the testbed in Section 3.1 is shown in Table 5-4. The total amount of data for each health state is balanced, since the unbalanced data sets will influence the effect of the label-based mini-batch

gradient descent. Nevertheless, the numbers vary slightly in each test, since the test duration time was controlled manually. Also, a few points at the start and end of the vibration signals were cut off to make the pattern locate randomly.

To validate the performance of the proposed method, as outlined in Section 5.2, each test data set was split into training and testing data sets with the ratio of 0.8 to 0.2, respectively. Thus, the sizes of training and testing data images were 23,077 and 5,769, respectively.

Table 5-4 Data size of each data set for each health state.

Health states	Set 1	Set 2	Set 3	Set 4	Set 5	Total
Normal	1441	1448	1450	1439	1441	7219
Rubbing	1443	1432	1444	1446	1438	7203
Misalignment	1443	1440	1452	1442	1442	7219
Oil whirl	1445	1437	1444	1438	1441	7205

5.4 Results of Label-based, Mini-batch Gradient D escent Methods

5.4.1 Performance of Label-based, Mini-batch Methods

The performance results of five label-based combinations were evaluated using the data set described in Section 5.3. The data sets of the four health states were used to train the convolutional neural network (CNN) described in Section 5.1.1. Relatively shallow CNNs were adopted in this research to allow us to clearly see the effect of the label-based mini-batches. To see the difference between the conventional random mini-batch and the proposed label-based mini-batch, both the training accuracies and the testing accuracies are presented. Figure 5-5 shows the results of the random mini-batch approach, using 1-layer and 2-layer CNN structures. Figure 5-7 shows the results of the label-based mini-batch approach, using 1-layer and 2-layer CNN structures.

Random mini-batches reached the maximum of both training and testing accuracies in a few epochs, as shown in Figure 5-5. The 1-layer and 2-layer CNN structure cases of random mini-batches converged to 100% accuracies at earlier epochs. The two cases generated several mini-batches that had missed at least one label, which had minor effect on the performance. However, as the ratio of such missing label mini-batches is increased by the imbalanced data size between training labels, the performance decreases and fluctuates as shown in Figure 5-6. This result shows that the random mini-batches may not be robust for certain cases.

Compared to the conventional random mini-batch case, the label-based mini-batch cases presented diverse results, except the equal mini-batch case shown in

Figure 5-7(e). The equal mini-batches represented a trend similar to random mini-batches for the balanced data set. The training and testing accuracies reached maximum accuracy within a few epochs; the cost values decreased steadily as well. Through the random and the equal mini-batch cases, we can see that even a shallow structure can learn to represent the four health states of the rotor system.

However, the other proposed label-based mini-batches failed to learn generalized patterns of the four health states. Single-sequential and double-sequential cases represent relatively lower testing accuracies throughout the entire epochs, as presented in Figure 5-7(a) and (b) respectively, for both CNN structures. Although the 2-layer CNN structure in Figure 5-7(b) presents 0.8 accuracy at around epoch 10, it decreases steadily and falls to a quite low accuracy in the end. In contrast, training of sequential order cases present high accuracy values, which indicates that the network has been overfitted to the training data set. In addition, single-mixed and double-mixed mini-batches show similar patterns to the sequential order cases. Although the training accuracy tends to follow the random case, testing accuracy starts to fall after 10 epochs, independent of CNN structure. Eventually, the network fails to predict more than half of the testing data. The mixed order cases present a typical overfitting tendency as well.

Through the case study, the effect of a label-based, mini-batch method for fault diagnosis performance is shown. Equal composition mini-batches showed a performance similar to the conventional random mini-batch method, while other combinations showed a deteriorated generalization ability. Since the performance relies heavily on the capability of the filter to capture patterns of images, the cause of the performance variation can be explained by assessing the capability of the

filters.

The capability of the filters can be evaluated qualitatively by visual inspection of filter images, as shown in Figure 5-8, which represents images of a few filters of the double-sequential case at epochs 1, 25, 30, and 45 using a 1-layer CNN. Each filter image is normalized to stress out the pattern of the filter images. At epoch 1, which is early in the training process, the filters present lots of noise as they are initialized randomly before training. As filters are updated through each epoch, some filters represent a pattern, such as that seen in Figure 5-8(a) and (c). In contrast, some other filters present a weak pattern, as presented in Figure 5-8(b) and (d). The weak pattern is not effective for representing the vibration images. Other types of filters also appear to be not effective, as a small part of a filter shows no variation, such as the gray part in filter number 15 in Figure 5-8(e). The major drawback of the ineffective filters shown in Figure 5-8(b), (d), and (e) is that the filters fail to be updated to represent a pattern if some part of the images becomes blank, which indicates less variation.

The images in Figure 5-9 and Figure 5-10 are filter images of each mini-batch case at the last training epoch for 1- and 2-layer CNN structures, respectively. Although the random mini-batch case seems to have few ineffective filters, the testing accuracy indicates that there are enough effective filters to recognize the vibration image patterns. Other combination cases, except the equal case, show that useless filters comprise most of the filters, which leads to the low performance results. However, visual inspection of filters is not accurate enough to decide whether the filter is effective or not. To quantitatively evaluate the performance of the filters, a quantified criterion of filter performance is proposed in the next section.

By using a quantified criterion of filter sensitivity, the number of effective and ineffective filters can be measured accurately.

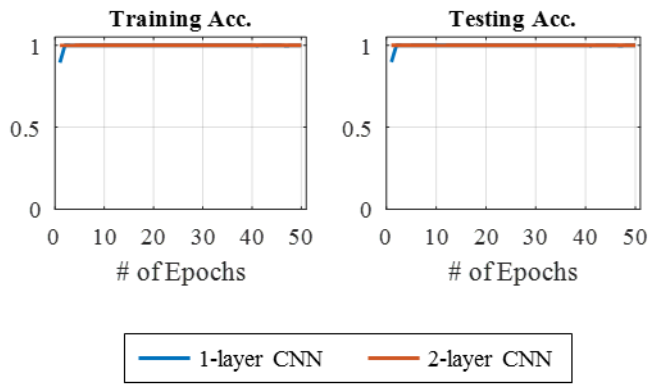


Figure 5-5 Training and testing accuracy of the random, mini-batch method.

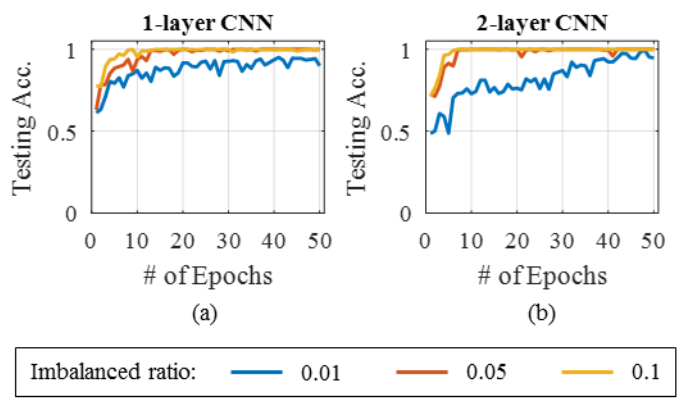


Figure 5-6 Testing accuracy of the random, mini-batch method for imbalanced data set using (a) 1-layer and (b) 2-layer CNN structures.

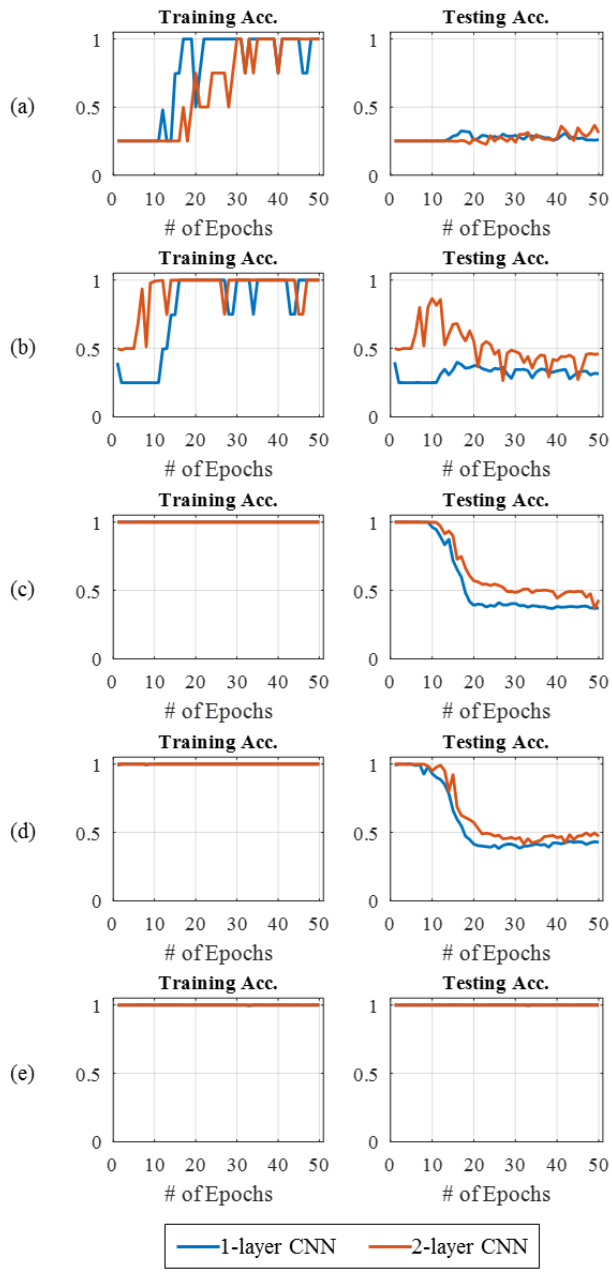


Figure 5-7 Training and testing accuracy of the label-based, mini-batch methods: (a) single-sequential, (b) double-sequential, (c) single-mixed, (d) double-mixed, and (e) equal.

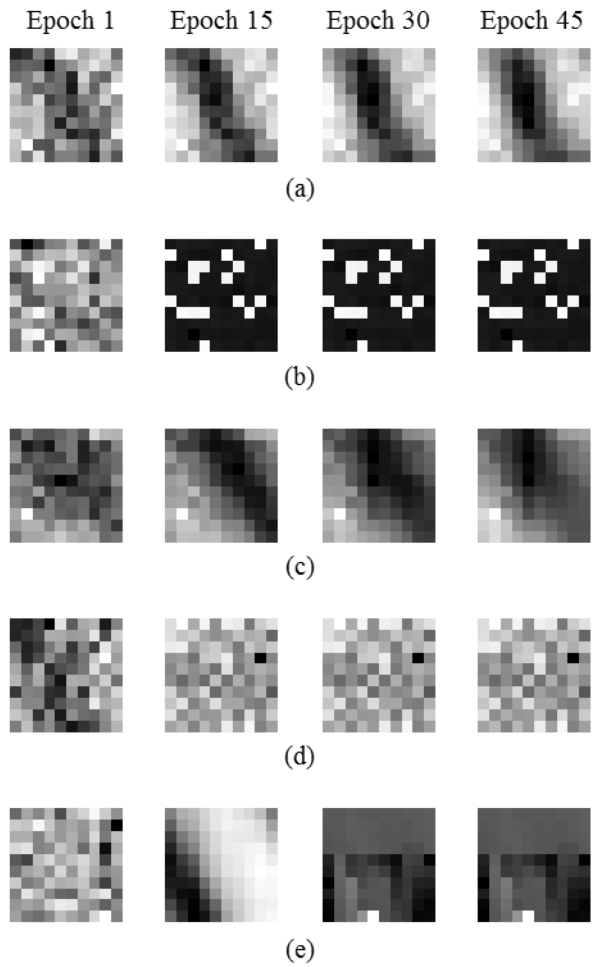


Figure 5-8 Filter images of double-sequential mini-batch of 1-layer CNN:

(a) filter #3, (b) filter #5, (c) filter #6, (d) filter #7, and (e) filter #15.

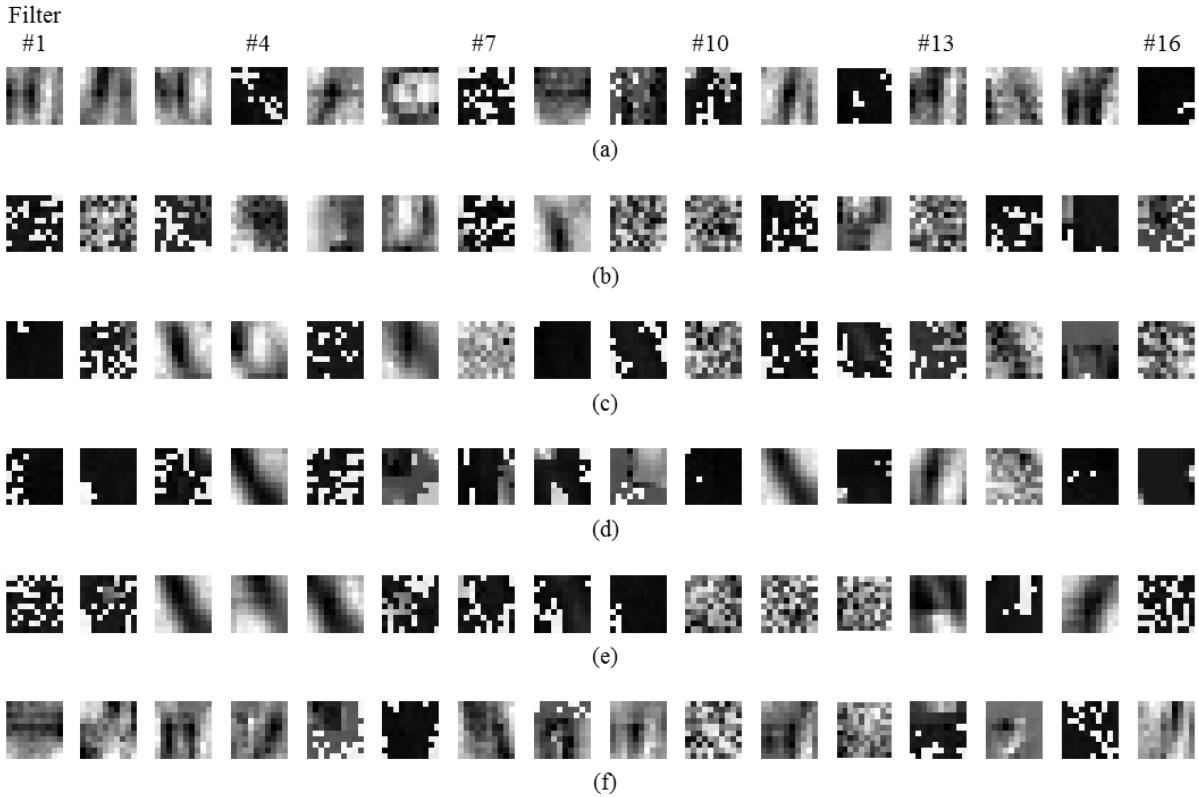


Figure 5-9 Trained filter images of 1-layer CNN: (a) random, (b) single-sequential, (c) double-sequential, (d) single-mixed, (e) double-mixed, and (f) equal.

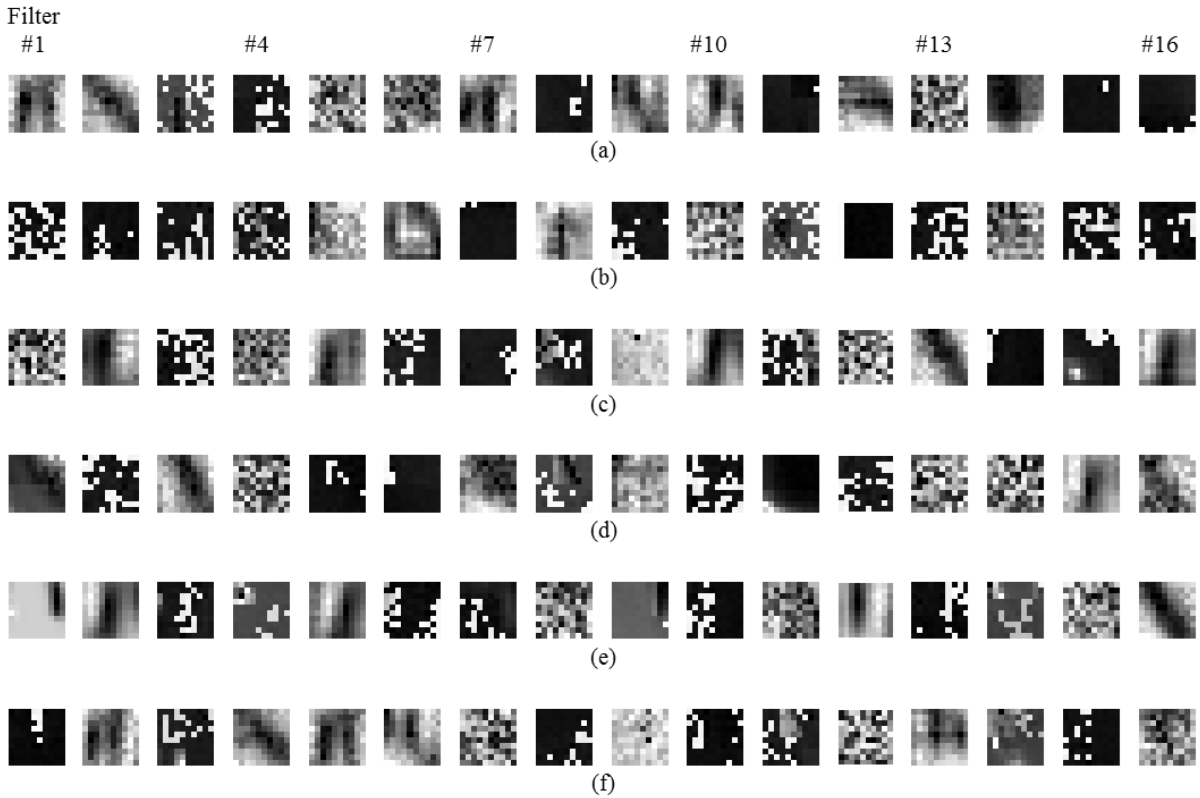


Figure 5-10 Trained filter images of 2-layer CNN: (a) random, (b) single-sequential, (c) double-sequential, (d) single-mixed, (e) double-mixed, and (f) equal.

5.4.2 Sensitivity of Filters for Label-based, Mini-batch Methods

The sensitivity of the filters is analyzed in this section, as described in Section 5.2.2. The dimension of a sensitivity vector equals the number of nodes at the last fully connected layer, where the output values of each node represent the probability of each health state. Since the four health states are trained, the dimension of the sensitivity vector is fixed to four. In this manner, every testing data set requires four sensitivity values. Nevertheless, it is complex to express the vectors of each testing data set, so the vectors are averaged throughout the same health state data. As a result, sensitivity vectors of the same health state of the testing data are averaged, which are reduced to four sensitivity vectors, presented as four lines in each sub-figure in Figure 5-11.

First, the high prediction accuracy of the 1-layer CNN structure random mini-batch approach can be supported by the sensitivity of filter numbers 3, 5, and 8, as presented in Figure 5-11. The three filters had the largest sensitivity values among the sixteen filters. Filter number 3 responded sensitively to normal data at node 1, which represents the probability of the normal health state. Similarly, filter number 8 responded sensitively to oil whirl states in node 4, and filter number 5 responded sensitively to rubbing and misalignment states in nodes 2 and 3, respectively; this can be interpreted as a case where the two health states share a similar pattern. Other filters showed relatively low sensitivity values; the images of those filters had ineffective filter images, such as those shown in Figure 5-8(b) and (e). Although sensitivity values of filter numbers 3, 5, and 8 are high, the filter images in Figure 5-12 presented vague patterns.

In contrast to the random mini-batch approach, the label-based, mini-batch

approaches showed different results. As shown in Figure 5-7, single-sequential and double-sequential approaches for the 1-layer CNN structure had low testing accuracies. Consistently, most of the filter images in Figure 5-9(b) and (c) presented low-variance filters. Sensitivity values of those filters from each approach are shown in Figure 5-13(a) and (b), respectively. Figure 5-13(a) presents one of the insensitive filters and Figure 5-13(b) presents an example of an incorrectly sensitive filter that is sensitive of all the nodes.

The single-mixed and double-mixed mini-batch methods for 1-layer CNN showed similar trends. As the testing accuracies of the testing data set decrease after several epochs, the sensitivity of some filters also shows high values in earlier epochs and low values in later epochs. Filter number 2 of the single-mixed, mini-batch case in Figure 5-13(c) responded sensitively to the oil whirl state at the corresponding node 4, but the filter became insensitive after 20 epochs. Likewise, filter number 5 of the double-mixed mini-batch in Figure 5-13(d) also showed relatively high sensitivity values for normal health state data; however, it did not give clear sensitivity results after 20 epochs. In addition, the equal mini-batch cases had similar sensitivity results as the random mini-batch case presented in Figure 5-13(e).

