University of Windsor

Scholarship at UWindsor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

4-27-2018

Fault Detection and Diagnosis of Electric Drives Using Intelligent Machine Learning Approaches

Shokoofeh Zare
University of Windsor

Fault Detection and Diagnosis of Electric Drives Using Intelligent Machine Learning Approaches

By

Shokoofeh Zare

A Thesis
Submitted to the Faculty of Graduate Studies
through the Department of Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science
at the University of Windsor

Windsor, Ontario, Canada

2018

Fault Detection and Diagnosis of Electric Drives Using Intelligent Machine Learning Approaches

by

Shokoofeh Zare

APPROVED BY:

J. Johrendt
Department of Mechanical,Automotive and Materials Engineering

---

M. Ahmadi
Department of Electrical and Computer Engineering

M. Saif, Advisor
Department of Electrical and Computer Engineering

April 23, 2018

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyones copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Electric motor condition monitoring can detect anomalies in the motor performance which have the potential to result in unexpected failure and financial loss. This study examines different fault detection and diagnosis approaches in induction motors and is presented in six chapters. First, an anomaly technique or outlier detection is applied to increase the accuracy of detecting broken rotor bars. It is shown how the proposed method can significantly improve network reliability by using one-class classification technique. Then, ensemble-based anomaly detection is utilized to compare different methods in ensemble learning in detection of broken rotor bars. Finally, a deep neural network is developed to extract significant features to be used as input parameters of the network. Deep autoencoder is then employed to build an advanced model to make predictions of broken rotor bars and bearing faults occurring in induction motors with a high accuracy.

DEDICATION

Firstly, thank God for your miracles that I always achieve my dreams. I would like to dedicate my thesis to my lovely parents, for their unconditional love, financial support, and kindness which every day make me more powerful and confident. To my smart brother, Behnam, whom I can never get by without his guidance and encouragement, and finally to my kind sister-in-law, Mahshid who has always made me happy.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my supervisor Dr. Mehrdad Saif, for his perpetual supports, patience, and inspiration during my Masters program at the University of Windsor. It was such an honor for me to know you, and be part of your research team. Thank you so much for providing me such an opportunity to learn from you. In addition, I would like to express my gratitude to my committee members Dr. Majid Ahmadi, and Dr. Jennifer Johrendt for their helpful advice and suggestions to my thesis. I would also like to express my special thanks to my friends and colleagues for helping me during the past two years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Electrical motors play a significant role in our daily lives. Therefore, it is very important that they do not fail unexpectedly [39]. Three main kinds of electrical motors used in industry are DC motors, synchronous motors, and induction motors. The most popular kind of electrical machines is the polyphase induction motor. A source of polyphase AC voltage applied to the stator winding is required for induction motors. This voltage produces a magnetic flux which rotates around the stator at synchronous speed. A magnetic flux produced by induced currents in the rotor winding combines with the stator flux to produce torque. Since these motors are subjected to sudden malfunction due to different reasons; it is a matter of high importance to recognize such faults to prevent unexpected failure. Various methods of fault detection and diagnosis of induction motors are described in this thesis.

Induction motors (IMs) are considered electromechanical energy transformation devices since they convert electrical energy into mechanical energy [44]. Reliability, simple design, construction, and cost effectiveness are the major reasons for the vast applications of IMs in industry [48]. Three possible types of faults that affect IMs performance include electrical, mechanical, and environmental faults. Major mechanical faults in IMs include bearing faults, stator faults, and rotor faults including broken rotor bars. Each of them may lead to the system failure [44]. The severity of such damages makes it absolutely essential to establish an accurate monitoring system to detect such incident [49, 44]. A significant number of issues can cause broken rotor bar (BRB) fault, including thermal, magnetic, residual, dynamic, environmental stresses, and mechanical defects generated by bearing faults [82]. Since the rotor is rotating quite quickly and it is difficult to attach transducers directly to the rotor body, fault detection and diagnosis is a challenging task. Therefore, indirect measurement techniques are required to detect rotor damage. One of the best measurement techniques for detecting rotor faults is stator current analysis. On the other hand, the bearings faults can be caused by a number of reasons, including material fatigue, overheating, harsh environments, corrosion, improper installation, poor lubrication which is the main cause of their failure, and so on. Vibration and stator current analysis are developed for the detection of bearing faults in IMs. Over the last decades, fault detection and condition monitoring systems have improved rapidly that help to increase the availability, and enhance the performance of the system. These studies are mainly based on the health management with fault detection and diagnostics (FDD) methods using condition-based maintenance (CBM) for IMs [48, 44]. Condition monitoring methods possess certain advantages including detection of the motor failure, improving the reliability, decreasing the maintenance cost and machine downtime [44]. The general approaches in condition monitoring are model-based and data-driven methods. This thesis focuses on data-driven techniques to detect broken rotor bars and bearing faults. The rest of the thesis is organized as follows. In Chapter 2, detailed information about fault detection and diagnosis approaches is given. Chapter 3 describes anomaly technique, whereas Chapter 4 presents the ensemble-based anomaly detection, Chapter 5 defines deep learning models to solve fault diagnosis problems. Finally, the thesis ends with some concluding remarks in Chapter 6.

CHAPTER 2

Fault Detection and Diagnosis Approaches

In order to prevent destructive unanticipated failures, a large number of studies in condition monitoring (CM), fault detection and diagnosis in the dynamic modeling of, for example, gears, bearings, rotor bars in IMs have been carried out to study. Fault detection and diagnosis approaches can be categorized into two types in CM named a model-based approach and data-driven approach [47, 64, 12]. A brief summary of prior work based on these approaches and the values of them are defined in this chapter.

2.1 Condition Monitoring Approaches

2.1.1 Model-Based Approach

The model-based fault detection and diagnosis approach is a mathematical model of the system under observation, in which a fault will cause deterministic changes in the model parameters. These techniques use usual differential equations and different elements of the model relating to actual results. They produce features like residuals r, parameter estimates Θ or state estimates $\overline{x}$, based on measured input signals U and output signals Y, identify the possible fault conditions, and extract useful information [16]. The main model-based techniques advantage is the ability to detect unexpected faults beside the replacement of hardware redundancy by diagnostic redundancy [83]. Many real-world applications are too complex to develop an accurate model. Therefore, model-based fault detection and diagnosis approaches are almost impractical to apply and other methods should be applied.

2.1.2 Data-Driven Approach

Extracted features from the measured process data are applied in data-driven approach in order to build a model that shows the process. The data-driven approach has been applied in many too complex real-world applications to develop an accurate model. A large number of techniques in data-driven approach have been applied to solve fault detection and diagnosis problems. Statistically based methods and those based on artificial intelligence (AI) techniques [42, 86] are different methods in the data-driven approach. As it is illustrated in Figure 2.1.1, after data collection and feature extraction, the intelligent detection and diagnosis will be employed.

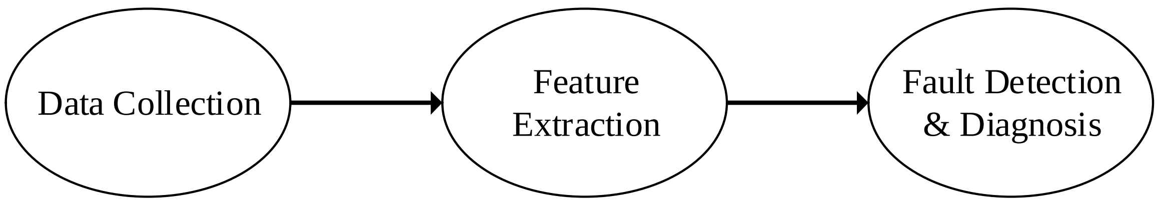

FIGURE 2.1.1: Fault detection and diagnosis step

Statistically-Based Approaches

Data-Driven based fault detection and diagnosis is a novel detection for CM, which recognizes any abnormalities between the features extracted from the measured data and the data measured under normal (healthy) operating conditions. Extracted features from a machine in its healthy state will have a distribution with a connected mean and variance. When the fault occurs, a variation in the mean and/or variance will appear. One of the earliest statistical fault detection techniques is statistical control charts (SCCs) [86]. SCCs monitor the distribution of the features and detect any changes in the distribution characteristics of the features will indicate the fault, termed outlier analysis. One of the intelligent detection and diagnosis examples in the statistically-based approaches is anomaly technique or outlier analysis.

Artificial Intelligence (AI)-Based Approaches

A computerized approach that applies knowledge to enable machines to perform tasks which humans perform using their intelligence [36] is artificial intelligence (AI). In order to enhance the accuracy and efficiency of fault detection and diagnosis of machines, AI techniques such as artificial neural networks, fuzzy logic and support vector machines (SVM) have been widely developed in recent years. In addition, the intelligent detection and diagnosis examples in AI-based approaches are training classifiers like artificial neural networks (ANN), also known as Multi-layer Perceptron (MLP) [30, 56], and support vector machine (SVM) [71] with these features. The accuracy of intelligent fault diagnosis with the help of their multilayer nonlinear mapping ability can be improved by using their multilayer nonlinear mapping ability which is named deep learning models.

Artificial Neural Network (ANN)

The artificial neural network is inspired by human brains. Data processing and learning ability of biological neurons are processed in the brain. ANN is used artificial neurons which use the functionality of both memory and computation [70, 19]. As a result, ANNs can play an important role in identifying and diagnosing faults in machinery. These intelligent fault diagnosis can propose a self-diagnostic procedure. They can be applied for a variety of applications to the area of intelligent condition monitoring, including function approximation, classification, pattern recognition, clustering, and forecasting [70].

Support Vector Machines (SVM)

One types of artificial intelligence methodology applied commonly for the classification and regression of data is support vector machine (SVM). In most neural network systems, SVMs are supervised learning methods resulting from statistical learning theory. Supervised learning is one of the machine learning methods which creates a clear map between the inputs and outputs in the training data. Normally, SVMs are applied for binary-class classification, but they can be used for multi-class classification problems [62, 58] by using some techniques. SVMs can predict the relationship between the input and output accurately by using a small amount of training information. For instance, SVM can classify a two-class dataset by finding a splitting plane between two classes. The splitting plane named decision boundary can be a linear or non-linear boundary.

SVMs applications:

- Mechanical fault diagnosis
- Data mining
- Text classification
- Facial recognition

2.2 Condition Monitoring Techniques

Various CM techniques have been applied for the purpose of rotating machinery health monitoring in recent decades. These techniques, including vibration, acoustic emission, motor current, lubricant analysis, and thermal monitoring. The most applicable techniques to use in different applications are described in this section.

2.2.1 Vibration Condition Monitoring

Fault detection and diagnosis techniques in various industrial applications [28] use vibration signal analysis. Each component's geometry and the rotational speed of the machine effect on each component's frequencies. Vibration signal analysis can determine the fault along with its cause and severity by using the relationship between the measured frequencies and expected faults, either by theoretical modeling of the machine or by measurement. When there is a surge in vibration level, it means that a fault occurs in a rotating machine. In order to analyze the vibration signal of rotating machines, different methods including, fast Fourier transform (FFT) for frequency analysis, empirical mode decomposition (EMD), wavelet analysis, and so on are applied. The main goal of signal processing is getting some useful information which cannot be received for any reasons from the initial signal. This goal is named feature extraction which is achieved by data mining. Data mining is extracting hidden data (features) from the signals. Figure 2.2.1 shows different categories of the signal processing step.

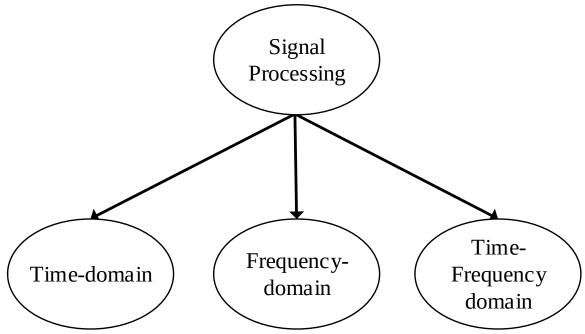

FIGURE 2.2.1: Signal processing step

Importance of signal processing:

- Remove and reduce the effect of noise
- Achieve hidden signal content
- Create a better signal for better data mining

When the signal is contaminated by noise, vibration condition monitoring can improve the signal-to-noise ratio. Therefore, faults can be detected efficiently.

Various methods of signal processing:

- Time-domain

  Almost all vibrating signals are initially time-domain. It means that the signal layers apart from what is measured is the nature of the time. In other words, the variation of signal's amplitude over time is referred to the time domain. In order to analyze mathematical functions, physical signals or time series of economic or environmental data, with respect to time, time domain technique can be used. Real-world signals in the time domain, including continuous and discrete time, can be visualized by a common tool named an oscilloscope. Fault detection and prognosis of control systems, CM, and time series are some time-domain signals' applications.

- Frequency-domain

  In general, all signals are composed of many sinusoidal signals with different frequencies (Fourier series). And in some cases, the frequency content of a signal contains an essential and necessary information of the signal. The presence of noise in the time-domain signals leads to some problems in fault detection and diagnosis. In order to solve these problems, signals should be converted to frequency-domain. Frequency-domain can remove noise from the signals. As a result, useful information can be achieved. The most common method in this domain is fast Fourier transform (FFT).

- Time-frequency domain
  Classical Fourier analysis assumes that signals are infinite in time or periodic, while many signals in practice are of short duration, and change substantially over their duration. As a result, time-frequency analysis should be applied. Time-frequency analysis is the study of the signal in both the time and frequency domains simultaneously. It means that time-frequency analysis studies a two-dimensional signal. Different kinds of time-frequency analysis methods are short-time Fourier transform (STFT), wavelet analysis, and empirical mode decomposition (EMD).

There is no particular mathematical-physical interpretation in these methods for signal processing step since they include a very large group of raw data. As a result, a number of signal features that are mathematically interpreted should be extracted. In other words, after employing one of these methods for signal processing step, fundamental information (fault features) should be extracted from the vibration signal of machines.

The properties and states of a signal cannot be highlighted by the low selection of features since it is impossible to distinguish between two different signals. Also, it is difficult to analyze a large number of features. As a result, the behavior of a signal cannot be accurately predicted in these two cases. In order to reduce data space, save time, and improve the performance, a large number of features should be extracted and then select some useful features from them. This selection of some useful features is named feature selection.

2.2.2 Acoustic Emission Condition Monitoring

The study of the generation, propagation, and reception of sound that is heard by a human being [63] is Acoustics. The sounds are divided into desirable and undesirable, which is traditionally known as noise. Only sound waves within a specific frequency range, between 20 Hz to 20 kHz, can be heard by human ears. Moreover, frequencies above 20 kHz are recognized as ultrasonic. High-frequency signals range from 100 kHz to 1 MHz [63, 76] can be dealt with the acoustic emission (AE) technique, which has more stable performance in fault detection. Therefore, the AE-based technique needs much higher sampling rates than vibration-based techniques. Most machines under normal operating conditions emit acoustic signatures and any variation in these signatures can show the start of corrosion of some components.

2.2.3 Motor Current Signature Analysis Condition Monitoring

Mechanical faults with electrical signatures can be detected by motor current signal analysis (MCSA). The stator current signal of the motor can be measured at distant locations from the motor because of the accessibility of the current-carrying conductor to the motor. Therefore, this sensorless technology does not need any transducers or measuring equipment to be installed on or near the monitored machine. A large number of faults in IMs can be detected by this technique, including broken rotor bars [23, 7], shorted windings, air-gap eccentricity [18], bearing faults [72], load faults, and so on.

Stator Current Analysis

Stator current analysis means filtering the stator current to remove the important frequency content that is irrelevant to faults occurring in IMs [75]. A baseline or reference model named autoregressive model can be trained by the filtered healthy current signal. When a fault occurs, the deviation in spectral content from its reference measurement is increased. This increase in spectral deviation can be used as the fault index. One reliable method for detecting faults at their early stages in IMs [27] is a CM technique based on statistical and numerical tools. In order to find the spectrum of the motor current, FFT can be used. And then wavelet function, a multi-resolution signal processing technique, can be applied on this spectrum to detect the significant peaks. In wavelet function, vibration signals are segmented into multi-level in order to analyze the simulated signals. These signals are analyzed using time and frequency-domain feature extraction techniques. Then, some statistical parameters, including mean value, root mean square (RMS), energy, entropy and so on are calculated from each segment and applied to detect faults at their early stages. Overall, some diagnostic techniques such as spectrum comparison, spectral kurtosis analysis, and envelope analysis can be applied to the vibration signals for fault detection method in IMs.

