People’s Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of KASDI MERBAH OUARGLA

Faculty of New Technologies of Information and Communication

Department of Electronic and Telecommunications

ACADEMIC MASTER’S DISSERTATION

field: Science and Technology

Sector: Electronics

Speciality: Electronics of Embedded Systems

By: Kahel Elhadj chikh

Bouzidi Abdelkahar

Theme

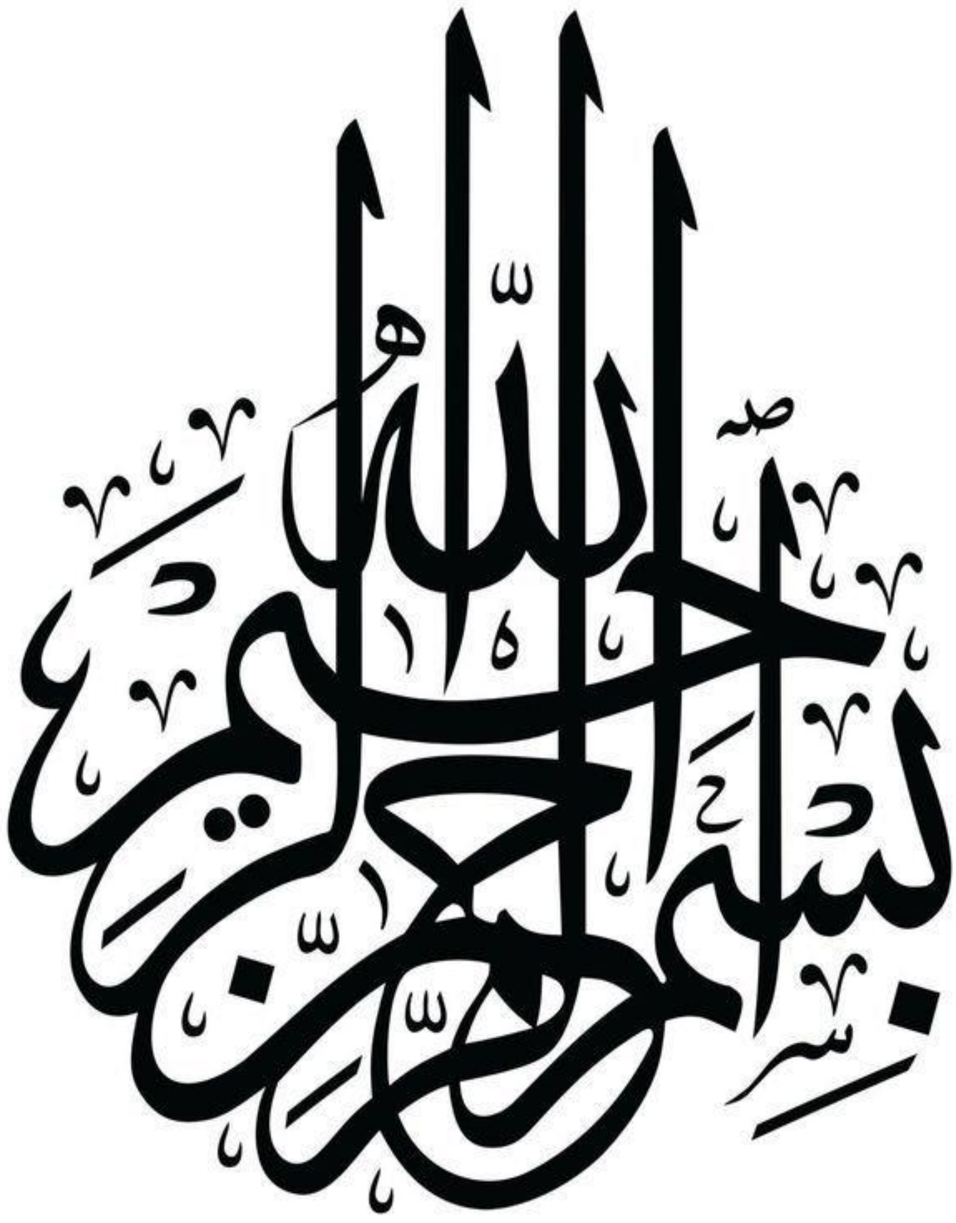
Deep Transfer Learning-Based Broken Rotor Fault Diagnosis for Induction Motors

Publically presented in the 02nd June 2025

Before jury:

Mr. Abderazzek Benchabane	MCA	Ouargla University	President
Mrs. Samira Souri	MAA	Ouargla University	Supervisor
Mr. Farid Kadri	MAA	Ouargla University	Examiner

ACADIMEC YEAR: 2024/2025


بسم الله الرحمن الرحيم

Dedication

First of all, I dedicate this work to my dear parents, my father **Abderrazzak** and my mother **Sabah,** who were everything to me. Their unwavering love and support have been a constant source of motivation throughout my academic journey. Their patience and encouragement have been helpful in my success, and I am forever grateful for their sincere and deep love. I would also like to express my appreciation to my brothers: **mohamad**, **Abdefattah** and **Taqi Eddin**, my sister **Rayhana**, and all my family for their unwavering support. Finally, I would like to thank all my colleagues for their valuable contributions and support.

Bouzidi Abdelkahar

Dedication

I dedicate this work to the memory of my beloved mother, **Malika**, whose love, strength, and unwavering support continue to guide me even though she is no longer with us. She was everything to me, and her memory has been a constant source of motivation throughout my academic journey. Her patience, encouragement, and deep love remain in my heart forever.

To my dear father, **Ismail**, thank you for your enduring support and wisdom. I am deeply grateful to my brothers, **Mohamed** and **Aissa**, and my sisters for their constant encouragement and presence in my life.

I also extend my heartfelt appreciation to all my family members for their steadfast support, and to my colleagues for their valuable contributions and camaraderie throughout this journey.

Kahel Elhadj Chikh

Acknowledgments

First and foremost, we thank God Almighty for granting us the strength, health, and perseverance to complete this research.

We sincerely thank our supervisor, **Mrs.Samira Souri**, for her continuous support, guidance, and encouragement throughout this work. Her dedication played a vital role in its successful completion.

We also extend our heartfelt thanks to the honorable members of the discussion committee, **Mr. Benchabane Abderazzek** and **Mr. Farid Kadri**, for their valuable time, constructive feedback, and insightful evaluation.

Our appreciation also goes to all the respected professors at the University of Kasdi Merbah, who has supported and guided us throughout our academic journey.

May peace and blessings be upon our Prophet Muhammad, his family, and his companions.

Contents

Contents

List of Figures

List of Tables

Acronyms

ABB ASEA Brown Boveri.

AI Artificial intelligence.

ANN Artificial Neural Network.

BRB Broken Rotor Bar.

CNN Convolutional Neural Network.

DL Deep Learning.

DWT Discrete Wavelet Transform.

EPRI Electric Power Research Institute.

FC Fully-Connected.

FFT Fast Fourier Transform.

HT Hilbert Transform.

IEEE Institute of Electrical and Electronics Engineers.

IM Induction Motor.

KNN K-Nearest Neighbors.

ML Machine Learning.

MCSA Motor Current Signature Analysis.

ReLU Rectified Linear Unit.

ResNet Residual Network.

STFT Short-Time Fourier Transform.

SVM Support Vector Machine.

General Introduction

Induction motors (IM) are the prime mover for various industrial applications such as conveyors, winders, pumps, and wind tunnels to produce torque. Induction motors serve as prime movers in the majority of electric cars. Despite being a strong, durable, and potent machine, IM is prone to failure because of many stresses that are created during operation, including mechanical, thermal, and electrical stressors. Increased downtime, monetary losses, plant shutdown, and possibly fatal accidents would all arise from an IM failure. However, comprehensive and efficient condition-based monitoring can avoid such unforeseen circumstances. Therefore, induction motor failure detection is crucial for both industrial applications and IM-based electric vehicles. Mechanical and electrical flaws are the two main categories into which induction motor defects fall. Electrical defects include broken rotor bars, inter-turn shorts, and stator damage, while mechanical faults include eccentricity, imbalance, and bearing problems. Vibration and current monitoring are useful tools for diagnosing mechanical and electrical problems, respectively. The most common broken rotor bar fault (BRB) in IM is caused by fatigue parts, loose lamination, mechanical stress from bearing failure, thermal stress, and magnetic stress.

There are two types of fault detection methods for induction motors (IMs): model-based methods and signal-based data-driven methods. Signal-based techniques involve obtaining real-time signals from the system, such as vibration data or stator current, and extracting characteristics from their time-domain, frequency-domain, or time-frequency representations. Typical signal processing techniques for examining these signals include the Fast Fourier Transform (FFT) and Discrete Wavelet Transform (DWT), which make it possible to discover typical problems like broken rotor bars. Artificial intelligence methods, such as machine learning (ML), deep learning (DL), and artificial neural networks (ANN), have become more popular in defect diagnosis due to increases in processing capacity because they provide greater automation and accuracy.[1]

Automated classification systems leveraging advanced techniques such as Convolutional Neural Networks (CNN), Support Vector Machines (SVM), and K-Nearest Neighbors (KNN) have gained increasing attention in the field of fault diagnosis for induction motors. Among these, dilated CNN architectures have emerged as promising tools for analyzing motor conditions, with potential applications in detecting faults like broken rotor bars and bearing defects. These developments highlight the growing role of deep learning and signal processing in enabling more effective and data-driven maintenance strategies.

We have chosen to focus our study on three main chapters:

Chapter I:

Provides an overview of industrial maintenance strategies for induction motors. It introduces the fundamental components of an induction motor and explains its operating principle. The discussion then focuses on common electrical and mechanical faults, their frequency in industrial applications. The text outlines key criteria for designing effective diagnostic systems and reviews various fault diagnosis methods, ranging from model-based techniques to model-free approaches such as signal processing and artificial intelligence.

Chapter II:

Explore AI fundamentals, focusing on Machine Learning (ML) and Deep Learning (DL). ML enables autonomous learning from data, with various algorithms and applications discussed. The focus then shifts to DL, particularly ANNs and CNNs for image and signal processing. We analyze their structures before detailing ResNet, specifically ResNet50, which solves vanishing/exploding gradient issues via residual connections for deeper, more accurate models.

Chapter III:

Present the experimental results of the proposed methodology, covering performance evaluation, work environment, parameter settings, and comparison with state-of-the-art methods. The dataset and workflow are introduced along with evaluation metrics, followed by hardware and software configurations. Key parameters are detailed for reproducibility, and the results are analyzed using accuracy and loss curves, confusion matrices, and classification reports. The chapter concludes with a comparative assessment and summary of key findings.

Chapter 1

Overview of Induction Motor Systems and Diagnostic Approaches

1.1 Introduction

This chapter provides an overview of the industrial maintenance of induction motors, highlighting the main types of maintenance: preventive, corrective, and predictive. It outlines the motor's key components (the stator, rotor, and mechanical parts) and explains its operating principle. Common electrical and mechanical faults, such as stator faults, broken rotor bars, bearing issues, and eccentricity, are discussed along with their frequency in industrial settings. The chapter also presents criteria for designing diagnostic systems and reviews fault diagnosis methods, including model-based and model-free approaches like signal processing and artificial intelligence.

1.2 Industrial Maintenance of Induction Motors

Any activity intended to keep an asset in a specific state or operating circumstances to guarantee dependable performance and fulfill a necessary function is referred to as maintenance. These activities include management, administrative, and technical duties. Maintenance tasks include troubleshooting, lubrication, inspections, repairs, and enhancements to retain an equipment's operating capability and guarantee continuous, high-quality production. Efficient upkeep guarantees that these procedures are performed to minimize total expenses. (see Figure 1.1) displays the types of industrial maintenance diagrams [2].