The results of filter sensitivity analysis for the 1-layer CNN structure indicate a correlation between the filter sensitivity and the testing accuracy. Few filters responded clearly to the health state data at proper nodes following the trend of testing accuracy, while other filters did not show any difference independent of testing accuracy. Although the correlation can be observed by analyzing sensitivity from all the filters, it gets more and more incomprehensible as the number of filters is increased. In addition, highly fluctuating results and small sensitivity values, such

as the 2-layer CNN cases presented in Figure 5-14, discourage finding the correlation between sensitivity and accuracy. Thus, a quantified measure of sensitivity analysis results is used in the following section.

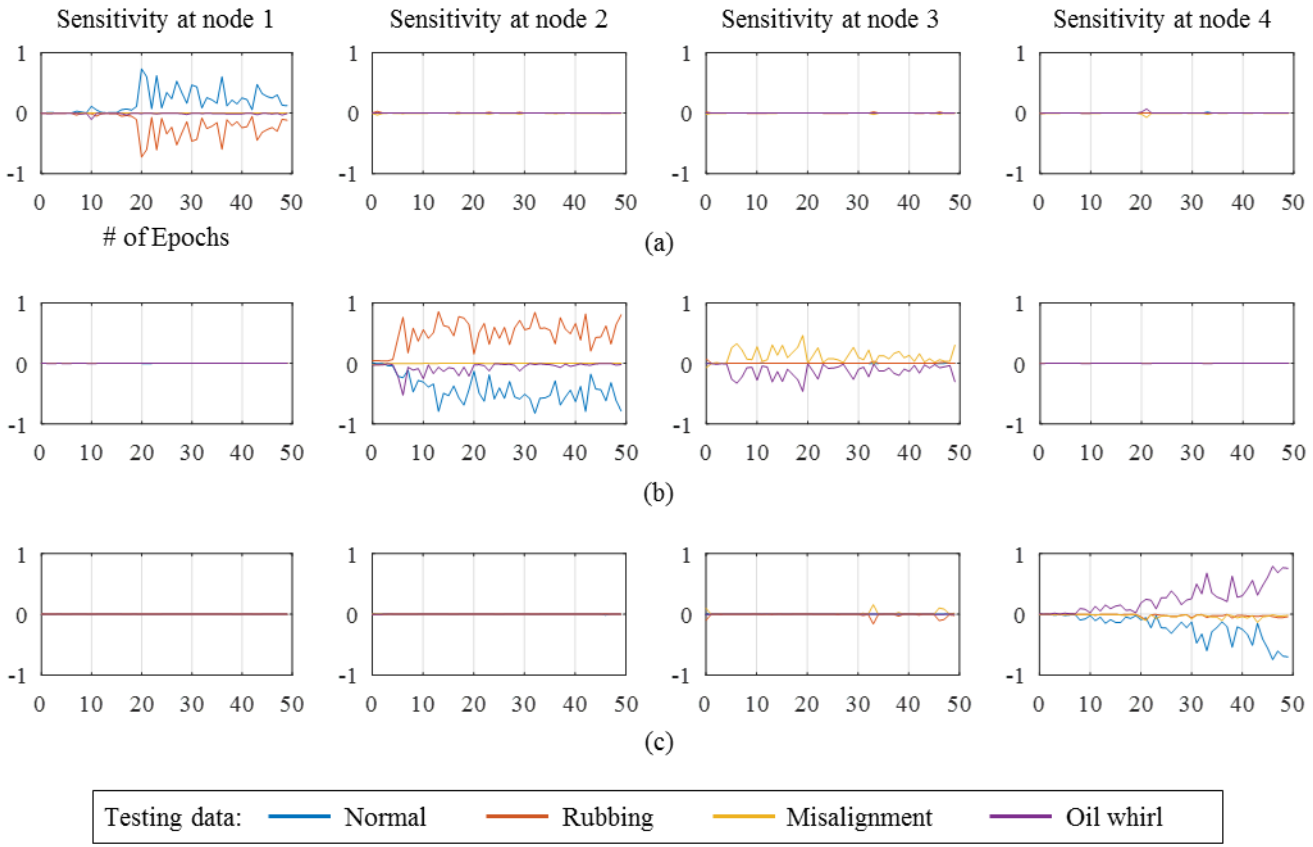


Figure 5-11 Filter sensitivity analysis of random, mini-batch case for 1-layer CNN: (a) filter #3, (b) filter #5, and (c) filter #8.

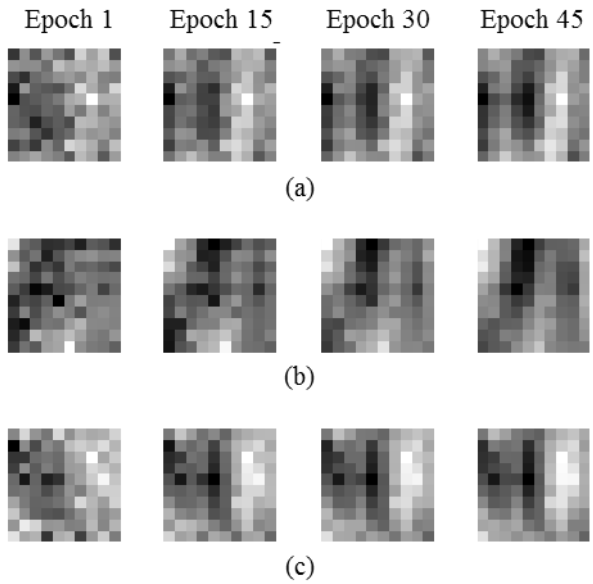


Figure 5-12 Filter images of random, mini-batch case for 1-layer CNN; (a) filter #3, (b) filter #5, and (c) filter #8.

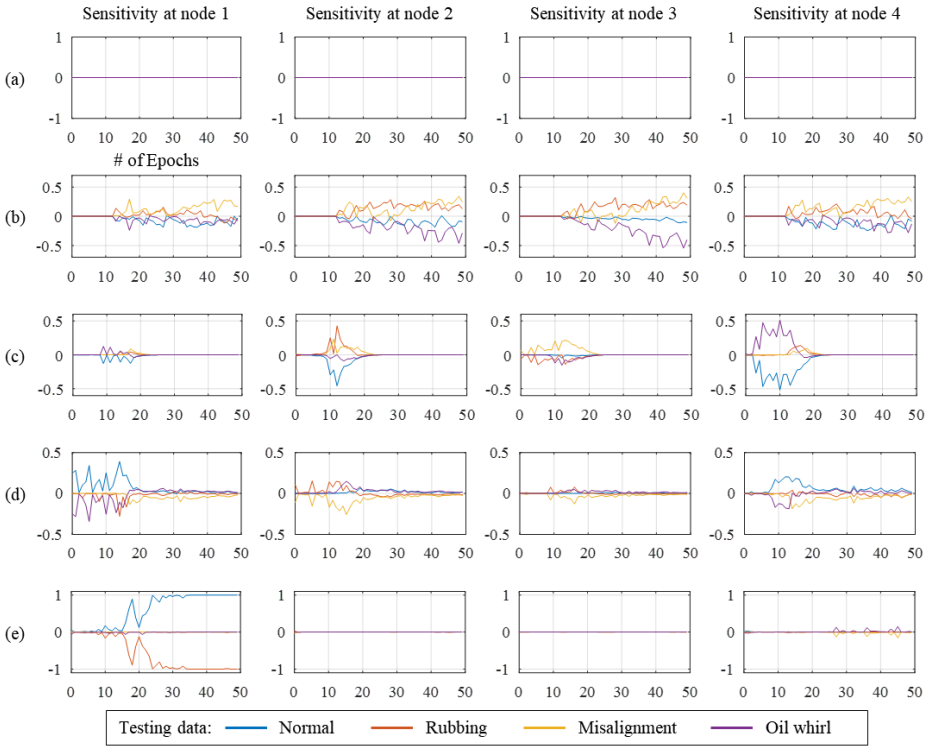


Figure 5-13 Filter sensitivity analysis of 1-layer CNN: (a) filter #3 for the single-sequential case, (b) filter #4 for the double-sequential case, (c) filter #2 for the single-mixed case, (d) filter #5 for the double-mixed case, and (e) filter #7 for the equal case.

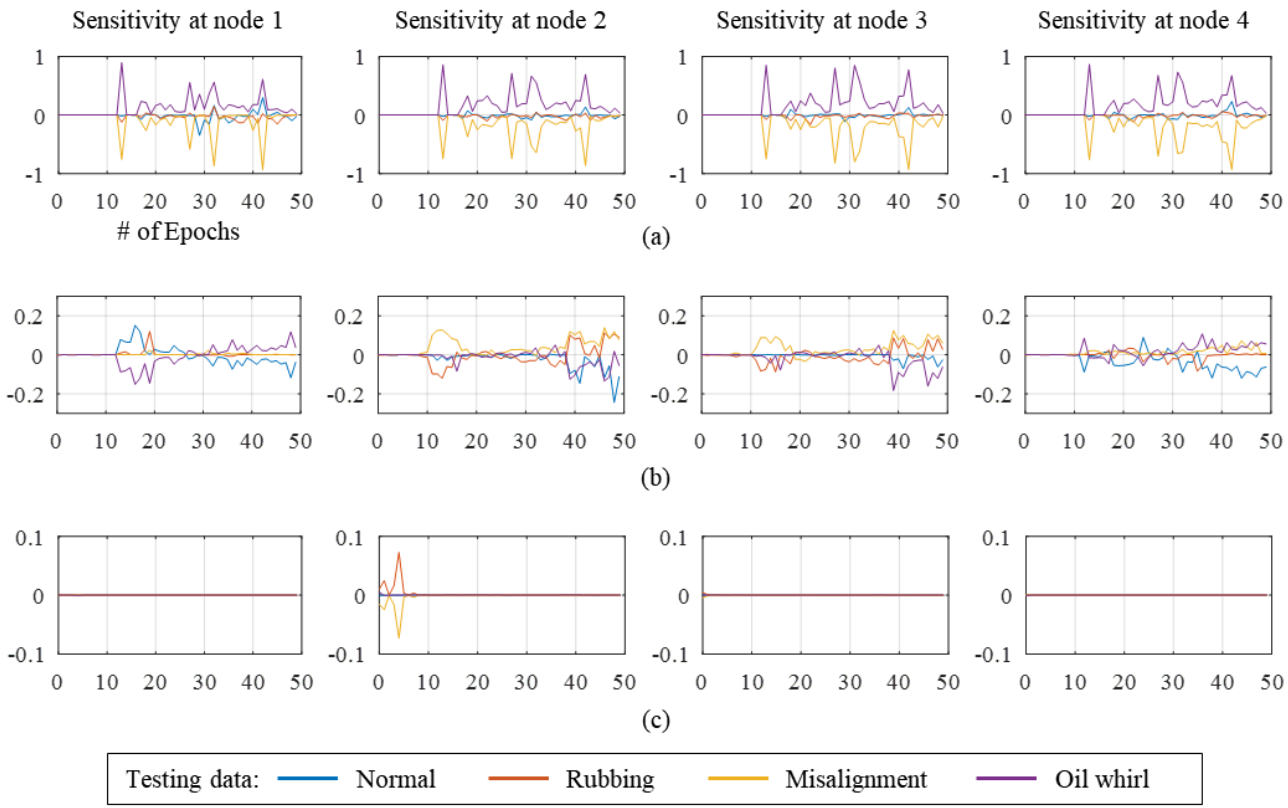


Figure 5-14 Filter sensitivity analysis of 2-layer CNN: (a) filter #2 for the single-sequential case, (b) filter #10 for the single-mixed case, and (c) filter #2 for the equal case.

5.4.3 Correlation between Performance and Sensitive Filters

The sensitivity of filters varies widely among mini-batch methods. Random and equal mini-batch methods hold several filters that are properly sensitive to health states, while other methods have filters that are mistakenly sensitive or insensitive. The result indicates that the quantity of the properly sensitive filters (PSF) is closely related to the performance of the CNNs. Although the correlation can be observed, a quantified measure of filter sensitivity will make the correlation more evident. Thus, this section evaluates the quantifies measure of PSF defined in Section 5.2.3 to show the correlation of the sensitivity and the accuracy.

An example of evaluation process of PSF is shown by Figure 5-11(a). The sensitivity values of filter #3 at node 1 show that the normal health state has the largest sensitivity values among those of the other health states in most of the epochs, which satisfies criteria 1. Also, the sensitivity values of the normal health state in other nodes are less than the value at node 1, which satisfies criteria 2 as well. Finally, the sensitivity values of the health state at node 1 are larger than 0, which is the condition of criteria 3. The processes are repeated at nodes 2, 3, and 4 of filter #3, which is followed by the same processes at all other filters.

Prediction accuracies of mini-batch methods shown in Section 5.4.1 are presented with the normalized quantity of PSF in Figure 5-15 and Figure 5-16. In the figures, the solid and dotted lines denote the prediction accuracy of the testing data set and the normalized quantity of properly sensitive filters, respectively. Prediction accuracy showed a very similar trend to the quantity of properly sensitive filters for both 1- and 2-layer CNN structures. For single-mixed and double-mixed batches, the large number of the normalized quantity led to a high prediction

accuracy in early epochs; however, in later epochs the quantity dropped, as well as the prediction accuracy. The single-sequential and double-sequential methods had consistently low amounts of properly sensitive filters along epochs, which generated similarly low prediction accuracies. In contrast, the prediction accuracies of random and equal mini-batch cases remain high, since the amount of properly sensitive filters is relatively larger than other mini-batch methods.

The results indicate a strong correlation between the quantity of properly sensitive filters and the performance of the network. As sensitivity denotes the ability of filters to capture good representations of the training data set, a large number of properly sensitive filters leads to better performance of the network. In contrast, the mini-batch methods with low performances had relatively fewer properly sensitive filters. By using PSF, the performance of the network was evaluated indirectly.

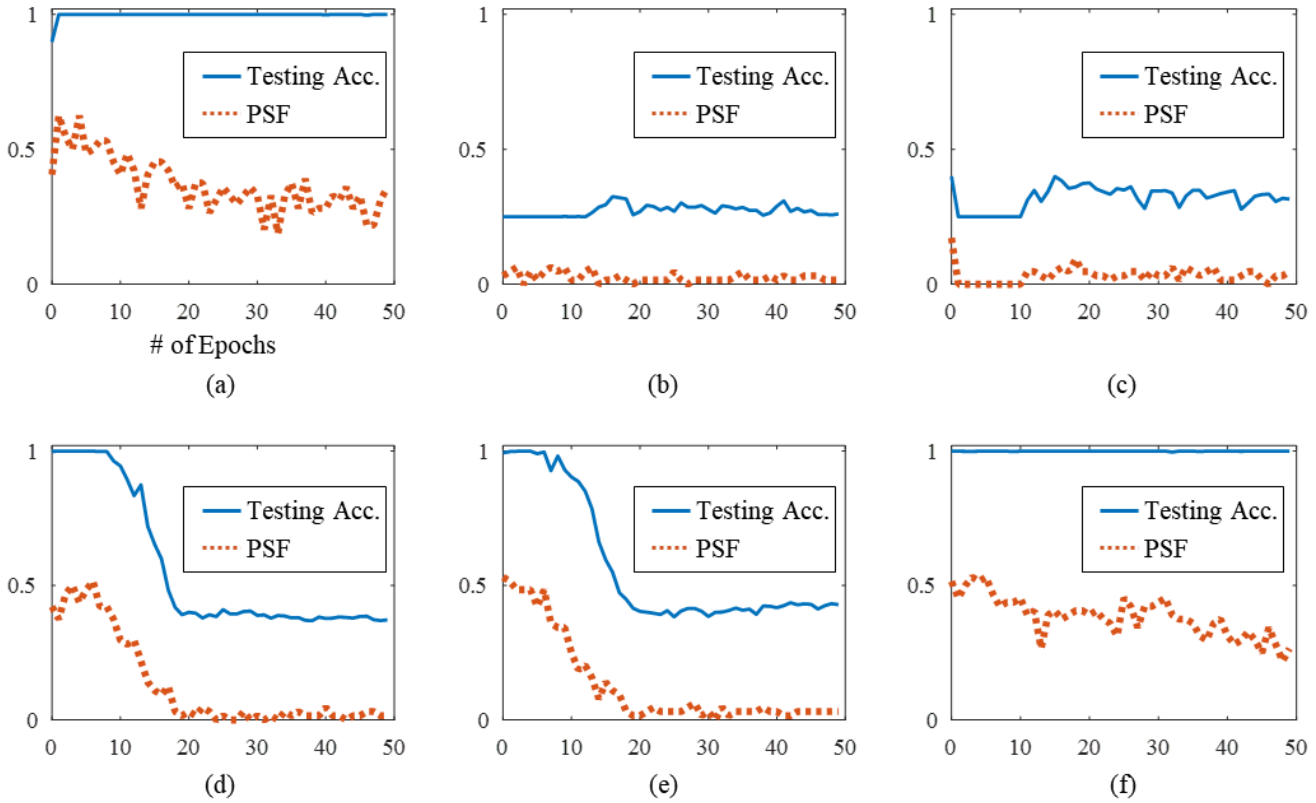


Figure 5-15 Testing accuracy and normalized quantity of Properly Sensitive Filters (PSF) for 1-layer CNN:

(a) random, (b) single-sequential, (c) double-sequential, (d) single-mixed, (e) double-mixed, and (f) equal.

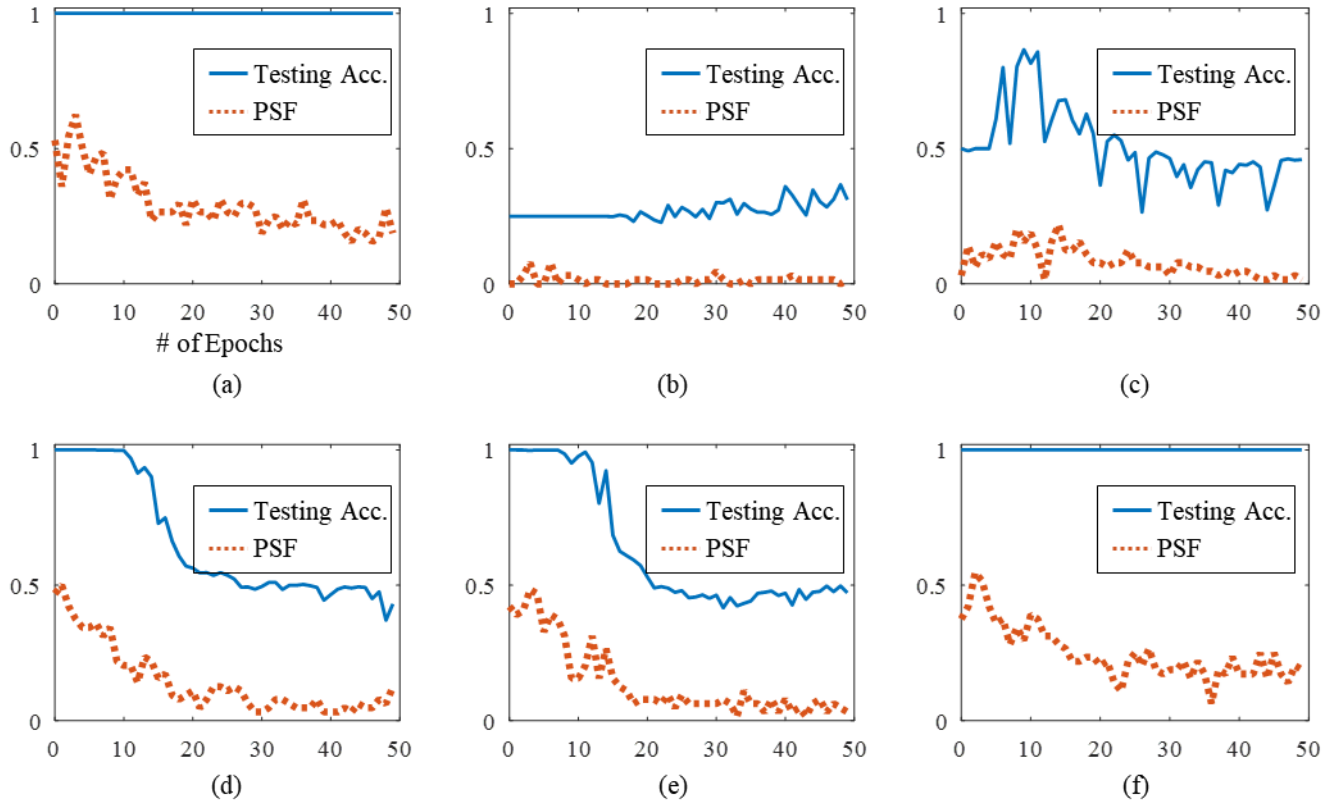


Figure 5-16 Testing accuracy and normalized quantity of Properly Sensitive Filters (PSF) for 2-layer CNN:

(a) random, (b) single-sequential, (c) double-sequential, (d) single-mixed, (e) double-mixed, and (f) equal.