Various feature functions:

- Mean value

  The mean value is good to be calculated when the defect affects the overall mean of the signal amplitude. As it is shown in the below equation, $X(n)$ is a value of the signal and N is a number of signal points at the time.
  $Mean = \frac{\sum_{n=1}^{N} X(n)}{N}$

- Root Mean Square (RMS)

  RMS is the effective amount of a signal. And it can be measured by the below equation:
  $RMS = \sqrt{\frac{\sum_{n=1}^{N} X(n)^2}{N}}$

- Energy

  The energy level of a signal indicates its degree of disturbance. Therefore, signal high energy indicates a phenomenon such as a system failure, an installation failure, and so on. The below formula shows it.
  $Energy = \sum_{n=1}^{N} X(n)^2$

- Entropy

  Entropy is one of the significant features for condition monitoring (CM).
  $Entropy = \sum_{n=1}^{N} (X(n) * log(\frac{1}{X(n)}))^2$

2.3 Fault Detection in IMs

2.3.1 Detection of Broken Rotor Bars

A number of work done have been noted that broken rotor bars cause a pulsation at twice the slip frequency in the stator current [32, 50]. In addition, others concluded that axial flux may be monitored in order to detect faults [81, 55]. Although the amplitude of vibrations caused by a damaged rotor is smaller compared with that caused by damaged bearings, vibration can also be utilized to detect rotor faults [81]. Parameter estimation is another technique for the detection of broken rotor bars [14, 24]. Model parameters are the measurements of current and voltage. A sensor should be connected to the motor supply terminals in order to measure the voltage. On the other hand, a current transformer or Hall effect transducer should be clamped to the motor supply cable to measure a current. It is used for stator current analysis. And when specific frequency components in the spectrum of the stator current exist, rotor faults can be detected [32].

2.3.2 Detection of Bearing Faults

The most of the failures occurring in IMs (about 40%) are related to the bearings [17]. These failures are so costly and time-consuming. They do not lead to an immediate breakdown. But, they evolve in time until they produce a critical failure of the machine. Rolling-element bearing fault detection in IMs using MCSA technique is described in [72]. First, [72] researched on the effects of different bearing faults on the stator current spectrum and the relationship between motor current and induced vibration, due to incipient bearing faults. The predicted relationship between the vibration and current frequencies showed that the stator current signature can be applied for a bearing fault detection. Overall, the bearing characteristic frequencies and the modes of failure are related to the bearings construction. As a result, vibration analysis is not always possible to diagnose the bearing faults because some vibration sensors and particular equipment for the CM are needed. Moreover, the stator current monitoring is more convenient because it needs only simple and cheap current sensors. Therefore, the stator current analysis can be applied in some specific situations. In general, the rolling-element bearing consists of two rings, between which a set of balls or rollers rotate in raceways. In most cases, fatigue failure begins with small cracks, which are located on the surfaces of the raceway and rolling elements under normal operating conditions. The cracks slowly expanded by the repetitive impacts between the components of the bearing and the faulted surfaces. These cracks cause an increase in vibrations and noise levels [33]. The position of the fault affects vibrations. The fault can be occurred in the inner race, the outer race, balls, and cage [72].

CHAPTER 3

Anomaly Technique

One of the data-driven techniques like anomaly technique or outlier analysis with their fusion in various configurations, are defined in this chapter. A data-driven diagnostic scheme to detect broken rotor bar by analyzing stator current signal is proposed. The primary goal of the proposed model is to create a proper feature subset that represents a precise index of the IMs operating conditions. More importantly, the proposed model has benefited from one-class classifiers (OCCs), which are ideal for fault detection purposes [87, 74]. OCCs are outlier detection techniques [37], which aims to detect the normal condition or the target class and reject abnormal samples or the outliers, in this case representive samples of broken rotor bar. OCCs could assist in fault detection process even when only the information about normal state of the system is available, which is frequently happens in real applications. They are mainly used to know the normal condition or class of target, and reject any other samples as fault or class of outlier [87, 74].

3.1 One-class classifiers (OCCs)

A one-class classifier aims to detect particular samples, which belong to the class of target, amongst all other samples that belong to the class of outlier. It is widely used to identify whether the new samples are similar to the sample of the target class, which the classifier already learned them during the training process. If a new testing sample is not the same as the training set, it will be called an outlier, or novelty or abnormality [13]. In this section, six state-of-the-art OCCs named Gaussian Distribution (GD), Parzen Density (PD), Nearest Neighbour (NN), k-Nearest Neighbour (kNN), k-means, and Angle-Based Outlier Fraction (ABOF) are discussed. These classifiers are working based on four different methods; the density-based estimation method (i.e., GD and PD), the distance or boundary methods (i.e., NN and kNN), the reconstruction method (i.e., k-means) and the last method which is based on the variance of the angles (i.e. ABOF).

3.1.1 Gaussian Density or Normal Distribution

Gaussian density is widely used to estimate the density of the probability functions [60]. This one-class classifier models the target class as a Gaussian distribution [79]. Given a test point x, the Gaussian method uses the Mahalanobis distance to measure the resemblance between x and all training samples. The measurement of the distance d from x to the target class, which is represented by the mean value of the training set μ_{tr}, is calculated by [20]:
$D^2 = (x - \mu_{tr})^T (cov_{tr})^{-1} (x - \mu_{tr})$
where, cov_{tr} stands for the covariance of the training set. One should apply a threshold like β, which describes the separation between the target and outlier classes. If $D^2 \leq \beta$, so $f(x) =$target, otherwise $f(x)$ is an outlier [77].

3.1.2 Parzen Density

Parzen method needs a large number of training objects to make a correct probability density estimation. A width parameter σ gives an information about the probability density distribution. In this method the width σ of the kernels has to be expected [78]:
$p(x) = \frac{1}{N} \sum_{x_{tr}} \frac{1}{(2\pi)^{\frac{d}{2}} \sigma^d} exp(\frac{-1}{2\sigma^2}(x - x_{tr})^2)$
This is the average of N Gaussian functions with each data point as a center, which can be used to model the training subset. x_{tr} is a data point in the training set and d is the dimensionality of the input space [78, 60]. This method can incorporate the outlier in its probability estimate [61]. This method sorts all training objects and, during the test phase, calculates and sorts the distances to all training objects. This might limit the applicability of the method, especially when large datasets in high dimensional feature spaces are considered [61].

3.1.3 Nearest Neighbour (NN)

The nearest neighbor data description (NNDD) is a one-class classification method that is on the basis of the distance between the object and its nearest object in the feature space [43]. Given a test object x, the first nearest objects of x like x_{tr1} will be selected, and the distance between x and its neighbor in the training set, x_{tr1} will be calculated and named d_1.

$||dist(x, x_{tr1}(x)\| = d_1$

After that, the distance between x_{tr1} and its nearest neighbor in the training set, x_{tr2} will be calculated and named d_2.

$||dist(x_{tr1}, x_{tr2})\| = d_2$

, where dist is the Euclidean distance between two objects [13].

NN method says if $\frac{d_1}{d_2} < 1$, x is accepted as a target. Otherwise, if $\frac{d_1}{d_2} > 1$, x is an outlier [80, 46, 13].

3.1.4 k-Nearest Neighbour (kNN)

The same as NN, k-Nearest neighbor data description is a one-class classification method, in which instead of choosing only the first nearest neighbor, it needs to select k nearest neighbors [80, 46, 13]. Where k is the number of nearest neighbors to an object detected by the classifier [2].

3.1.5 Angle-Based Outlier Fraction (ABOF)

Another method to detect outliers is the Angle-Based outlier detection approach. ABOF, instead of using distance-based methods to detect outliers, makes use of the variance of angles. These angles are more stable than distances in high dimensional space [52]. The Angle-Based Outlier Factor ABOF ($\overrightarrow{A}$) is the variance over the angles between the different vectors of $\overrightarrow{A}$ to all pairs of nearest neighbors, e.g., $\overrightarrow{B}$, $\overrightarrow{C} \in N_k(\overrightarrow{A})$, weighted by the distance of the samples, where $N_k(\overrightarrow{A})$ of $\overrightarrow{A}$ stands for the k nearest neighbors of $\overrightarrow{A}$:

$$ABOF(\overrightarrow{A}) = VAR_{\overrightarrow{B},\overrightarrow{C} \in N_k(\overrightarrow{A})} \left(\frac{\langle \overline{AB}, \overline{AC} \rangle}{||\overline{AB}||^2 . ||\overline{AC}||^2} \right) \tag{3.1.1}$$

The ABOD algorithm assigns the angle-based outlier factor ABOF to each point in the dataset and returns the list of points sorted according to their ABOF [53]. The variance of the angles for points outside of the cluster is the smallest and, thus, these points are assigned as outliers.

3.1.6 k-means

The simplest reconstruction method is the k-means clustering [9]. Clustering algorithms are widely divided into two methods: hierarchical and partitional. Among different methods, k-means is the simplest partitional algorithm [41]. k-means forms k clusters whose points have maximum interior cluster similarity (minimum distance) and also minimum similarity with points inside the other clusters. In this algorithm, k initial "means" or "cluster centers" are randomly created within each cluster. Therefore, assume X=x_i, $i = 1, ..., n$ as the set of n d-dimensional points to be clustered into a set of k clusters, c_j, $j = 1, ..., k$ [41].
The distance metric can be found by measuring the similarity between interior points and the mean of the cluster. Within cluster distance, it can be defined as:
$E_1 = \sum_{x_i \in c_j} dist(x_i, \mu_j) = \sum_{x_i \in c_j} \|x_i - \mu_j\|^2$
, where μ_j is the mean of cluster c_j. Minimizing the sum of this distance over all k clusters is the main aim of k-means.

$$E_2 = \sum_{j=1}^{k} \sum_{x_i \in c_j} dist(x_i, \mu_j) = \sum_{j=1}^{k} \sum_{x_i \in c_j} \|x_i - \mu_j\|^2 \tag{3.1.2}$$

In the k-means classifier, it is assumed that if $E_2 \leq \theta$, $f(x)$ =target, otherwise $f(x)$ is an outlier (θ is a preset threshold) [77].

3.2 Design of The Fault Detection

The goal of this chapter is to design a fault detection model to identify broken rotor bar of IMs. Figure 3.2.1 shows the proposed fault detection scheme, which consists of three sub-modules of feature extraction (FE), feature selection (FS), and fault classifiers. In the first module named feature extraction (FE) is applied to the stator current signals. In other words, vibration signals including normal and faulty conditions are segmented into different parts, then seven time-domain features, including root mean square, mean value, shape factor, energy, entropy, peak to peak, and variance of each segment are calculated. The normalization is also applied on the extracted features to create a well-processed dataset [68]. Once feature extraction task is completed, two different scenarios are considered. In the first scenario, the extracted features are directly fed to the six different state-of-the-art OCCs to discover if the motor is in the normal condition or not. In the second scenario, the correlation-based feature selection method, adopted from [31], is applied on the extracted features before the classification task. This feature selection technique tries to find a subset of features, which is the most correlated to IM operating condition, while it also considers the degree of redundancy between the features. The Best-first search starts with an empty set and then, searches forward through the feature space [65].

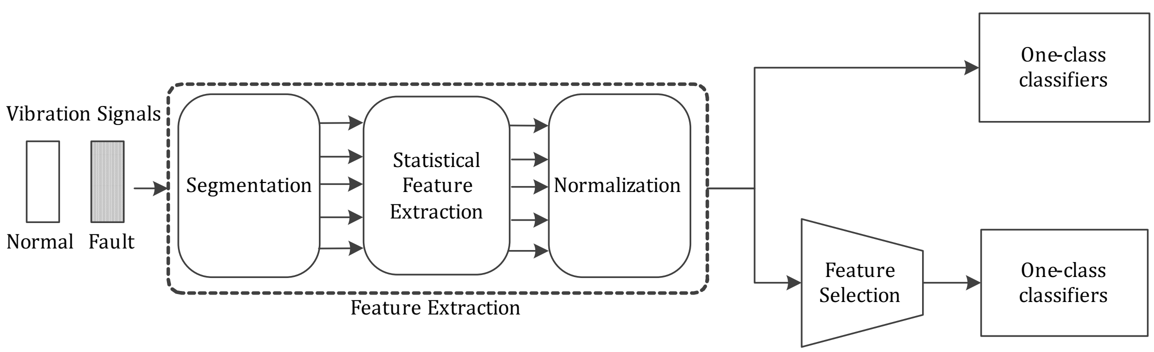

FIGURE 3.2.1: Block diagram of the proposed fault detection scheme

3.3 Experimental Results

In this study, a three-phase, 1.2 KW, 380 volt, 50 Hz, 1400 rpm, four-pole induction motor is used to collect experimental data. The stator current signal is analyzed with the proposed fault detection scheme. The experimental results obtained from two different scenarios, with/without applying feature selection method, are evaluated and compared. Moreover, the performance metrics (i.e., accuracy and f-measure) for each state-of-the-art one-class classifiers (i.e., GD, PD, NN, kNN, k-means and ABOF) are calculated to determine the best technique. It is also noticeable that 10-fold cross validation is considered to provide a reliable evaluation. The proposed fault detection scenarios are compared with each other in Figure 3.3.1. In this figure, solid blue circles show the accuracy and f-measure of the six different OCCs participating in each scenario, the white squares show the average performance, and the red crosses represent the outliers. From this it can be concluded that the use of selected features by means of FS (i.e., mean and entropy) could improve the average performance of the OCCs from about 0.67 in the first scenario to around 0.96 in the second scenario. In other words, the applied feature selection method could effectively enhance the accuracy of the fault detection scheme about 0.29. Figure 3.3.2 is also provided to take a closer look at efficiency of each one-class classifier after feature selection. This figure shows that kNN has the best performance since it has the highest mean value. In addition, performance of ABOF and k-means as fault detectors are very close to each other and can be placed in second and third ranks, respectively. The highest variation and the least stable results corresponds to GD and NN classifiers.

For a better comparison, the decision boundaries of six classifiers are shown in Figure 3.3.3. Considering kNN, ABOF and k-means panels in this figure, there are a very few samples of outlier (broken rotor bar), which are located inside the normal region (i.e., representative of the missed alarms). Moreover, the number of misclassified samples of normal condition, which results in false alarms, is almost so small. As represented in the figure, GD has the highest amount of the false alarms and NN has missed many alarms, i.e., samples of faulty class. As a result, NN and GD are also ranked as fifth and sixth, respectively.