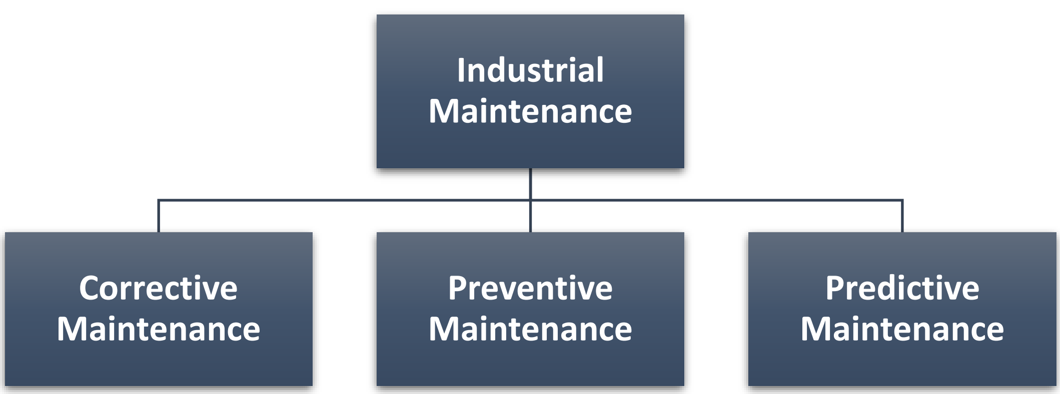


Figure1.1: Types of Industrial Maintenance

1.2.1 Types of Maintenance

1.2.1.1 Preventive Maintenance

There are numerous definitions for preventive maintenance. When taken literally, the term describes a maintenance program aimed at avoiding or doing away with corrective and breakdown maintenance. A comprehensive preventive maintenance program will include routine inspections of critical plant machinery, equipment, and systems to spot potential problems and quickly create repair plans that will halt any decline in operational condition.

Most plants just need to be lubricated occasionally as a preventative measure. To fully support all plant production or manufacturing systems, a thorough preventive maintenance program will include predictive maintenance, time-driven maintenance chores, and corrective maintenance. [3]

1.2.1.2 Corrective Maintenance

The goal of corrective maintenance is to maintain all essential plant machinery and systems operating at peak efficiency through regular, planned tasks. The efficacy of maintenance is assessed based on the life-cycle costs of critical plant machinery, equipment, and systems rather than how quickly a malfunctioning unit can be fixed. The fundamental tenet of corrective maintenance is to fully and appropriately address any new problems when they arise.

Before reusing the machine or system, each repair is meticulously planned, executed by knowledgeable craftspeople, and examined. Mechanical and electrical problems are not the only problems. On the contrary, any deviation from optimal operating conditions—that is, productivity, output, and product quality fixed.[3]

1.2.1.3 Predictive Maintenance

Predictive maintenance considers the status of the equipment when planning maintenance, starting as soon as the equipment deviates from typical operating circumstances and before a breakdown happens. This allows the equipment to be maintained as needed, preventing the needless replacement of parts, hence increasing component life and reducing machine downtime. The massive volume of data collected by sensors enables predictive maintenance, which can be viewed as an expansion and advancement of Condition Monitoring. Due to its advantages, the industry is paying attention to predictive maintenance, which is the most effective kind of maintenance. However, without the right hardware and analysis skills, predictive maintenance could be costly and challenging to build if previous data is lacking.[4]

1.3 Constitution of the Induction Motor

The asynchronous computing device has two windings. The first principal winding is placed at the stator. It is powered by a constant-frequency electrical network. The second, so-called secondary winding, is positioned at the rotor. It is closed on itself or on resistances (see Figure 1.2)

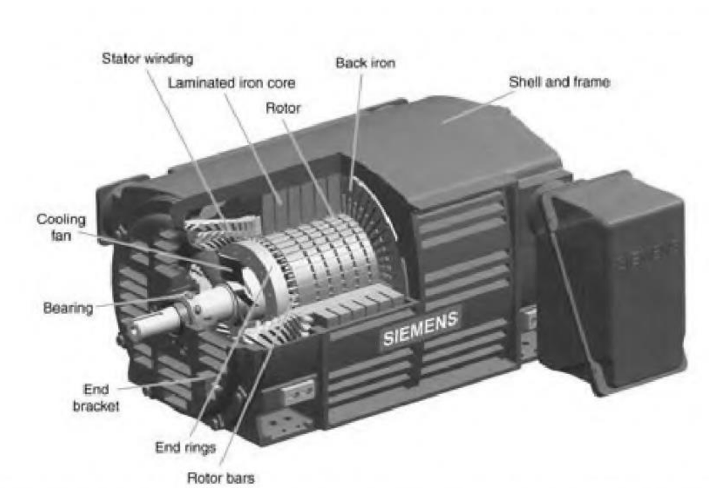


Figure 1.2: All of the parts of a three-phase induction motor are visible Siemens

1.3.1 Stator

An asynchronous machine's stator, also called the inductor, is powered by a three-phase frequency voltage system and contains three phases, each of which is composed of coils. The inductor features a ferromagnetic cylinder with notches to make room for the windings. The ferromagnetic cylinder is constructed from a stack of laminated, notched sheet metal plates to lessen the impact of eddy currents. Slot conductors and coil heads are characteristics of the stator winding. The magnetic field in the air gap is produced by notch conductors, and coil heads cut off currents in the coil to guarantee a closed circuit of the windings and, as a result, the flow of current from one notch conductor to another (see Figure 1.3). [5]

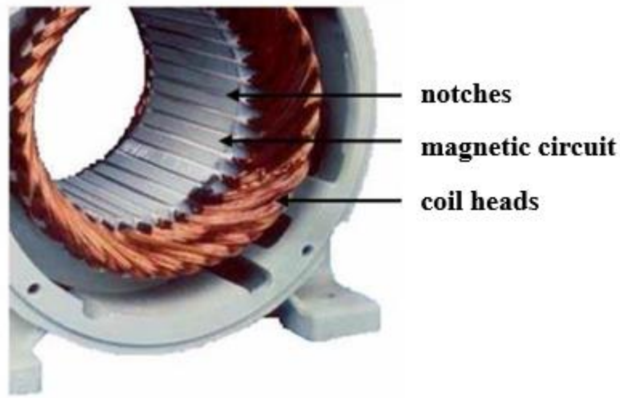


Figure1.3: An asynchronous machine's stator

1.3.2 Rotor

The rotor of the most popular induction motors is a squirrel cage. Heavy-gauge copper or aluminum bars that are short-circuited by end caps are used in its construction. It is shaped like a squirrel cage or a hamster wheel. The conductor bars are placed inside a steel lamination stack. By reducing air reluctance, the steel core aids in concentrating the flux lines between the rotor conductors. The bars minor skew lessens the humming noise. The shaft is fixedly attached to the rotor (see Figure 1.4).[6]

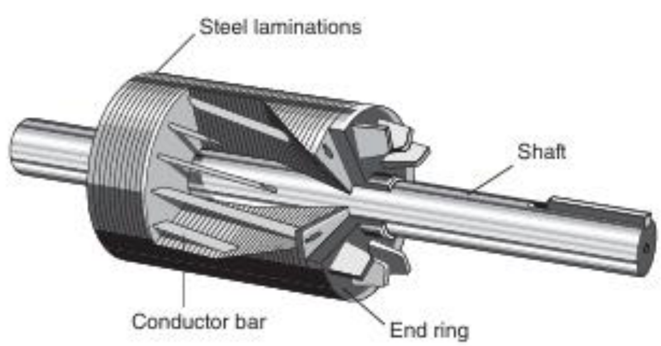


Figure1.4: AC induction motor's squirrel cage rotor

1.3.3 Mechanical components

The body serves as support. It offers defense against the outside world and functions as an envelope. One component of a gearbox is the shaft. It consists of a shaft extension with a coupling half attached to it and a central component supporting the rotor body. Steel that has been cast or forged is typically used to make it. Bending forces (centrifugal force acting on it, radial magnetic pull, etc.), torsional forces (electromagnetic torque transmitted in steady and transient modes), and radial and tangential forces resulting from centrifugal forces all affect its dimensions. At least one bearing supports it. These bearings support the rotor, which also guarantees free rotation. The second bearing is free to allow for the shaft's thermal expansion.

Shaft currents caused by magnetic circuit reluctance asymmetries are eliminated when one of the bearings is electrically insulated. Typically, bearings are utilized for machines with little to medium power.[7]

1.4 Operating principle of the asynchronous machine

Currents flow through the rotor's conductors as a result of the stator magnetic field. This current circulates the end rings and the rotor bars. Each of the rotor conductor bars consequently experiences the creation of a magnetic field (see Figure 1.5) by the stator's spinning field, and the cage turns into an electromagnet with alternating N and S Poles. As the stator's magnetic field increases, the rotor's magnetic field attempts to keep up. The torque produced by their interaction powers the motor. The rotor bars generate voltages due to the stator's revolving magnetic field.

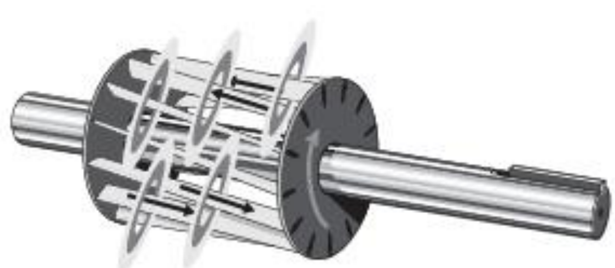


Figure 1.5: Magnetic fields produced by induced rotor currents

According to Faraday's induction law, the induced voltage is directly proportional to the rate of change of the flux lines that cut the rotor bars over time. If both spin at the same speed, there won't be any relative motion between the rotor and the stator field. Consequently, no induced voltage (or torque) would be generated, and the motor would not operate. Consequently, the stator's speed is higher than the rotor's. The speed difference between the stator and rotor is known as slip. The synchronous speed, or stator magnetic field speed (ns), is determined by Equation (1.1):

$$n_s = \frac{f}{p}\ tr/s \quad \text{or } n_s = \frac{60f}{p}\ \text{tr/min} \tag{1.1}$$

The synchronous speed and the rotor's actual speed (n) will be marginally different. For induction motors, the speed is indicated on the name plate. To express slip, use the following formula (1.2):

$$s = \frac{n_s - n}{n_s} \tag{1.2}$$

Upon starting the motor, the rotor stays still while the stator windings create a simultaneously rapid magnetic field. Therefore, the initial slip is s = 1.0 (100%) after all. The motor consumes a significant amount of inrush current during startup. This current can be up to six times the full-load current shown on the nameplate and is also referred to as the locked-rotor current.

The motor current drops as the motor steadily increases speed. The torque that the load requires and the torque that the motor produces are identical when the rotor speed reaches that point. At this stage, as long as the burden stays the same, the motor speed stays constant.

The motor speed must be lower than the synchronous speed to deliver usable torque and adapt to loading. A three-phase, 60 Hz motor's slip typically ranges from 0.02 to 0.03 (2 to 3%). Since they don't run at synchronous speed, induction motors are also known as asynchronous motors.[8]

1.5 Different types of faults

Sensor faults: Sensors measure system states directly or generate state estimates for the control law; therefore, the presence of sensor faults can degrade state estimates, resulting in inaccurate diagnosis. Sensors represent the output interface of a system and transmit information about its behavior; therefore, sensor faults can degrade process performance and compromise data integrity, which in turn distorts decision-making.

Actuator faults: Many systems do not allow the immediate application of the controller's command signals to the system. Actuators are required to convert them into suitable actuation signals. However, a complete loss of control may result from the emergence of abnormalities in this component.

Process faults: Process failures typically arise when specific system conditions change, making the system's nominal dynamic equation incorrect.[9]

The types of flaws that were shown are summarized in the figure below.