Sections of this chapter have been published or submitted as the following journal articles:

1) **Joon Ha Jung**, Myungyon Kim, Jin Uk Ko, Hyeon Bae Kong, Byeng D. Youn, and Kyungho Sun, "Performance Evaluation of CNN Based Diagnostics by Filter Sensitivity Analysis with Label-based, Mini-batch Combinations ," *Knowledge-Based Systems*, 2019. (Submitted)

Chapter 6

Retraining the Minor Class Scheme for Imbalanced Data Sets

The amount of data in each class can have a considerable effect on the performance of deep learning based diagnosis of a system. In industry sites, where acquisition of data is hindered by various complicated issues, the size of the data set can vary from class to class; this is called am ‘imbalanced data set problem.’ The imbalanced problem frequently arises in large-scale machinery. Generally, large-scale machinery is maintained carefully via strict regulations and has maintenance sessions periodically, even if the machine is in a healthy state. This periodic-based maintenance strategy is retained, along with condition-based maintenance, because an accident or a sudden breakdown can cause substantial loss and may even cause casualties. In addition, inducing a fault on a real system may produce unwanted results, which can also damage the system permanently. Thus, conservative operation of large-scale machinery makes data acquisition for anomaly states problematic.

Fluid-film bearing rotor systems are mostly large-scale, since the fluid can support high loads and speeds. Hence, data-imbalanced problems are in common for fluid-film bearing rotor systems, as well as other large-scale machinery. For example,

the steam turbine in a power plant operates 24 hours per day, 7 days per week in normal conditions, as the cost of cool-down and restarting is extremely high. Thus, vibration data of the normal state can be acquired easily; whereas, the chance of acquiring vibration data of an anomaly state is low.

As the imbalanced ratio between classes becomes greater, the conventional deep learning based diagnosis process for balanced data sets should be revised to achieve moderate performance. The widely used random mini-batch gradient descent method randomly samples data from the imbalanced data set, which makes the classifier biased towards major classes. Specifically, the training of the convolutional neural network will update the filters to be sensitive to the major classes. For example, if the two class data set has 99 and 1 samples for the major and minor classes, respectively, conventional training will mostly focus on estimating the samples of the major class, as shown in Figure 6-1(a). Although the overall accuracy is 0.99, the accuracy of the minor class is 0, which may allow a desirable result in the fault diagnosis. In contrast, in some cases, the prediction accuracy of the minor class can be weighted further, as shown in Figure 6-1(b), which has lower overall accuracy but higher accuracy of the minor class prediction. As failure of a large-scale, fluid-film bearing rotor system can result catastrophic damage, the issues arising from the imbalanced data set must be considered during the training process in which the classes of anomalies are usually minor.

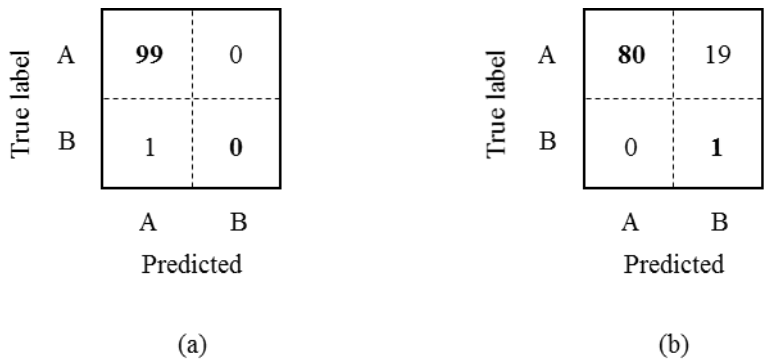


Figure 6-1 Confusion matrix of (a) the conventional training method and (b) the modified method, considering an imbalanced data set.

Training schemes for an imbalanced data set can be grouped into three types: data-level, algorithm-level, and cost-sensitive approaches, as described in the literature [e.g., 45, 46]. In this study, a hybrid level approach that combines data-level and algorithm-level approaches is proposed to efficiently learn features of the minor class. As stated in Section 4.2.3, the size of images and filters for this study were set at 28-by-28 pixels and 10-by-10 pixels, respectively. In addition, the sensitivity of the filters, as proposed in Section 5.2.2, was used in this research thrust.

The rest of this chapter is organized as follows. Section 6.1 defines the imbalanced data sets used in this study and the preliminary result of the imbalanced data set is shown. Section 6.2 describes the proposed retraining approach that uses equally labeled mini-batches, as described in the previous section. In Section 6.3, the results of different imbalanced ratios are analyzed.

6.1 Preliminary Study of the Imbalanced Data Set Problem

This section describes the overall issue of the imbalanced data set problem in convolutional neural network (CNN) based diagnosis. Section 6.1.1 defines an imbalanced data set and describes the performance of the conventional approach. Through a case study of fluid-film bearing rotor system data, the imbalanced data set is shown to affect the performance of learning the representation from samples of the minor class. However, the equally labeled mini-batch method proposed in Section 5.2 can address some case studies that have up to a certain level of imbalanced ratio. The performance of the equally labeled mini-batch method on several imbalanced ratio cases is shown in Section 6.1.2.

6.1.1 Imbalanced Data Sets

The imbalanced data sets used in this research include four different health states; normal, rubbing, misalignment, and oil whirl. The imbalanced data set was randomly sampled from the balanced data set used in Chapter 5. As a normal condition is the prevailing state of a fluid-film bearing rotor system, the normal state is fixed as the majority class. The size of the other three anomaly states is reduced by the imbalanced ratio (ρ), which can be defined as the ratio between the size of the minority class (N_{min}) and the majority class (N_{max}).

The imbalanced ratio of the three anomaly states was fixed at the same ratio for the preliminary study. The three imbalanced ratios–0.1 , 0.01, and 0.001–were tested with the balanced testing data set using the conventional random mini-batch gradient descent method. A convolutional neural network (CNN) was used to learn the

representation of four health states of the fluid-film bearing rotor system. The prediction accuracy of the testing data set is shown in Figure 6-2. As the number of epochs are increased, the cases of an imbalanced ratio of 0.1 and 0.01 present better prediction accuracy, while the case of a 0.001 ratio presents a low prediction accuracy of 0.25. In addition, the prediction accuracies at the last epoch for each case show a clear decrease as the imbalanced ratio decreases.

The substantial prediction gap between the cases of 0.01 and 0.001 is due to the number of minority class samples. Since the number of normal state (majority class) samples is about 5,000, the size of anomaly states (the minority class) is set as 50 and 5 for each case, respectively. Clearly, for the 0.001 ratio case, the probability of learning features of the anomaly states is low. In other words, without considering the imbalanced training data set, most mini-batches in each epoch miss samples from the minority classes, which leads to a decrease in performance. As the imbalanced ratio decreases, the prediction accuracy also decreases as the chance of missing the minority class sample in a mini-batch increases. This trend is clearly shown in Figure 6-3 and Figure 6-4. The confusion matrices shown in Figure 6-5 and Figure 6-6 indicates that only the normal state has been diagnosed correctly in both the 1-layer and 2-layer CNN structures.

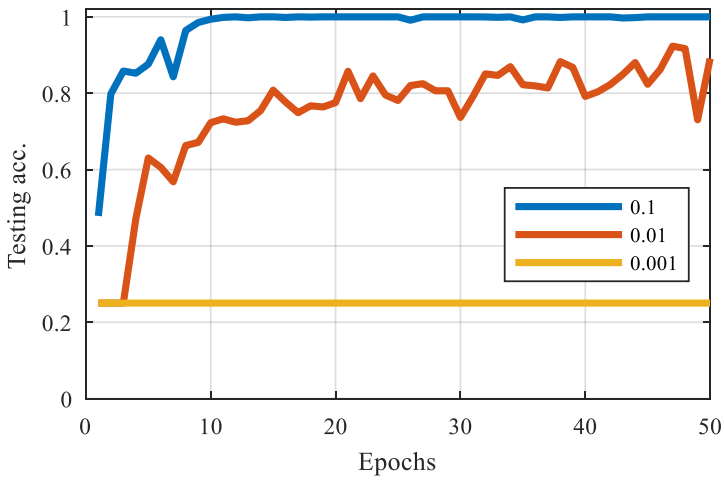


Figure 6-2 Prediction accuracy of using a training data set of various imbalanced ratios.

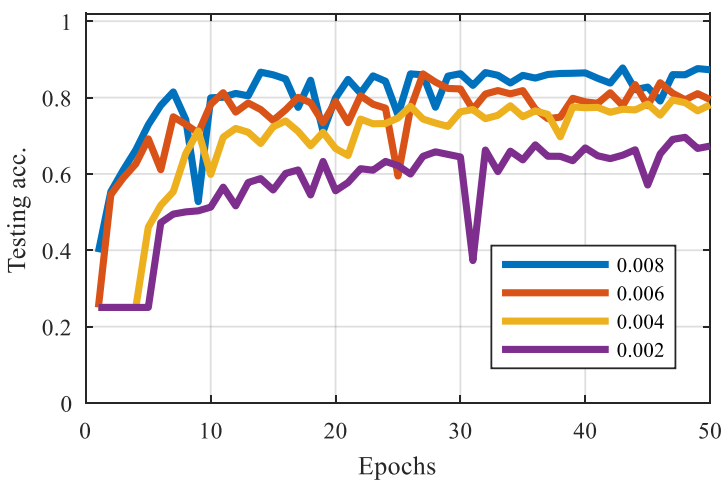


Figure 6-3 Prediction accuracy of using a training data set with an imbalanced ratio between 0.002 and 0.008 using 1-layer CNN.

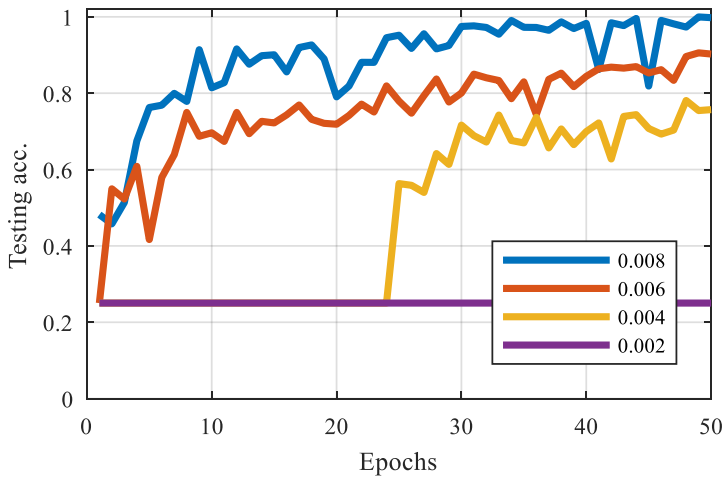


Figure 6-4 Prediction accuracy of using a training data set with an imbalanced ratio between 0.002 and 0.008 using 2-layer CNN.

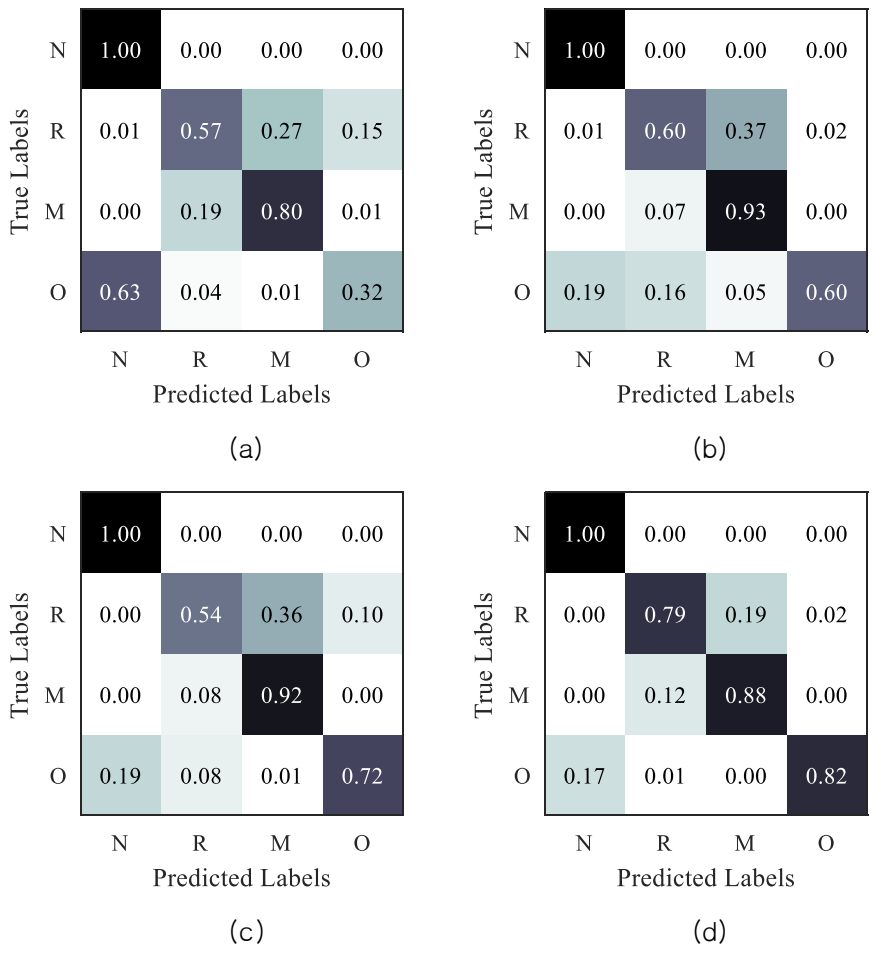


Figure 6-5 Confusion matrix of the last epoch of imbalanced ratio cases using 1-layer CNN; (a) 0.002, (b) 0.004, (c) 0.006, and (d) 0.008.

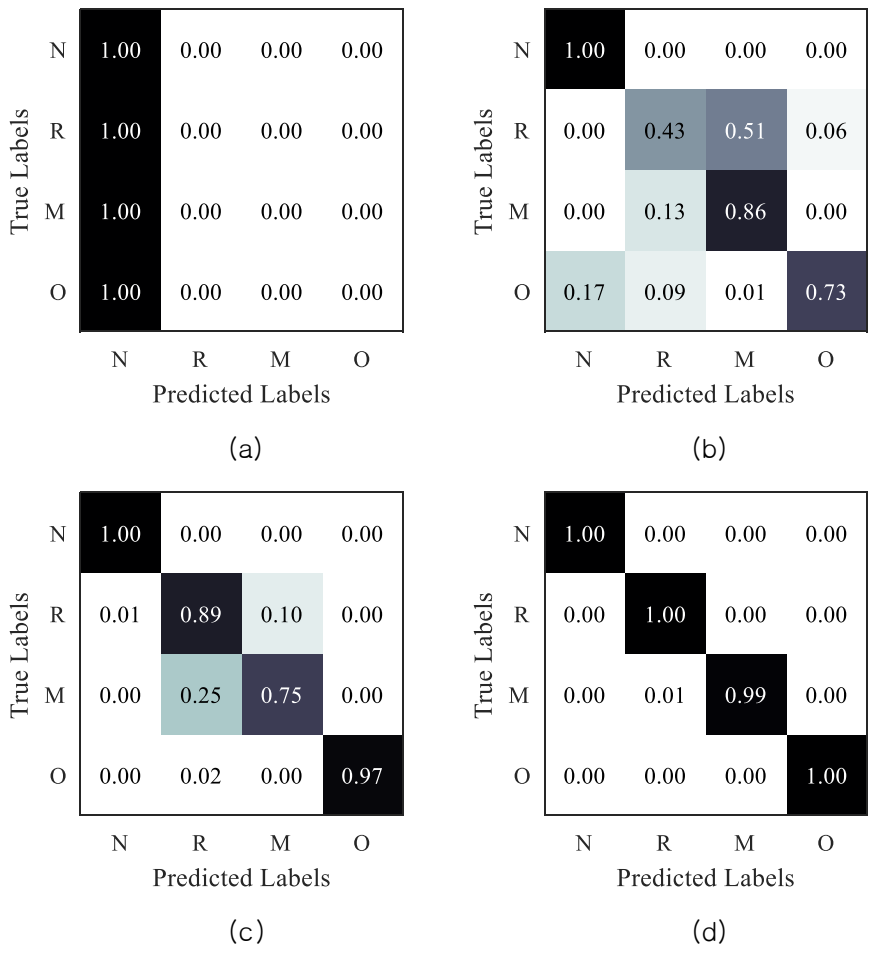


Figure 6-6 Confusion matrix of the last epoch of imbalanced ratio cases using 2-layer CNN; (a) 0.002, (b) 0.004, (c) 0.006, and (d) 0.008

6.1.2 Equally Labeled Mini-batch by Oversampling

The proposed equally labeled mini-batch method showed consistent performance for the balanced data set presented in Section 0. However, this mini-batch may not be useful for the imbalanced data set, as the size of the minority classes is not large

enough to make all of the mini-batches in an epoch. For example, if the size of mini-batch has been set as 100 for the four class problem, the size of each minority class needed to form an equally labeled mini-batch should be at least 25 (=100/4). In addition, some portion of the mini-batches in an epoch could miss the minority class due to the imbalanced data set. Thus, oversampling of the minority class is required to make the mini-batches.

The combination of imbalanced ratios of 0.1, 0.1, 0.01 for the three anomaly states–rubbing, misalignment, oil whirl–was selected to see the effect of oversampling in the equally labeled mini-batch method. For each anomaly state, the imbalanced ratio of 0.01 covered the following three cases. For example, the first case stands for a relative imbalanced ratio of 1, 0.01, 0.1, and 0.1, which represents the relative size of samples for normal, rubbing, misalignment, and oil whirl, respectively. The prediction accuracy of a balanced testing data set is presented in Figure 6-7. Compared to the random mini-batch method, the equally labeled mini-batch method retains greater performance throughout epochs. Although the random mini-batch method starts to learn presentations of the minority class after 20 epochs, the prediction accuracy does not converge to that of the equally labeled mini-batch method. The other two cases are presented in Figure 6-8 and Figure 6-9. A similar trend is shown with the other two cases as well.

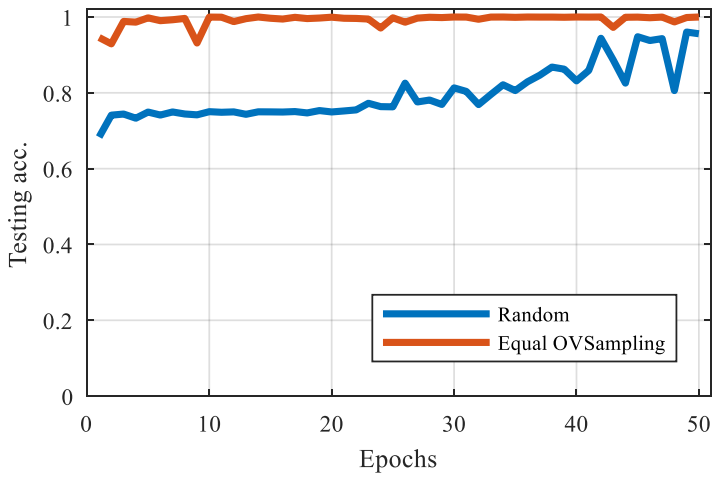


Figure 6-7 Prediction accuracy of the random mini-batch method and the equally labeled mini-batch method using oversampling for the imbalanced data set (1, 0.01, 0.1, 0.1).

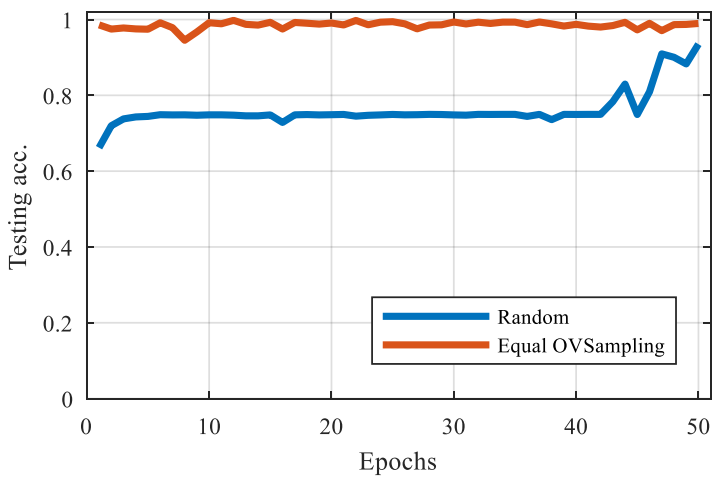


Figure 6-8 Prediction accuracy of the random mini-batch method and the equally labeled mini-batch method using oversampling for the imbalanced data set (1, 0.1, 0.01, 0.1).