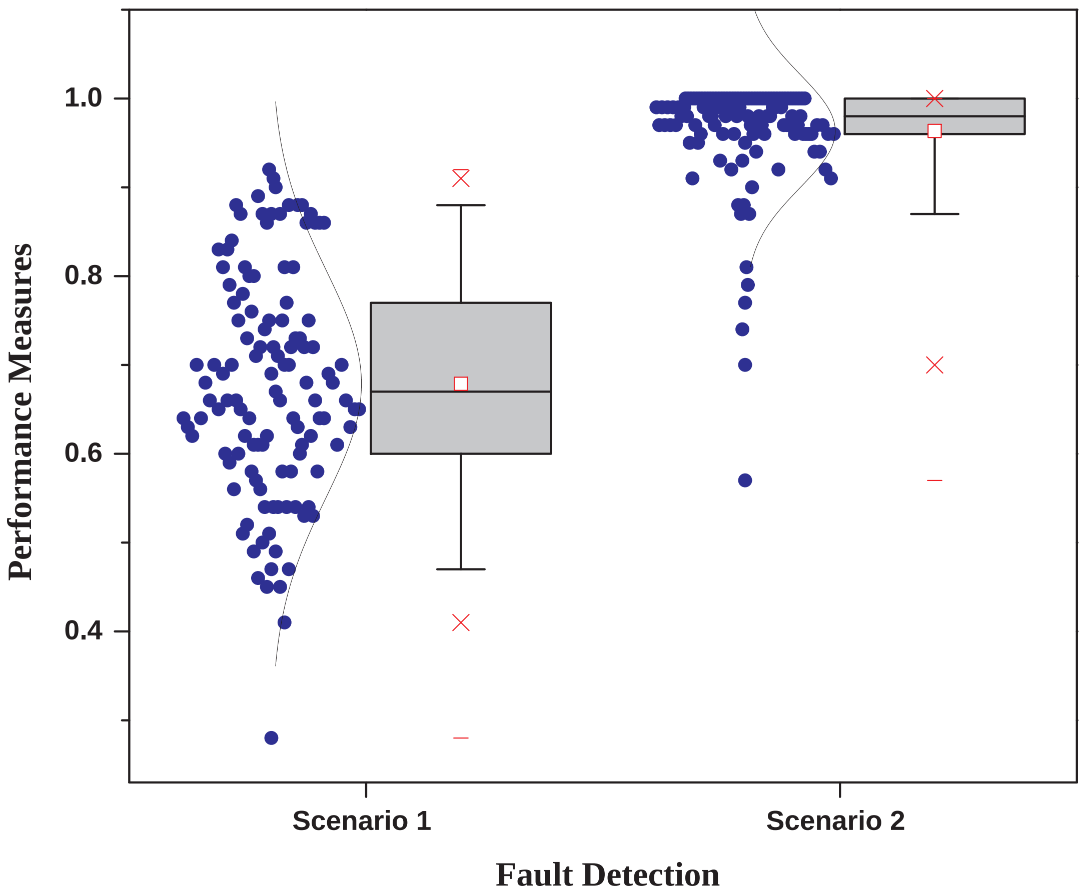

FIGURE 3.3.1: Performance measures (i.e., accuracy and f-measure) obtained by each scenario.

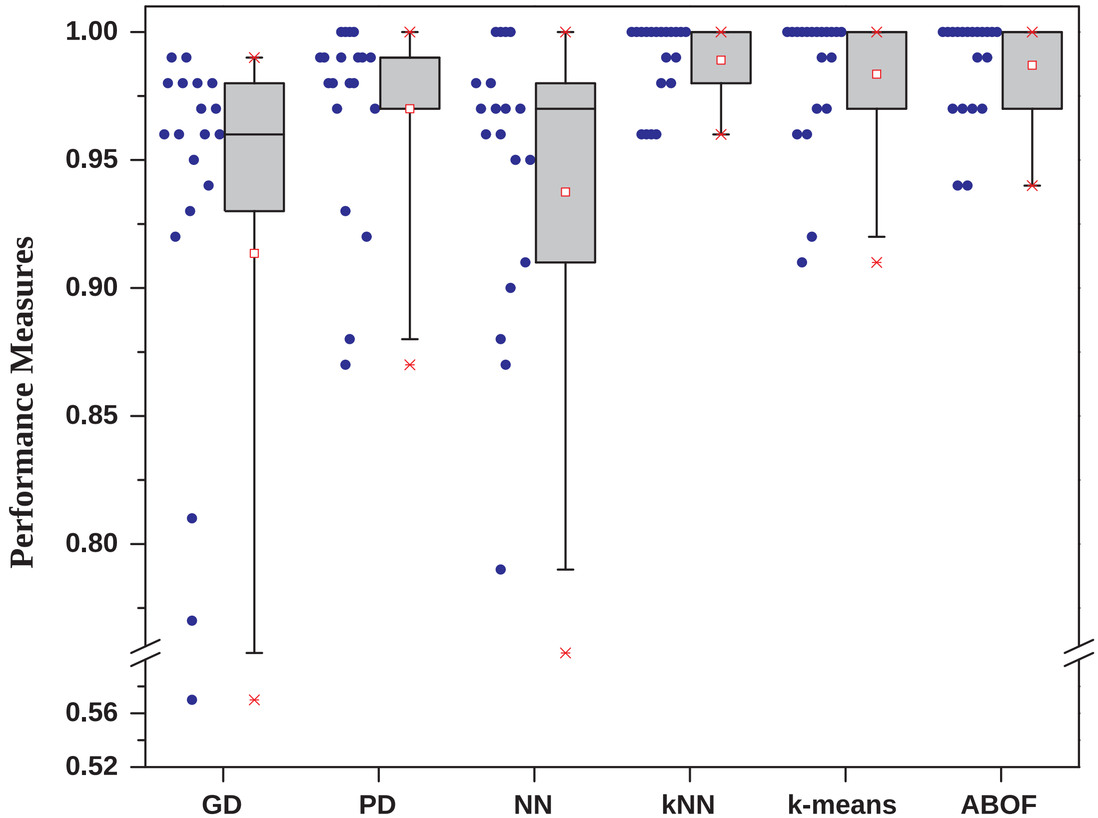

FIGURE 3.3.2: Performance measures (accuracy and f-measure) obtained by each one-class classifier after feature selection.

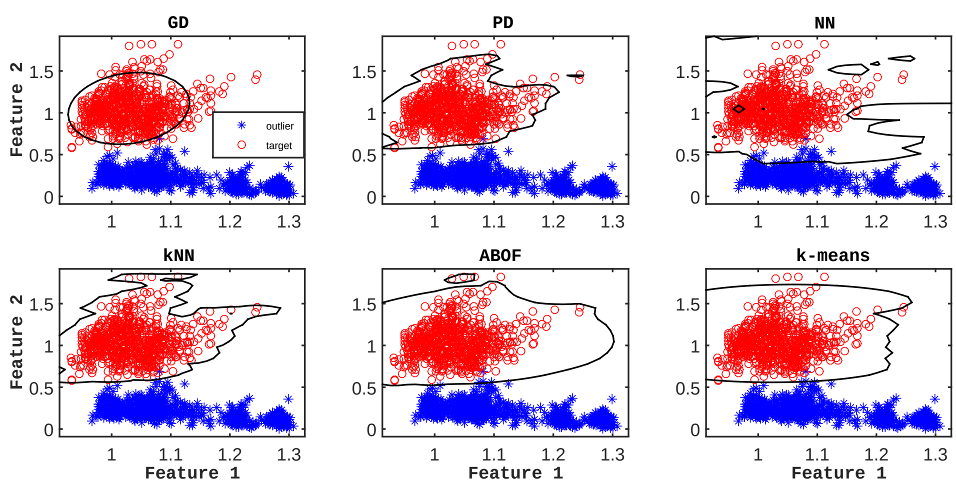

FIGURE 3.3.3: Performance measures trends for each one-class classifier after feature selection.

In addition, it is common in OCCs to determine a threshold parameter like β (i.e., fraction of target rejected or missed alarm) which describes the separation between the target and outlier class during learning process. Figure 3.3.4 is also provided to present the classifiers' performances obtained by varying β from 0.01 to 0.9 after feature selection. From this figure, one may conclude that by increasing β the average performance measures are decreased among all classifiers. kNN, ABOF, and k-means have very similar trend. They have the best rank among others. The difference between them is just standard deviation. PD, NN, and GD have some variation in results (they do not have a stable results in this range).

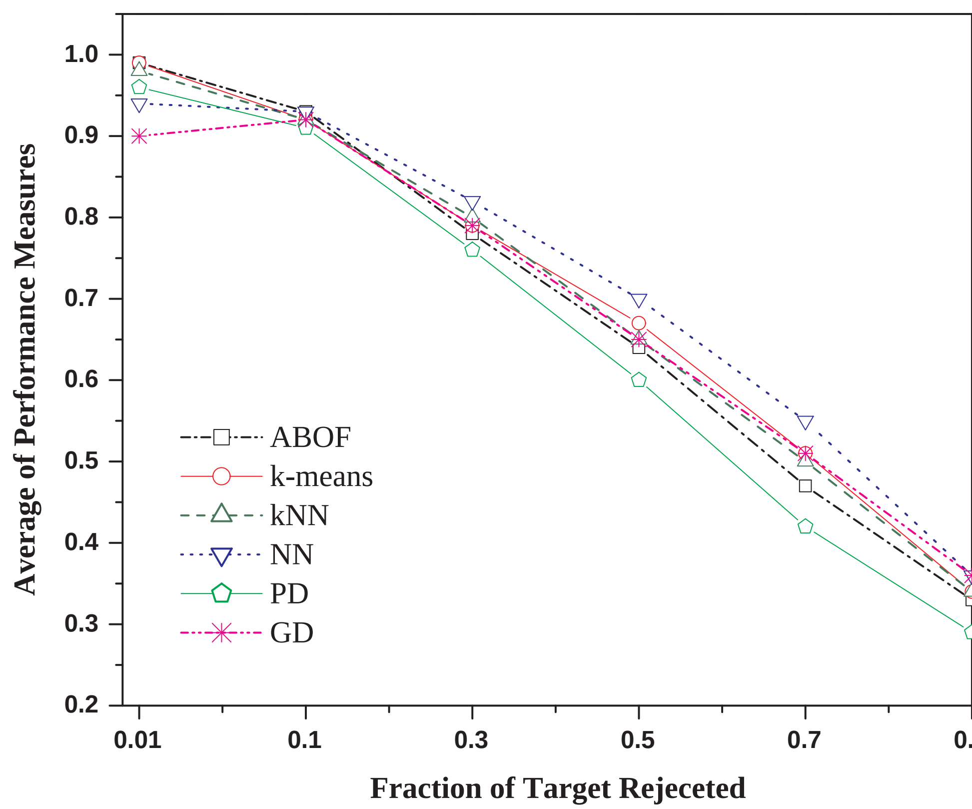

FIGURE 3.3.4: Performance measures trends for each One-class classifiers after feature selection.

In addition, the obtained accuracy and f-measure of the classifiers for both scenarios are presented in Table 3.3.1. The performance metrics are formatted as mean $\pm$ standard deviation of the 10-fold cross validation. The classifiers are ranked according to their performance measures, from 1 which means the best performance to the 6, in the last column of the table.

TABLE 3.3.1: Classifiers' performances with/ without feature selection

First scenario	Accuracy	F-measure	Rank
GD	0.71 ± 0.17	0.76 ± 0.12	1
ABOF	0.50 ± 0.01	0.66 ± 0.02	2
NN	0.50 ± 0.02	0.66 ± 0.02	3
k-means	0.51 ± 0.03	0.67 ± 0.03	3
kNN	0.50 ± 0.03	0.66 ± 0.03	4
PD	0.64 ± 0.17	0.70 ± 0.15	5
Second scenario	Accuracy	F-measure	Rank
kNN	0.98 ± 0.03	0.98 ± 0.04	1
ABOF	0.99 ± 0.02	0.99 ± 0.02	2
k-means	0.99 ± 0.02	0.99 ± 0.03	3
PD	0.97 ± 0.05	0.96 ± 0.06	4
NN	0.94 ± 0.08	0.95 ± 0.06	5
GD	0.93 ± 0.09	0.91 ± 0.13	6

CHAPTER 4

Ensemble-based Anomaly Detection

4.1 Problem statement

Ensemble methods using OCCs:

This chapter aims to study the use of the ensemble of OCCs for detecting broken rotor bars in IMs. Various configurations of the ensemble are made and studied in this chapter in order to design of the fault detection system. These fault detection configurations are indeed multiple classifier systems (MCSs) that aim to combine the outputs of various individual OCCs. These OCCs can be merged together through mean voting, majority or plural voting, and random subspace [10, 26]. These are indeed different sets of diverse models [3]. This diversity usually results in a better performance.

4.2 Ensemble-based systems

The main idea in MCSs is creating different subsets of data and, then, train a number of OCCs based on each. Various factors play important roles in designing a learning system including classifier parameters and training sets. This diversity can result in less estimation error. This section initially explains these ensemble techniques, including random subspace, bagging, and boosting of similar and different OCCs.

4.2.1 Bagging

The Bootstrap Aggregation (bagging) algorithm creates a numerous bootstrapped training sets repeatedly in a random manner to train individual models. This means that some training instances may be selected several times and others may not be selected at all. This algorithm applies the Majority Voting technique for aggregation. This means that at least more than half of the classifiers should return a label so that label can be assigned to that sample [3]. Moreover, the Parallel and Stacked combination have also been utilized in order to combine classifiers [21]. Bagging reduces the variance, while boosting reduces both bias and variance [10, 6]. Bagging algorithm as illustrated in Figure 4.2.1.

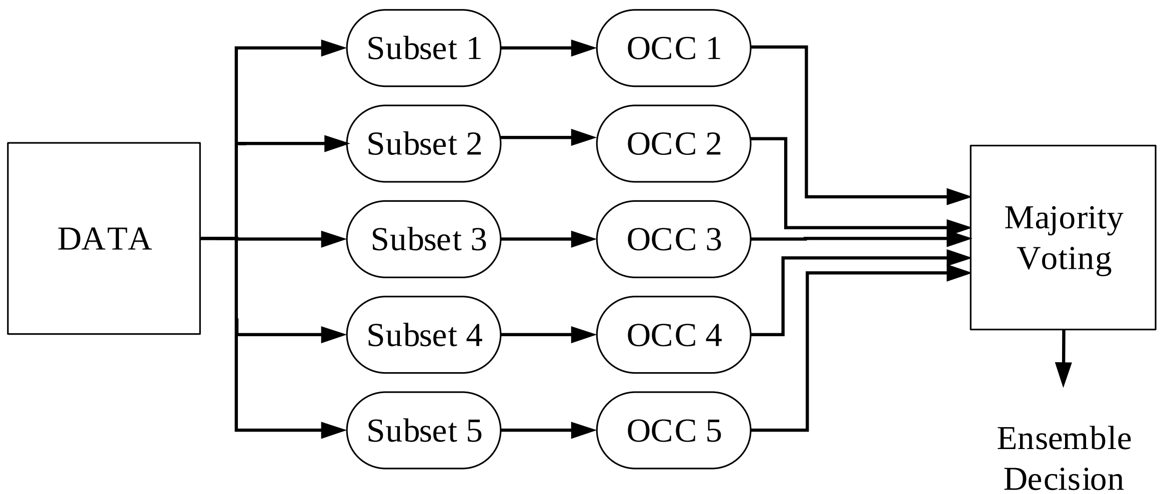

FIGURE 4.2.1: General scheme of the bagged of OCCs [5]

.

4.2.2 Random Subspace

Random subspace method is one of the ensemble learning methods, which is similar to bagging algorithm, called feature bagging. The only difference is in selecting random features. This method has a better performance for the high-dimensional data, where the number of features is much greater than the number of samples [73, 38]. Figure 4.2.2 illustrates the general scheme of the random subspace ensemble of OCCs.

4.2.3 Boosting

Boosting creates and trains a number of weak learners, which perform slightly better than random guessing. It modifies the input subsets of the upcoming OCC with increasing the weight of samples that are misclassified by means of previous OCCs and, thus, return the

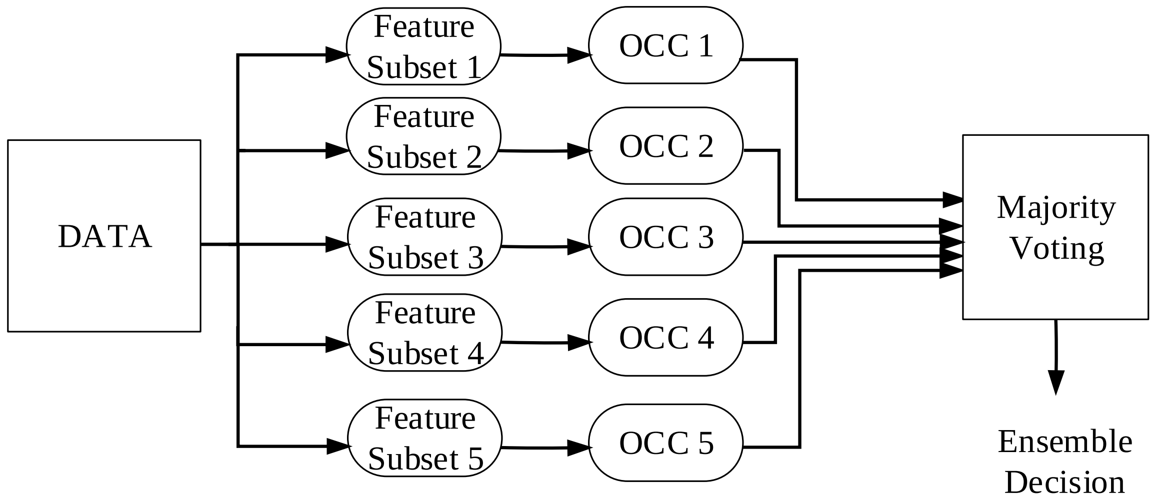

FIGURE 4.2.2: General scheme of the random subspace ensemble of OCCs.

total error [25]. The main objective of boosting is combining a number of weak learners to achieve a strong learner with the desired accuracy. There exists three different methods boosting strategies [51]:

- **Filtering**

  In this method, selected samples of the large dataset are deleted or returned to the dataset.