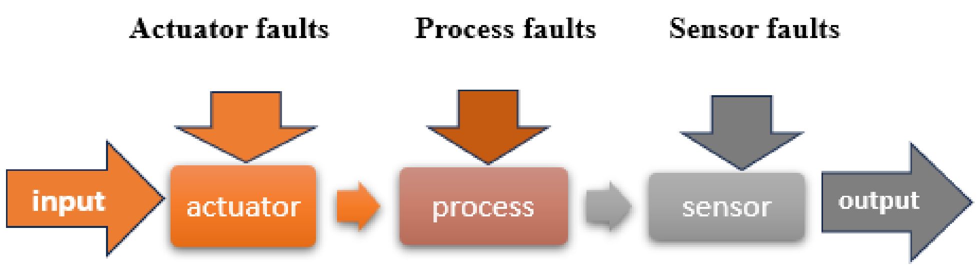


Figure 1.6: Different kinds of physical system flaws

1.6 Common Faults in Induction Motors

The two most prevalent types of induction motor defects are mechanical and electrical. For simplicity's sake, this study classifies motor failures based on the component that causes the defect, even though there are other fault classifications. The rotor, rolling bearings, and stator are the three primary sections of the rotatory machine that have the most frequent failures.[10]

1.6.1 Statistical overview of common faults in induction motors

Statistical analyses of the prevalence of various machine defects have been conducted. The Electric Power Research Institute (EPRI), ASEA Brown Boveri (ABB), and the Institute of Electrical and Electronics Engineers (IEEE) have based their conclusions on these findings. Table 1 provides an overview of the study's results.[11]

Table 1.1: Potential induction motor failure

Studied by	bearing fault	stator fault	rotor fault	Others
IEEE	41%	28%	9%	22%
EPRI	42%	36%	8%	14%
ABB	51%	16%	5%	10%

1.6.2 Electronics faults

1.6.2.1 Stator faults

It is generally accepted that the bulk of stator faults are caused by damage to the insulation of winding turns. The term "stator turn defect" refers to this type of flaw. When a stator turn fault occurs in a symmetrical induction motor, an excessive amount of current flows through the turns, producing excessive heat in the shorted turns.[12]

Until a catastrophic failure, such as a phase failure, phase-to-phase failure, phase-to-ground failure, or phase-to-phase-to-ground failure, occurs, the additional local heat between two or more shorted turns would further deteriorate the insulation of the adjacent turn. Nonetheless, the stator problems fall into one of two general categories:

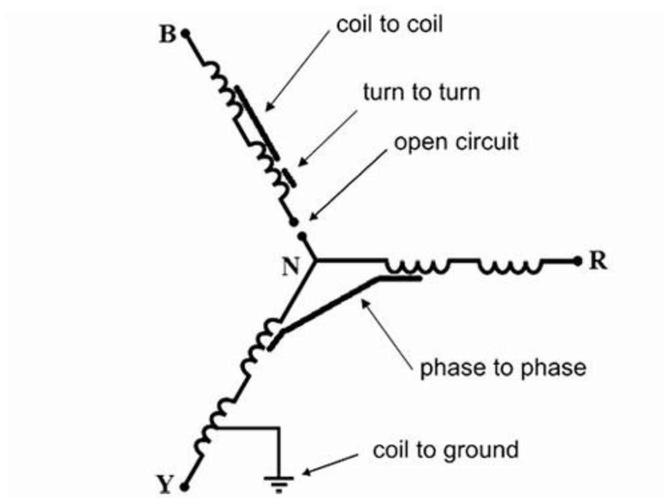


Figure 1.7: A stator with a star connection displaying several stator winding problems

• Defects in the frame (vibration, circulation currents, coolant loss, earth faults, etc.) and laminations (core hot spot, core slackening, etc.).

• Defects with stator windings: The end winding or slot portion of stator windings is the most frequently found defect.

Failures of the end winding portion include local insulation damage, insulation fretting, insulation contamination by moisture, oil, or dirt, connection damage, turn-to-turn faults (whereas slot portion problems include insulation fretting), and conductor displacement. The bulk of these flaws are caused by several stresses applied to the stator and can be categorized as mechanical, electrical, thermal, and environmental. One defect most prevalent in induction motors is inter-turn short circuits in the stator windings. Short circuits usually happen in stator windings between one phase's turns, two phases' turns, or all phases' turns. Additionally, short circuits between the stator core and winding conductors can also happen.[13]

Figure 1.8: Examples of damage brought on by short circuits in IM stators

1.6.2.2 Rotor faults

Broken bar faults

Bar breakage usually happens in highly pressured conditions and in applications that specifically require high-power motors. Under such conditions, oscillatory loads, improper motor installation, mechanical stresses, and hysteresis stresses can all weaken the joints connecting the bars and end rings. Therefore, it is possible to unhook weakly linked bars from end-rings on one or both motor ends. Occasionally, the end-ring may also crack or break, however, this can happen midway between two bars rather than at the bar/end-ring junction. This type is usually caused by a casting process that was done incorrectly during the motor's manufacture. Another important aspect that could increase the risk of fracture is air bubbles within the materials or joints. As the name suggests, this type of issue only affects motors that have a squirrel-cage rotor (see Figure 1.9).[14]

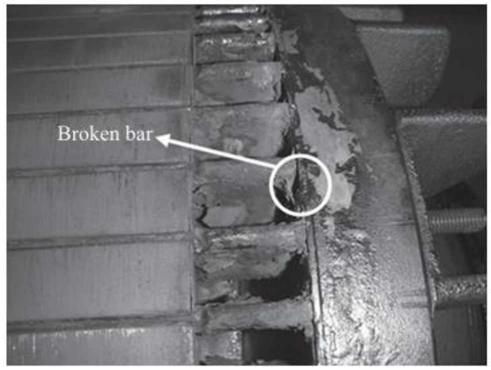


Figure 1.9: A damaged bar in the squirrel cage of an induction motor

Breakage of a short-circuit ring portion faults

One fault that happens just as frequently as bar breakage is ring section breakage. The short-circuit rings conduct higher currents than the rotor bars, which makes these failures even more likely to be caused by bubbles in the casting or by differing expansions between the bars and the rings. Due to this phenomenon, ring failure may happen from inadequate ring sizing, deteriorating operating conditions, torque overload, and consequently, current overload.[15]

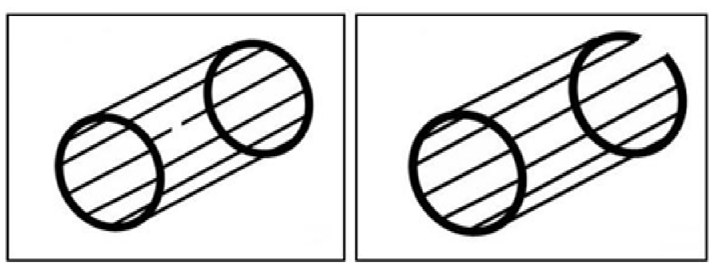


Figure 1.10: Broken bar and ring with a short circuit

Similar to the effect of bar breakage, the amplitude modulation effect on stator currents is produced when a ring section breaks because it throws off the balance of current distribution in the rotor bars.

1.6.3 Mechanical Faults

1.6.3.1 Bearing Faults

Bearing defect is the phrase used to describe any physical damage to the balls' surface, inner race, or outer race. To sustain the revolving shaft of an induction motor, two sets of bearings are positioned at either end of the rotor. By reducing friction, they helped the rotor revolve freely while holding it in place. A collection of rolling components known as balls sits between the inner and outer rings, known as races, that make up each bearing.

To reduce friction, a motor's inner race is typically coupled to the shaft, and the load is transferred through the revolving balls. Bearings are the weakest part of an induction motor when it comes to failure.

The following are some reasons and consequences of bearing failure:

- Excessive loads, tight fits, and excessive temperature rise.
- Fatigue failure.
- Corrosion (Bearing corrosion occurs if lubricants degrade or if the bearings are handled negligently during installation)
- Contamination (Most frequently found in industrial settings, dirt and other foreign particles contaminate lubricants. Contamination results in wear and high vibration)
- Lubricant failure.
- Misalignment of bearings. [16]

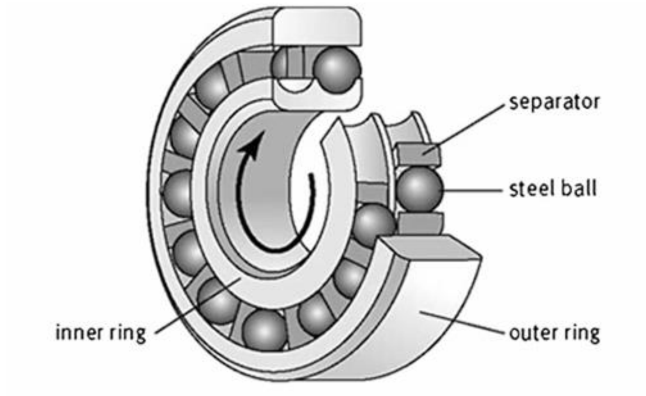


Figure 1.11: ball bearing (dissected)

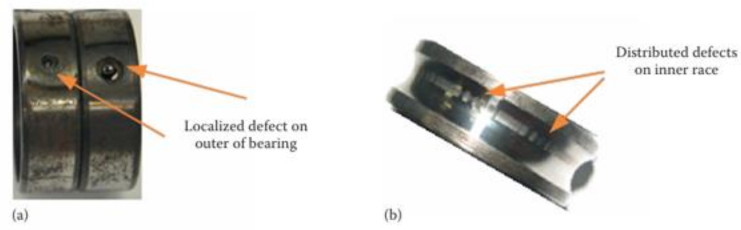


Figure 1.12: A bearing example. (a) Local defects. (a) Defects that are distributed

1.6.3.2 Eccentricity Faults

A situation called air-gap eccentricity happens when the rotor and stator in the air-gap are not at the same distance. When the air-gap is eccentric, different inductances lead to an imbalanced magnetic flux inside the gap, which produces fault harmonics in the line current that are visible in the spectrum. The current spectrum can reveal imbalances in the current flow caused by fluctuating magnetic flux within the air-gap if the distance between the stator bore and rotor is not constant throughout the machine. Misalignment, rotor unbalance, loose or missing bolts, or improper mounting can all contribute to air-gap eccentricity.[17][18]

A machine may experience one of three forms of eccentricity (see Figure 1.13):

1. **Static eccentricity:** the rotor shaft's center of rotation differs from the machine's geometric center.
2. **Dynamic eccentricity**: the rotor shaft's center of rotation revolves around the machine's geometric center.

3. **mixed eccentricity:** the total of the two before it. When currents at specific frequencies produce harmonics in their spectra, eccentricity is present.

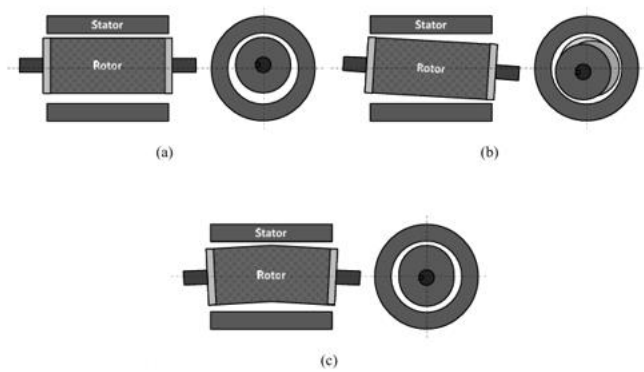


Figure 1.13: Types of eccentricity. Faults include static (a), dynamic (b), and mixed (c)

1.8 Criteria for developing a diagnostic system

We must first establish the evaluation criteria for a diagnostic system to guarantee that the system we create operates as efficiently as feasible. In general, we can classify a detection system's different performance requirements as follows:

- detectability.
- Isolability.
- Sensitivity.
- robustness.

Detectability: is the diagnostic system's capacity to identify the existence of a process flaw.

Isolability: is the diagnostic system's capacity to identify the root cause of the issue directly. The structure of the residues and the actual detection process are connected to the isolability property.

Sensitivity: describes the system's capacity to identify problems with a specific amplitude.