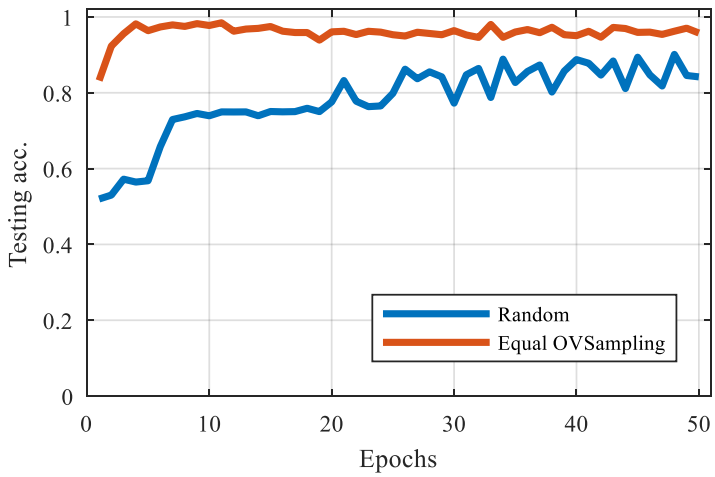


Figure 6-9 Prediction accuracy of the random mini-batch method and the equally labeled mini-batch method using oversampling for the imbalanced data set (1, 0.1, 0.1, 0.01).

6.2 Retraining Scheme for the Minor Class

The cases of an imbalanced ratio, including 0.01, were effectively trained using equally labeled mini-batches with the oversampling technique. However, if the least imbalanced ratio is decreased to 0.001, the effect of the method is decreased, as shown in Figure 6-10. The confusion matrix in Figure 6-11 indicates that the overall prediction accuracy is decreased due to the least minor class, which is the rubbing state. Other states in which the imbalanced ratio is equal or larger than 0.01 achieved high accuracy. Thus, to increase the overall performance of such cases, a retraining scheme for the least minor class is suggested in this research.

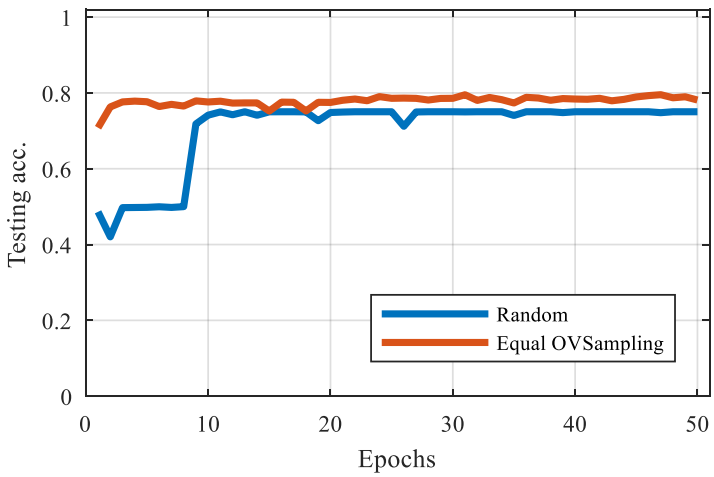


Figure 6-10 Prediction accuracy of the random mini-batch method and the equally labeled mini-batch method using oversampling for the imbalanced data set (1, 0.001, 0.01, 0.1).

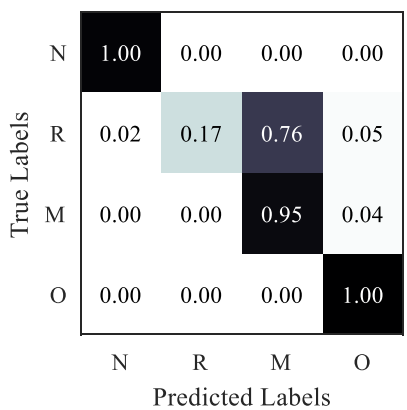


Figure 6-11 Confusion matrix of the equally labeled mini-batch method for an imbalanced data set (1, 0.001, 0.01, 0.1) at epoch 50.

6.2.1 Equally Labeled Mini-batch Method using Oversampling

In this study, the training is split into two phases; training and re-training, as shown in Figure 6-12. This proposed method can be considered a hybrid of the data-level and algorithm-level approaches, as both approaches are used. In the first phase, the weight of the CNN is trained by using equally labeled mini-batches. As the number of the minority classes are less than the majority class, the minority samples are oversampled to generate mini-batches. The result of the training phase I example is shown in Figure 6-11. As stated above, the result shows that the health states of a 0.001 imbalanced ratio had the lowest prediction accuracy. In other words, the network has not completed learning the features of the least minor class. Since the size of the normal state was around 5,000, the rubbing state had only 5 samples, which deterred the network from complete learning. Also, just increasing number of epochs does not help to avoid the method from learning the biased classifier due to the limited number of samples.

6.2.2 Retraining Low-sensitive Filters for Minor Class Recognition

If the performance of phase I does not satisfy the criteria, retraining part of the filters is performed in phase II. Rather than developing a complex algorithm or modifying the cost function, a simple method of retraining the CNN is performed in phase II. The first step is to analyze the sensitivity of the trained filters, as suggested in Section 5.2.2. As the overall performance is determined by the ability of the filter to learn the representation of each health state, the ability of filters is evaluated through filter sensitivity analysis. Then, filters that respond sensitively to one of the health states

are frozen, while filters that are not sensitive to any of the health states – or are improperly sensitive – are trained again. The criteria of properly sensitive filters are defined in Section 5.2.3. Note that if the number of frozen filters is too large or too small, the retraining process may not be effective. Thus, in this research, the number of frozen filters is set as 1/4 of total number of filters in a layer.

The next step is the retraining process, which also uses the equally labeled mini-batch approach to avoid the weights being updated by biased data. However, before the actual re-training, the labels of the data are temporary turned into two-classes, as shown in Figure 6-13. The least minor class is used as one class, and other classes are combined as the other class. By turning the training process into a binary classification problem, the unfrozen filters in the network can be updated to better represent the least minor class samples.

The final step is to fine-tune all of the filters with a smaller learning rate than the one used in the retraining process. The classes of training samples are turned back into the original ones since the final goal is to predict all classes. By fine-tuning the filters in the CNN layers and the weights in the fully-connected layers, the filters are updated to represent all of the classes again, as in phase I.

As the training that occurs in phase I and II can be regarded as source and target tasks, respectively, the proposed retraining scheme can be referred to as ‘weak’ transfer learning [116]. The term ‘weak’ transfer learning intends to signify that the training tasks in phase I and II are different, but closely related. Typical transfer learning uses information gained from the source task to train efficiently and helpfully for the target task. The suggested method can be viewed as instance transfer

learning in the same domain (inductive). Specifically, the information that is passed from phase I training to phase II training, is the index of frozen filters. In this case, the domains are the same, as the same training data set is used. Yet, the distributions are different, as the label information has been changed in phase II. Thus, the suggested method can be viewed as an instance and inductive transfer.

Note that the unfrozen filters and weights of the fully connected layers are initialized before the retraining. As noted in [117], in transfer learning, the generalization ability can be increased by initializing the network before training. Therefore, the initialization step precedes the actual training in phase II.

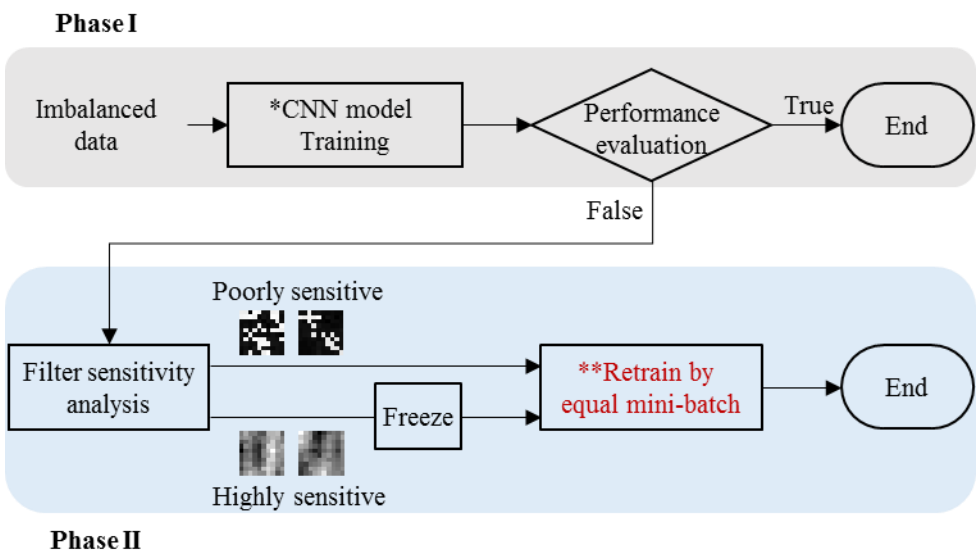


Figure 6-12 Two-phase training scheme for the minor class in an imbalanced data set.

6.3 Results of the Proposed Minor Class Retraining Scheme

The results of proposed retraining scheme are shown in this section. Among many cases of various imbalanced ratios, six different cases are examined in this study. Considering the fluid-film bearing rotor kit data set described in Section 5.3, the imbalanced ratio of the least minor state was set as 0.001 the size of normal state, and the number of the least minor state was fixed to one. To make the problem more diverse, the imbalanced ratios of the other two anomaly states were set as 0.1 and 0.01, respectively. Thus, all states had a different number of samples. The six cases of the imbalanced data set are presented in Table 6-1.

The CNN structures used in this section had four different combinations of hyper-parameters, as presented in Table 6-2. The 1-layer and 2-layer networks with 16 filters were used in this study, as in Chapter 5. A layer includes three functional sub-layers: a convolutional layer, a pooling layer, and an activation function. Max pooling and the rectified linear unit (RELU) method were used for the pooling layer and the activation function. The two fully connected layers are attached at the end, which has 16 and 4 nodes, respectively. The Adam optimizer with a learning rate of 0.001 was used [118]. Note that a batch size of 100 was added for 1-layer and 2-layer cases, which makes four combinations in total.

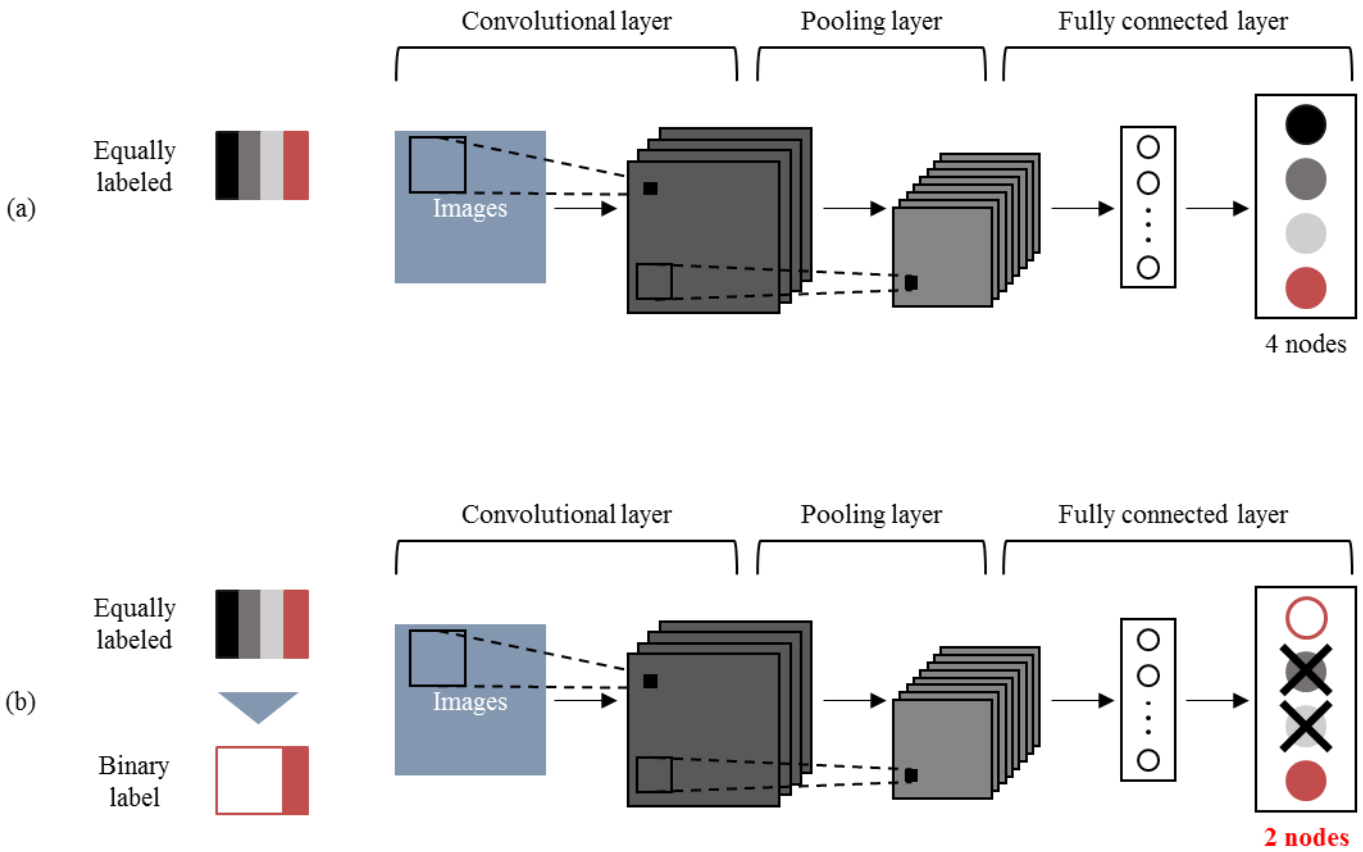


Figure 6-13 Training of the CNN in (a) phase I and (b) phase II

Table 6-1 Case description of various imbalanced ratios of health states.

	Relative ratio of data size per each health state (normal, rubbing, misalignment, oil whirl)
Case 1	(1, 0.1, 0.01, 0.001)
Case 2	(1, 0.01, 0.1, 0.001)
Case 3	(1, 0.1, 0.001, 0.01)
Case 4	(1, 0.01, 0.001, 0.1)
Case 5	(1, 0.001, 0.1, 0.01)
Case 6	(1, 0.001, 0.01, 0.1)

Table 6-2 Hyper-parameters of the convolutional neural network.

Number of convolutional layers	1		2	
Number of filters	Layer 1: 16		Layer 1: 16 Layer 2: 16	
Batch size	12	100	12	100
Fully connected (FC) layer	FC layer 1: 16 nodes FC layer 2: 4 nodes			
Learning rate	0.001			
Activation function	ReLU			
Pooling function	Max pooling			
Dropout ratio	0.3			
Optimizer	Adam			

6.3.1 Overall Performance of the Proposed Method for Retraining the Minor Class Scheme

The results of the proposed retraining scheme for the four different CNN structures is shown in Figure 6-14 to Figure 6-17. Specifically, each sub-figure represents the overall prediction accuracy of the six imbalanced ratio combinations described in Table 6-1, respectively. The x-axis indicates the training epochs; the green line represents the conventional random mini-batch method. The red and yellow lines represent the results of phase I and phase II, respectively.

All cases illustrate that the equally labeled mini-batch method, which is the result of phase I training, improves the overall performance. For mini-batch size 12 with a 1-layer CNN (shown in Figure 6-14), the performance increase varied from 0.03 to 0.2. The oversampled, equally-labeled mini-batch methods showed about 0.8 accuracy, and the random mini-batch method varied from 0.6 to 0.78, which caused the variations. Thus, the equally-labeled mini-batch method assures consistent results regardless of the data set, and performance improvements can be achieved, as compared to the conventional method. In addition, the result of phase II denotes a slight increase compared to that of phase I. The improvement varied from 0.01 to 0.04, which is a relatively smaller improvement than that observed for the equal mini-batch method. Although the increase of accuracy is not substantial, the trend was verified in all of the cases. The limited increase is due to the lack of the minority class data samples, as learning of the generalized features of such minority classes depends heavily on the distribution of the data samples. If the distribution of such minority samples is bounded in a small region of that class, the training process may end up with over-fitted or incorrect results.

The batch size 100 case shown in Figure 6-15 shows similar performance results to that of batch size 12. The equally labeled mini-batch method in phase I of the retraining scheme showed a 0.05 to 0.3 increase in accuracy, as compared to the conventional random mini-batch methods. Some random cases failed to learn the features of all class, as the overall accuracy stopped at 0.5. This is because the number of iterations is less than that of the batch size 12 cases. The cases shown in Figure 6-14 (a) and (d) indicate that if training continues for more epochs for batch size 100 cases, the overall prediction accuracy may start to increase. Generally, this is another proof that large mini-batches are not favorable for learning representation features [119]. In addition, the results in the sub-figures indicate that the performance of phase II retraining increase the performance by 0.02 to 0.07.

Next, the 2-layer CNNs with batch sizes of 12 and 100 are shown in Figure 6-16 and Figure 6-17. All the random mini-batch cases reached a prediction accuracy of 0.75, which can be interpreted as indicating that the least minority health state was incorrectly diagnosed. Compared to the 1-layer CNN cases, the conventional random mini-batch method appears to have learned the features of other health states than the least minority one. This shows that as the structure gets deeper, the performance can also be enhanced. Yet, even as the number of layers continually increases until 4-layers, the performance shows almost the same results as observed in the 2-layer CNN. In other words, for the four-class, fluid-film bearing rotor system data sets, the 2-layer CNN is enough to learn generalized features. In addition, the retraining results also showed a larger increase of prediction accuracy, as compared to the 1-layer CNN cases. For a batch size of 12, the average increase of the 2-layer CNN in the six cases was 0.0358, while the average of the 1-layer CNN in the six cases was

0.019. For the batch size of 100, the average increase of the 2-layer CNN was 0.0476, while the average of the 1-layer CNN was 0.0345. For most of the 2-layer cases, the enhanced performance of retraining can be seen in the figures as well, which was not clearly shown for the 1-layer CNN cases.

In short, the retraining scheme has increased the overall prediction accuracy for the testing data set, as shown in Figure 6-18. The oversampled, equally labeled mini-batch method used in phase I clearly increased the performance. In addition, the retraining of the least minority class via the temporal binary problem also showed a slight increase, which may have not been possible through the use of the conventional training scheme or the equally labeled mini-batch method. To validate which class contributed to the increase of overall performance, the prediction accuracy of the least minority class will be analyzed for every epoch in the next section.

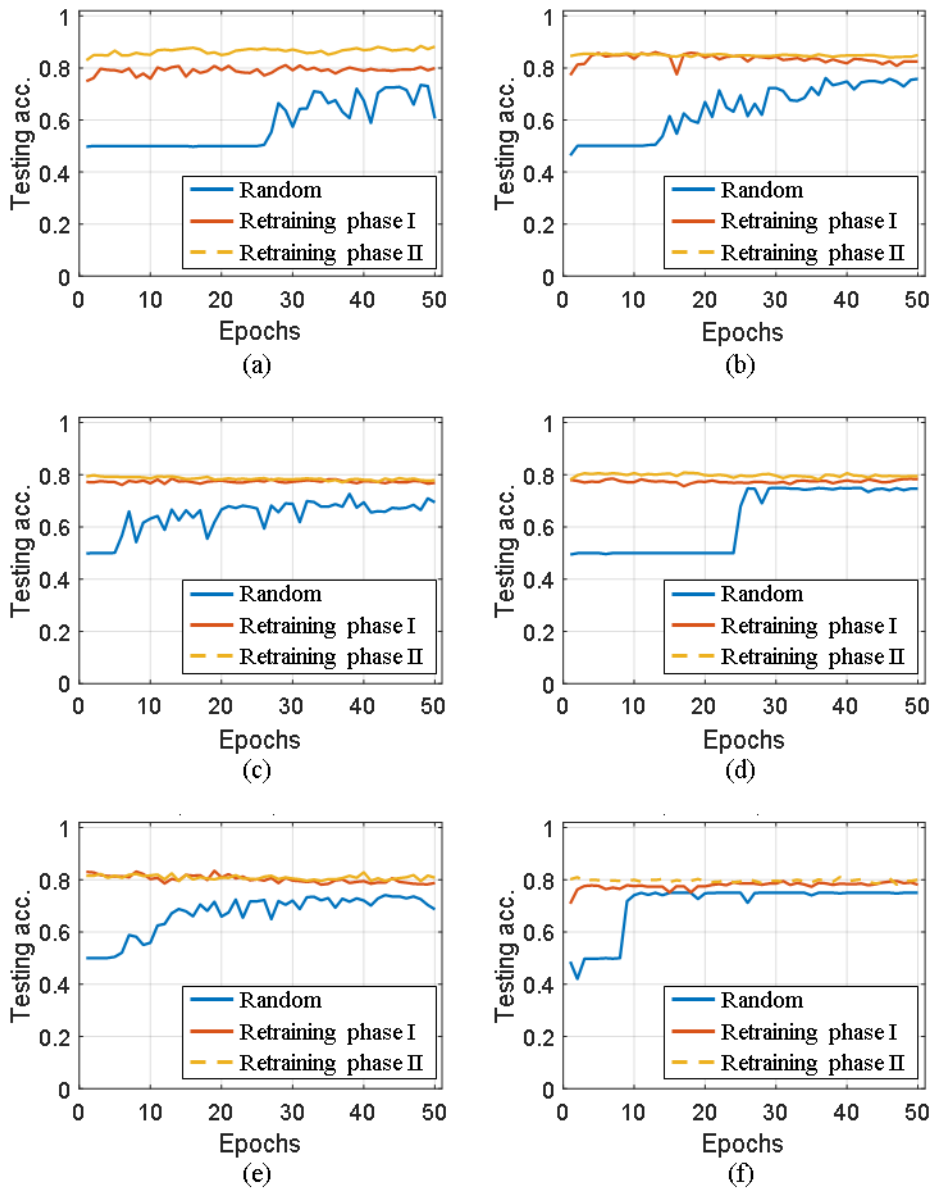


Figure 6-14 Overall prediction accuracy for the testing data set using 1-layer CNN with a batch size of 12: (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, and (f) case 6.