- **Sub-sampling**

  It is applicable over a constant dataset, where datasets will be resampled with replacement by using a probability distribution to their weights.

- **Reweighting**

  It is similar to the sub-sampling approach. This strategy aims to re-weight the samples according to the classification of the samples in previous iterations.

In this work, AdaBoost (M1) is used to begin a fault detection system by means of OCCs.

AdaBoost is an algorithm which utilizes a reweighting method to choose the training subsets. If a sample is misclassified by a weak OCC, the probability distribution (the weights) of selecting that sample for the next weak OCC will be increased. Otherwise, the probability will be decreased. The final hypothesis can be calculated by a weighted majority voting algorithm [59] over all OCCs. This method then focuses on misclassified samples. The number of classifiers is an important parameter of the AdaBoost algorithm. Figure 4.2.3 illustrates the general scheme of the boosted OCCs.

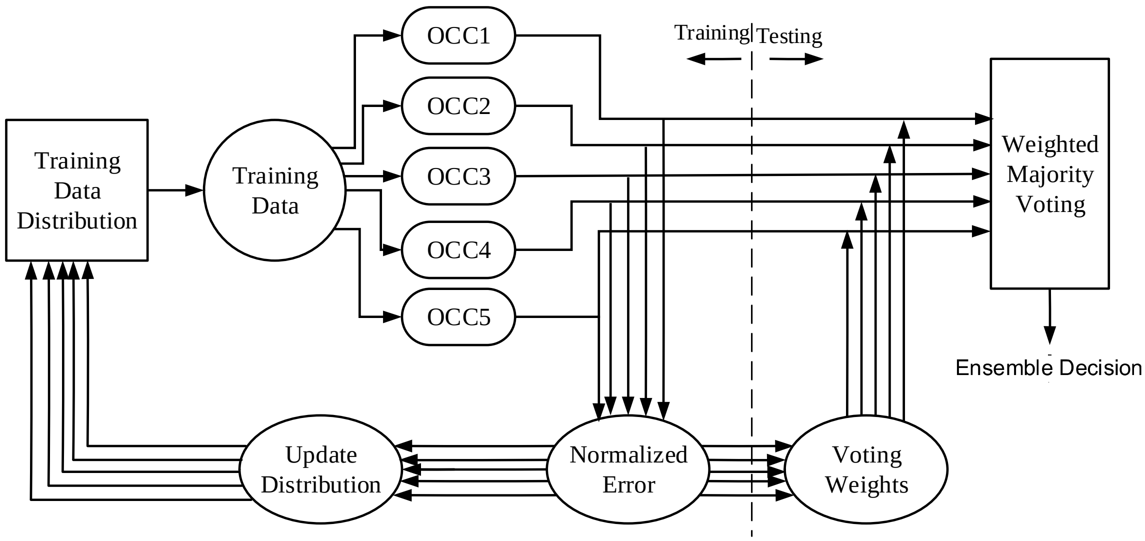

FIGURE 4.2.3: General scheme of the boosted OCCs [66].

4.2.4 One-Class Fault Classifiers

In this work, five OCCs, including GD, PD, NN, kNN, and k-means, are applied to design fault detection systems by constructing various type of ensemble. These OCCs techniques were individually used to detect BRB in Chapter 3 [68].

4.3 Experimental Results

In this section, a 3-phase, 50 Hz, 380 volts, 1.2 KW, 1400 rpm, 4-pole induction motor is used to gather experimental data. First of all, the stator current signal in normal and faulty conditions are segmented into various non-overlapping parts. Consequently, seven statistical features are extracted from each segment forming a feature set of statistical measures. These statistical features are root mean square, mean value, shape factor, energy, entropy, peak to peak and variance. The resulted sets contain seven features and less number of samples that is equal to number of non-overlapping segments. The normalization is also

TABLE 4.2.1: Performance measures obtained by each ensemble-based systems.

Ensembles	Performance Measures	GD	PD	NN	kNN	k-means	Combining heterogeneous classifiers
Random Subspace	Accuracy	0.96	0.99	0.64	1	0.99	0.70
	F-Measure	0.96	0.99	0.44	1	0.99	0.70
	Average	0.96	0.99	0.54	**1**	0.99	0.70
Bagging	Accuracy	0.98	0.99	0.95	1	0.99	1
	F-Measure	0.98	0.99	0.95	1	0.99	1
	Average	0.98	0.99	0.95	**1**	0.99	**1**
Adaboost	Accuracy	0.97	0.95	0.92	0.99	0.97	0.98
	F-Measure	0.98	0.96	0.94	0.99	0.93	0.99
	Average	0.975	0.955	0.93	**0.99**	0.95	0.985

applied to the extracted features to produce a well-processed dataset. The extracted set of normalized features is then used as inputs to construct the ensemble of OCCs for the sake of fault detection in IMs. Three different ensemble algorithms of random subspace, bagging and boosting with feature selection and a 5-fold cross validation is assessed and compared with each other in this section. Moreover, the performance metrics (i.e., Accuracy and F-Measure) for each method are measured to find the best detection scheme. First of all, the random subspace method is applied which generates features randomly and, then, trains five homogeneous and one heterogeneous OCCs. The results are summarized in Table 4.2.1. The best first feature selection strategy is used for selecting proper set of feature for each ensemble of OCCs. Various ensemble models, including bagging, boosting, and random subspace are constructed in homogeneous and heterogeneous configurations by means of the five stated OCCs. The attained results, i.e., performance measures by each ensemble model are reported in Table 4.2.1. As it can be seen in the table, the random subspace ensembles of NNs and heterogeneous OCCs do not perform well unlike bagged and boosted ensembles that result in satisfactory measures. This enlightens the fact that the random selection of features in relatively low dimensional data decreases the efficiency of the algorithms. Besides, homogeneous and heterogeneous bagged of OCCs slightly outperform homogeneous and heterogeneous boosted ensemble of OCCs. Moreover, homogeneous bagged of kNN and heterogeneous bagged of OCCs outperform other techniques. The random subspace of kNN has also achieved a very promising performance. The attained results also show that kNN is the best OCC to generate the ensemble schemes compared to other individual OCCs.

CHAPTER 5

Deep Neural Network

5.1 Artificial Neural Network

In recent decades, articial intelligence (AI) has played a significant role in the creation of machines that function as closely as possible to human brains as well as researching in some dynamic topics. Humans solve intuitive tasks easily, but describing that intuitive process is difficult. Therefore, AIs main applications include cognition and machine learning abilities. The machine is an intelligent computer that collects data from experience, learns complicated concepts and then makes an accurate decision. Deep learning is a subset of machine learning, which itself falls under the category of AI [8]. AI takes input data from the environment, and processes it for the purpose of decision making. The main goal of AI is simulating and understanding of human behavior. AI has a variety of applications, including robotics, natural language recognition, computer games, economics, behaviour recognition, and fault detection and diagnosis.

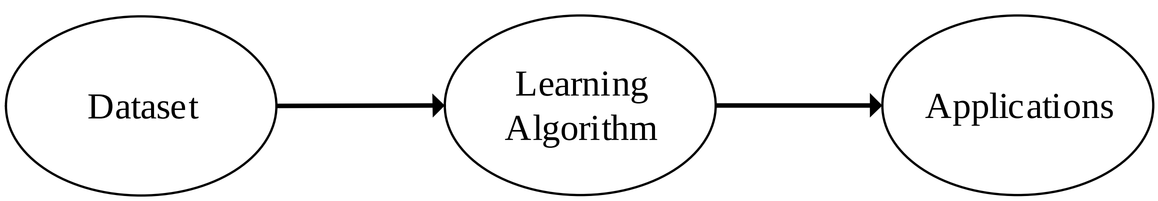

FIGURE 5.1.1: General scheme of machine learning techniques

As it is illustrated in Figure 5.1.1, the dataset is introduced to the machine learning algorithm and in the next process, the algorithm will be trained to get a target goal using this dataset. Once the algorithm is completed, it will be used for desired applications. A simple machine learning algorithm called representation learning is used to extract the right set of features [8]. Representation learning is also known as feature extraction. The model is trained by these extracted features and then performs one of two tasks: classification or regression. The model's task is to match each of the input data to the class related to it. Figure 5.1.2 shows this process. As it is shown in Figure 5.1.3, feature extraction is di-

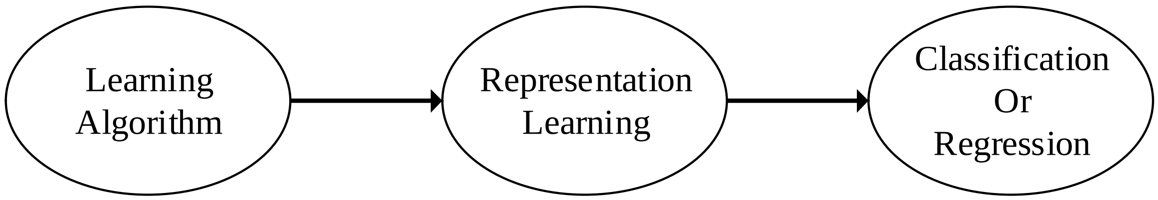

FIGURE 5.1.2: Steps in machine learning techniques

vided into two categories named automatic feature extraction (representation learning like deep learning, neural network, clustering, and so on) that results in much more acceptable performance compared with the traditional method (manual one) using formulas or predetermined methods. In representation learning methods, the algorithm itself learns which features are appropriate and how to extract them. Then, these extracted features are fed to the classifier to perform classification task or diagnose results.

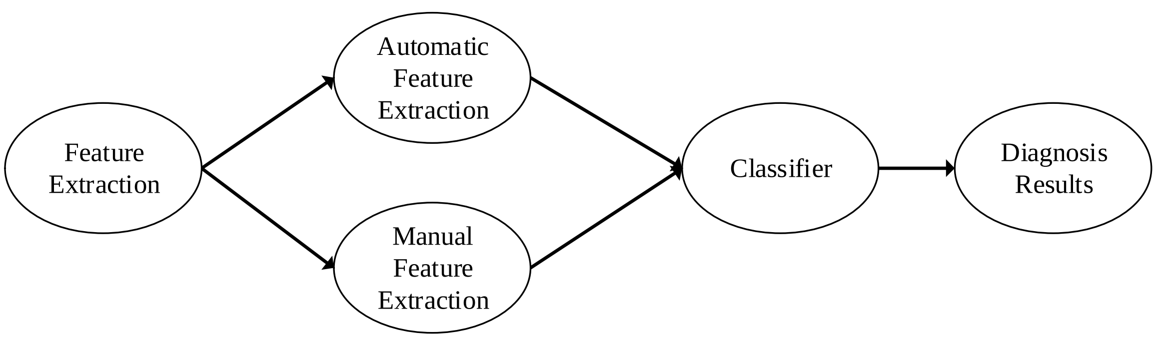

FIGURE 5.1.3: Different kinds of feature extraction

In the 1950s, linear models or the perceptron [69] were the simplest models in the artificial neural network which were inspired by human brains. In these models, information or data are transmitted or removed from the cell as electrical pulses or signals. These electrical signals from different neurons are entered into a core of neurons by dendrites. In the cell body, all inputs are added together, and then this data is processed to create a new signal which transmits along the cell's axon and sends to other neurons. During passing through the cell, some processes are done along the human's life which is similar to training the neural network over life. This structure of the human brain is interpreted to an artificial neural network for the computers. For example, in Figure 5.1.4, the n-dimensional input for the artificial neuron is assumed and the inputs are X_1,X_2,...,X_n, which are multiplied by a specific weight, W_1,W_2,...,W_n. The summation of weighted inputs and a bias are then passed through an activation function $F(z)$, which is a non-linear function, to create the output that is sent to other neurons. These linear classifiers could separate two different categories of inputs by learning the weights of inputs from each category [11, 8].

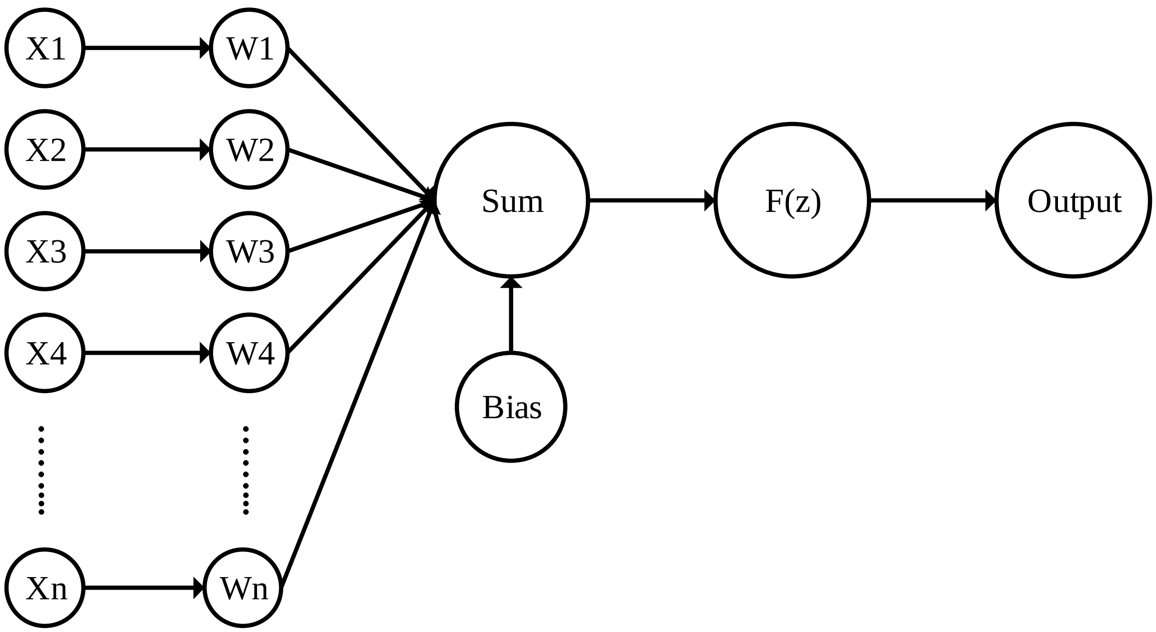

FIGURE 5.1.4: Schematic of a neuron in an artificial neural network

The below formula is used to show a neuron in an artificial neural network:
$u_k = \sum_{j=1}^{n} W_{kj} X_{kj}$
and
$y_k = f(u_k + b_k)$
In mathematical terms, $X_1, X_2, ..., X_n$ are input signals; $W_1, W_2..., W_n$ are the synaptic weights of neuron k; u_k is the linear combiner output due to the input signals; the bias is b_k ; $f(.)$ is the activation function; and the output signal of the neuron is y_k. Depending on whether the applied bias is positive or negative, respectively it can either increase or decrease the net input of the activation function, and change the output u_k [34].

In the Figure 5.1.4, the input to the j^{th} layer of the network is assummed as a vector $X = \begin{bmatrix} X_1 & X_2 & X_3 & ... & X_n \end{bmatrix}$ that propagates through the neurons and then the vector output $Y = \begin{bmatrix} Y_1 & Y_2 & ... & Y_n \end{bmatrix}$ will be produced by multiplying a weight matrix W and X and then add vector b. Therefore, the output of the neuron as $y_k = \sum_{j=1}^{n} W_{kj} X_{kj} + b_k$ is defined by the activation function $f(.)$, which implements a mathematical function on its input [11, 34].