Robustness: describes how the diagnostic system is not sensitive to the uncertainties of the model system or to deterministic or random disruptions.[19]

1.9 Fault diagnosis methods for induction motors

1.9.1 Model-based methods

These diagnostic techniques are predicated on an understanding of the system's physical model that has to be observed. They are regarded as substitutes for physical redundancy or hardware. The idea behind this kind of diagnosis is to determine if an anomaly is present or not by comparing the system's actual behavior to the behavior that its model predicted, and if a failure is present, to identify it. We mention two methods that can be applied to parametric model-based fault diagnosis.[20]

1.9.1.1 Parametric estimation approach

This method is predicated on the idea that the system's physical characteristics, including mass, friction, resistance, viscosity, inductance, capacitance, etc., represent the defects. The detection method's core idea is to make repeated online estimates of the real process's parameters using well-known parameter estimation techniques, then compare the findings with the parameters that the reference model first obtained under the fault-free situation. Any significant difference is flagged as a mistake.[21]

1.9.1.2 Observer-based approach

The observer-based approach is suitable for identifying and isolating additive problems, particularly when the faults are linked to modifications in actuators, sensors, or unquantifiable state variables. To give the states in the state-space equations a physical interpretation, a thorough mathematical model of the plant is needed, ideally one that is developed from first principles. A Kalman filter or Luenberger observer is used to rebuild the unmeasured states from the measurable input and output variables.[22]

1.9.1.3 Parity Space Approach

The foundation of parity equation techniques is the confirmation of a static or dynamic relationship between measurements. The space of potential residuals can be thought of as the parity space. The orthogonal of the observability matrix, which removes the state's impact on the residual, is a common definition. The most common method for perturbation decoupling is projection, which is equivalent to a direct decoupling technique (after projection, only a portion of the original system is used; the other portion is discarded).[23]

1.9.2 Model-free methods

These techniques focus on identifying fault signatures rather than a high-fidelity model of the system. There are two types of model-free methods: the first is based on processing signals from sensors of electrical or mechanical qualities affected by faults. Fault signatures, which can be obtained by modeling or experimental measurement, are often recorded in a database. The latter employs artificial intelligence-based expert systems.

1.9.2.1 Signal Processing

One benefit of techniques based on acquisition signal analysis is their independence from internal system oscillations. This indicates that there are no modeling errors affecting the information in the signals. The spectrum content of different signals from electrical machines, such as currents, powers, torque, speed, flux, vibrations, etc., has long been analyzed using a variety of signal processing techniques. Fast Fourier Transform (FFT), Discrete Wavelet Transform (DWT), and Hilbert Transform (HT) are the most widely used techniques for evaluating diagnostic data.[11][24]

These diagnostic techniques look at the following flaws:

- Broken bars or portions of rotor rings.
- Short-circuits between turns on the stator.
- Static and dynamic eccentricity.

1.9.2.2 Artificial Intelligence

Artificial intelligence encompasses many diagnostic-related techniques, including neural networks, fuzzy logic, expert systems, genetic algorithms, particle swarm optimization, and pattern recognition, which can be applied singly or in combination to increase diagnostic effectiveness.[24]

1.10 Fault detection methods

Various monitoring approaches for induction motors have been developed by different researchers using different machine parameters to detect the aforementioned defects. machine specifications. They fit the following description:

- Motor Current Signature Analysis (MCSA)
- Flux analysis
- Vibration and acoustic signal analysis
- Temperature analysis
- Electromagnetic torque analysis
- Instantaneous power analysis
- Partial discharge analysis

1.11 Conclusion

In this chapter, we presented the foundational concepts related to the industrial maintenance of induction motors. We discussed the main types of maintenance and examined the internal structure of induction motors, including the stator, rotor, and mechanical components. The operating principle of asynchronous motors was outlined, followed by an overview of common electrical and mechanical faults. We also explored the criteria for developing diagnostic systems and reviewed both model-based and model-free fault diagnosis methods, setting the stage for advanced fault detection approaches in the following chapters.

Chapter 2

Work Background

2.1. Introduction

This chapter introduces the foundational concepts of Artificial Intelligence (AI), focusing on its key branches: Machine Learning (ML) and Deep Learning. We begin with an overview of AI and explain how ML enables systems to learn from data without explicit programming, covering core ML algorithms and their real-world uses. We then explore Deep Learning, highlighting neural networks like Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs), especially in image and signal processing. Finally, we examine ResNet, a deep CNN architecture, focusing on ResNet-50 and its solutions to challenges like vanishing gradients in deep network training.

2.2 Artificial intelligence

Artificial intelligence is defined as the fusion of computer science and psychological intelligence. In simple terms, it is the computational part of the ability to achieve a goal. Intelligence is the ability to think, imagine, remember, and understand things, recognize patterns, make decisions, adapt to changes, and learn from experience. Artificial intelligence strives to give computers human behavior.[25]

2.3 Machine learning

Machine learning is a subfield of computer science that focuses on creating algorithms that depend on a set of instances of a phenomenon to be effective. These examples may be produced by an algorithm, handmade by humans, or derived from nature. Another definition of machine learning is the process of solving a real-world issue by compiling a dataset and using that dataset to create a statistical model algorithmically. It is assumed that the practical problem will be resolved in some way using that statistical model. I use the phrases "learning" and "machine learning" interchangeably to save keystrokes.[26]

2.3.1 Types of machine learning algorithms

Machine learning algorithms, such as classification, regression, machine translation, and anomaly detection, can be used to solve problems that are difficult for traditional programming to handle.

Machine learning algorithms are often divided into three categories: supervised learning, unsupervised learning, and Reinforcement learning, depending on the kind of data being used (see Figure 2.1)

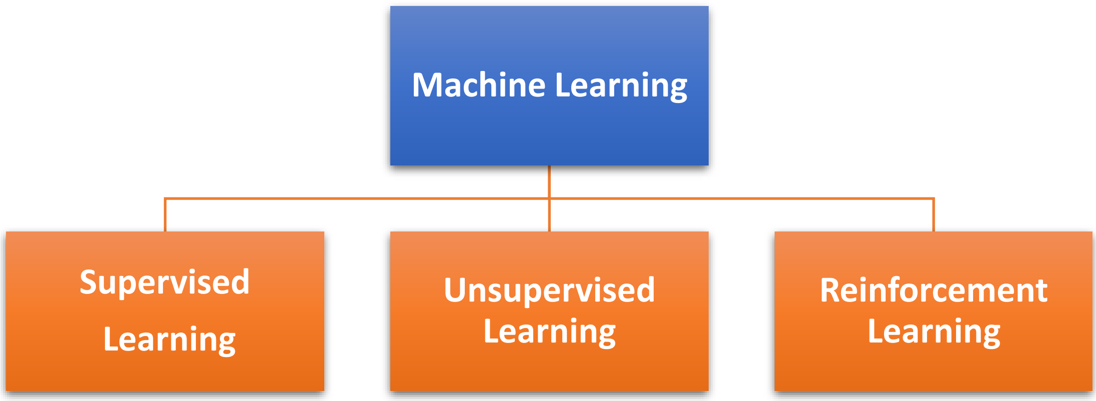


Figure 2.1: Types of Machine Learning algorithms

2.3.1.1 Supervised learning

Supervised learning uses input-output pairings to train a function that transforms an input into an output. Labeled training data, which is a set of training samples, is used to infer the function. A supervised learning algorithm analyzes the training data and produces an inferred function to map new samples.[26]

2.3.1.2 Unsupervised learning

Machine learning that searches a dataset for previously unidentified patterns without labels and with little to no human oversight is known as unsupervised learning. Unsupervised learning, also known as self-organization, can be used to model probability densities across inputs, unlike supervised learning, which usually employs data that has been human-labeled.[26]

2.3.1.3 Reinforcement learning

Reinforcement learning aims to maximize the concept of cumulative reward by determining how software agents should act in a certain environment. Reinforcement learning differs from supervised learning in that it does not require the explicit correction of poor behaviors or the presentation of labeled input/output pairings. Instead, the focus is on finding a balance between using what is already known and exploring new areas.[26]

2.4 Deep Learning

Deep learning is a subfield of machine learning (see in Figure 2.2) that uses numerous hidden layers to mine data for information and patterns. Such consecutive layers of increasingly relevant representations are used to produce learning in deep learning, which focuses on hierarchical representation learning. As an example, in image processing, lower layers might detect borders, whereas higher layers might detect human-relevant concepts like faces, letters, or numbers.[4][26]

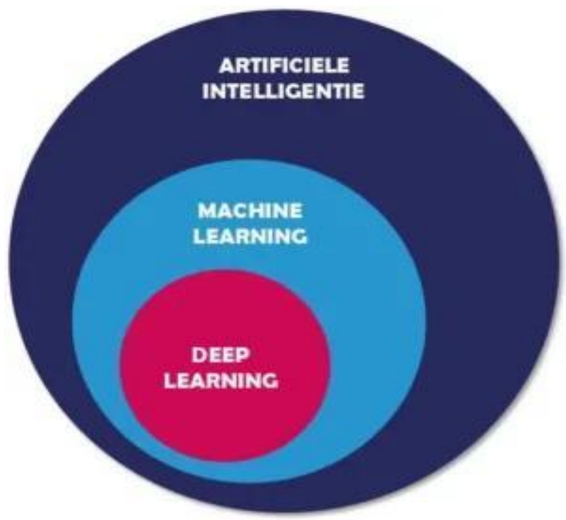


Figure 2.2: Relationship among the three ideas: DL, ML, and AI

2.5 Artificial Neural Network (ANN)

An artificial neural network (ANN) is a type of computer network that is modeled after biological neural networks, which are the intricate networks of neurons seen in human brains. Since the nodes produced by the ANN are purportedly designed to function similarly to real neurons, they are considered artificial neurons. The network of nodes, or artificial neurons, that comprise the artificial neural network is seen in (see Figure 2.3.)

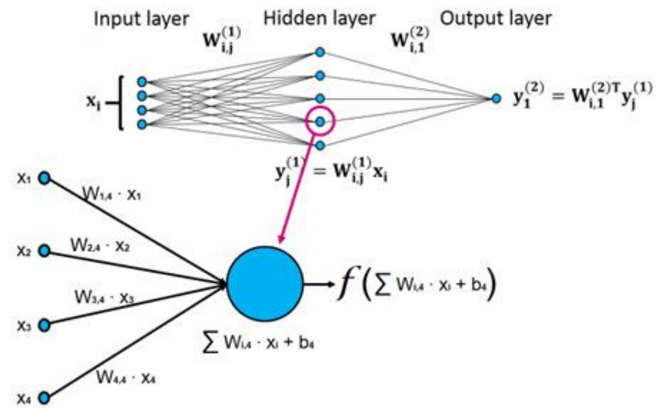


Figure 2.3: Structure of an Artificial Neural Network (ANN)

An artificial neural network's primary structural element may be the number of layers and the number of neurons or nodes in each layer. At first, the bias and weights (which show the connections) are insufficient to make the judgment (classification, etc.). It resembles the brain of a newborn with no past knowledge. Experiences teach a newborn how to make wise decisions (classifier). Labeled experiences and data aid in fine-tuning the neural weights and biases in the brain's neural network. The same procedure is followed by the artificial neural network. To produce a good classifier, the weights are adjusted each iteration. Algorithms are used to carry out these tasks because manually adjusting and determining the appropriate weights for thousands of neurons takes a lot of time. Learning or training is the term used to describe that weight-tuning process. This is identical to what people do every day.

A conventional neural network consists of many artificial neurons or units. The input, hidden, and output layers are the three layers of a neural network (Figure 2.4). Because neural networks are interconnected, every neuron in the hidden layer is completely coupled to every other neuron in both the input layer and the output layer that follows. To achieve the best outcomes, a neural network learns by iteratively modifying the weights and biases in each layer.[27]

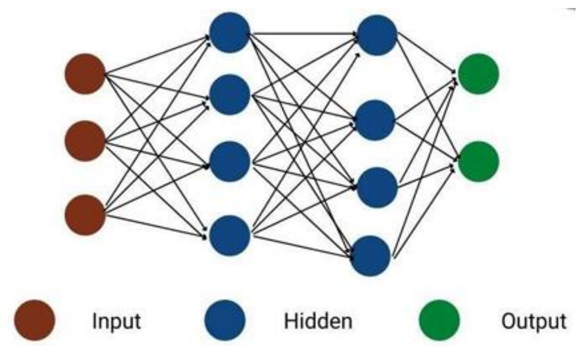


Figure 2.4: Basic Structure of an Artificial Neural Network

2.6 Convolutional Neural Network CNN

A CNN uses convolutions to try to learn higher-order features in the data. They routinely win picture categorization contests and are excellent at item recognition using pictures. Faces, people, street signs, platypuses, and a variety of other visual data elements can all be recognized by them. CNNs can also be used to analyze words as separate textual units, however, they overlap with text analysis by optical character recognition. They are adept at sound analysis as well. One of the primary reasons why deep learning is so powerful is that CNNs are so good at picture recognition. CNNs ability to construct location and (relatively) rotation-invariant features from raw visual input is seen in Figure 2.5.