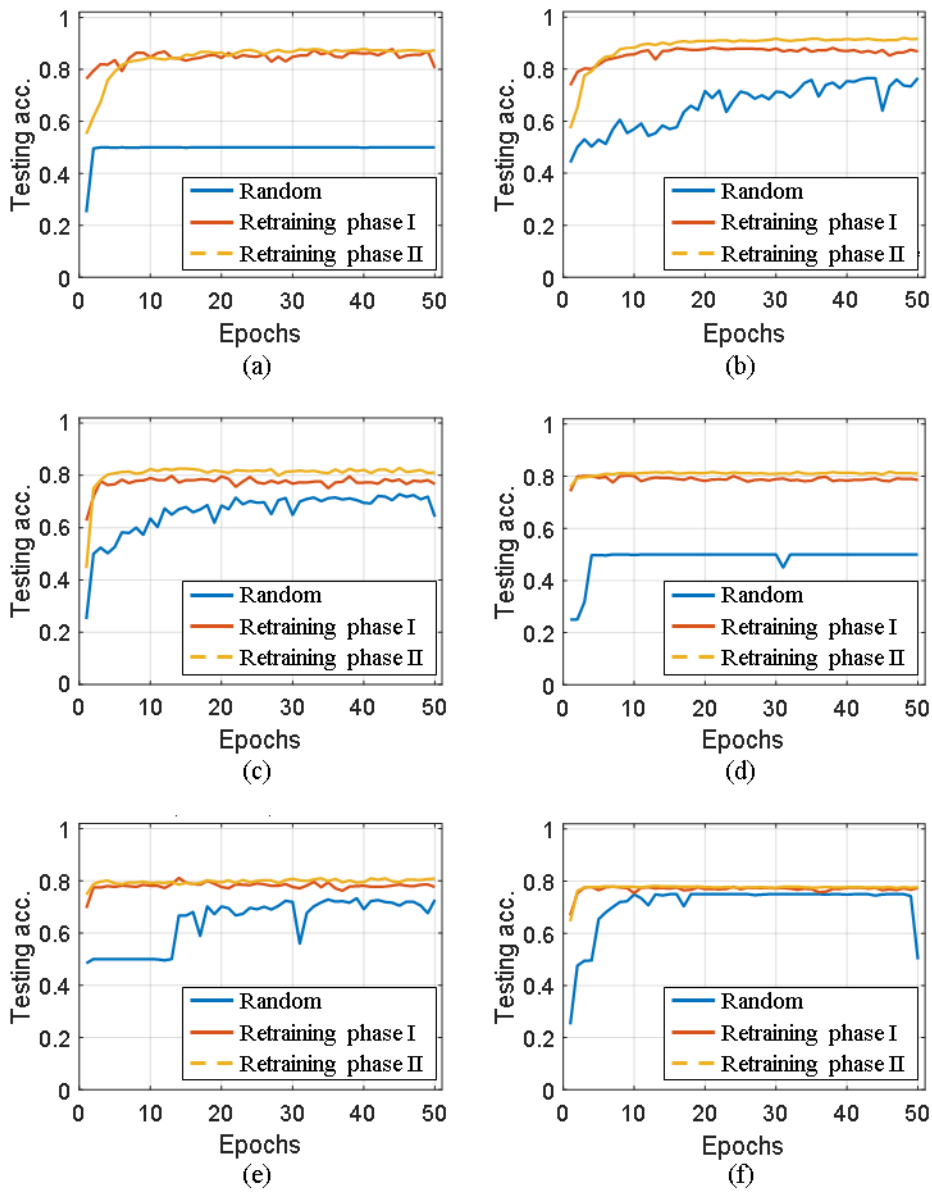


Figure 6-15 Overall prediction accuracy for the testing data set using 1-layer CNN with a batch size of 100: (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, and (f) case 6.

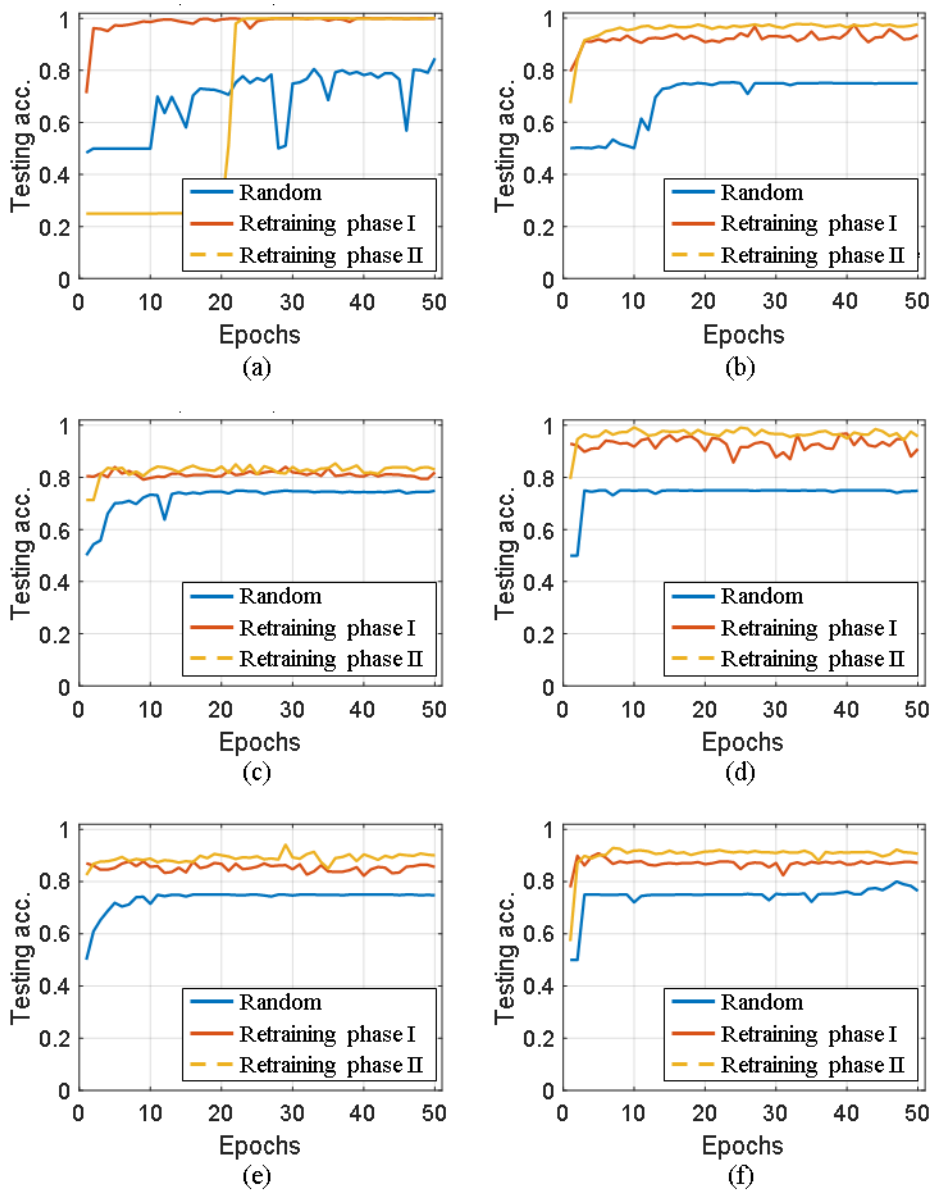


Figure 6-16 Overall prediction accuracy for the testing data set using 2-layer CNN with a batch size of 12: (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, and (f) case 6.

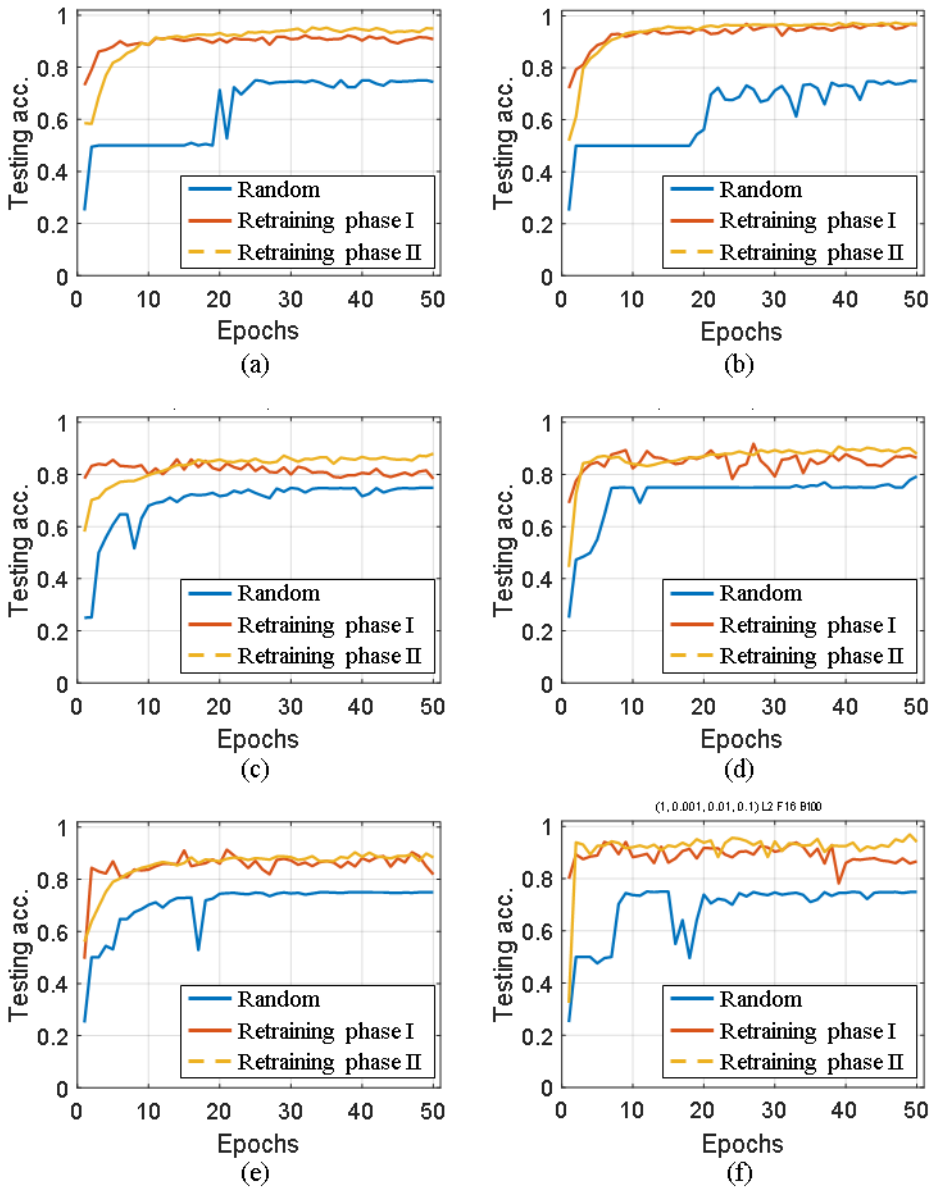


Figure 6-17 Overall prediction accuracy for the testing data set using 2-layer CNN with a batch size of 100: (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, and (f) case 6.

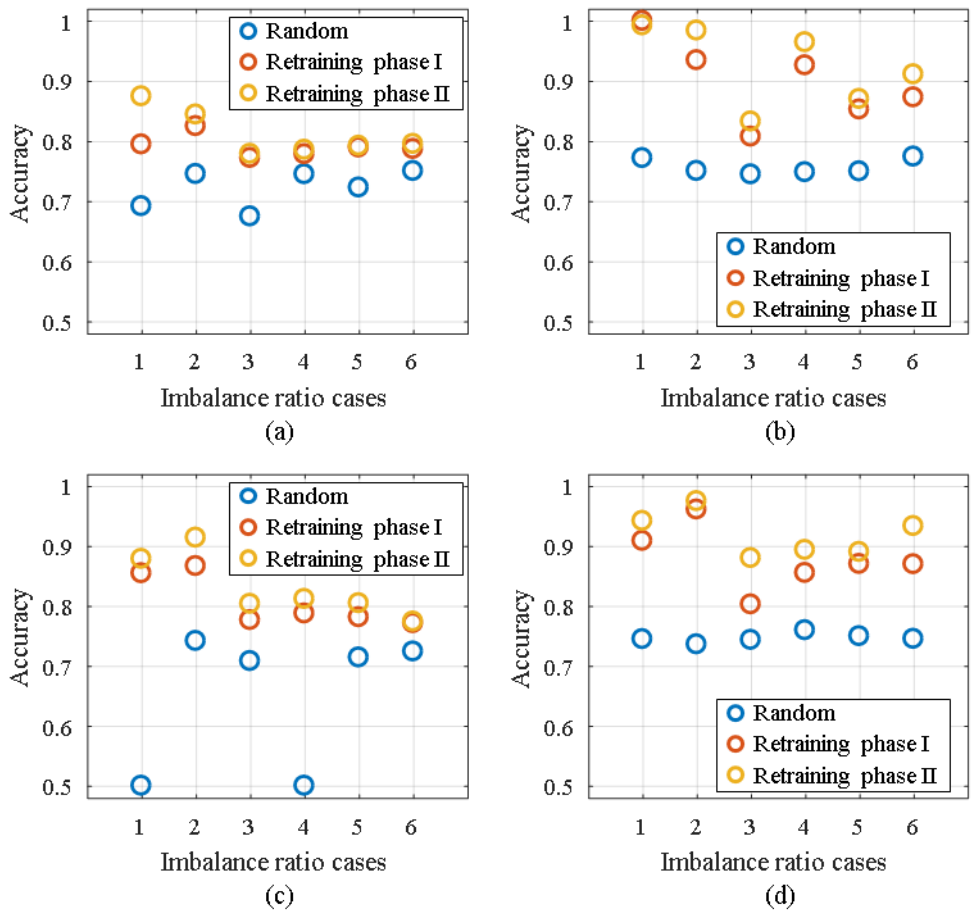


Figure 6-18 Average of the last 10 epochs' prediction accuracy for the testing data set using (a) 1-layer CNN with a batch size of 12, (2) 1-layer CNN with a batch size of 100, (c) 2-layer CNN with a batch size of 12, and (d) 2-layer CNN with a batch size of 100.

6.3.2 Performance of Minor Class Prediction Accuracy

The difference of overall prediction accuracy between the random mini-batch and the equally labeled mini-batch methods is the result of the least minor class prediction accuracy. As the retraining scheme increases the accuracy of the least minor class prediction accuracy, the difference in the overall performance can be explained by the difference of the minority class performance, as shown in Figure 6-19, Figure 6-20, Figure 6-21, and Figure 6-22. The results denote the prediction accuracy on the least minor class within the balanced testing data set. For example, the least minority class of cases 1 and 2 is the oil whirl state, and cases 3 and 4 have misalignment for the least minority class. Lastly, the training data set for cases 5 and 6 had the rubbing state as the least minority class.

As stated in previous section, oversampling the equally labeled mini-batch increased the overall prediction accuracy considerably, as compared to the random mini-batch method. Most of random mini-batch cases failed to learn representation for the least minor state, which resulted in 0 accuracy, regardless of the various CNN structures. Only a few cases showed 0.1 accuracy, as seen in Figure 6-19 (b), and 0.4 accuracy, as seen in Figure 6-21 (a). These results indicate that the random mini-batches, which are biased due to scarcity of the least minority class, cannot learn representations from such a small number of samples. However, by making the unbiasedly labeled mini-batch to be the equally-labeled mini-batch, the network can be trained to a certain level. In addition, results of the retraining in phase I proved that the 2-layer CNN can learn representation better than the 1-layer CNN. The 2-layer CNN structure of batch sizes 12 and 100 had better prediction accuracy than the 1-layer CNN structures of batch size 12 and 100, respectively.

The results of phase II of the least minority class prediction accuracy approach also match those of the overall class prediction accuracy. The increase of accuracy in phase II retraining varied by case for the 1-layer CNN structure, as presented in Figure 6-19 and Figure 6-20. For the batch size of 12, the first imbalanced case – 1, 0.1, 0.01, 0.001 – showed the largest increase of 0.3, whereas other cases showed less than a 0.03 increase. For the batch size of 100, all cases except one case showed more than a 0.1 increase when the retraining scheme was used. The mean of increased accuracy for the batch sizes of 12 and 100 are 0.056 and 0.105, respectively. The batch size 100 cases have relatively higher increased accuracy than the batch size 12 cases, because all of the least minority class samples are included in a mini-batch. As the number of the least minority classes are fixed to five (by the imbalanced ratio of 0.001 that is tied to the size of the normal state), all five samples can be included in a mini-batch, which has 25 samples per class. In other words, all of the least minority class samples can be used for every iteration.

Additionally, the 2-layer cases present the same trend as the 1-layer cases. Batch size 12 and batch size 100 CNNs are shown in Figure 6-21 and Figure 6-22, respectively. For most of the cases, the retraining scheme results, given by the yellow line, exceed the performance of the phase I training results, except in the imbalanced case of (1, 0.1, 0.01, 0.001) for batch size 12, which has already reached an accuracy of 1.0 in phase I training. Excluding this case, the mean of increased accuracy for batch sizes 12 and 100 are 0.144 and 0.190, respectively. The difference of 0.045 accuracy arises for the same reason as in the 1-layer CNN cases. Likewise, both of the averaged accuracies show a greater value than that observed in the 1-layer cases.

The averaged accuracy of the last 10 epochs concisely summarizes the results. As

deep layers are used, the network is more capable of learning representations. If the network has learned the representation of the minority class, the sensitivity of the filters on that class will also increase. Thus, in the next section, the sensitivity of the filters for the least minority class is examined.

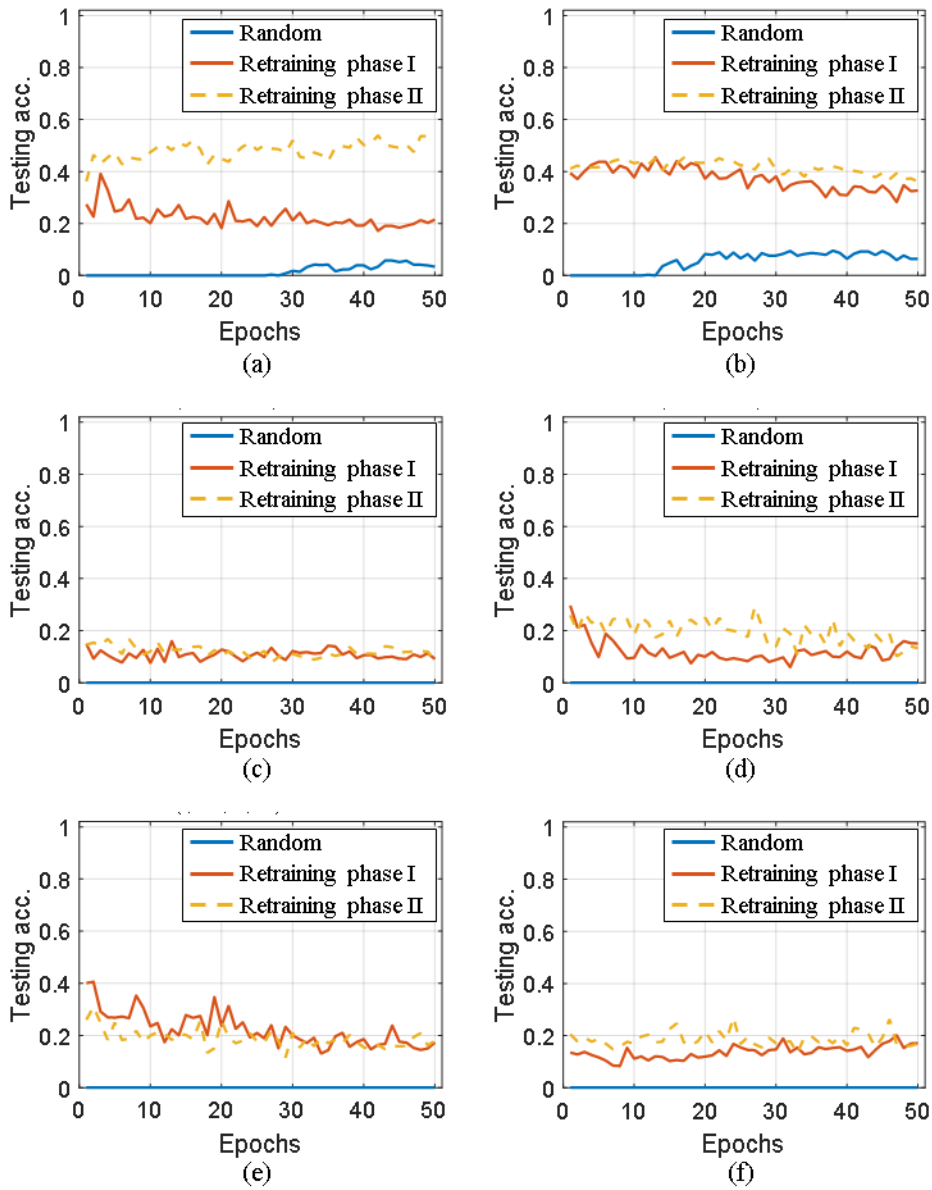


Figure 6-19 The least minority class prediction accuracy for the testing data set using 1-layer CNN with a batch size of 12: (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, and (f) case 6.