5.2 Activation Function

There are various forms of nonlinear neurons in the hidden layer. In this section, three major types will be introduced. The first of these is the sigmoid neuron, which uses the following non-linear activation function: $f(z) = \frac{1}{1+e^{-z}}$
Its graph is s-shaped. It is also defined as a firmly increasing function that shows a smooth balance between linear and nonlinear actions. Its procedure has realized a breakdown, because its outputs are not zero-centred and it is likely to overload, which decreases its learning capacity. Another type of nonlinear neurons is Hyperbolic tangent neurons stating an s-shape neuron, the only difference is ranging boundary, the output of Hyperbolic tangent neurons range from -1 to 1 and it is zero-centered. Therefore, the Hyperbolic tangent neuron is often better than the sigmoid neuron [45]. Restricted linear unit (ReLU) neuron uses a different kind of nonlinearity with the function $f(z) = max(0, z)$, which states a specific hockey stick shaped response. And it changes the negative inputs to zero. A large number of neurons never influences the output of the neural network in this activation function. Therefore, it finds applications mostly in computer vision. The output layers, however, mostly use nodes with linear functions while input layer acts as a buffer.

5.3 Feed-Forward Networks

One layer in the neural network has one or more neurons [34]. There are three types of layers:

- **Input Layers**

They are connected to the inputs of the model.

- **Hidden Layers**

  They are not visible in the training set.

- **Output Layers**

  They present the output of the model.

The depth of the model is known as the number of layers, while the number of neurons in a layer is referred to as the width of the model. At least a depth of three including input, single hidden, and output layer is called deep learning. ANNs are divided into two layered network including feedforward and recurrent neural networks based on the connections between the layers. Figure 5.3.1 shows single-layer feed-forward networks, one of the simplest form of a layered network, which has an input layer that affects on an output layer of neurons. This network is firmly a feed-forward or acyclic type [34]. Moreover, input layers are not necessarily connected to one neuron. This means that these layers can be connected to multiple neurons with various weights. For instance, the three-dimensional input layer can be connected to four different hidden or output layer neurons. As it is illustalated in Figure 5.3.2, the inputs are mapped from three-dimensional to four-dimensional space which is considered as the features space. It means that the input mapped to a series of useful features. This process is the same as feature extraction. Feed-forward neural networks contain zero or more hidden layers, where all of the leaving connections from layer N will influence layer $N + 1$. While, recurrent neural networks learn from sequences instead of discrete training examples by using an additional feedback loop [35]. Nonlinear data can be classified by using deep networks [54]. In addition, the ideal instance of a deep learning model known as a multi-layer perceptron (MLP) or multilayer feed-forward networks that consist one or more hidden layers were formed. Hidden layers get involved in the external input and the network output in some useful manner. Useful features from the input can be learned by using hidden layers [11, 34]. For instance, as it is shown in Figure 5.3.2, the network is enabled to extract higher-order statistics [15]. In Figure 5.3.3, after feature extraction, features are used as inputs of the classification task to find the classes of the input. Classification method can be added as a layer to the network. This classifying a set of inputs is called forward propagation neural network. Higher-order statistics can be extracted by hidden neurons and when the size of the input layer is large, these layers are mainly valued [34]. This figure illustrates the framework of a multilayer feed-forward neural network for the case of a single hidden layer. In addition, this figure is considered as a 3-4-2 network because it has 3 input neurons, 4 hidden neurons, and 2 output neurons. Generally, in feed-forward networks, only the first layer is connected to the second layer which means that neurons of the same layer are not connected, and there are no connections that transmit data from the second layer to the first layer.

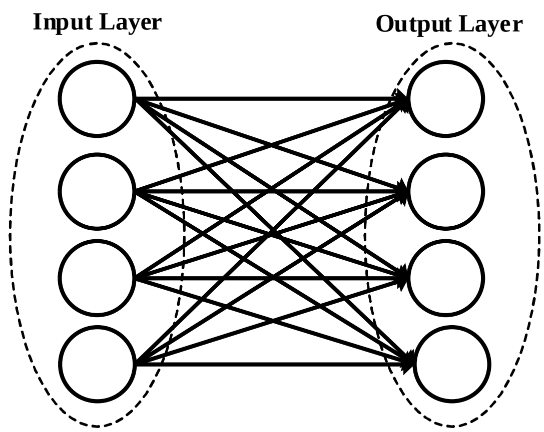

FIGURE 5.3.1: Single-layer feed-forward fully connected networks

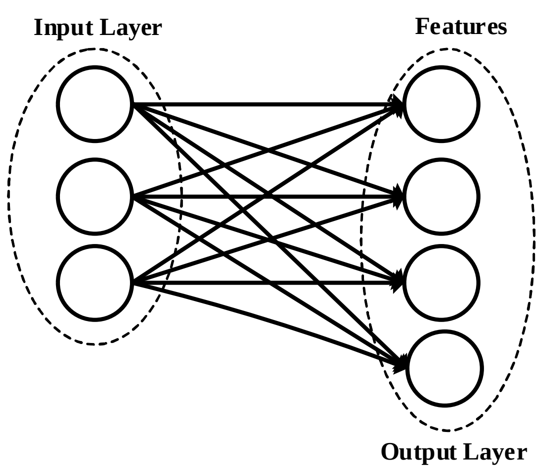

FIGURE 5.3.2: Example of single-layer feed-forward networks

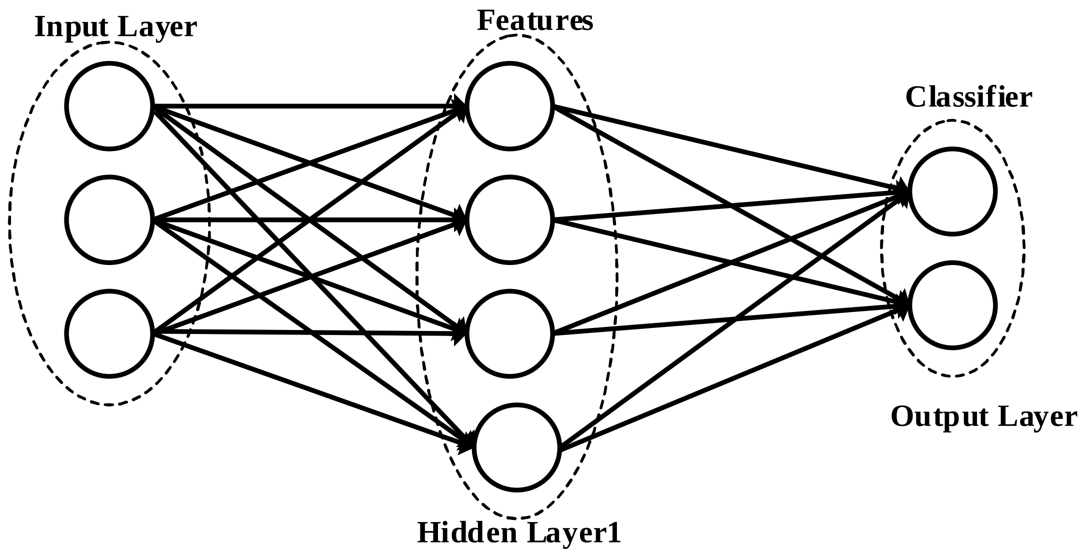

FIGURE 5.3.3: Multilayer feed-forward networks

5.4 Multiple Layer Perceptron (MLP)

Figure 5.5.1 represents multiple layers of perceptrons. This neural network shows that every neuron in each layer of the network is connected to every other neuron in the next forward layer and it is referred to as fully connected. However, the network is partially connected if some of these synaptic connections are missing from the network [34]. MLP networks are feed-forward direction or forward propagation neural networks which represent the relation between inputs and outputs and also consist multiple layers of neurons. The MLP structure consists three layers of input, multiple hidden, and output [11, 8]. In order to improve the performance of expectations, MLP networks are widely utilized for a variety of purposes, including pattern recognition, condition monitoring, fault diagnosis, function approximation, and many other purposes [85]. In order to learn weights of MLP networks, backpropagation (BP) will be applied. In BP, gradient descent will be used to minimize the square error of outputs of the network and target values. BP can show hidden layers in MLP.

5.5 Deep Feed-Forward Neural Networks

Deep neural networks (DNNs) or feed-forward neural networks are similar to MLP that contain multiple hidden layers. Feed-forward neural networks estimate a function $f(.)$, which represents the relation between the input vector x and the output vector y [29]. This means, the behaviour of the output layers is described by the training set from the values in the input layer. DNNs can be applied for both classification and regression problems. For early fault detection in industrial systems, unlabeled sensor data have been used in DNNs [4]. The normal operating data is applied to train the DNN in order to predict a measured parameter based on a wide range of measured features. Then, the model can make predictions of the measured parameters, which can be compared to the actual measurements of that parameter.

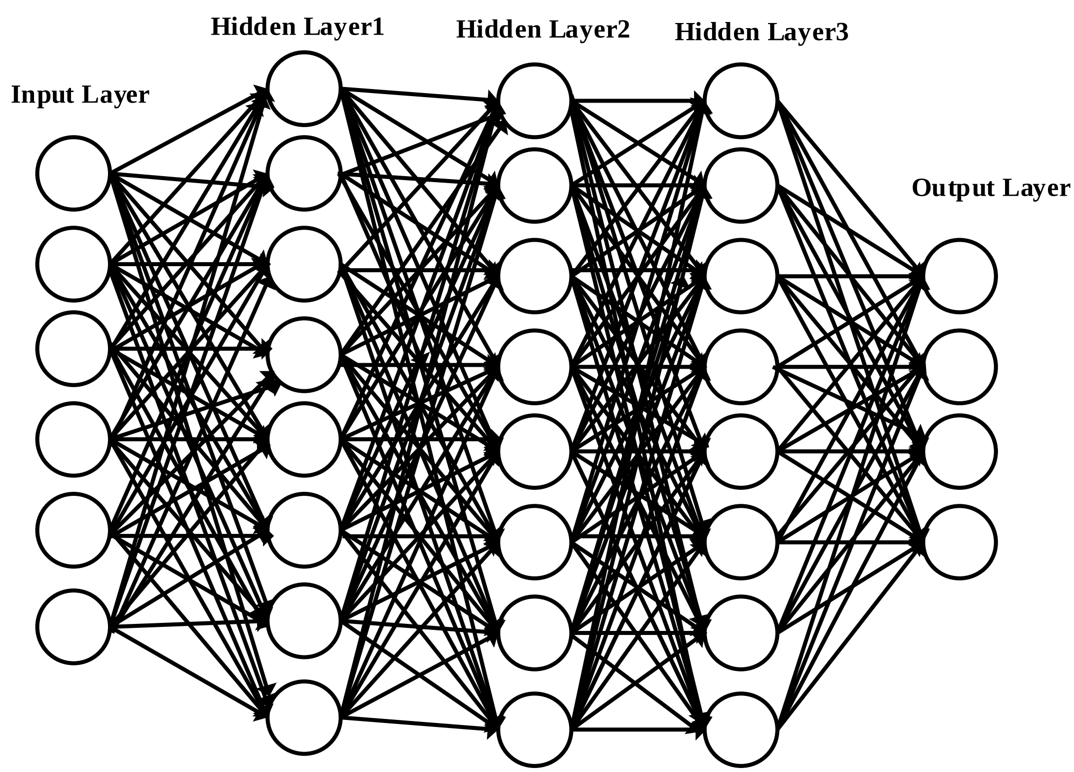

FIGURE 5.5.1: Multiple layer perceptron structure

5.6 Deep Learning Models

Computers are able to create more complicated and reliable models because of deep learning techniques, one representation of learning algorithms. The complex models are divided into sequences of simple patterns in deep learning models. In other words, in MLP structure, all neurons in each layer are fully connected to the others in the next layer. But, deep learning method applies various models for the connection of neurons, including stacked autoencoder (SAE), deep belief networks (DBN), convolutional neural networks (CNN), and recurrent neural networks (RNN). Deep learning models are widely functional for a variety of purposes in data mining, computer vision, video games, medical, natural language processing, and robotics. Deep learning has a large number of advantages that include the automatic learning of features, multi-layer features learning, high accuracy and generalization ability, hardware and software support, and the potential for more capabilities. On the other hand, the challenges of that are the weakness of the theory, high computational cost, requires vast amount of data, difficulty adjusting the parameters, and training problems like overfitting. In other words, deep learning is used to avoid overfitting in the training and to increase performance. Methods available in deep learning include supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. In this chapter, stacked autoencoder (SAE), which uses the unsupervised learning method in training, and makes use of a large amount of data to reflect hidden features, has been applied and compared with Support Vector Machine (SVM).

5.7 Autoencoder

Autoencoders are the first models in deep learning which use one of the learning algorithms of unsupervised learning. In addition, these models are one of the ideal examples of the representation learning algorithm. Simple autoencoders are feed-forward neural networks, containing an input layer, one or more hidden layers, and an output layer. Autoencoders are the mixture of the encoder and decoder functions. In most of the autoencoders, the input data is transformed into various features in feature space by using encoder function and then these new features are decoded to the original format. The only difference between autoencoders and feed-forward neural networks is the number of neurons in the output layer, which is equal to the number of neurons in the input layer. As a result, they learn an estimation of the activation function by reconstructing the input vector at the output as illustrated in Figure 5.7.1. For more explanation, encoding defines the mapping of the inputs to the various features and the mapping of these extracted features to the outputs is considered as decoding. Autoencoders calculate W_1,W_2,W_3,W_4 and b_1,b_2,b_3,b_4 by using stochastic gradient descent method. Autoencoders are divided into two models of linear and nonlinear. As it can be seen in Figure 5.7.1, 4-dimensional data is mapped into 2-dimensional space by using a neural network with one hidden layer, called linear autoencoder. In these autoencoders, the linear activation function is used. However, nonlinear or deep autoencoder is used for nonlinear data which requires more hidden layers adding to the network [54]. In addition, simple autoencoder with a single hidden sigmoid layer is comparable with its counterpart principal component analysis (PCA), a data preprocessing method.

Autoencoder Applications

- Denoising:

  Denoising autoencoders can remove noise from the input data to reconstruct data without noise at the output.

- Data compression:

  In this method, autoencoders can reduce the dimension and new data or features can be applied as compressed data. Moreover, effective features can be learned automatically from the data, but this only can compress similar patterns in order to be trained on. Furthermore, autoencoders can encode the random inputs, and cannot be applied in low-dimensional representations. Although, it can be applied for data compression, it is more normally used for data denoising and dimensionality reduction.

- Unsupervised learning:

  In this case, a number of useful features can be applied by using unsupervised learning (unlabeled data). Unsupervised learning is a machine learning process without human guidance in which only input data is accessible. The main use of that is finding predictabilities in the input without any labels. On the other hand, in supervised learning, supervisor creates some exact values related to the outputs in order to find the relation between the input and output [1]. This machine learning problem has a variety of applications including learning the essential similarities in the data and their clustering, feature extraction like dimensionality reduction, and so on.