Figure 2.5: CNNs and computer vision.

CNNs are generating significant advancements in machine vision, which has clear uses in robotics, drones, self-driving automobiles, and visually impaired treatment.[28]

2.6.1 Building blocks of CNN architecture

2.6.1.1 The Convolution Layer

This layer is the first one an image goes through following input, because it is where most of the calculation in any specific CNN takes place. A subset of the image is scanned by filters within a convolutional layer. Although the height and width of each filter are not very large, they all span the entire length of this layer. For example, consider the scenario where we are attempting to categorize an image as either a 1 or a 0, however, the image is a 1. Additionally, consider a picture with a black backdrop and a white pixel outline around the digit. An illustration of how this image might be described is shown in (see Figure 2.6).

The computer will recognize that the black pixels have a value of -1 and the white pixels have a value of 1. The convolutional layer takes this image as input and uses it to extract the distinctive elements of the image, which are typically the colors, shapes, and edges that make up a particular image. After determining the features of a training image, we apply what is referred to as filtering to the input image.

Figure 2.6: A sample picture of "1"

The method of filtering involves matching an input picture feature in this case, a 3 × 3 pixel square with a patch of that image, which is likewise a 3 x 3 pixel square. (see Figure 2.7) shows us what the filtering process looks like. Next, we increase the feature pixel number by the picture patch pixel number associated with it.[29]

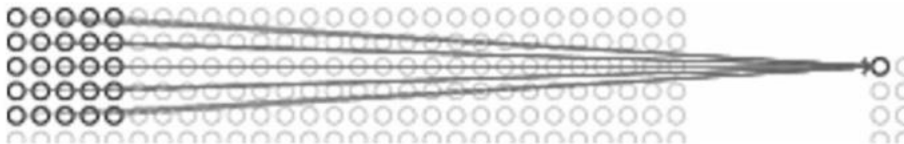


Figure 2.7: Example of a filter

2.6.1.2 Activation Function (ReLU)

The activation function is also known as a rectifier. We usually apply the subsequent function to this layer's inputs.

$$f(x) = \max(0, x)$$

x = input to a neuron.

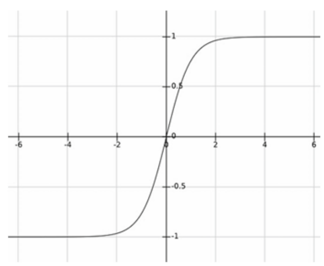


Figure 2.8: Graph of tanh-activation function

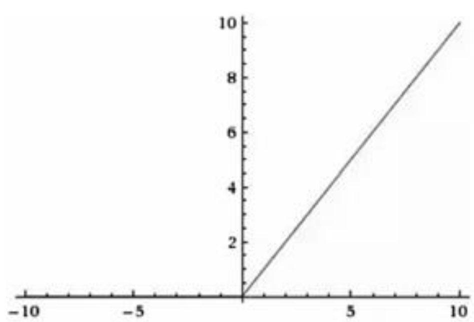


Figure 2.9: Rectified Linear Unit (ReLU) activation function

Any feature map values that would have been negative now might be imagined to be zero when applied to the feature map. This specifically helps draw the feature map closer to the image with which it is most commonly linked. This is done to every feature map to have a "stack" of pictures.[29]

2.6.1.3 Pooling layer

Pooling is a widely used technique that has been incorporated into numerous image processing techniques. Pooling layers combine nearby pixels in a particular image region into a single representative value, which reduces the size of the image. Set the representative value and specify how to choose the pooled pixels from the image in order to carry out the actions in the pooling layer. Typically, a square matrix is used to choose neighboring pixels, and the problem determines how many pixels are combined. Typically, the average or maximum value of the chosen pixels is used to define the representative value. Surprisingly, a pooling layer operates quite simply. This is a two-dimensional operation; thus, describing it verbally could make things more confusing. There are primarily two kinds of pooling:

Max pooling: The filter chooses the pixel with the highest value and adds it to the output array as it iterates over the input values.

Average pooling: The filter calculates the average value in the receptive field as it passes over the input and sends that value to the output array.[30]

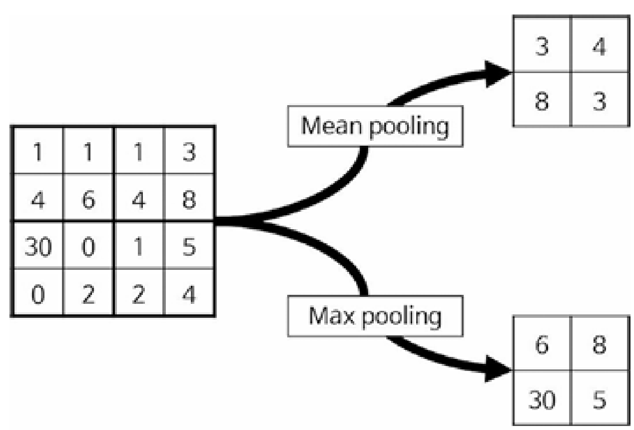


Figure 2.10: The resulting instances of pooling using two distinct approaches

2.6.1.4 Fully Connected (FC) Layer

Each activation map in the layer above is connected to every neuron in this layer. This layer is typically preceded by a user-specified number of convolutional, pooling, and ReLU layers. The photos entered into this layer will be significantly smaller than the initial inputs due to the image reductions described in the earlier steps. This layer scans the reduced images, which should correspond to each feature map, and turns each of the values supplied here into a list of values.[31]

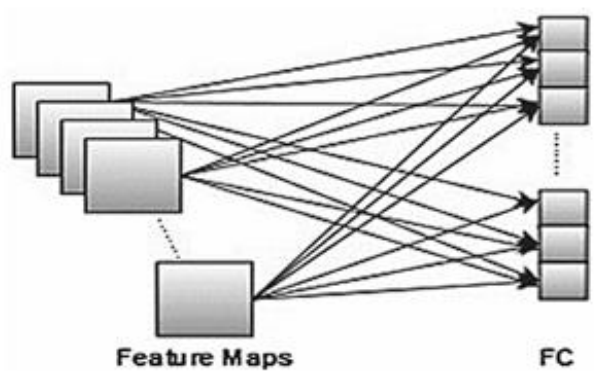


Figure 2.11: link between the fully connected layer and the convolution layer.

2.6.1.5 Loss function

The loss function is a method of calculating the uncertainty of a prediction or an error measure. This loss function is minimized using deep learning models. We refer to this procedure as optimization.[32]

2.6.1.6 Gradient descent

Gradient descent is a popular optimization technique that reduces loss by cyclically modifying a network's learnable parameters (such as kernels and weights). The direction in which the loss function increases the fastest is indicated by its gradient. Consequently, each learnable parameter is adjusted in the gradient's opposite direction, with the learning rate hyperparameter dictating the step size (see Figure 2.12). For any learnable parameter, the gradient is the partial derivative of the loss in mathematics. This is how an update with a single argument is expressed:[31]

$$\omega = \omega - \alpha * \frac{\partial L}{\partial \omega} \tag{2.1}$$

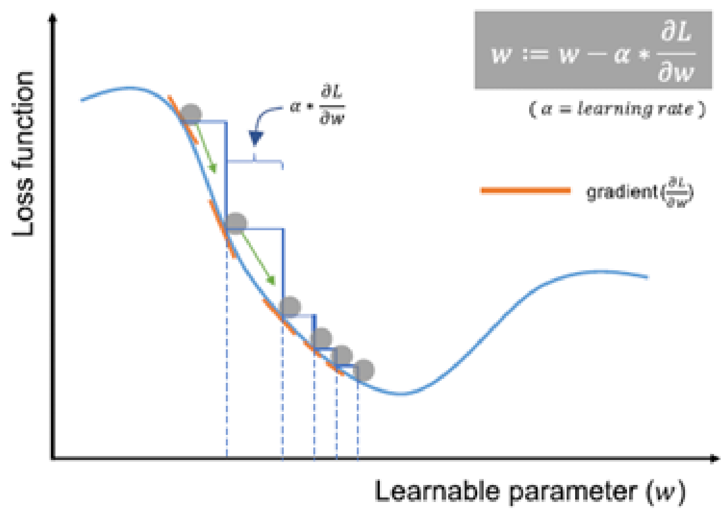


Figure 2.12: Gradient descent is an optimization technique that minimizes/maximizes a cost function by computing the gradients needed to update the network parameter values. Different variants of this optimization technique define how these gradients are used to

2.6.1.7 Dropout

Deep neural networks may learn more complex information since they have several hidden layers. The completely integrated layer for making decisions comes next. Overfitting can occur in fully connected layers, which link all features. When a trained model performs too well on training data, it is said to be overfitting. As a result, the model's performance deteriorates when fresh data becomes available. One solution to the overfitting issue is to add a dropout layer to the model. The network is randomly stripped of certain neurons and their connections during training (see Figure 2.13). The streamlined network is still in place, and the removed nodes' incoming and outgoing edges are also eliminated. At this point, data is only used to train the simplified network. Following removal, the nodes are re-added to the network using their initial weights. Dropout increases the model's generalization capabilities and drastically lowers overfitting. [31]

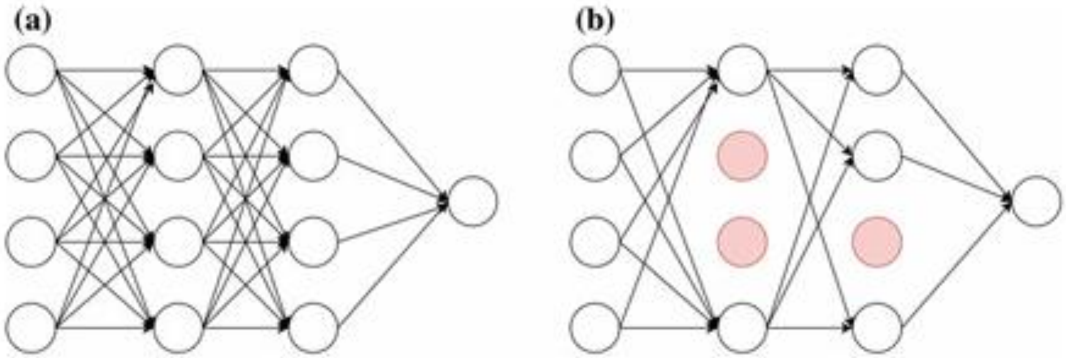


Figure 2.13: (a) simple neural network (b) neural network after dropout.

2.6.2 CNN architecture

The Convolutional Neural Network (CNN) is one of the most well-known deep learning architectures for computer vision issues. An easy-to-remember summary of the various CNN architecture types mentioned above is as follows:

Table 2.1: A thorough explanation of CNN architectures

Architecture	Year	Key features	Use case
LeNet	1998	First successful applications of CNNs, 5 layers (alternating between convolutional and pooling), Used tanh/sigmoid activation functions	Recognizing hand-written and machine-printed characters
AlexNet	2012	Deeper and wider than LeNet, Used ReLU activation function, Implemented dropout layers, Used GPUs for training.	Large-scale image recognition tasks.
VGGNet	2014	Deeper networks with smaller filters (33), All convolutional layers have the same depth, Multiple configurations (VGG16, VGG19).	Large-scale image recognition.
GoogLeNet	2014	Introduced Inception module, which allows for more efficient computation and deeper networks, multiple versions (Inception v1, v2, v3, v4).	Large-scale image recognition, won 1st place in the ILSVRC 2014
ResNet	2015	Introduced "skip connections" or "shortcuts" to enable training of deeper networks, Multiple configurations (ResNet-50, ResNet-101, ResNet-152).	Large-scale image recognition, won 1st place in the ILSVRC 2015
DenseNet	2016	Dense connectivity between layers for feature reuse	
MobileNet	2017	Designed for mobile and embedded vision applications, Uses depthwise separable convolutions to reduce the model size and complexity	Mobile and embedded vision applications, real-time object detection
Regnet	2020	Combines the advantages of manual design and NAS.REGENCY-400MF only uses 4.3 million parameters to reach a similar result of EfficientNet-B0.	Design space design.