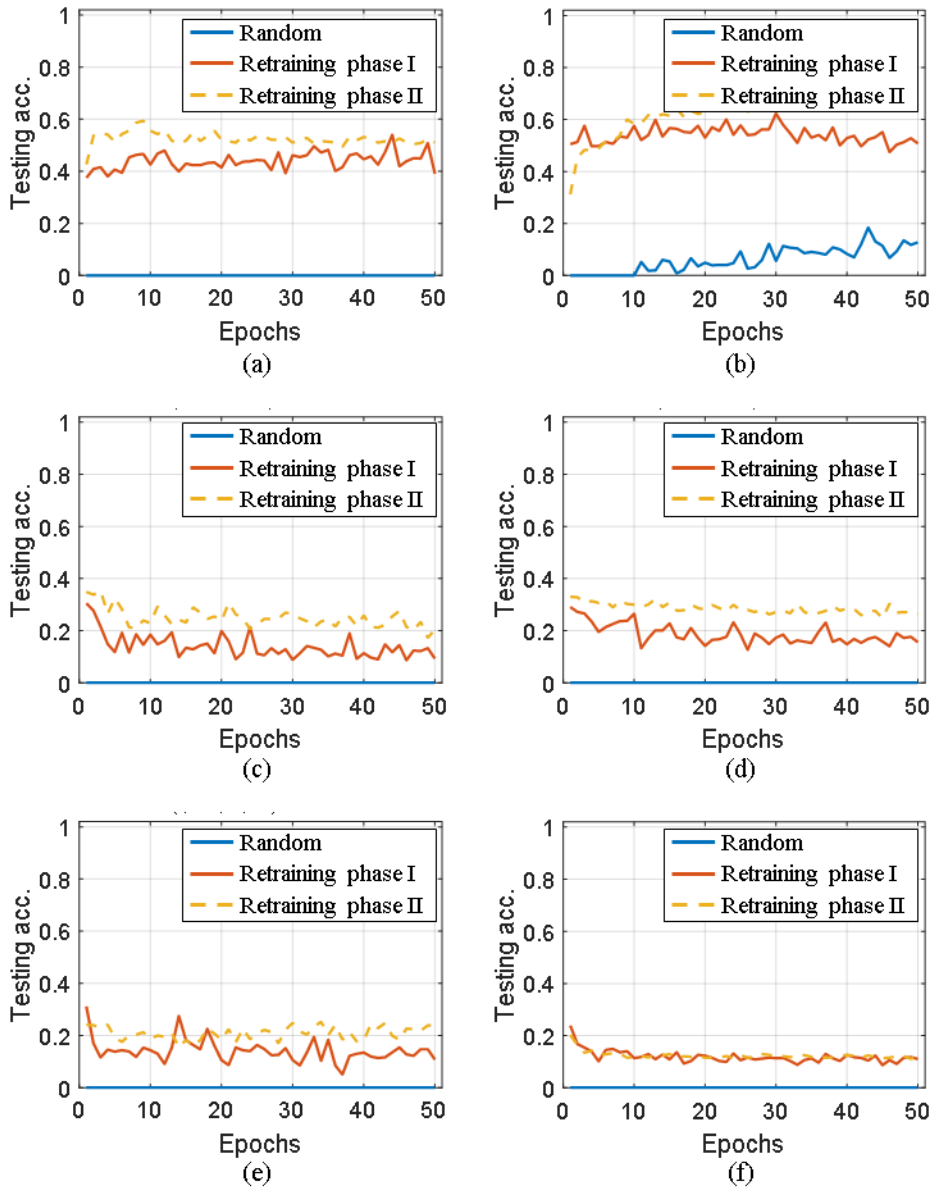


Figure 6-20 The least minority class prediction accuracy for the testing data set using 1-layer CNN with a batch size of 100: (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, and (f) case 6.

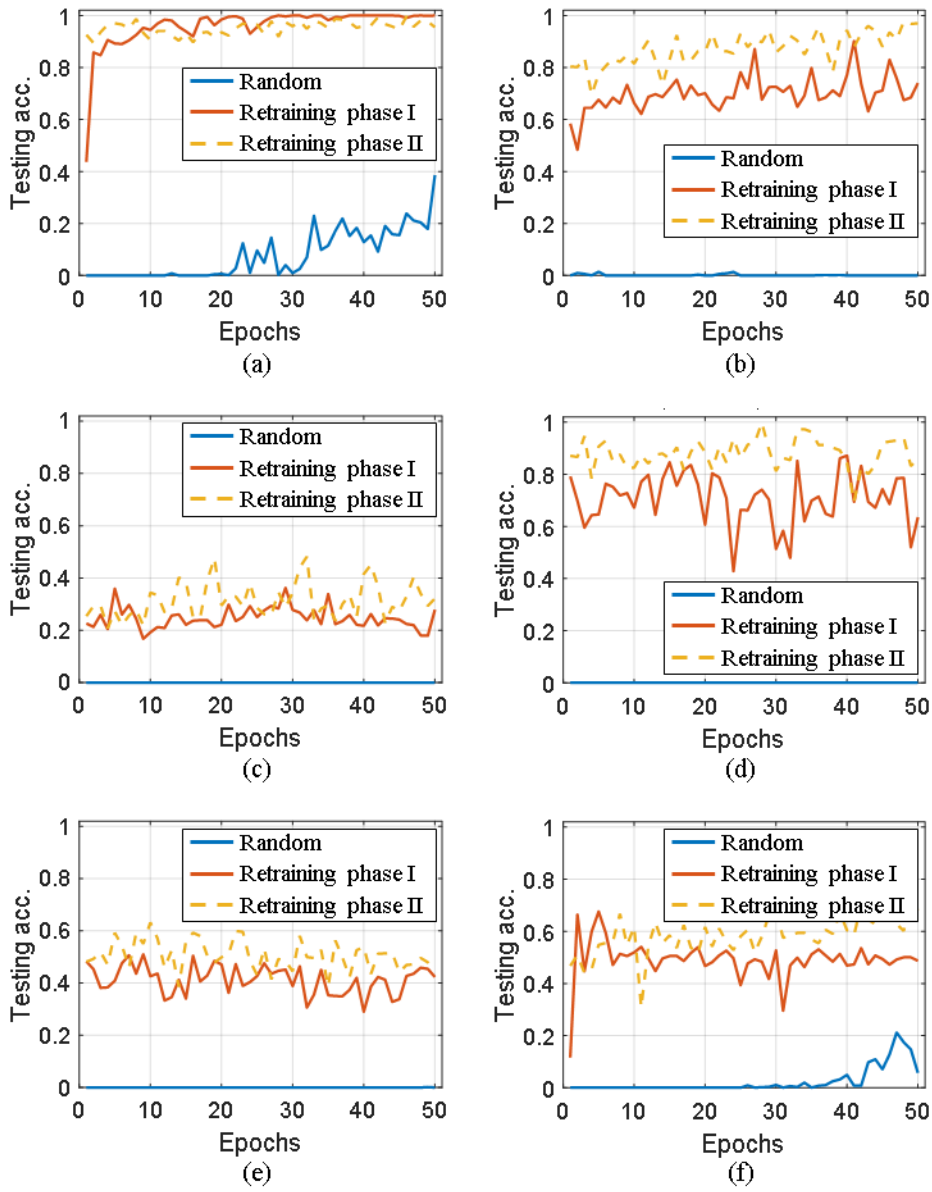


Figure 6-21 The least minority class prediction accuracy for the testing data set using 2-layer CNN with a batch size of 12: (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, and (f) case 6.

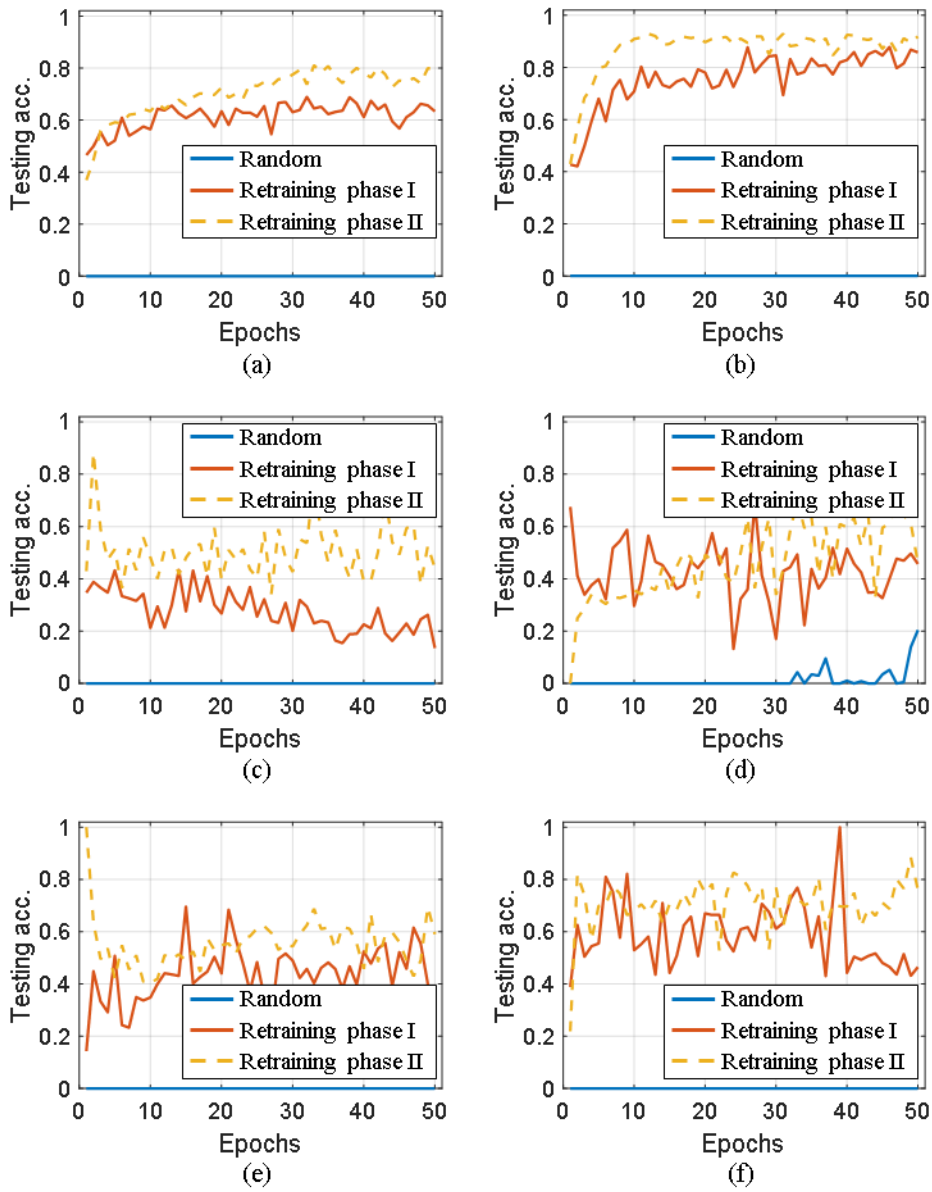


Figure 6-22 The least minority class prediction accuracy for the testing data set using 2-layer CNN with a batch size of 100: (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, and (f) case 6.

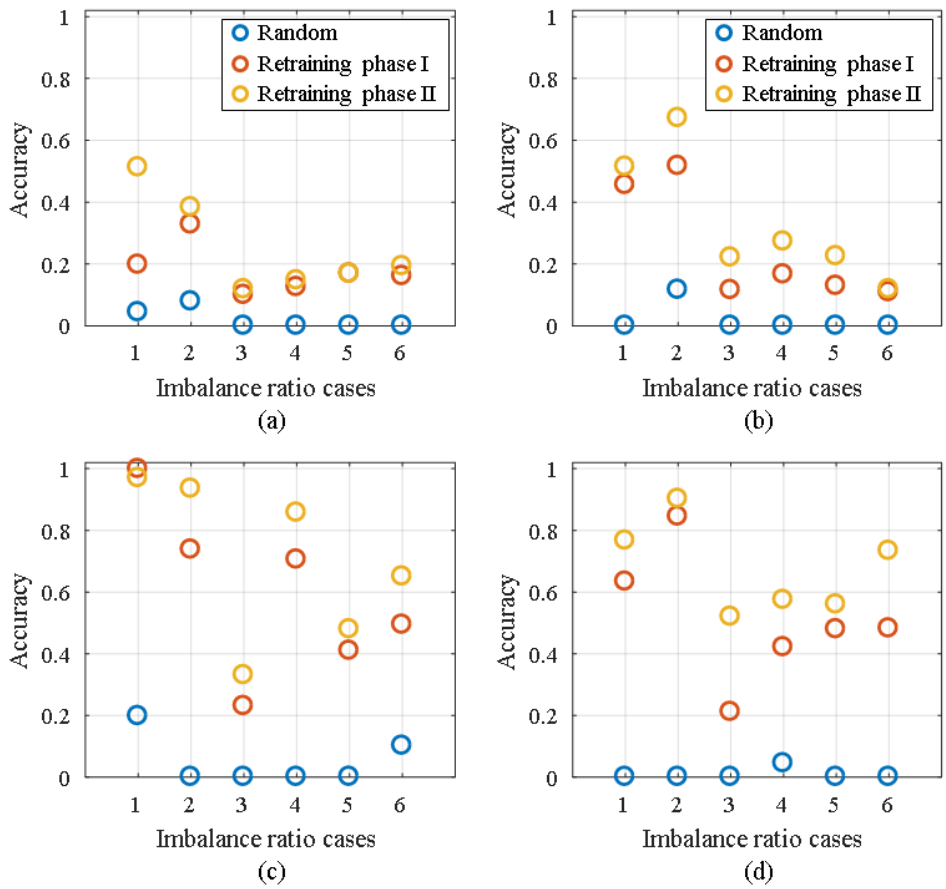


Figure 6-23 Average of the last 10 epochs' prediction accuracy of the least minority class for the testing data set using (a) 1-layer CNN with a batch size of 12, (2) 1-layer CNN with a batch size of 100, (c) 2-layer CNN with a batch size of 12, and (d) 2-layer CNN with a batch size of 100.

6.3.3 Filter Sensitivity Analysis for the Minor Class

The sensitivity for the least minority class samples can present the recognition ability of the overall network, as stated in Section 0. Generally, large portions of filters are insensitive or slightly sensitive; however, a few filters respond sensitively to one of the classes. Thus, the sum of the sensitivity values for the least minority class is analyzed in this section, as shown in Figure 6-24, Figure 6-25, Figure 6-26, and Figure 6-27.

In most of the cases, oversampling the equally labeled mini-batch resulted in a larger sum of sensitivity than observed for the random mini-batch; this is expected by the prediction accuracy of the least minority class. In addition, the retraining results of phase II had a larger sum of sensitivity than observed for the equally labeled mini-batch and the random mini-batch cases. For example, the 1-layer CNN structure of the batch size 12 case $1 - (1, 0.1, 0.01, 0.001) -$ showed a similar trend between the sum of sensitivity and the prediction accuracy for the minority class. In other words, the increased sensitivity for the least minority class resulted in the increase in overall performance. However, a few cases do not show a trend similar to the stated example. For example, the random mini-batch method results in a larger sum of sensitivity than the retraining phase I, even though the retraining phase I prediction accuracy is larger.

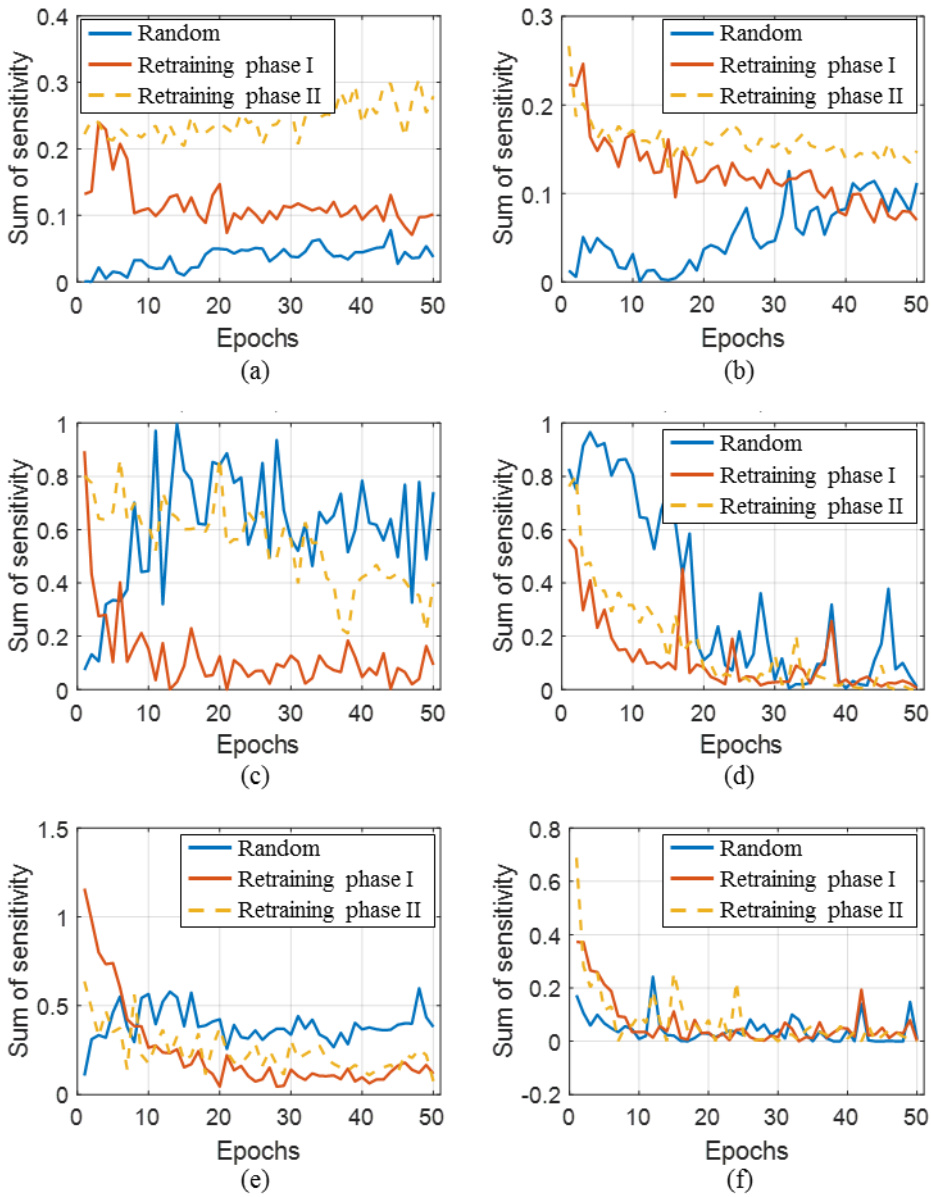


Figure 6-24 Sum of filter sensitivity for the least minority class for the testing data set using 1-layer CNN with a batch size of 12: (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, and (f) case 6.