Different kinds of Autoencoders:

- Stacked autoendocer
- Denoising autoendocer
- Sparse autoendocer
- Contractive autoendocer
- Convolutional autoendocer
- Variational autoendocer

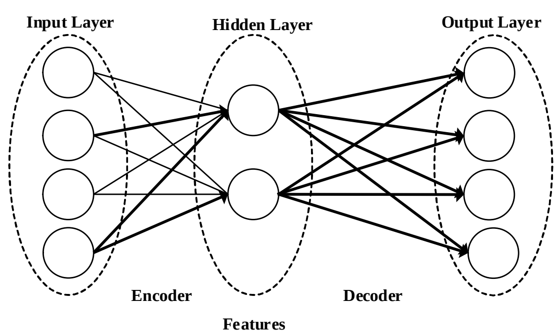

FIGURE 5.7.1: The general structure of Autoencoder

5.7.1 Deep Autoencoder (DAE)

Deep autoencoder (DAE) or stacked autoencoder (SAE) consists of several autoencoders arranged side by side which have several encoders and decoders as it is shown in Figure

5.7.2. SAE features can be trained , extracted by raw data, and retrained.

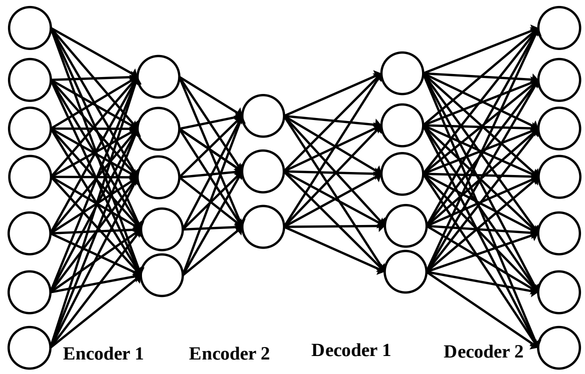

FIGURE 5.7.2: Deep autoencoder (DAE) structure

5.7.2 Training Autoencoder

The process of adjusting an important part, weights, and biases, in order to compare and match the expectation to the correct output is called training algorithm. Autoencoders are trained by this adjusting to minimize the reconstruction error between the input vector x and its reconstruction at the output vector $\overline{x}$ [22]:

$$min\|x - \overline{x}\|^2$$

In order to solve the challenges in the training of a deep autoencoder, greedy layer-wise training algorithm is introduced in which each layer of the network is trained individually in one autoencoder and then the training layers are stacked together. This algorithm of training can build the better network by using a large number of unlabeled data and identify the better parameter space for the weights of each layer after training [84].

Greedy layer-wise training algorithm

The first layer is shown in Figure 5.7.3, which is a simple autoencoder with three different layers including input layer x, hidden layer or features a, and output layer $\overline{x}$ as reconstructed inputs. Then, this autoencoder is being trained in such a way that some features a are created by x in encoder part and in the decoder part, these features are decoded to create

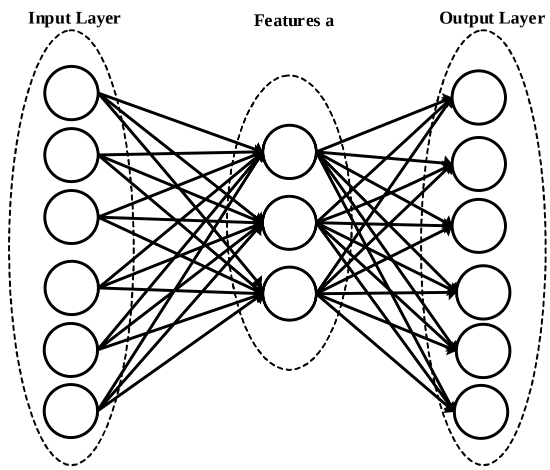

FIGURE 5.7.3: The first layer

$\overline{x}$. After the network has been trained, these suitable features (features a) are being produced, and then the decoder part is set aside. As it is shown in Figure 5.7.4, if it is needed to add a new layer to the network, the main input is set aside and made another autoencoder with features a. This network is being trained to turn inputs (features a) into features b, and then features a can be reconstructed by the features b. Similarly, after the training of the network and setting aside the decoder part, features b will be created. Figure 5.7.5 shows one trained deep autoencoder as two-layer features are extracted from the inputs.

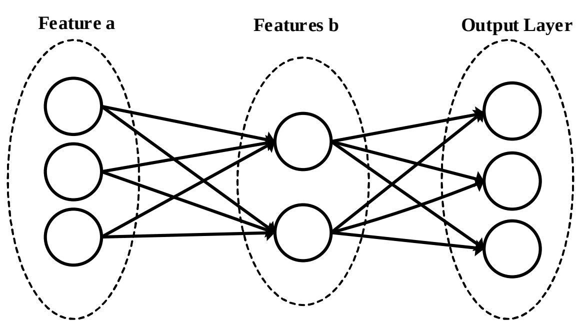

FIGURE 5.7.4: The second layer

Deep Autoencoder Applications [54]:

- **Feature Extraction using unsupervised data**

  This method is a pretraining step in which deep autoencoder is trained, using unsupervised data, and finally, a number of features are extracted from the inputs. Feature

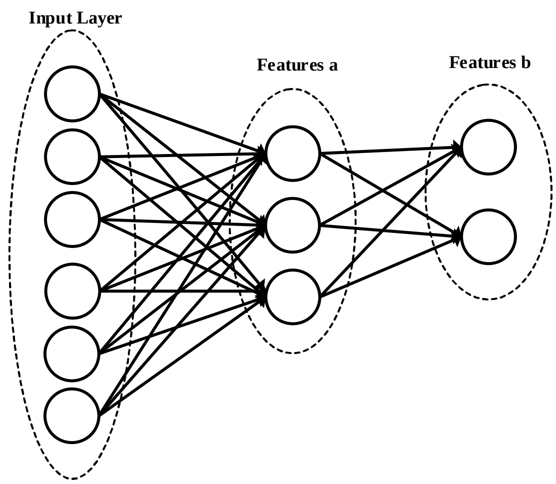

FIGURE 5.7.5: Trained deep autoencoder

extraction for autoencoders can also be named dimensionality reduction.

- **Fine-tuning of a pre-trained network using supervised leaning**

  In this method, a pre-trained network or the last layer is trained again by using labeled data to solve classification problems and it can improve the performance of deep neural network. This method is illustrated in Figure 5.7.6.

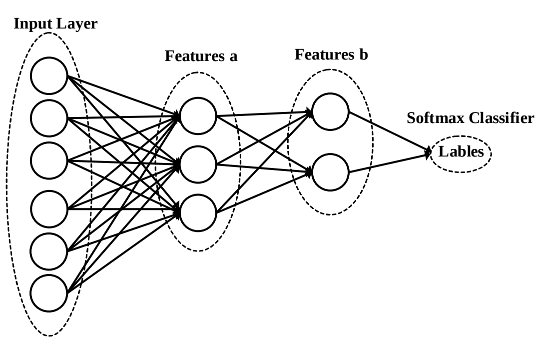

FIGURE 5.7.6: Fine-tune algorithm

- **Reconstructing Data or Denoising Autoencoder**

  As it can be seen in Figure 5.7.7, the decoder part is added to the encoder and the inputs are given and outputs reconstruct the data. Usually, this is a network-style for data modification, such as input noise data and as a result, data output includes no noise. In general, denoising autoencoder learns to perform a noise cancellation process. Backpropagation algorithm (BP) is used in these autoencoders. It means that this algorithm is able to set the output values to be equal to the input values.

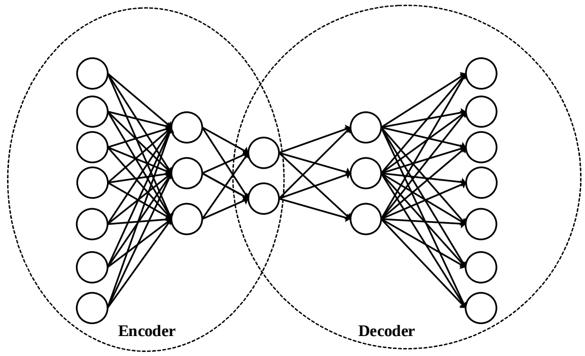

FIGURE 5.7.7: Denoising autoencoder

In this chapter, the unsupervised pretraining method, and fine-tune algorithm using supervised classification network based on the softmax function, the output, have been applied. The supervised learning stage can decrease the training error by applying a small amount of labeled data.

Softmax Classifier

As it can be seen, SAE can be connected with two kinds of classifiers, named logistic classifier and softmax classifier, to complete the network. In order to have more accurate predictions, a special layer called a softmax output layer, which is commonly used in neural networks for multi-class classification, can be applied. On the other hand, the logistic classifier can be used for binary classification [84]. Probability distribution is used in this classifier. Therefore, the desired output vector is as below, where [11]

$\sum_{i=0}^{9} P_i = 1$

$\begin{bmatrix} P_0 & P_1 & P_2 & ... & P_9 \end{bmatrix}$

As it can be seen in the above formula, the sum of all the outputs should be equal to 1. As a result, the outputs of all the other neurons affects on the output of a neuron in a softmax layer. Assume z_i be the logit of the i^{th} softmax neuron, set its output, and this normalization is achieved:

$y_i = \frac{e^{z_i}}{\sum_j e^{z_i}}$

A probability distribution over a set of mutually exclusive labels is mostly employed in Image Recognition. In general, deep learning based on fault diagnosis is put forward, cosists of feature learning by stacked sparse autoencoder and fault classification by softmax classifier. Back Propagation optimization algorithm is also used to train the softmax classifier [67].

5.8 The Comparison between two different methods of Induction Motor Fault Diagnosis

In this section, two methods are applied in induction motor fault diagnosis, including deep autoencoder, and Support Vector Machine (SVM). Two different parts of the induction motor, including rotor bar and bearing, are used to simulate fault diagnosis models. These parts are discussed in two case studies. As it is mentioned before, the main application of deep autoencoder is extracting useful data from a large amount of unlabeled data and preprocessing with it. The model can be trained with that autoencoder which uses softmax classifier (supervised learning) to do classification task. Moreover, it can be compared with SVM, a supervised learning algorithm, which is one of the nonlinear detection methods. In other words, autoencoders are used for feature extraction and SVMs for anomaly detection.

5.8.1 Case Study I

Availability of Data

In this study, a three-phase, 1.2 KW, 380 volts, 50 Hz, 1400 rpm, the four-pole induction motor is used to collect experimental data. Broken rotor bars detection is described in this case study. The stator current signal is recorded in three different conditions, normal, one broken bar and two broken bars. Figure 5.8.1 and Figure 5.8.2 show different conditions of the stator current signal.

Discussion and Comparison of Results

This acquired rotor bar data contains three classes including data of normal operation, data of one broken bar, and data of two broken bars. In this simulation experiment, the stator current signal is segmented into 15000 samples. 100 points of current are recorded for each

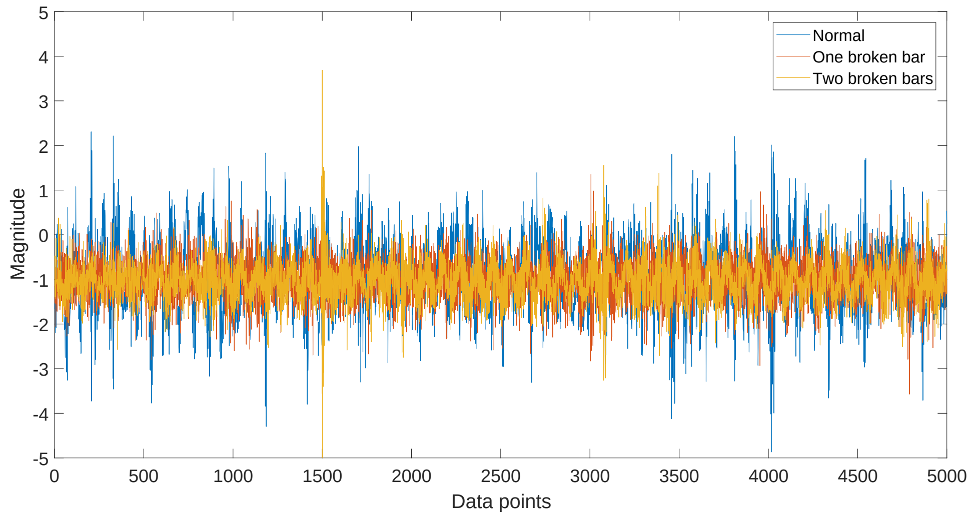

FIGURE 5.8.1: The comparison between various conditions

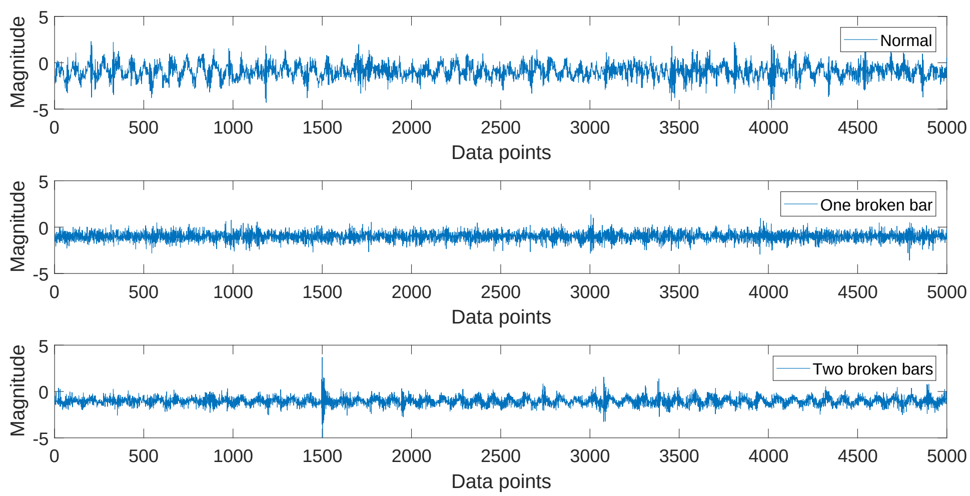

FIGURE 5.8.2: The comparison between various conditions

TABLE 5.8.1: Parameters in experiments

Parameters	Value
Input Units	100
Output Units	3
Number of Hidden Layer	2
Number of Neurons In Each Hidden Layer	[70,10]

TABLE 5.8.2: Samples of simulation experiment

Number of Hidden Layer	Hidden Layer I	Hidden Layer II	Accuracy (%)
2	50	10	62.6
2	70	10	84.6
2	70	5	74.9

sample before fault or during normal or healthy operation. Then, 100 points of current are entered into a network as the input layer. So, this layer contains 100 neurons. The number of output neurons is given by 3 since the data is divided into three classes. This network of fault diagnosis in a simulation is shown in Figure 5.8.3.

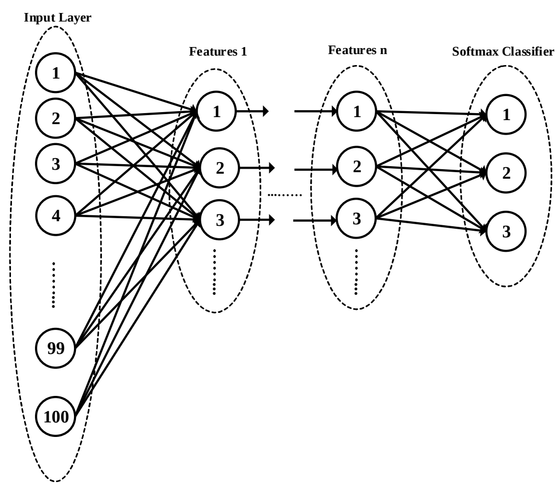

FIGURE 5.8.3: Deep fault diagnosis model

Table 5.8.2 represents the best parameters chosen for the experiment. The fault diagnosis performance is shown by accuracy rate of diagnosis, which is calculated by the different number of each hidden layer. In Matlab environment, these simulation experiments represent that three hidden layers cannot considerably improve the accuracy, only two hidden layers of autoencoder are enough. Moreover, this table shows the power of a number of each hidden layer on accuracy rate of diagnosis.

Multi-Class Confusion Matrices Case Study I

A confusion matrix is applied to summarize the performance of a classification algorithm. If there are more than two classes in the data, classification accuracy cannot represent what the classification model is getting right and what types of errors it is making. In other words, the performance of your model cannot be diagnosed. As a result, confusion matrix can be calculated. Classification accuracy can be measured by the ratio of correct predictions to total predictions made. It is normally shown as a percentage by multiplying the result by 100. Misclassification rate or error rate can be calculated by

Error rate = 1 - classification accuracy.

The main aim of the confusion matrix is summarizing the number of correct and incorrect predictions of each class. Moreover, it shows not only the errors made by the classifier but also, the types of errors are diagnosed. Figure 5.8.4, 5.8.5, and 5.8.6 show confusion matrices obtained from the Table 5.8.2, respectively. These diagnosis confusion matrices represent how the classification of the different conditions is done.