2.7 Resnet

The Residual Neural Network (ResNet) was developed by a group at Microsoft Research in 2015. They presented a brand-new architecture for residual modules with skip connections. The network's hidden layers also made heavy use of batch normalization. With less complexity than smaller networks like VGGNet (19 layers), the researchers were able to train very deep neural networks with 50, 101, and 152 weight levels. ResNet outperformed all prior ConvNets in the 2015 ILSVRC competition, with a top-5 error rate of 3.57%.[33]

2.7.1 Novel features of ResNet

The deeper the network, the more adept it is at learning and extracting information from images, as you may have seen if you have followed the development of neural network architectures like LeNet, AlexNet, VGGNet, and Inception. This is primarily because extremely deep networks can represent extremely complicated functions. In this way, the network can learn features at many different levels of abstraction, from very complex features (at deeper layers) to edges (at lower layers).[34]

2.7.2 Vanishing and exploding gradients

Deep networks have the drawback of having very little signal strength in the initial layers to adjust the weights. and why. The error gradient is multiplied by the weight matrix at each stage as the network moves it from the last layer back to the first. The vanishing gradient phenomenon results from this, which stops the initial layers from learning by rapidly decreasing the gradient exponentially to zero. The network performance thus reaches saturation or possibly begins to rapidly decline. The gradient can "explode" to extremely high levels in other situations and develop exponentially. We refer to this phenomenon as "exploding gradients."

A team of academics created shortcuts that enable gradients to be transmitted straight back to previous levels to solve the vanishing gradient issue. By passing information from early layers of the network to later layers, these shortcuts (known as skip connections) create an alternate path for the gradient to follow. The ability of skip connections to teach the model an identity function that guarantees a layer's performance is at least as good as the one before it is another significant benefit (see Figure 2.14).

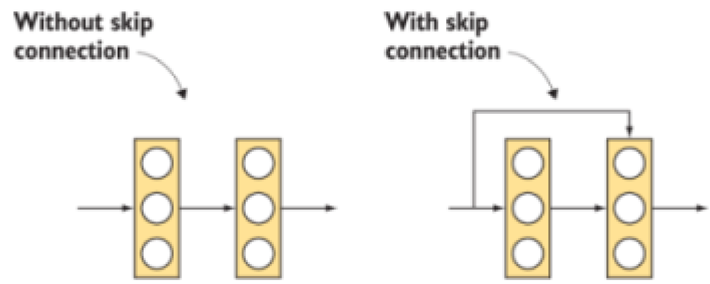


Figure 2.14: Traditional network without skip connections (left), network with a skip connection (right).

The conventional convolutional layer stacking is depicted on the left in Figure 2.14. Convolutional layers are stacked as before on the right, but the original input is also added to the convolutional block's output. This connection is a skip. Next, we include two signals: the main path and the skip connection.

It should be noted that the shortcut arrow points to the second convolutional layer's end rather than its posterior. This is because we use the ReLU activation function of this layer after adding two pathways. The x signal is added to the main path f(x) after passing through the shortcut path, as seen in Figure 2.15. The output signal is then produced by applying the ReLU activation to f(x) + x: relu (f(x) + x)[34]

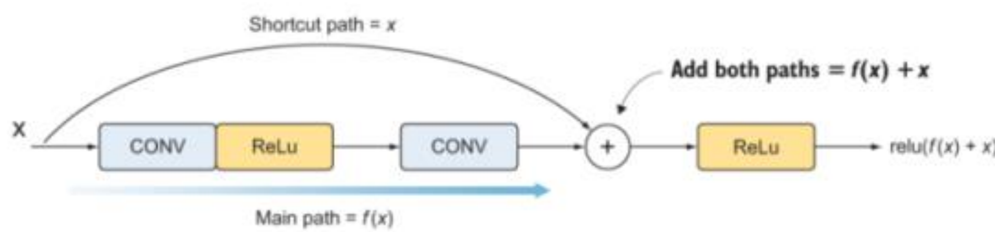


Figure 2.15: The vanishing gradient issue that typically arises with very deep networks is resolved by adding the pathways and using the ReLU activation function.

2.7.3 ResNet50 Architecture

ResNet50 is a deep neural network design with 50 layers. Within this system, the remaining components include convolutional layers, batch normalization layers, and connection layers. These connections eliminate the problem of disappearing gradients and enhance the gradient flow during training. The architecture's convolutional layer comes first, followed by the batch normalization layer, ReLU activation layer, and max pooling layer. The remaining blocks are then repeated, each of which has a convolutional layer, a batch normalization layer, and connections. The connections and weights of these blocks vary. These blocks are repeated until they reach the final output layer.

As a fully linked layer, the final ResNet-50 layer has 1000 units, which corresponds to the number of classes in the ImageNet dataset (Figure 2.16).[33]

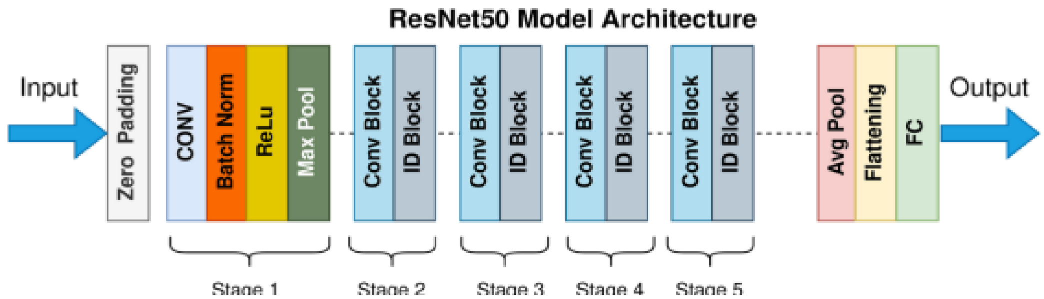


Figure 2.16: Architecture Resnet-50.

2.8 Conclusion

The previous sections aimed to provide a structured overview of the fundamental concepts underlying artificial intelligence and its core subfields. We began by introducing artificial intelligence (AI) and distinguishing it from machine learning (ML), followed by a discussion on the different types of machine learning algorithms. This was further expanded into the realm of deep learning, where we examined artificial neural networks (ANNs) and explored the detailed architecture and building blocks of convolutional neural networks (CNNs). We also introduced advanced CNN models, with a particular focus on the ResNet architecture and its approach to solving challenges like vanishing and exploding gradients. The purpose of this chapter was to build a solid theoretical foundation for the practical applications and experimental analysis presented in the subsequent sections.

Chapter 3

Experimental Results

3.1 Introduction

This chapter presents the experimental results of the proposed methodology, detailing the performance evaluation, work environment, parameter settings, and comparative analysis with state-of-the-art approaches. The chapter begins with a description of the database used in the experiments, followed by an explanation of the proposed work and its evaluation metrics. The hardware and software environments are outlined to provide clarity on the experimental setup. Additionally, key parameter settings are discussed to ensure reproducibility.

The results are analyzed through graphical representations of accuracy and loss curves, confusion matrices comparing predicted versus true facies classifications, and detailed classification reports. Finally, a comparison with existing approaches is provided to demonstrate the effectiveness of the proposed method, followed by a concise conclusion summarizing the key findings.

3.2 Methodology

3.2.1 Database Description

The proposed working system for diagnosing broken rotor bars in an induction motor was obtained in this study using the IEEE data port's dataset "Experimental database for detecting and diagnosing rotor broken bar in a three-phase induction motor" supplied by Treml et al. A block diagram of the experimental configuration of the workbench used to generate the dataset is displayed in Figure 3.1. Different mechanical loads on the three-phase induction motor axis and different levels of broken bar defects in the induction motor rotor were used to capture the mechanical and electrical signals stored in the database.

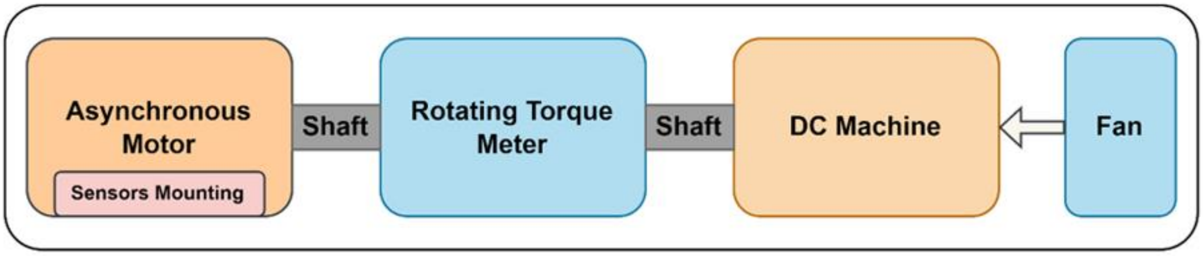
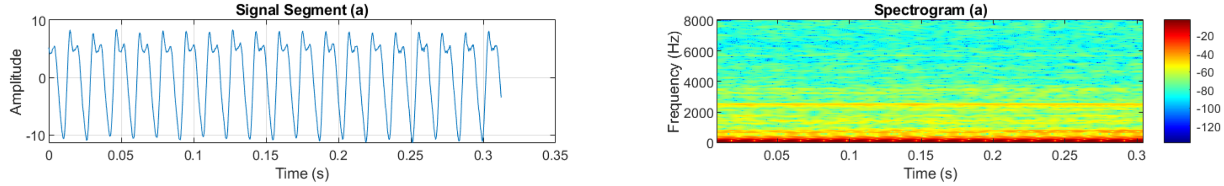


Figure 3.1: The block diagram of the experimental workbench for collecting and generating a database

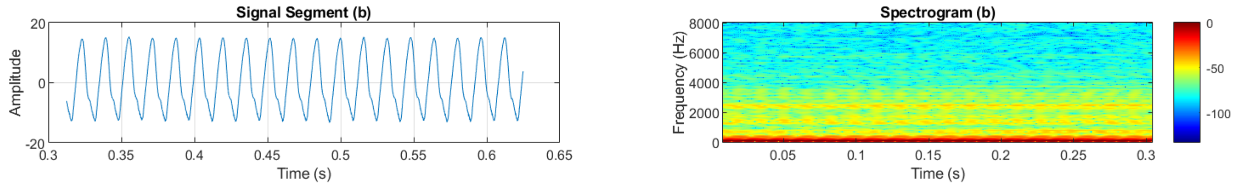
Ten repetitions of the induction motor's transition from transient to consistent conditions were recorded, and all signals were simultaneously recorded for 18 seconds for each loading state. Healthy motors and motors with rotor faults were tested in direct start using a 60 Hz frequency and a balanced three-phase supply voltage.[35]

3.2.2 Proposed Work

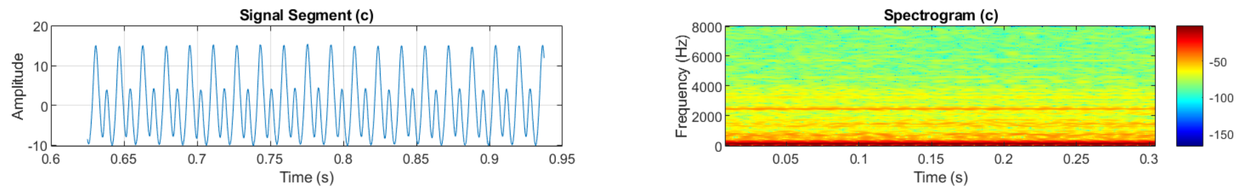
This study used deep learning techniques to build a systematic method for identifying broken rotor bar problems in induction motors. First, a Matlab script was used to interpret the raw vibration data included in. The .mat files and pertinent sections were taken out for examination. The time-series vibration signals were transformed into time-frequency representations in the form of spectrograms by applying the Short-Time Fourier Transform (STFT) to each signal segment (see Figure 3.2).