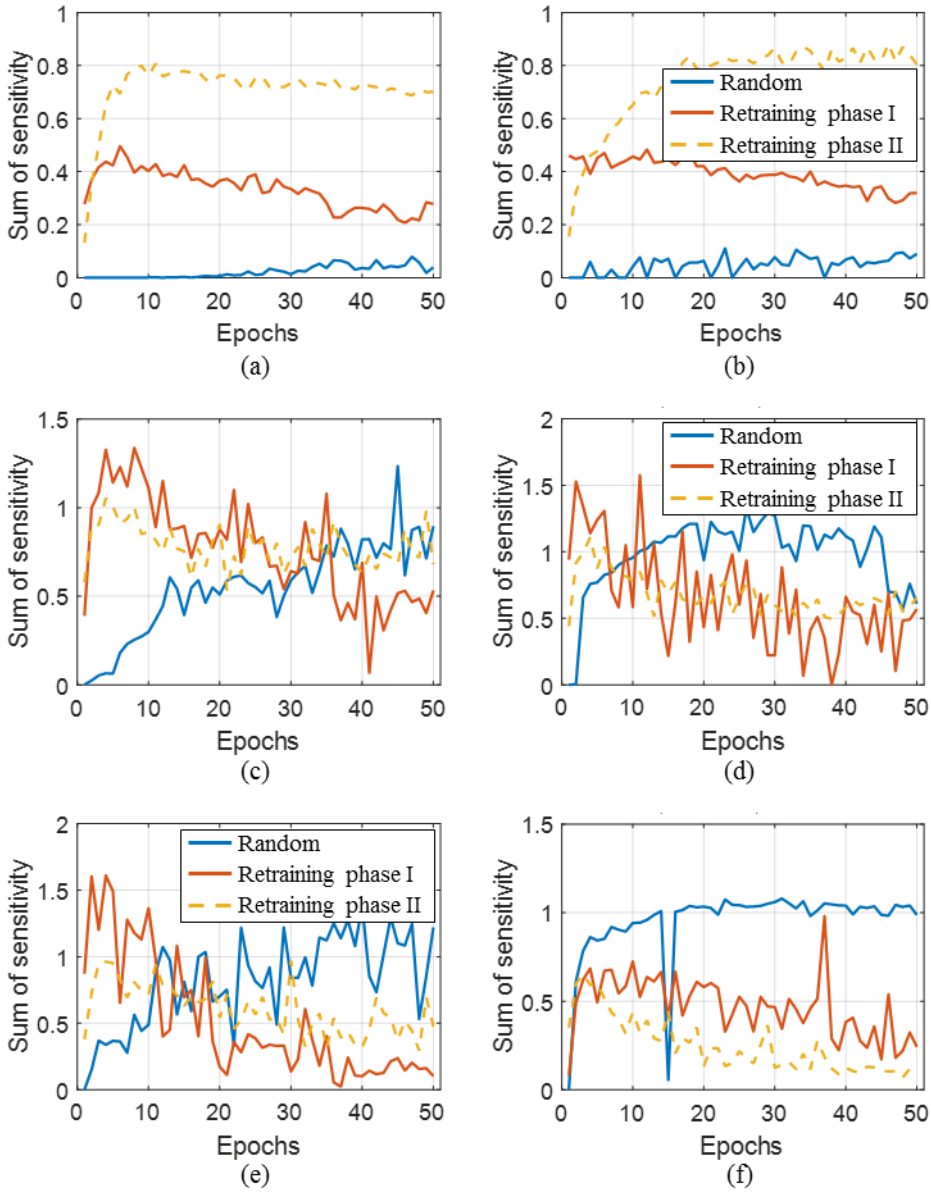


Figure 6-25 Sum of filter sensitivity for the least minority class for the testing data set using 1-layer CNN with a batch size of 100: (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, and (f) case 6.

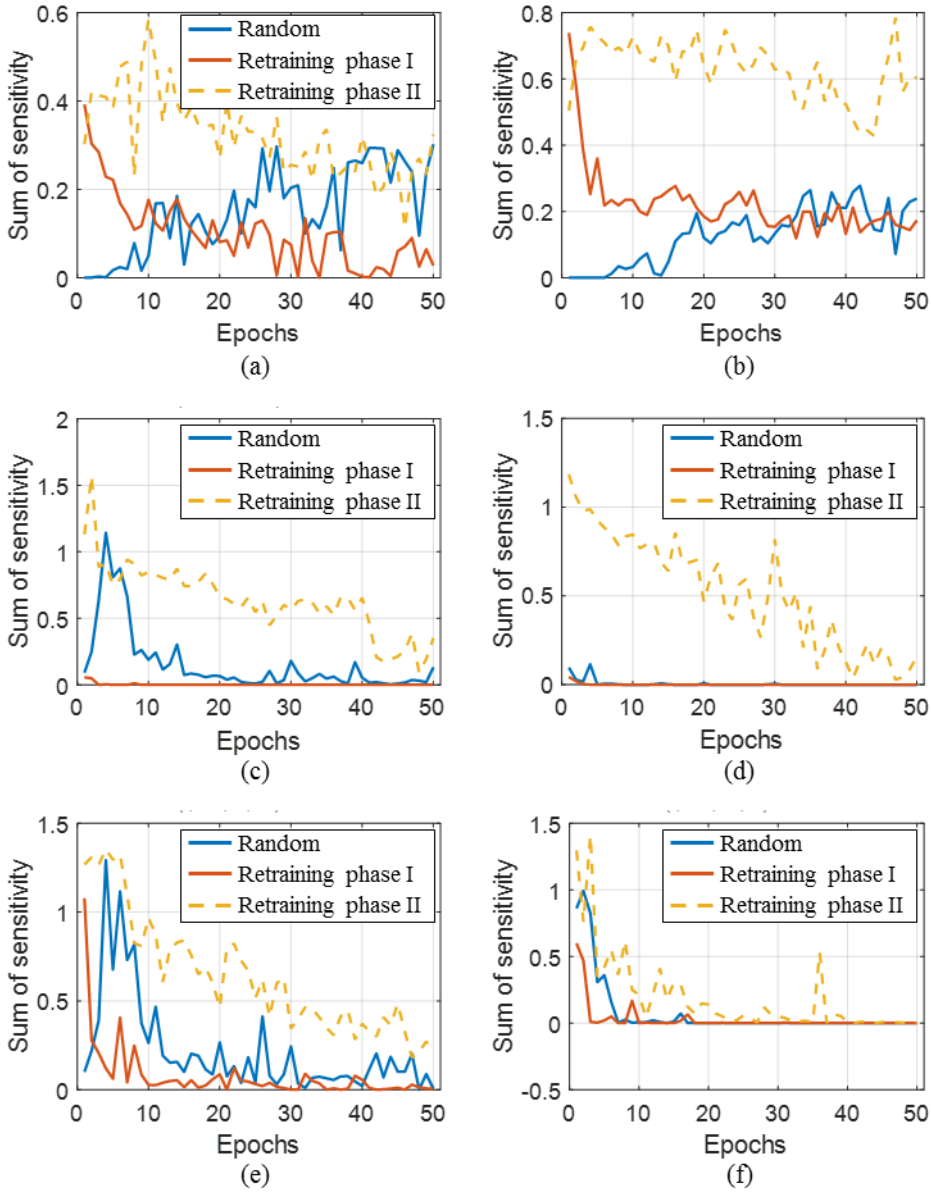


Figure 6-26 Sum of filter sensitivity for the least minority class for the testing data set using 2-layer CNN with a batch size of 12: (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, and (f) case 6.

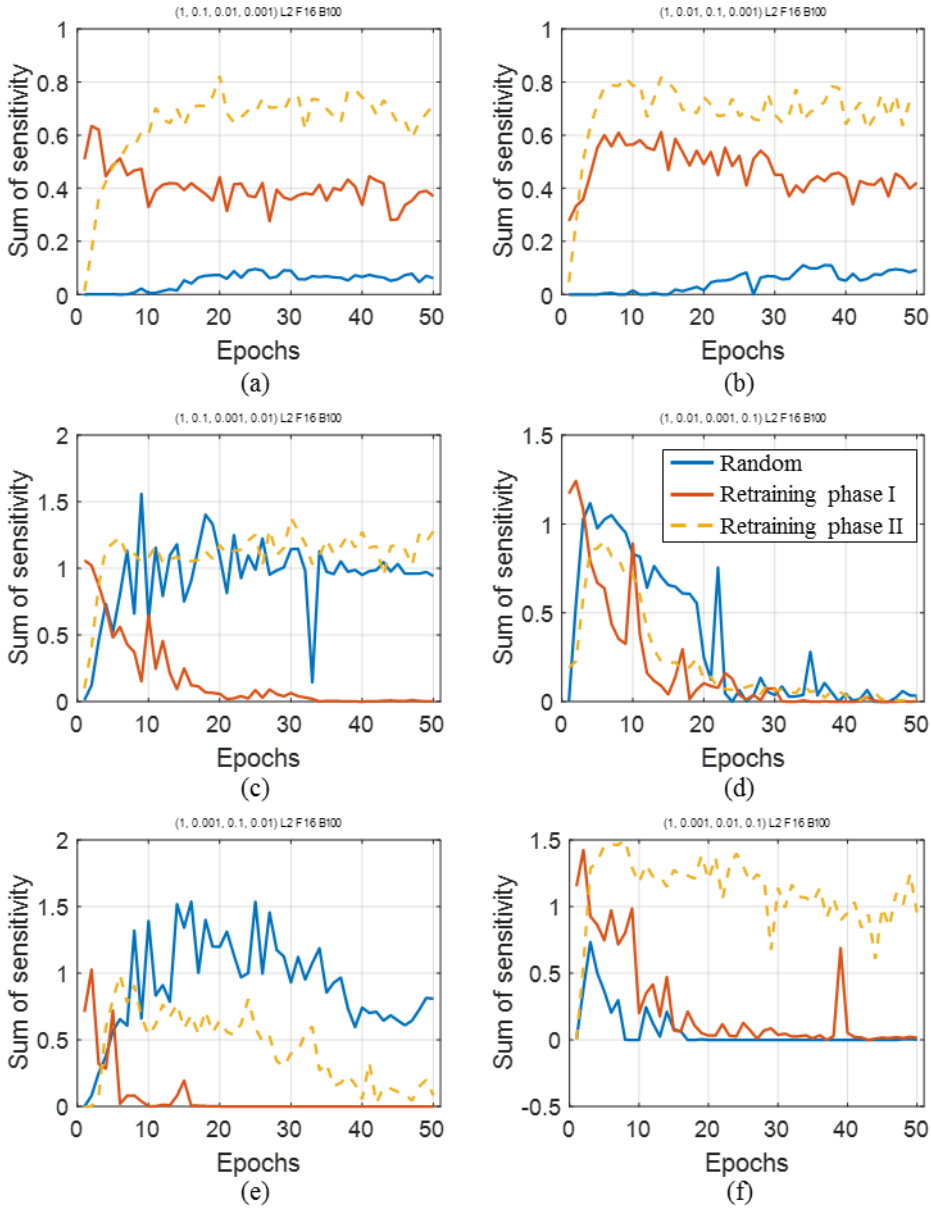


Figure 6-27 Sum of filter sensitivity for the least minority class for the testing data set using 2-layer CNN with a batch size of 100: (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, and (f) case 6.

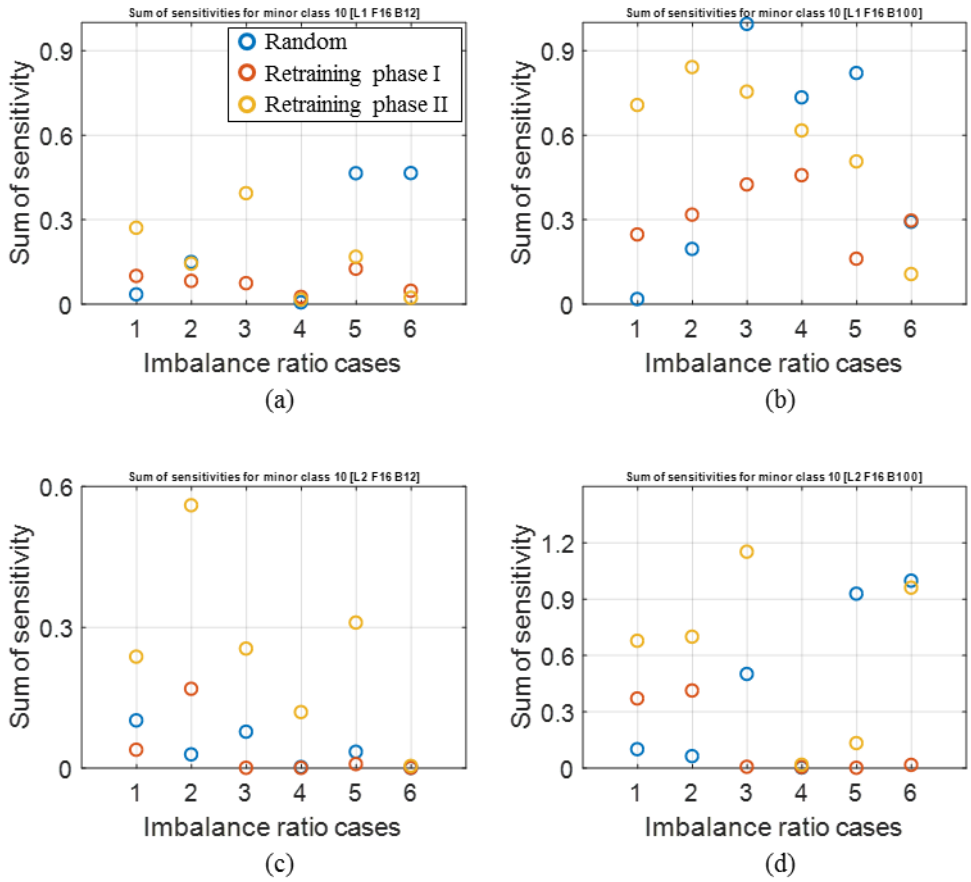


Figure 6-28 Average of the last 10 epochs' filter sensitivity of the least minority class for the testing data set using (a) 1-layer CNN with a batch size of 12, (2) 1-layer CNN with a batch size of 100, (c) 2-layer CNN with a batch size of 12, and (d) 2-layer CNN with a batch size of 100.

6.3.4 Summary and Discussion

An imbalanced data set may deteriorate the diagnosis performance considerably as the mini-batches are randomly generated. Due to the imbalanced ratio, the mini-batches are highly biased. Thus, a retraining scheme based on equally labeled mini-batches has been proposed in this section. The proposed two-phase retraining scheme boosted the overall performance. The increase in the performance of the least minority class gets greater in the deeper CNN structure. Also, if the batch size is large enough to include all of the least minority class samples, the increase of prediction accuracy is larger. This proposed approach improves the overall performance by using the given samples without the need for complex data augmentation techniques, which require careful consideration of the physics. In further research, a method for validating the augmented data can be developed to increase the minority class samples, which will eventually lead to an increase of generalization ability.

Chapter 7

Conclusions

7.1 Contributions and Significance

The research investigates the preprocessing of convolutional neural network (CNN) based diagnosis for fluid-film bearing rotor systems. The dissertation describes three research thrusts: (1) the optimal size of the vibration images, considering filter size; (2) a study of label-based, mini-batch combinations using filter sensitivity analysis; and (3) investigation of a technique for retraining minor classes in imbalanced data sets. The three thrusts of the research are expected to offer the following contributions.

Contribution 1: Suggestion of Criteria for an Optimal Vibration Image Size , Considering Size of the Filter in a CNN

This study suggests criteria for deciding the optimal size of input images with respect to the filter size of a CNN. To the best of author's knowledge, the filter size and the input image size were determined empirically in the previous studies. Thus,

considering the gray-scaled gradient vibration images, the magnitude of the image gradient was used to propose a metric for the optimal ratio of image size to filter size. This criteria can be used to determine the filter size or the image size for recognizing similar patterns of an image.

Contribution 2: Suggestion of a Label-based, Mini-batch Gradient Descent Method for Training CNNs

This dissertation proposes a label-based, mini-batch gradient descent method for training CNNs. As the large variance between the mini-batches prevents learning of a general representation of the training images, methods that reduce the variance of the mini-batches have been proposed in prior work. However, the variance can also be controlled using the suggested method. Further, the suggested label-based mini-batches can be used to tackle imbalanced problems by making the mini-batch biased towards the minor health state. Since labels provide valuable information for the diagnosis of rotor systems, the suggested mini-batch concept can be used for various purposes.

Contribution 3: Definition of the Quantified Capability of Filters in a CNN: Sensitivity Analysis of Filters

The performance of a CNN depends on the ability of filters to capture the patterns in the training data set. Thus, this dissertation suggests a method of sensitivity analysis

for CNN filters. To the best of author's knowledge, the previous studies have not considered using sensitivity of CNN filters. The suggested sensitivity analysis method can be used to evaluate whether or not each filter has been trained properly. As the number of properly sensitive filters is strongly correlated to the overall performance, the sensitivity criteria can be used to infer the performance of the trained network. Also, insensitive or erroneously sensitive filters can be distinguished; these can then be modified to improve the trained network.

Contribution 4: Suggestion of a Retraining Scheme for Minor Class Recognition in Imbalanced Data Sets

This dissertation suggests a novel method to address imbalanced data set problems. In the suggested approach, an equally labeled mini-batch is generated by oversampling if the minority class samples are not enough. Previous studies have not considered label information in generating mini-batches from imbalanced data sets. The proposed method is effective for diagnosis of the minority class samples, which frequently happens in large machinery systems. In addition, the retraining phase improves the prediction accuracy of the minority class by simply formulating the training labels. Although the amount of improved accuracy is not great, the clear trend of improvement shows the validity of the suggested method. This hierarchical approach may be applied effectively to other imbalanced data sets as well.

7.2 Suggestions for Future Research

This study offers various essential preprocessing steps for CNN-based diagnosis of fluid-film bearing rotor systems. This work focuses on deciding the optimal size of the vibration image with respect to the filter size. Label-based, mini-batch methods are analyzed using the proposed filter sensitivity method, which can be used to evaluate the network performance. Lastly, a novel two-phase training scheme is proposed to improve learning features for the minority classes. Although the three proposed methods clearly improve the overall performance, a few other topics can be explored in future work to further improve the performance.

Suggestion 1: Determining the criteria for the optimal size ratio by considering magnitude and direction of the image gradient

The criteria for deciding the optimal ratio of image size to filter size can reduce the cost of finding the hyper-parameter of a CNN. The criteria proposed in this research uses the magnitude of the image gradients; the popular histogram of oriented gradients (HOG) method uses the direction of the image gradients. Since the proposed method is suitable for gray-scaled gradient images, a new criteria may be required for other kinds of images. For example, a new criteria using both the magnitude and the direction can be investigated for use with images in which directions are important.

Suggestion 2: More ‘sensitive’ sensitivity for filters in deeper layers

In this study, a method was proposed to analyze the sensitivity of the CNN filters. As the analysis was targeted to the filters in the first layer, further studies can be conducted to analyze the filters in the second or deeper layers. In deeper layers of a CNN, the input images are translated into features that may not be interpretable. Also the magnitude of sensitivity will be relatively minor compared to that of the first layer. Thus, more 'sensitive' sensitivity criteria may be required to properly analyze the trained network. In addition, correlation of filters in the deep layers may also be considered to develop a robust sensitivity method.

Suggestion 3: Data augmentation for imbalanced data set problems

The proposed method that enhanced the imbalanced data set problem is based on a hybrid approach that combines data-level and algorithm-level approaches. Recently, in vision recognition research, novel techniques have been developed for data augmentation. As both a large quantity and good quality are keys to addressing the data imbalanced problem, various data augmentation approaches have been proposed. For example, a generative adversarial network (GAN) can generate images of the minority health state, which are analogous to the original ones [120]. Also, variational autoencoders (VAE) can generate images by training the manifold of the minority health state [121, 122]. These data-level approaches for generating additional images can surely increase the size of the minority health state. However, it must be confirmed that the generated images represent the health state of the rotor system, which is another topic for future research. The quality problem of the data set will always exist in these types of generative techniques.

References

[1] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213-237, 2019.

[2] H. Oh, J. H. Jung, B. C. Jeon, and B. D. Youn, "Scalable and Unsupervised Feature Engineering Using Vibration-Imaging and Deep Learning for Rotor System Diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3539-3549, 2018.

[3] C. Cecati, "A survey of fault diagnosis and fault-tolerant techniques—Part II: Fault diagnosis with knowledge-based and hybrid/active approaches," *IEEE Transactions on Industrial Electronics*, 2015.

[4] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757-3767, 2015.

[5] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6418-6428, 2014.

[6] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: An overview," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 657-667, 2015.

[7] F. Immovilli, C. Bianchini, M. Cocconcelli, A. Bellini, and R. Rubini, "Bearing Fault Model for Induction Motor With Externally Induced Vibration," *IEEE Transactions on Industrial Electronics,* vol. 60, no. 8, pp. 3408-3418, 2013.

[8] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation,* vol. 18, no. 7, pp. 1527-1554, 2006.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature,* vol. 521, no. 7553, p. 436, 2015.

[10] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks,* vol. 61, pp. 85-117, 2015.