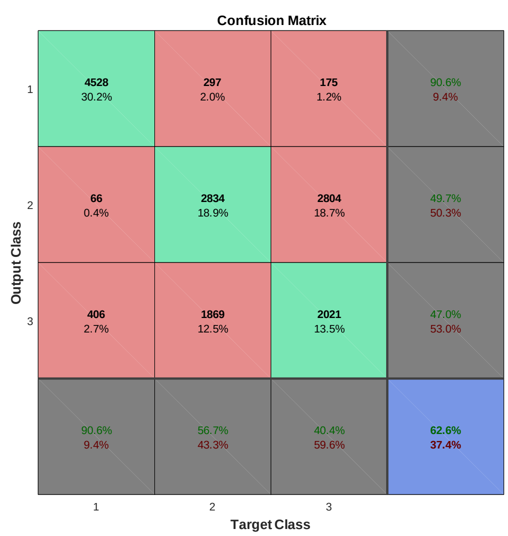

FIGURE 5.8.4: Confusion matrix I

Figure 5.8.5 represents the best number of each hidden layer, the first hidden layer has 70 neurons and the second one has 10 neurons. In this figure, the number and percentage of correct classifications by the trained network are shown by the first two diagonal cells. For instance, 4752 samples are correctly classified as a class of normal. This corresponds

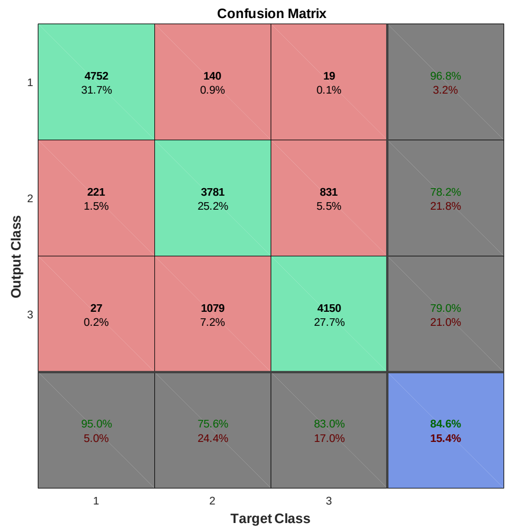

FIGURE 5.8.5: Confusion matrix II

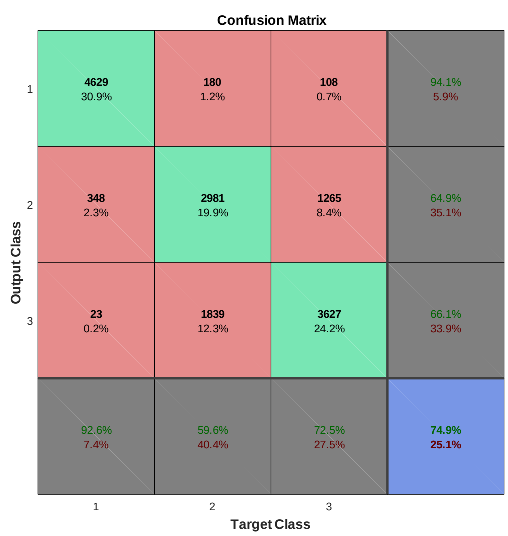

FIGURE 5.8.6: Confusion matrix III

to 31.7% of all 15000 samples. Similarly, 3781 cases are correctly classified as one broken bar. This corresponds to 25.2% of all samples. In addition, 4150 cases are correctly classified as two broken bars. This corresponds to 27.7% of all samples. 140 of the normal condition are incorrectly classified as one broken bar and this corresponds to 0.9% of all 15000 samples in the data. And, 19 of the normal condition are incorrectly classified as two broken bars and this corresponds to 0.1% of all 15000 samples in the data. 221 of the normal condition are incorrectly classified as one broken bar and this corresponds to 1.5% of all data. Also, 831 of two broken bars are incorrectly classified as one broken bar and this corresponds to 5.5% of all data. Similarly, 27 of normal operation are incorrectly classified as two broken bars and this corresponds to 0.2% of all data. 1079 of one broken bar are incorrectly classified as two broken bars and this corresponds to 7.2% of all data. Out of 4911 normal predictions, 96.8% are correct and 3.2% are wrong. Out of 4833 one broken bar predictions, 78.2% are correct and 21.8% are wrong. Out of 5256 two broken bars predictions, 79% are correct and 21% are wrong. Out of 5000 normal cases, 95% are correctly predicted as normal and 5% are predicted as other classes. Out of 5000 one broken bar cases, 75.6% are correctly classified as one broken bar and 24.4% are classified as other classes. Out of 5000 two broken bars cases, 83% are correctly classified as two broken bars and 17% are classified as other classes. Overall, 84.6% of the predictions are correct and 15.4% are wrong classifications.

The comparison between deep autoencoder and SVM

Table 5.8.3 represents the results obtained by deep autoencoder and SVM. As it is illustrated, in deep autoencoder, 96.8% are correctly classified as the normal class, 78.2% as one broken bar, 79.0% as two broken bars, and overall, its accuracy is 84.6%. On the other hand, in SVM, 87.71% are correctly classified as the normal, 69.02% as one broken bar, 83.09% as two broken bars, and overall, its accuracy is 79.94%. As a result, deep fault diagnosis model has a better performance in diagnosing different faults occurring in rotor compared with SVM.

TABLE 5.8.3: The comparison between deep autoencoder and SVM

Broken Rotor Bar Conditions	Deep Autoencoder (Accuracy %)	SVM (Accuracy %)
Normal condition	96.8	87.71
One broken bar	78.2	69.02
Two broken bars	79.0	83.09
Overall	84.6	79.94

5.8.2 Case Study II

Availability of Data

Ball bearing data from Case Western Reserve University is used. The experimental setup consisted of a 2hp (horsepower) motor (1750 rpm), a torque converter/encoder, a dynamometer and control circuits. Vibration signals considered in this study include the normal, an inner race fault, and outer race fault signals were sampled at the 12kHz frequency. Drive end accelerometer data with fault diameter of 0.07 inch is studied. Figure 5.8.7 and 5.8.8 show different conditions of the vibration signal.

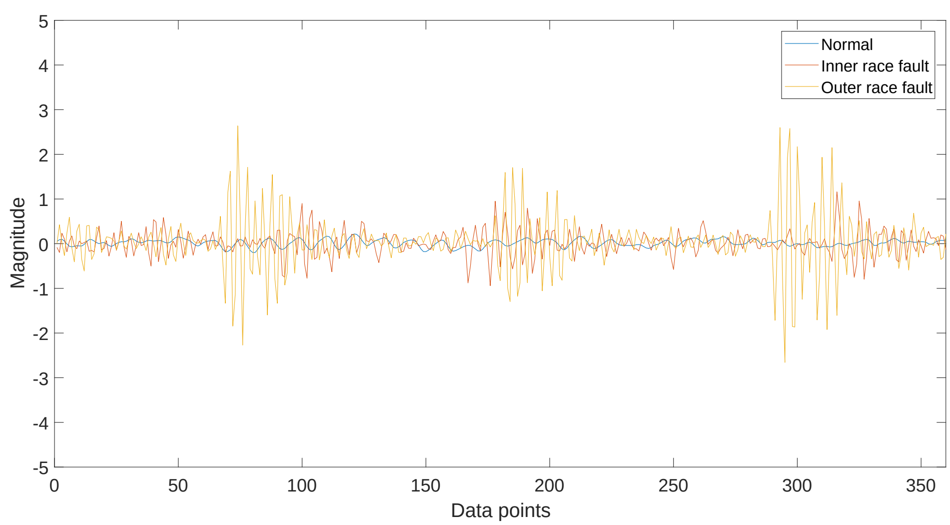

FIGURE 5.8.7: The comparison between various conditions

Discussion and Comparison of Results

This acquired bearing data contains three classes including data of normal operation, data of inner race fault, and data of outer race fault. In this simulation experiment, the vibration signal is segmented into 1080 samples. 100 points are recorded for a sample before fault

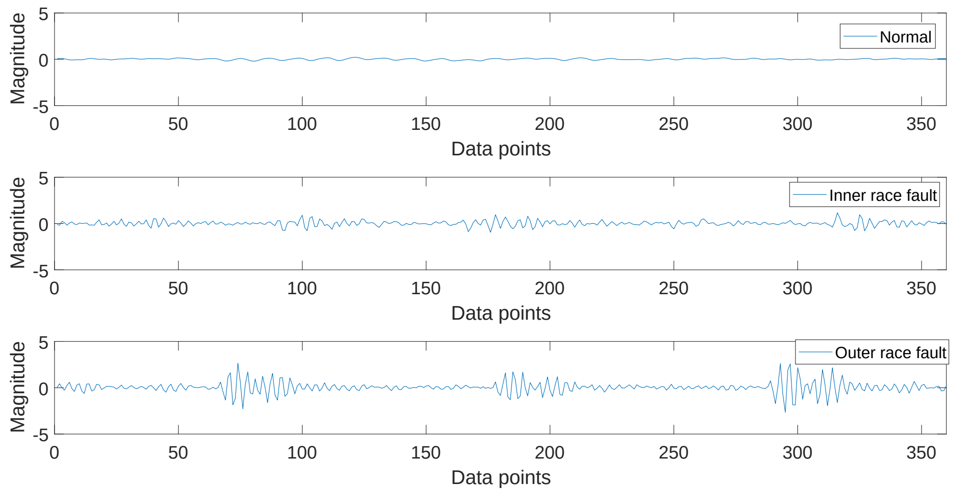

FIGURE 5.8.8: The comparison between various conditions

TABLE 5.8.4: Parameters in experiments

Parameters	Value
Input Units	100
Output Units	3
Number of Hidden Layer	2
Number of Neurons In Each Hidden Layer	[70,5]

or during normal operation. Then, 100 points of each sample are entered into a network as the input layer. So, this layer contains 100 neurons. The number of output neurons is given by 3 since the data is divided into three classes.

Multi-Class Confusion Matrices Case Study II

Table 5.8.4 represents the best parameters chosen for the experiment. The fault diagnosis performance is shown by accuracy rate of diagnosis, which is calculated by the different number of each hidden layer. In Matlab environment, these simulation experiments represent that three hidden layers cannot considerably improve the accuracy, only two hidden layers of autoencoder are enough. Moreover, this table shows the power of a number of each hidden layer on accuracy rate of diagnosis.

Fig 5.8.11 represents the best number of each hidden layer, the first hidden layer has 70 neurons and the second one has 5 neurons. In this figure, 360 samples are correctly classified as a normal condition. This corresponds to 33.3% of all 1080 samples. Similarly,

TABLE 5.8.5: Samples of simulation experiment

Number of Hidden Layer	Hidden Layer I	Hidden Layer II	Accuracy (%)
2	50	10	98.1
2	70	10	98.4
2	70	5	99.0

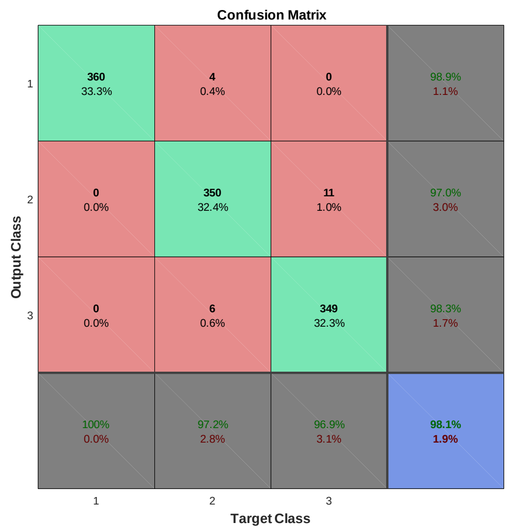

FIGURE 5.8.9: Confusion matrix I

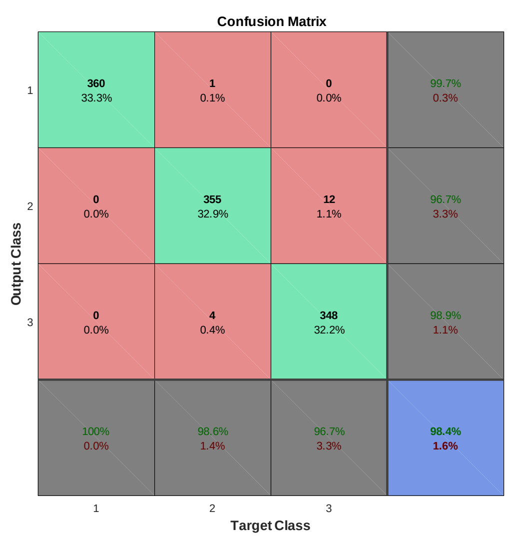

FIGURE 5.8.10: Confusion matrix II

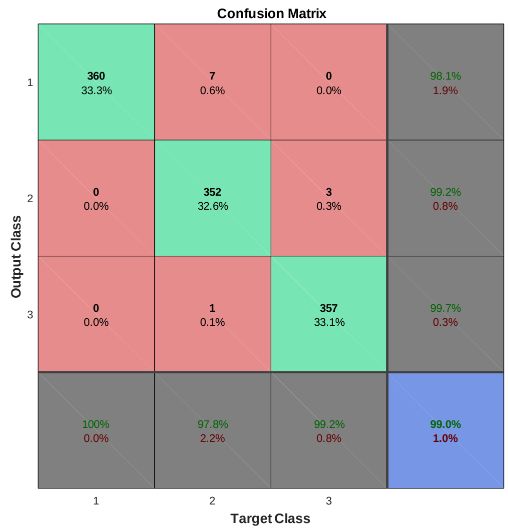

FIGURE 5.8.11: Confusion matrix III

352 cases are correctly classified as inner race fault. This corresponds to 32.6% of all samples. In addition, 357 cases are correctly classified as outer race fault. This corresponds to 33.1% of all samples. Seven of the normal class are incorrectly classified as inner race fault and this corresponds to 0.6% of all 1080 samples in the data. And, none of the normal conditions are incorrectly classified as outer race fault. None of the normal conditions are incorrectly classified as inner race fault. Also, three of outer race fault. are incorrectly classified as inner race fault. and this corresponds to 0.3% of all data. Similarly, none of the normal conditions are incorrectly classified as outer race fault. One of the inner race fault is incorrectly classified as outer race fault and this corresponds to 0.1% of all data. Out of 367 the normal predictions, 98.1% are correct and 1.9% are wrong. Out of 355 inner race fault predictions, 99.2% are correct and 0.8% are wrong. Out of 358 outer race fault predictions, 99.7% are correct and 0.3% are wrong. Out of 360 the normal cases, 100% are correctly predicted as the normal. Out of 360 inner race fault cases, 97.8% are correctly classified as inner race fault and 2.2% are classified as other classes. Out of 360 outer race fault cases, 99.2% are correctly classified as outer race fault and 0.8% are classified as other classes. Overall, 99% of the predictions are correct and 1% are wrong classifications.

TABLE 5.8.6: The comparison between deep autoencoder and SVM

Bearing Conditions	Deep Autoencoder (Accuracy %)	SVM (Accuracy %)
Normal condition	98.1	100
Inner race fault	99.2	26.85
Outer race fault	99.7	71.30
Overall	99.0	66.05

The comparison between deep autoencoder and SVM

Table 5.8.6 shows the results obtained by Deep Autoencoder and SVM. As it is illustrated, in Deep Autoencoder, 98.1% are correctly classified as the normal condition, 99.2% as inner race fault, 99.7% as outer race fault, and overall, its accuracy is 99%. On the other hand, in SVM, 100% are correctly classified as the normal, 26.85% as inner race fault, 71.30% as outer race fault, and overall, its accuracy is 66.05%. Therefore, deep fault diagnosis model has a better performance in diagnosing different faults occurring in bearing compared with SVM.