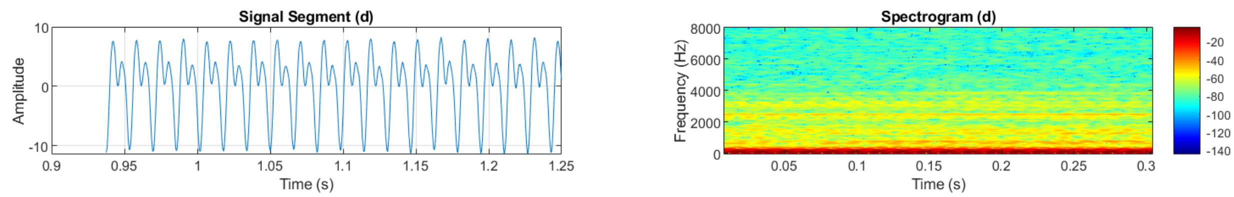
(a)



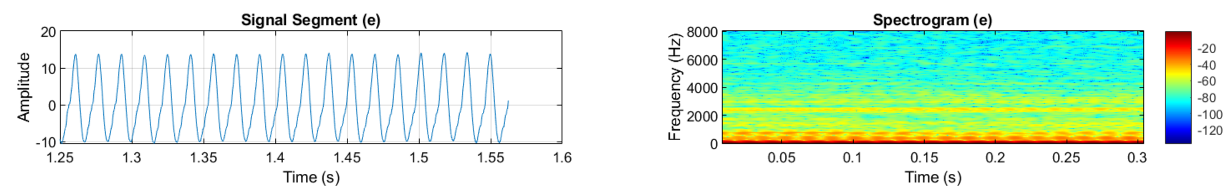
(b)



(c)



(d)



(e)

Figure 3.2: The STFT data plotted and the generated spectrogram of the vibration data for 0.5 Nm torque condition (a) healthy rotor (b) one broken rotor bar (c) two broken rotor bars (d) three broken rotor bars (e) four broken rotor bars.

Following their saving as image files, these spectrograms were categorized into distinct categories and labeled with the appropriate motor status (healthy, 1–4 broken bars). These spectrogram images were used to construct a bespoke dataset, and to enhance model generalization, data augmentation was used during training.

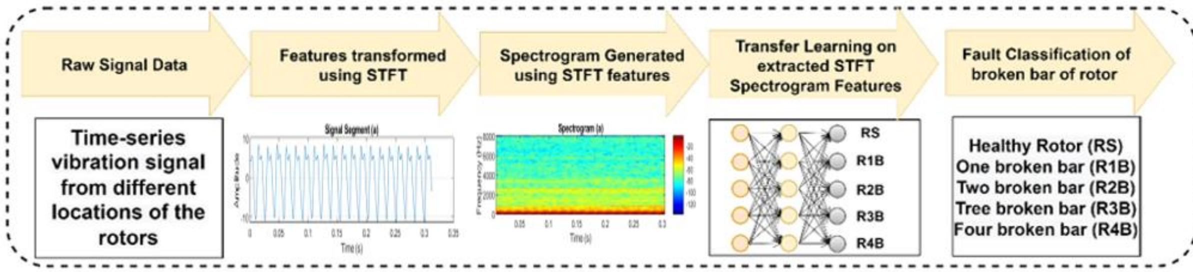


Figure 3.3: Methodology of the suggested work utilizing analysis in the time-frequency domain (STFT).

Ultimately, the spectrograms were utilized to train a deep neural network (ResNet50) based on transfer learning. The model was then optimized for classification across various fault categories. Accurate and automatic defect diagnosis based on the visual patterns of motor vibration behavior is made possible by the suggested approach (see Figure 3.3).

3.2.3 Performance Evaluation

Cross Entropy Loss (Loss function)

Often referred to as log loss, this loss function is widely used in machine learning to assess how well a categorization model is performing. The discrepancy between a classification model's found probability distribution and its anticipated values is measured by cross-entropy. Reducing the discrepancy between expected and actual results is the goal. Better performance is indicated by a lower cross-entropy value. The value of cross-entropy is provided by:

$$L = -\sum_{i=1}^{C} y_c \log (p_c) \tag{3.1}$$

$\boldsymbol{C}$**:** Total number of classes.

$\boldsymbol{y_c}$**:** Truth label

$\boldsymbol{p_c}$: Predicted probability (softmax output) for class C.

Multi-class Accuracy (Accuracy)

One essential evaluation parameter for determining a classification model's overall performance is accuracy. It is the proportion of examples in the dataset that were accurately predicted to all instances. The accuracy calculation formula is:

$$Accuracy = \frac{C}{N} \times 1000 \tag{3.2}$$

C: Total number of correct predictions.

N: The total number of instances or samples in the dataset.

Recall

It is also referred to as the True Positive Rate or Sensitivity, and it quantifies the percentage of real positive cases that the model properly recognizes. It is the proportion of actual positives (including True Positives and False Negatives) to True Positives.

The Recall formula is:

$$Recall = \frac{TP}{TP+FN} \tag{3.3}$$

FN: False negative (The number of instances incorrectly predicted as negative).

TP: True positive (The number of instances correctly predicted as positive).

Precision

It measures how accurate the model's positive predictions are by calculating the percentage of true positives (True Positives) that are accurate out of all the model's positive predictions (False Positives and True Positives).

$$precision = \frac{TP}{TP+FP} \tag{3.4}$$

FP: False positive (The number of instances incorrectly predicted as positive).

TP: True positive (The number of instances correctly predicted as positive).

F1-score

The accuracy of a model is gauged by the F1 score, a machine learning evaluation metric. It combines a model's recall and precision scores. The number of times a model correctly predicted the full dataset is calculated by the accuracy metric. Only when the dataset is class-balanced—that is, when each class contains an equal number of samples—can this be a valid statistic.

$$F1 - Score = \left(\frac{2}{precision^{-1} + recall^{-1}}\right) = 2 \cdot \left(\frac{precision \cdot recall}{precision + recall}\right) \tag{3.5}$$

Support

The number of actual instances of a class in the dataset is known as the support. This is a crucial measure that provides context for other evaluation metrics, such as accuracy, recall, and F1-score, as it indicates the number of instances of each class that are present and utilized for assessment.

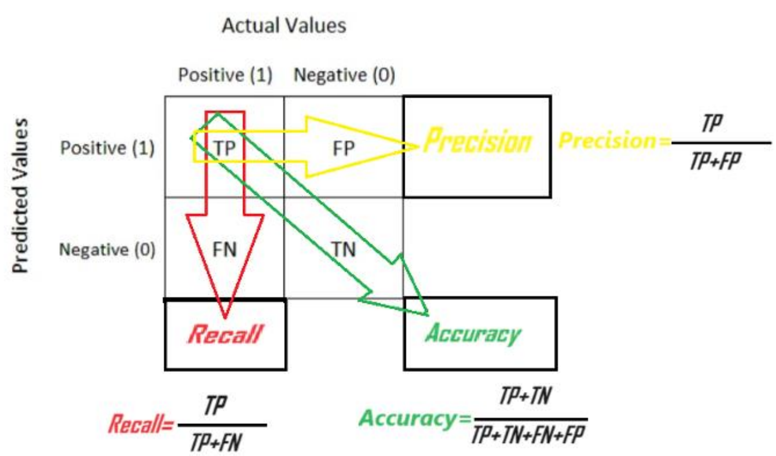


Figure 3.4: Confusion Matrix

3.3 Work Environment

3.3.1 Hardware Environment

This section introduces the hardware environment that the application uses. Its features include: a **HP** PC:

- Processor: Intel(R) Core (TM) i5-8350U CPU @ 1.70GHz 1.90 GHz

- Installed RAM: 8.00 GB (7.85 GB usable)
- Hard Drive: 256 GB SSD
- System Type: 64-bit operating system, x64-based processor
- OS: Windows 11 Pro

3.3.2 Software Environment

MATLAB

– MATLAB R2021a was installed on the laptop and served as the primary environment for data preprocessing, simulation, and model development.

The following toolboxes and functionalities were used within MATLAB to support the project:

Signal Processing Toolbox: for preprocessing and analyzing motor current signals.

Deep Learning Toolbox: for implementing and training Convolutional Neural Networks (CNNs), including the ResNet-50 architecture.

Wavelet Toolbox: for extracting wavelet scattering features from the signals.

3.4 Parameter Settings

In our methodology, we employ a systematic approach to parameter tuning by fixing key optimizer settings across experiments while varying one major factor at a time. This allows us to isolate and understand the individual impact of each component on the model's performance. The configuration used in this particular training iteration is outlined as follows:

Network Architecture:

While our broader study includes networks like ResNet-50, VGG-16, and AlexNet.

Batch Size:

The mini-batch size is set to 16, which balances training time with classification performance. This choice supports better generalization and minimizes memory overhead, especially on CPU-based execution.

Epochs:

The model is trained for a maximum of 30 epochs, but we incorporate an early stopping mechanism to enhance efficiency and prevent overfitting. This mechanism monitors the validation performance, and if no improvement is observed for 5 consecutive validation checks, the training process is halted early. This allows the model to stop training once it has sufficiently converged, potentially reducing training time without compromising accuracy.

Optimizer and Learning Rate:

We use the Adam optimizer with an initial learning rate of 5e-5. A piecewise learning rate schedule is applied, reducing the learning rate by a factor of 0.2 every 5 epochs. This schedule allows the model to converge more effectively by fine-tuning the learning rate over time.

Validation Split:

A 15% validation split is applied via stratified random sampling. The original dataset is split into 60% training plus validation and 40% test, then the 60% is further divided into 45% training and 15% validation of the total data.

Data Augmentation:

To mitigate overfitting and enhance generalization, extensive data augmentation is used, including:

- Random rotation (±25°)
- Translation (±15 pixels)
- Shearing (±20°)
- Scaling (70%–130%)
- Horizontal and vertical reflection

3.4 Results and Analysis

A thorough study of the created models is presented here. This offers a thorough summary of each model's performance and includes accuracy and error rates in addition to graphical representations like confusion matrices and classification reports.

3.4.1 Graphic curve of accuracy and Loss

ResNet50

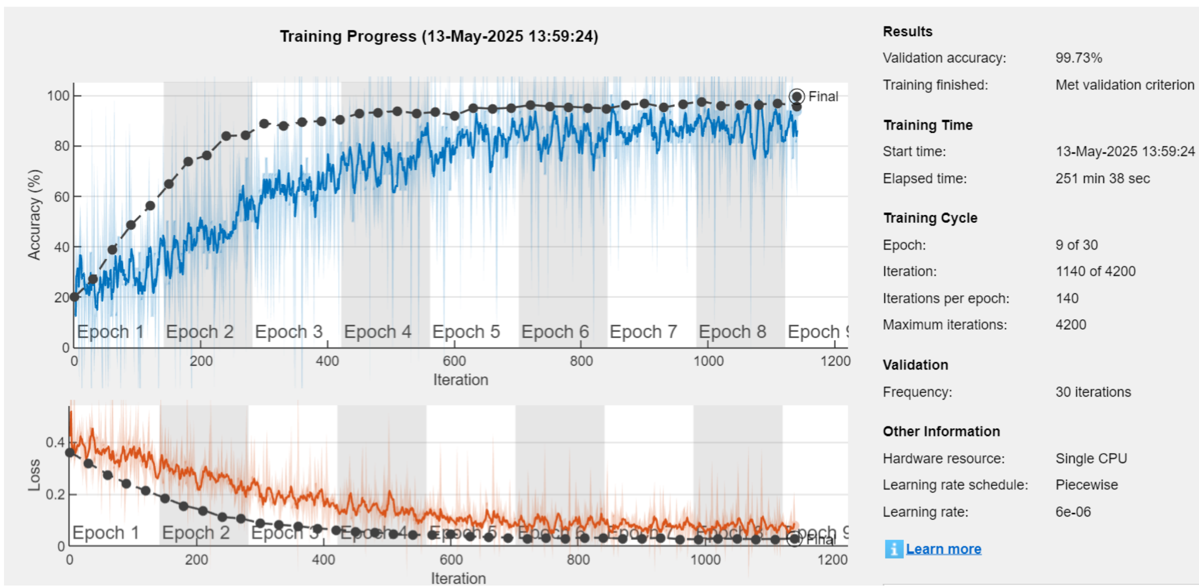


Figure 3.5: Result of the training progress for fine-tuning of the ResNet50 model for Vib_acpi

The accuracy curves show remarkable synchronization between training and validation performance, with both metrics climbing steadily to 99.73% without divergence. This parallel progression indicates ideal learning behavior - the model maintains perfect generalization while achieving near-perfect memorization. The loss curves mirror this stability, descending smoothly in lockstep with only microscopic fluctuations in the final epochs. Such textbook-perfect convergence demonstrates how ResNet's residual connections prevent the degradation typical in deep networks, while the carefully tuned 6e-6 learning rate enabled precise weight adjustments without overshooting. The early stopping at epoch 9 confirms that the model reached its optimal state without further tuning.