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

[12] J. H. Jung, B. C. Jeon, B. D. Youn, M. Kim, D. Kim, and Y. Kim, "Omnidirectional regeneration (ODR) of proximity sensor signals for robust diagnosis of journal bearing systems," *Mechanical Systems and Signal Processing,* vol. 90, pp. 189-207, 2017.

[13] 전병철, "Unsupervised Learning and Diagnosis Method for Journal Bearing System in a Large-scale Power Plant," 서울대학교 대학원, 2016.

[14] F. Y. Zeidan and B. S. Herbage, "Fluid Film Bearing Fundamentals and Failure Analysis," in *Proceedings of the Twentieth Turbomachinery*

Symposium, 1991.

[15] M. He, C. H. Cloud, and J. M. Byrne, "Fundamentals of Fluid Film Journal Bearing Operation and Modeling," in *Proceedings of the Thirty-Fourth Turbomachinery Symposium*, pp. 155-175, 2005.

[16] K. Shiraki and H. Kanki, "A new vibration criteria for high speed/large capacity turbo machinery," in *Proceedings of Eighth Turbomachinery Symposium*, pp. 59-70, 1979.

[17] D. H. Diamond and P. S. Heyns, "A novel method for the design of proximity sensor configuration for rotor blade tip timing," *Journal of Vibration and Acoustics*, vol. 140, no. 6, p. 061003, 2018.

[18] T. H. Patel and A. K. Darpe, "Experimental investigations on vibration response of misaligned rotors," *Mechanical Systems and Signal Processing*, vol. 23, no. 7, pp. 2236-2252, 2009.

[19] J. M. Ha, H. Oh, J. Park, and B. D. Youn, "Classification of operating conditions of wind turbines for a class-wise condition monitoring strategy," *Renewable energy*, vol. 103, pp. 594-605, 2017.

[20] A. Muszynska, *Rotordynamics*. CRC press, 2005.

[21] W. Hongjun, W. Hongfeng, and J. Yongjian, "Orbit identification method based on ISOMAP for rotor system fault diagnosis," in *Electronic Measurement & Instruments (ICEMI), 2013 IEEE 11th International Conference on*, vol. 2, pp. 668-671, 2013.

[22] C. Wang, J. Zhou, P. Kou, Z. Luo, and Y. Zhang, "Identification of shaft orbit for hydraulic generator unit using chain code and probability neural network," *Applied Soft Computing,* vol. 12, no. 1, pp. 423-429, 2012.

[23] C. Yan, H. Zhang, H. Li, Y. Li, and W. Huang, "Automatic identification of shaft orbits for steam turbine generator sets," in *WRI Global Congress on Intelligent Systems, 2009. GCIS '09.* , vol. 4, pp. 53-57, 2009.

[24] H. Wang, H. Wang, and Y. Ji, "Orbit identification method based on ISOMAP for rotor system fault diagnosis," in *2013 IEEE 11th International Conference on Electronic Measurement & Instruments (ICEMI)*, vol. 2, pp. 668-671, 2013.

[25] N. Bachschmid, P. Pennacchi, and A. Vania, "Diagnostic significance of orbit shape analysis and its application to improve machine fault detection," *Journal of the Brazilian Society of Mechanical Sciences and Engineering,* vol. 26, pp. 200-208, 2004.

[26] F. Bo, Z. Jian-zhong, C. Wen-qing, and Y. Bing-Hui, "Identification of the shaft orbits for turbine rotor by modified Fourier descriptors," in *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, vol. 2, pp. 1162-1167 vol.2, 2004.

[27] C. YAN, H. ZHANG, and L. WU, "Automatic Recognition of Orbit Shape for Fault Diagnosis in Steam Turbine Generator Sets," *Journal of Computational Information Systems,* vol. 6, no. 6, pp. 1995-2008, 2010.

[28] W. Fengqi and G. Meng, "Compound rub malfunctions feature extraction based on full-spectrum cascade analysis and SVM," *Mechanical Systems and Signal Processing,* vol. 20, no. 8, pp. 2007-2021, 2006.

[29] X. Zhao, T. H. Patel, and M. J. Zuo, "Multivariate EMD and full spectrum based condition monitoring for rotating machinery," *Mechanical Systems and Signal Processing,* vol. 27, no. 0, pp. 712-728, 2012.

[30] P. Goldman and A. Muszynska, "Application of full spectrum to rotating machinery diagnostics," *Orbit,* vol. 20, no. 1, pp. 17-21, 1999.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097-1105, 2012.

[32] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725-1732, 2014.

[33] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5325-5334, 2015.

[34] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, no. 1, pp. 221-231, 2013.

[35] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning*, pp. 807-814, 2010.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016.

[38] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1-9, 2015.

[39] X. Guo, L. Chen, and C. Shen, "Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis," *Measurement*, vol. 93, pp. 490-502, 2016.

[40] O. Janssens *et al.*, "Convolutional Neural Network Based Fault Detection for Rotating Machinery," *Journal of Sound and Vibration*, vol. 377, pp. 331-345, 2016.

[41] Z. Chen, C. Li, and R.-V. Sanchez, "Gearbox fault identification and classification with convolutional neural networks," *Shock and Vibration*, vol. 2015, 2015.

[42] 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, 2016.

[43] J. Wang, J. Zhuang, L. Duan, and W. Cheng, "A multi-scale convolution neural network for featureless fault diagnosis," in *2016 International Symposium on Flexible Automation (ISFA)*, pp. 65-70, 2016.

[44] J. Yan, H. Zhu, X. Yang, Y. Cao, and L. Shao, "Research on fault diagnosis of hydraulic pump using convolutional neural network," *Journal of Vibroengineering*, vol. 18, no. 8, 2016.

[45] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-Sensitive Learning of Deep Feature Representations From Imbalanced Data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1-15, 2017.

[46] R. Razavi-Far, M. Farajzadeh-Zanjani, and M. Saif, "An Integrated Class-Imbalance Learning Scheme for Diagnosing Bearing Defects in Induction Motors," *IEEE Transactions on Industrial Informatics*, 2017.

[47] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge & Data Engineering*, no. 9, pp. 1263-1284, 2008.

[48] W. W. Y. Ng, G. Zeng, J. Zhang, D. S. Yeung, and W. Pedrycz, "Dual autoencoders features for imbalance classification problem," *Pattern Recognition*, vol. 60, no. Supplement C, pp. 875-889, 2016.

[49] W. W. Y. Ng, J. Hu, D. S. Yeung, S. Yin, and F. Roli, "Diversified Sensitivity-Based Undersampling for Imbalance Classification Problems," *IEEE Transactions on Cybernetics,* vol. 45, no. 11, pp. 2402-2412, 2015.

[50] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research,* vol. 16, pp. 321-357, 2002.

[51] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," in *International Conference on Intelligent Computing*, pp. 878-887, 2005.

[52] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Pacific-Asia conference on knowledge discovery and data mining*, pp. 475-482, 2009.

[53] C. Seiffert, T. M. Khoshgoftaar, J. V. Hulse, and A. Napolitano, "RUSBoost: A Hybrid Approach to Alleviating Class Imbalance," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans,* vol. 40, no. 1, pp. 185-197, 2010.

[54] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *European conference on principles of data mining and knowledge discovery*, pp. 107-119, 2003.

[55] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera, "EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling," *Pattern Recognition,* vol. 46, no. 12, pp. 3460-3471, 2013.

[56] M. A. Tahir, J. Kittler, and F. Yan, "Inverse random under sampling for class imbalance problem and its application to multi-label classification," *Pattern Recognition,* vol. 45, no. 10, pp. 3738-3750, 2012.

[57] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter,* vol. 6, no. 1, pp. 20-29, 2004.

[58] G. Wu and E. Y. Chang, "KBA: Kernel boundary alignment considering imbalanced data distribution," *IEEE Transactions on knowledge and data engineering,* vol. 17, no. 6, pp. 786-795, 2005.

[59] B. X. Wang and N. Japkowicz, "Boosting support vector machines for imbalanced data sets," *Knowledge and information systems,* vol. 25, no. 1, pp. 1-20, 2010.

[60] Y. Zhang, P. Fu, W. Liu, and G. Chen, "Imbalanced data classification based on scaling kernel-based support vector machine," *Neural Computing and Applications,* vol. 25, no. 3-4, pp. 927-935, 2014.

[61] M. Gao, X. Hong, and C. J. Harris, "Construction of neurofuzzy models for imbalanced data classification," *IEEE Transactions on Fuzzy Systems,* vol. 22, no. 6, pp. 1472-1488, 2014.

[62] K. Li, X. Kong, Z. Lu, L. Wenyin, and J. Yin, "Boosting weighted ELM for imbalanced learning," *Neurocomputing,* vol. 128, pp. 15-21, 2014.

[63] S. Yue, "Imbalanced Malware Images Classification: a CNN based Approach," *arXiv preprint arXiv:1708.08042,* 2017.

[64] A. Sze-To and A. K. C. Wong, "A Weight-Selection Strategy on Training Deep Neural Networks for Imbalanced Classification," in *Image Analysis and Recognition: 14th International Conference, ICIAR 2017, Montreal, QC, Canada, July 5–7, 2017, Proceedings*, F. Karray, A. Campilho, and F. Cheriet, Eds. Cham: Springer International Publishing, 2017, pp. 3-10.

[65] K. Sozykin, A. M. Khan, S. Protasov, and R. Hussain, "Multi-label Class-imbalanced Action Recognition in Hockey Videos via 3D Convolutional Neural Networks," *arXiv preprint arXiv:1709.01421,* 2017.

[66] C. Zhang, K. C. Tan, and R. Ren, "Training cost-sensitive Deep Belief Networks on imbalance data problems," in *Neural Networks (IJCNN), 2016 International Joint Conference on*, pp. 4362-4367, 2016.

[67] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," in *Neural Networks (IJCNN), 2016 International Joint Conference on*, pp. 4368-4374, 2016.

[68] A. Dalyac, M. Shanahan, and J. Kelly, "Tackling class imbalance with deep convolutional neural networks," *Imperial College,* pp. 30-35, 2014.

[69] S. Guan, M. Chen, H. Y. Ha, S. C. Chen, M. L. Shyu, and C. Zhang, "Deep

Learning with MCA-based Instance Selection and Bootstrapping for Imbalanced Data Classification," in *2015 IEEE Conference on Collaboration and Internet Computing (CIC)*, pp. 288-295, 2015.

[70] Y. Yan, M. Chen, M. L. Shyu, and S. C. Chen, "Deep Learning for Imbalanced Multimedia Data Classification," in *2015 IEEE International Symposium on Multimedia (ISM)*, pp. 483-488, 2015.

[71] C. Huang, Y. Li, C. Change Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5375-5384, 2016.

[72] C. Elkan, "The foundations of cost-sensitive learning," in *International joint conference on artificial intelligence*, vol. 17, pp. 973-978, 2001.

[73] K. M. Ting, "An instance-weighting method to induce cost-sensitive trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 3, pp. 659-665, 2002.

[74] M. A. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown," in *ICML-2003 workshop on learning from imbalanced data sets II*, vol. 2, pp. 2-1, 2003.

[75] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 63-77, 2006.

[76] G. M. Weiss, "Mining with rarity: a unifying framework," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 7-19, 2004.

[77] 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, 1998.

[78] D. E. Bently, C. T. Hatch, and B. Grissom, *Fundamentals of rotating machinery diagnostics*. Bently Pressurized Bearing Press, 2002.

[79] 전병철, "Unsupervised learning and diagnosis method for journal bearing system in a large-scale power plant / 전병철," 대형 발전소 저널베어링 회전체 시스템의 비지도 학습 및 진단 기법 연구, 서울 : 서울대학교 대학원, 서울, 2016.

[80] B. O. Al-bedoor, "Transient torsional and lateral vibrations of unbalanced rotors with rotor-to-stator rubbing," *Journal of Sound and vibration*, vol. 229, no. 3, pp. 627-645, 2000.

[81] I. Standard, "Mechanical Vibration-Evaluation of Machine Vibration by Measurements on Non-Rotating Parts," *ISO/IS*, vol. 10816, 1996.

[82] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, pp. 1137-1145, 1995.

[83] Y. Bengio and Y. Grandvalet, "No unbiased estimator of the variance of k-fold cross-validation," *Journal of machine learning research*, vol. 5, no. Sep, pp. 1089-1105, 2004.

[84] H. Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 26, no. 6, pp. 508-517, 1978.

[85] X. Ding and Q. He, "Energy-fluctuated multiscale feature learning with deep convnet for intelligent spindle bearing fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 8, pp. 1926-1935, 2017.

[86] J. R. Dim and T. Takamura, "Alternative approach for satellite cloud classification: edge gradient application," *Advances in Meteorology*, vol. 2013, 2013.

[87] I. Sobel, "An isotropic 3× 3 image gradient operater," *Machine vision for three-dimensional scenes*, pp. 376-379, 1990.

[88] J. M. Prewitt, "Object enhancement and extraction," *Picture processing and Psychopictorics*, vol. 10, no. 1, pp. 15-19, 1970.

[89] L. G. Roberts, "Machine perception of three-dimensional solids," Massachusetts Institute of Technology, 1963.

[90] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886-893, 2005.

[91] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[92] C. Lu, Z. Wang, and B. Zhou, "Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification," *Advanced Engineering Informatics,* vol. 32, pp. 139-151, 2017.

[93] Y. Xie and T. Zhang, "Fault diagnosis for rotating machinery based on convolutional neural network and empirical mode decomposition," *Shock and Vibration,* vol. 2017, 2017.

[94] W. You, C. Shen, X. Guo, X. Jiang, J. Shi, and Z. Zhu, "A hybrid technique based on convolutional neural network and support vector regression for intelligent diagnosis of rotating machinery," *Advances in Mechanical Engineering,* vol. 9, no. 6, pp. 1-17, 2017.

[95] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, and D. J. Inman, "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks," *Journal of Sound and Vibration,* vol. 388, pp. 154-170, 2017.

[96] K. B. Lee, S. Cheon, and C. O. Kim, "A Convolutional Neural Network for Fault Classification and Diagnosis in Semiconductor Manufacturing Processes," *IEEE Transactions on Semiconductor Manufacturing,* vol. 30, no. 2, pp. 135-142, 2017.

[97] 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,* 2016.

[98] M. Moller, "Supervised learning on large redundant training sets," in *Proceedings of the 1992 IEEE Workshop on Neural Networks for Signal Processing II*, pp. 79-89, 1992.

[99] G. B. Orr, "Removing noise in on-line search using adaptive batch sizes," in *Advances in Neural Information Processing Systems*, pp. 232-238, 1997.

[100] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade: Second Edition*, G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9-48.

[101] R. H. Byrd, G. M. Chin, J. Nocedal, and Y. Wu, "Sample size selection in optimization methods for machine learning," *Mathematical Programming*, vol. 134, no. 1, pp. 127-155, 2012.

[102] S. De, A. Yadav, D. Jacobs, and T. Goldstein, "Automated inference with adaptive batches," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, pp. 1504--1513, 2017.

[103] P. Zhao and T. Zhang, "Stochastic optimization with importance sampling for regularized loss minimization," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1-9, 2015.

[104] C. Zhang, H. Kjellström, and S. Mandt, "Determinantal point processes for mini-batch diversification," in *33rd Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, 11 August 2017 through 15*

August 2017, 2017.

[105] F. Chollet, "Keras," ed, 2015.

[106] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[107] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223-311, 2018.

[108] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*, 2016.

[109] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of the COMPSTAT*: Springer, 2010, pp. 177-186.

[110] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Advances in Neural Information Processing Systems*, pp. 315-323, 2013.

[111] C. Wang, X. Chen, A. J. Smola, and E. P. Xing, "Variance reduction for stochastic gradient optimization," in *Advances in Neural Information Processing Systems*, pp. 181-189, 2013.

[112] M. Gan, C. Wang, and C. a. Zhu, "Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern

recognition of rolling element bearings," *Mechanical Systems and Signal Processing,* vol. 72–73, pp. 92-104, 2016.

[113] T. de Bruin, K. Verbert, and R. Babuška, "Railway track circuit fault diagnosis using recurrent neural networks," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 28, no. 3, pp. 523-533, 2017.

[114] X. Guo, C. Shen, and L. Chen, "Deep Fault Recognizer: An Integrated Model to Denoise and Extract Features for Fault Diagnosis in Rotating Machinery," *Applied Sciences,* vol. 7, no. 1, p. 41, 2017.

[115] M. Xia, T. Li, L. Xu, L. Liu, and C. W. d. Silva, "Fault Diagnosis for Rotating Machinery Using Multiple Sensors and Convolutional Neural Networks," *IEEE/ASME Transactions on Mechatronics,* vol. PP, no. 99, pp. 1-1, 2017.

[116] 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, 2010.

[117] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, pp. 3320-3328, 2014.

[118] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[119] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp

minima," in *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[120] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672-2680, 2014.

[121] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[122] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.

국문 초록

딥러닝 기반 회전기계 진단을 위한 진동신호 전처리 및 변환 연구

대형 회전체 시스템은 사고 발생시 막대한 피해를 발생시킬 수 있으므로 안전성이 최우선으로 고려된다. 따라서, 발전소의 증기터빈과 같은 대형 회전체 시스템의 안전한 작동을 위해 상태를 정확하게 진단할 수 있는 기술이 필요하다. 다양한 상태진단 방법 중 데이터기반 방법은 학계와 산업에서 많은 관심을 받고 있다. 특히 딥러닝을 적용한 상태진단 논문의 수가 최근 몇 년간 급격하게 증가하고 있고, 앞으로도 늘어날 것으로 예상된다. 딥러닝을 적용한 연구가 활발한 이유로는 복잡해지는 기계 시스템의 물리적인 이해가 점점 더 어려워져 정확한 상태를 파악하기 어려워지고 있기 때문이다. 딥러닝은 물리적인 이해가 조금 부족하더라도 데이터를 기반으로 강력한 성능을 발휘하기 때문에, 이를 활용한 연구가 많이 진행되고 있다. 따라서 본 학위논문도 딥러닝 기반 회전체의 강건한 상태진단을 위한 진동신호 전처리 및 변환 연구를 주제로 정하였다. 주로 유막베어링 회전체 시스템을 대상으로 연구를 진행하였다.

본 학위논문에서는 진동신호 전처리 및 변환 연구로 총 3 가지를 연구하였다. (1) 합성곱신경망의 필터 크기를 고려한 최적의 이미지

크기를 결정하는 연구, (2) 필터의 민감도 분석을 통한 표지기반 mini-batch 방법의 효과 분석 연구, (3) 불균형 데이터의 학습을 위한 재학습 방법 연구를 통하여 딥러닝 회전체 상태진단이 최적의 성능을 낼 수 있도록 진동신호 전처리 기법 등을 제안하였다.

첫 번째 연구는 합성곱신경망의 입력 이미지의 크기에 관한 연구이다. 유막베어링 회전체의 일부 이상상태에서는 방향성을 갖는 상태가 존재하는데, 이를 강건하게 진단하기 위해서 진동신호로부터 이미지를 생성하는 방안을 제안하였다. 제안된 진동 이미지 생성방법을 통해 다양한 크기의 이미지와 합성곱신경망의 필터의 크기를 변화해가면서 진단 정확도를 비교하였다. 본 논문에서 제안한 입력 이미지의 구배(gradient)의 크기를 활용한 RSG 값을 기준으로 1.0 인 경우 최적의 결과를 얻을 수 있음을 확인하였다.

두 번째로는 기존의 무작위로 생성하던 mini-batch 를 학습데이터의 지표를 기준으로 생성하는 방안을 제안하였다. 다양한 조합을 대상으로 진단 성능을 평가해본 결과 동등한 수의 지표를 갖는 mini-batch 가 가장 좋은 성능을 나타내었다. 즉, Mini-batch 간의 분산이 가장 적은 조합이 합성곱신경망 학습시 하나의 지표에 치우치지 않게 하여 높은 진단 정확도를 나타내었다. 각각의 진단 결과의 차이는 본 논문에서 제안한 합성곱신경망 필터의 민감도로 분석하였다.

마지막 연구로는 실제 현장에서 자주 발생하는 데이터 불균형 문제를 해결하기 위한 방법을 제안하였다. 첫 번째 연구에서 제안한 입력 이미지의 최적의 크기를 적용하였고, 두 번째 연구에서 제안한 필터의 민감도 분석을 사용하였다. 제안한 재학습 방법의 두 단계 중 첫 번째

단계에서는 동등한 수의 지표를 갖는 mini-batch 를 과표본화를 적용하여 생성한다. 이 때의 진단 정확도가 높지 않은 경우, 두 번째 단계에서는 민감도가 낮은 필터들만 사용하여 다시 학습하면 더 정확한 진단 결과를 확보할 수 있음을 증명하였다.

주요어: **유막베어링 회전체 시스템**
합성곱신경망기반 진단
진동이미지 크기
지표기반 mini-batch
데이터 불균형 문제

학번: 2013-20715