CHAPTER 6

Conclusions and Future Works

6.1 Contributions

This thesis dealt with fault detection and diagnosis approaches in IMs. These approaches were proposed in the thesis in order to solve major problems with high computational cost of pre-processing techniques along with anomaly technique, ensemble-based anomaly detection, and learn the deep architectures of fault data by using a deep neural network. Moreover, various novelty detection techniques are applied to broken rotor bars and bearing faults by analyzing the stator current and vibration signals. In Chapter 3, pre-processing tasks including feature extraction and feature selection were followed with the one-class classification techniques to detect broken rotor bars in IMs. The results showed that the combination of feature selection and kNN one-class classifier provides the highest accuracy among all other techniques. From the experimental results, it was concluded that the proposed method can detect broken rotor bars with about 0.99 percent accuracy. Chapter 4 studied the use of ensemble techniques for fault detection in IMs. The system was specifically designed to detect and identify broken rotor bars in IMs. For this purpose, one-class classification techniques were used to construct the ensemble. The proposed scheme included a pre-processing step to extract and select proper sets of features. Then, five OCCs, including GD, PD, NN, kNN, and k-means were applied to train ensemble schemes. Three methods of random subspace, bagging, and boosting were applied for combining the classifiers. It was shown that bagging of homogenous kNN and five heterogenous OCCs outperform other models and result in a promising detection accuracy. The established fault detection and diagnosis system in this chapter was capable of detecting broken rotor bars in order to enhance the benefits of IMs. Chapter 5 presented the application of deep neural network for IMs fault diagnosis. Two different parts of the induction motor, including rotor bar and bearing, were used to simulate fault diagnosis models. Deep autoencoder and support vector machine (SVM) were applied and compared with each other in order to simulate these fault diagnosis models. Deep autoencoder could extract suitable features that were initially ignored by statistic techniques. In addition, the deep autoencoder has a strong learning ability for detecting different kinds of faults by using a softmax classifier. Softmax classifier reflects the types and possibility of diagnosis. As a result, this deep learning technique can improve the performance of fault detection and diagnosis in IMs. It was concluded that the performance of deep neural network was generally better than SVM. Due to a high computational load in the presence of a large number of training data and the absence of control over the number of data, SVM cannot perform accurately on the some datasets.

6.2 Future Works

In this study, the focus was on the condition monitoring of the induction motor and for that, many algorithms and techniques were studied and applied. In order to create a more accurate design and optimization of other electrical machines, more advanced methods are introduced by the researchers for numerous purposes. Therefore, a significant future work is investigating other signal processing techniques for all aspects of data pre-processing, including short-time Fourier transform (STFT), wavelet analysis, and empirical mode decomposition (EMD). Moreover, the applications and comparisons of different methods in a deep neural network including a deep belief network (DBN) and a convolutional neural network (CNN) in other electrical machines fault diagnosis can be applied. In addition, the future studies can also focus on fault prognosis and Remaining Useful Life estimation (RUL). Fault prognosis and Remaining Useful Life estimation (RUL) depend on the data availability. In other words, when a specific fault has occurred, the trend of data can be classified as the specific fault and models can be trained to detect it as opposed to normal behavior trends in fault detection and diagnosis step. Then, faulty data shows a degradation of the system or components performance, it can be used to estimate remaining useful life of various components which may lead to prognostic, predictive maintenance, and finally reduction of operational costs.

REFERENCES

[1] Alpaydin, E. (2014). *Introduction to machine learning*. MIT press.

[2] Alves, V., Oliveira, A. L., Silva Jr, E., and Mello, C. A. (2008). Handwritten digit segmentation in images of historical documents with one-class classifiers. In *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on*, volume 2, pages 41–44. IEEE.

[3] Bahler, D. and Navarro, L. (2000). Methods for combining heterogeneous sets of classifiers. In *17th Natl. Conf. on Artificial Intelligence (AAAI), Workshop on New Research Problems for Machine Learning*.

[4] Bangalore, P. and Tjernberg, L. B. (2015). An artificial neural network approach for early fault detection of gearbox bearings. *IEEE Transactions on Smart Grid*, 6(2):980–987.

[5] Baraldi, P., Razavi-Far, R., and Zio, E. (2011). Bagged ensemble of fuzzy c-means classifiers for nuclear transient identification. *Annals of Nuclear Energy*, 38(5):1161–1171.

[6] Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1-2):105–139.

[7] Bellini, A., Concari, C., Franceschini, G., Lorenzani, E., Tassoni, C., and Toscani, A. (2006). Thorough understanding and experimental validation of current sideband components in induction machines rotor monitoring. In *IEEE Industrial Electronics, IECON 2006-32nd Annual Conference on*, pages 4957–4962. IEEE.

[8] Bengio, Y., Goodfellow, I. J., and Courville, A. (2015). Deep learning. *Nature*, 521:436–444.

[9] Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

[10] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

[11] Buduma, N. and Locascio, N. (2017). *Fundamentals of Deep Learning: Designing Next-generation Machine Intelligence Algorithms*. " O'Reilly Media, Inc.".

[12] Butler, S. (2012). Prognostic algorithms for condition monitoring and remaining useful life estimation. *National University of Ireland, Maynooth*.

[13] Cabral, G. G., Oliveira, A. L., and Cahú, C. B. (2009). Combining nearest neighbor data description and structural risk minimization for one-class classification. *Neural Computing and Applications*, 18(2):175–183.

[14] Cho, K. R., Lang, J. H., and Umans, S. D. (1992). Detection of broken rotor bars in induction motors using state and parameter estimation. *IEEE Transactions on Industry Applications*, 28(3):702–709.

[15] Churchland, P. and Sejnowski, T. (1992). Computational neuroscience. the computational brain.

[16] Ding, S. X. (2008). *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Science & Business Media.

[17] Donnel, P. (1987). Report of large motor reliability survey of industrial and commercial installations: Part i and ii. *IEEE Trans. IA*, 23(1):153–158.

[18] Dorrell, D. G., Thomson, W. T., and Roach, S. (1997). Analysis of airgap flux, current, and vibration signals as a function of the combination of static and dynamic airgap eccentricity in 3-phase induction motors. *IEEE Transactions on Industry applications*, 33(1):24–34.

[19] Dreyfus, G. (2005). *Neural networks: methodology and applications*. Springer Science & Business Media.

[20] Duda, R. O. and Hart, P. E. (1973). Pattern recognition and scene analysis.

[21] Duin, R. P. and Tax, D. M. (2000). Experiments with classifier combining rules. In *International Workshop on Multiple Classifier Systems*, pages 16–29. Springer.

[22] Erfani, S. M., Rajasegarar, S., Karunasekera, S., and Leckie, C. (2016). High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134.

[23] Filippetti, F., Franceschini, G., and Tassoni, C. (1995). Neural networks aided on-line diagnostics of induction motor rotor faults. *IEEE Transactions on Industry Applications*, 31(4):892–899.

[24] Firoozian, R. and Lee, C. (1994). Condition monitoring of automatic control systems using parameter and systems identification: a numerical investigation. *Transactions of the Institute of Measurement and Control*, 16(2):86–98.

[25] Freund, Y. and Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer.

[26] Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *Icml*, volume 96, pages 148–156. Bari, Italy.

[27] García-Escudero, L. A., Duque-Perez, O., Morinigo-Sotelo, D., and Perez-Alonso, M. (2011). Robust condition monitoring for early detection of broken rotor bars in induction motors. *Expert Systems with Applications*, 38(3):2653–2660.

[28] Gelman, L., Jennions, I., and Petrunin, I. (2011). Detection of chipped tooth in gears by the novel vibration residual technology. *International Journal of Prognostics and Health Management*, 2(2):32–37.

[29] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

[30] Hajnayeb, A., Ghasemloonia, A., Khadem, S., and Moradi, M. (2011). Application and comparison of an ann-based feature selection method and the genetic algorithm in gearbox fault diagnosis. *Expert Systems with Applications*, 38(8):10205–10209.

[31] Hall, M. (1998). Correlation-based feature subset selection for machine learning. *Thesis submitted in partial fulfillment of the requirements of the degree of Doctor of Philosophy at the University of Waikato*.

[32] Hargis, C., Gaydon, B., and Kamash, K. (1982). The detection of rotor defects in induction motors. In *Proc. 1982 IEE Int. Conf. Electrical Machines, Design and Application*, pages 216–220.

[33] Harlisca, C. and Szabó, L. (2012). Bearing faults condition monitoring-a literature survey. *Journal of Computer Science and Control Systems*, 5(2):19.

[34] Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition.

[35] Haykin, S. S., Haykin, S. S., Haykin, S. S., and Haykin, S. S. (2009). *Neural networks and learning machines*, volume 3. Pearson Upper Saddle River, NJ, USA:.

[36] Heng, A. S. Y. (2009). *Intelligent prognostics of machinery health utilising suspended condition monitoring data*. PhD thesis, Queensland University of Technology.

[37] Hill, D. J., Minsker, B. S., and Amir, E. (2007). Real-time bayesian anomaly detection for environmental sensor data. In *Proceedings of the Congress-International Association for Hydraulic Research*, volume 32, page 503. Citeseer.

[38] Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844.

[39] Innes, A. and Langman, R. (1994). The detection of broken rotor bars in variable speed induction motor drives. *ICEM94*, pages 558–565.

[40] Isermann, R. (2006). *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media.

[41] Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666.

[42] Jardine, A. K., Lin, D., and Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, 20(7):1483–1510.

[43] Karacali, B. and Krim, H. (2003). Fast minimization of structural risk by nearest neighbor rule. *IEEE transactions on neural networks*, 14(1):127–137.

[44] Karmakar, S., Chattopadhyay, S., Mitra, M., and Sengupta, S. (2016). Induction motor fault diagnosis: General discussion and research scope. In *Induction Motor Fault Diagnosis*, pages 153–158. Springer.

[45] Karpathy, A. (2016). Cs231n convolutional neural networks for visual recognition. *Neural networks*, 1.

[46] Khan, S. S. (2010). Kernels for one-class nearest neighbour classification and comparison of chemical spectral data. *College of Engineering and Informatics, National University of Ireland*.

[47] Kim, H.-E. (2010). *Machine prognostics based on health state probability estimation*. PhD thesis, Queensland University of Technology.

[48] Kim, K. and Parlos, A. G. (2002). Induction motor fault diagnosis based on neuro-predictors and wavelet signal processing. *IEEE/ASME Transactions on mechatronics*, 7(2):201–219.

[49] Kliman, G. and Stein, J. (1990). Induction motor fault detection via passive current monitoring-a brief survey. In *Proc. 44th Meeting of the Mechanical Failures Prevention Group*, pages 49–65. DTIC Document.

[50] Kliman, G. B. (1991). Spectral analysis of induction motor current to detect rotor faults with reduced false alarms. US Patent 5,049,815.

[51] Kotsiantis, S. and Pintelas, P. (2004). Combining bagging and boosting. *International Journal of Computational Intelligence*, 1(4):324–333.

[52] Kriegel, H.-P., Kröger, P., and Zimek, A. (2010). Outlier detection techniques. *Tutorial at KDD*, 10.

[53] Kriegel, H.-P., Zimek, A., et al. (2008). Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 444–452. ACM.

[54] Le, Q. V. et al. (2015). A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks. *Google Brain*.

[55] Leonard, R. (1984). Vibration and stray flux monitoring for unbalanced supply and inter-turn winding fault diagnosis in induction motors. *Condition Monitoring84*, pages 340–354.

[56] Li, B., Chow, M.-Y., Tipsuwan, Y., and Hung, J. C. (2000). Neural-network-based motor rolling bearing fault diagnosis. *IEEE transactions on industrial electronics*, 47(5):1060–1069.

[57] Li, X. and Zhao, H. (2009). Weighted random subspace method for high dimensional data classification. *Statistics and its Interface*, 2(2):153.

[58] Lim, W. L. (2009). *The application of artificial neural networks for sensor validation in diesel engine condition monitoring and fault diagnosis*. PhD thesis, University of Newcastle upon Tyne.

[59] Littlestone, N. and Warmuth, M. K. (1989). The weighted majority algorithm. In *Foundations of Computer Science, 1989., 30th Annual Symposium on*, pages 256–261. IEEE.

[60] Marsland, S. (2003). Novelty detection in learning systems. *Neural computing surveys*, 3(2):157–195.

[61] Martinus, D. and Tax, J. (2001). One-class classification: Concept-learning in the absence of counterexamples. *Delft University of Technology*.

[62] Marwala, T. (2012). *Condition monitoring using computational intelligence methods: applications in mechanical and electrical systems*. Springer Science & Business Media.

[63] Mohanty, A. R. (2014). *Machinery condition monitoring: Principles and practices*. CRC Press.

[64] Park, H. G. and Zak, M. (2003). Gray-box approach for fault detection of dynamical systems. *Journal of dynamic systems, measurement, and control*, 125(3):451–454.

[65] Pearl, J. (1984). Heuristics: intelligent search strategies for computer problem solving.

[66] Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45.

[67] Qi, Y., Shen, C., Wang, D., Shi, J., Jiang, X., and Zhu, Z. (2017). Stacked sparse autoencoder-based deep network for fault diagnosis of rotating machinery. *IEEE Access*, 5:15066–15079.

[68] Razavi-Far, R., Farajzadeh-Zanjani, M., Zare, S., Saif, M., and Zarei, J. (2017). One-class classifiers for detecting faults in induction motors. In *Electrical and Computer Engineering (CCECE), 2017 IEEE 30th Canadian Conference on*, pages 1–5. IEEE.

[69] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[70] Samarasinghe, S. (2016). *Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition*. CRC Press.

[71] Santos, P., Villa, L. F., Reñones, A., Bustillo, A., and Maudes, J. (2015). An svm-based solution for fault detection in wind turbines. *Sensors*, 15(3):5627–5648.

[72] Schoen, R. R., Habetler, T. G., Kamran, F., and Bartfield, R. (1995). Motor bearing damage detection using stator current monitoring. *IEEE transactions on industry applications*, 31(6):1274–1279.

[73] Skurichina, M. and Duin, R. P. (2002). Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5(2):121–135.

[74] Smart, E., Brown, D., and Axel-Berg, L. (2013). Comparing one and two class classification methods for multiple fault detection on an induction motor. In *Industrial Electronics and Applications (ISIEA), 2013 IEEE Symposium on*, pages 132–137. IEEE.

[75] Stack, J. R., Habetler, T. G., and Harley, R. G. (2004). Bearing fault detection via autoregressive stator current modeling. *IEEE Transactions on Industry Applications*, 40(3):740–747.

[76] Stamboliska, Z., Rusiński, E., and Moczko, P. (2015). Condition monitoring techniques for low-speed machines. In *Proactive Condition Monitoring of Low-Speed Machines*, pages 69–119. Springer.

[77] Tax, D. (2006). Data description toolbox.

[78] Tax, D. M. and Duin, R. P. (1998). Outlier detection using classifier instability. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 593–601. Springer.

[79] Tax, D. M. and Duin, R. P. (2000). Data description in subspaces. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 2, pages 672–675. IEEE.

[80] Tax, D. M. and Duin, R. P. (2005). Characterizing one-class datasets. In *Proceedings of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa*, pages 21–26.

[81] Thomson, W., Leonard, R., Milne, A., and Penman, J. (1984). Failure identification of offshore induction motor systems using on-condition monitoring. *Reliability engineering*, 9(1):49–64.

[82] Thomson, W. T. and Fenger, M. (2001). Current signature analysis to detect induction motor faults. *IEEE Industry Applications Magazine*, 7(4):26–34.

[83] Vachtsevanos, G. J., Lewis, F., Hess, A., and Wu, B. (2006). *Intelligent fault diagnosis and prognosis for engineering systems*. Wiley Online Library.

[84] Wang, Y., Liu, M., and Bao, Z. (2016). Deep learning neural network for power system fault diagnosis.

[85] Waqar, T. and Demetgul, M. (2016). Thermal analysis mlp neural network based fault diagnosis on worm gears. *Measurement*, 86:56–66.

[86] Yadav, S. K. and Kalra, P. K. (2010). Condition monitoring of internal combustion engine using emd and hmm. In *Intelligent autonomous systems*, pages 167–185. Springer.

[87] Zgarni, S. E. and Braham, A. (2016). Intelligent induction motor diagnosis system: A new challenge. In *Advanced Technologies for Signal and Image Processing (ATSIP), 2016 2nd International Conference on*, pages 408–413. IEEE.

VITA AUCTORIS

NAME:	Shokoofeh Zare
PLACE OF BIRTH:	Shiraz, Fars, Iran.
EDUCATION:	Bachelor of Science in Computers and Electrical Engineering, Azad university, Shiraz, Fars, Iran, 2011.
	Master of Applied Science in Computers and Electrical Engineering, University of Windsor, Windsor, Ontario, Canada, 2018.