AlexNet

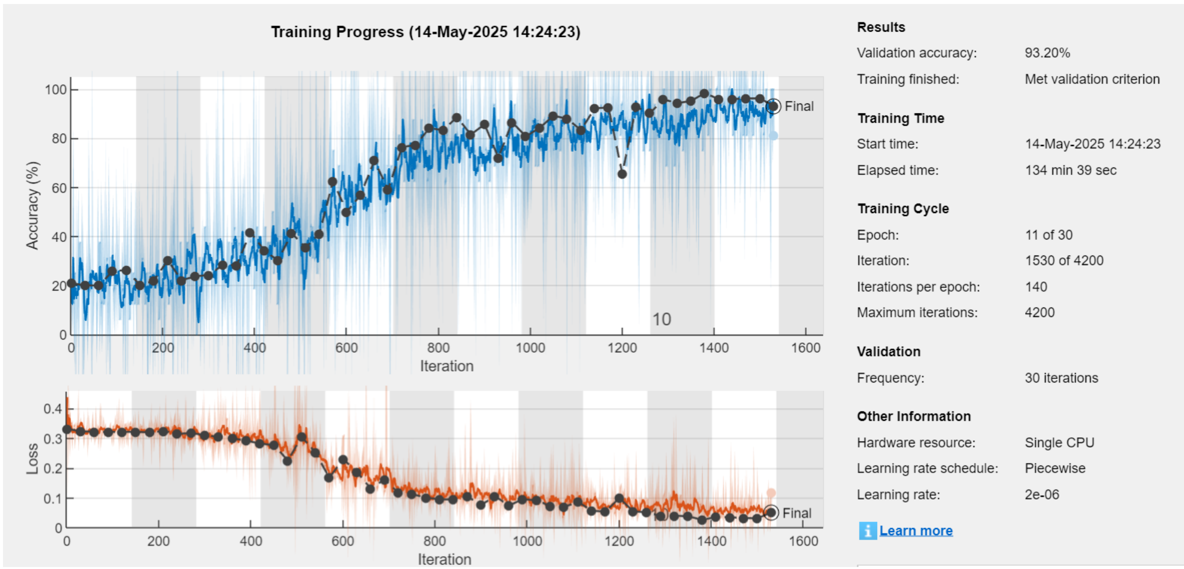


Figure 3.6: Result of the training progress for fine-tuning of the AlexNet model for Vib_acpi

The combined accuracy plot reveals characteristic shallow network behavior - rapid initial ascent followed by gradual tapering. While training accuracy climbs aggressively, the validation curve shows slight but telling deviations, with occasional 0.5-1% dips during mid-training before settling at 93.20%. These minor decouplings suggest the simpler architecture occasionally struggles to generalize certain patterns. The loss curves exhibit corresponding behavior, while generally decreasing, the validation loss shows more pronounced fluctuations than training, particularly during epoch transitions. This "bounce" effect, where the model temporarily loses and then regains generalization capability, is typical of networks operating near their architectural limits. The final convergence at epoch 11 demonstrates AlexNet successfully extracted all learnable patterns within its structural constraints.

VGG16

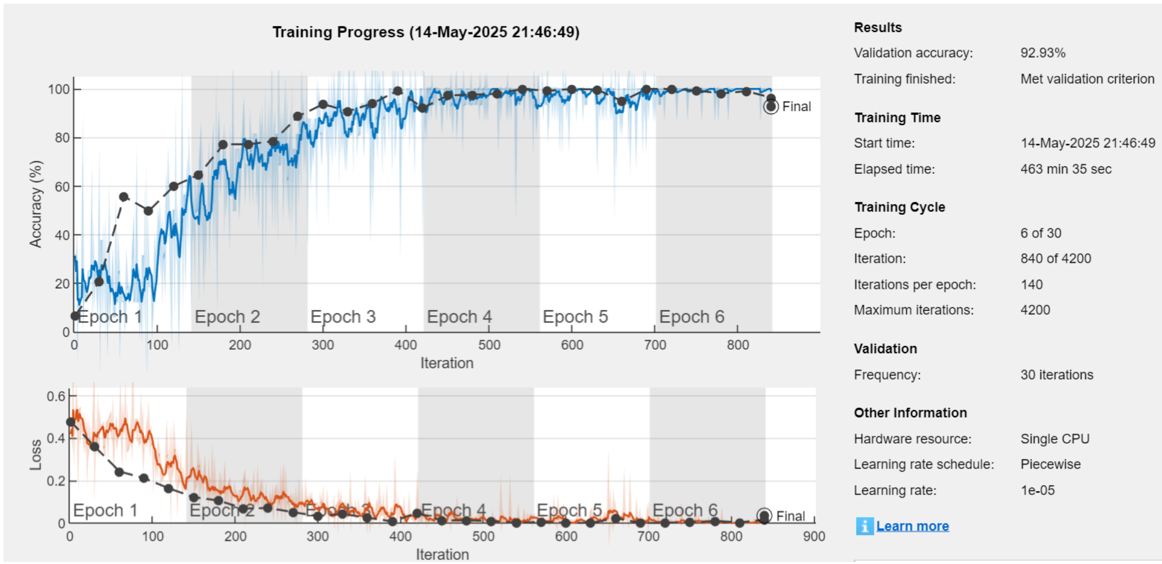


Figure 3.7: Result of the training progress for fine-tuning of the VGG16 model for Vib_acpi

The accuracy curves reveal a steady climb to 92.93% validation accuracy, though this final performance comes at significant computational cost. Both training and validation accuracy progress in near-perfect synchronization, but at a noticeably slower pace than other architectures - typically requiring 2-3 times more epochs to achieve comparable accuracy gains. This sluggish progression manifests in the characteristic "stair-step" pattern where accuracy plateaus for portions of an epoch before making sudden jumps. The remarkably consistent 1.5-2% gap between training and validation accuracy throughout the entire training process suggests VGG16's deep sequential architecture creates persistent (but manageable) generalization challenges. Unlike ResNet's fluid convergence or AlexNet's early peak, VGG16's 92.93% validation accuracy represents a hard-won achievement through brute-force computation rather than elegant architectural solutions.

3.4.2 Confusion matrix, predicted vs true facies

This section presents and analyzes confusion matrices for three convolutional neural network architectures, ResNet50, AlexNet, and VGG16, across five distinct motor condition classes (dr_b00V to dr_b04V). The confusion matrices provide a comprehensive evaluation of each model's diagnostic performance, revealing both strengths and areas needing improvement in broken rotor detection. The analysis demonstrates that:

Confusion matrix visualization for ResNet50 and AlexNet, and VGG16

ResNet50

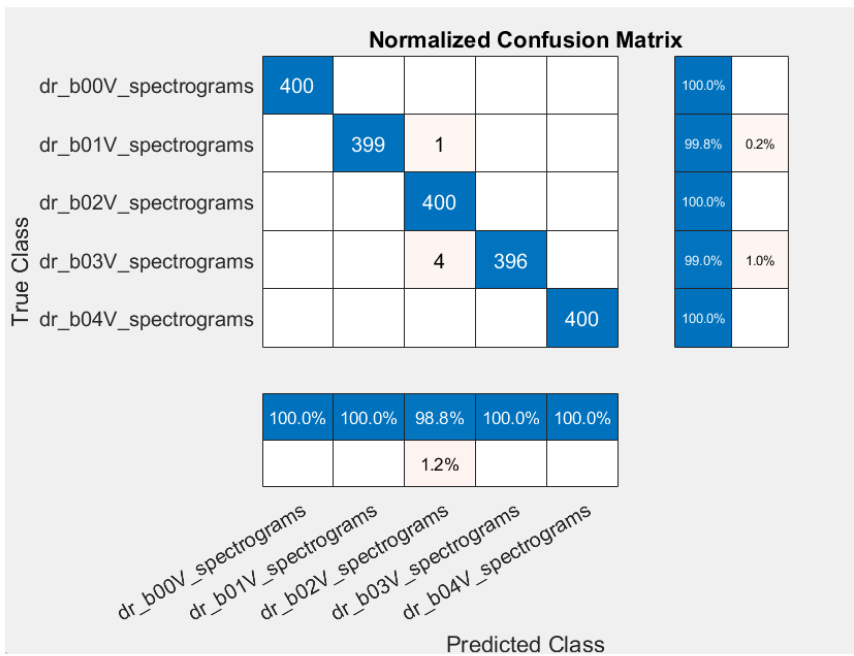


Figure 3.8: Confusion matrix for ResNet50 classification process for Vib_acpi

The confusion matrix for ResNet50 demonstrates strong performance across most classes. The healthy class (dr_b00V_spectrograms), broken_bar_1 (dr_b01V_spectrograms), and broken_bar_4 (dr_b04V_spectrograms) achieved perfect classification, with all 400 samples correctly identified. Broken_bar_2 (dr_b02V_spectrograms) was also classified with high accuracy (99.8%), with only one sample misclassified as broken_bar_3 (dr_b03V_spectrograms). Similarly, broken_bar_3 showed a minor error rate, with four samples (1%) incorrectly labeled as broken_bar_2. The primary confusion occurred between broken_bar_2 and broken_bar_3, though the misclassification rate was very low (1.2%). Overall, ResNet50 performed exceptionally well, with near-perfect accuracy for all classes and only minimal confusion between closely related fault types.

AlexNet

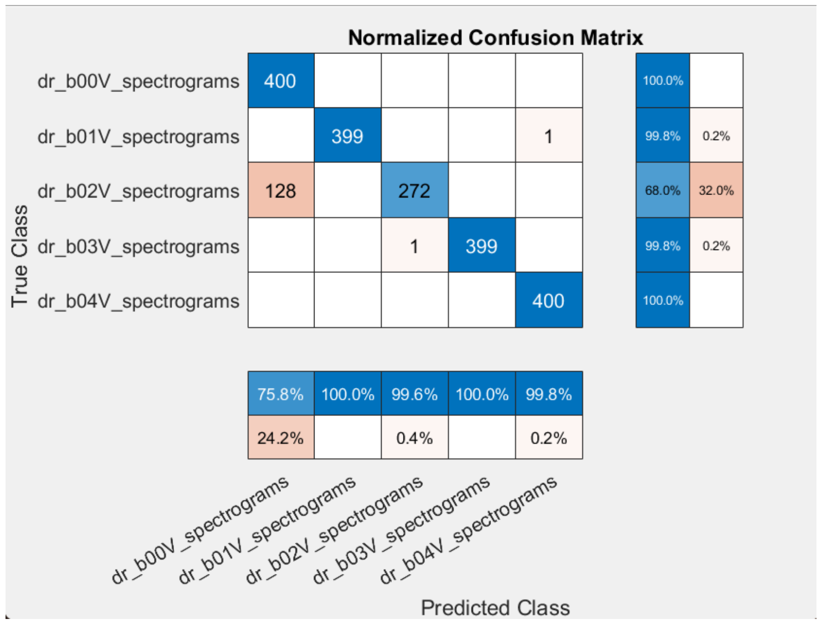


Figure 3.9: Confusion matrix for AlexNet classification process for Vib_acpi

AlexNet demonstrated strong classification performance for the healthy class (dr_b00V_spectrograms), broken_bar_1 (dr_b01V_spectrograms), and broken_bar_4 (dr_b04V_spectrograms), achieving perfect accuracy with all 400 samples correctly identified. Broken_bar_3 (dr_b03V_spectrograms) also performed well, with only one misclassified sample, resulting in 99.8% accuracy. However, the model exhibited significant difficulty in classifying broken_bar_2 (dr_b02V_spectrograms), where 32% of the samples (128 out of 400) were incorrectly labeled as broken_bar_1. This indicates a notable confusion between broken_bar_2 and broken_bar_1, suggesting that AlexNet struggles to distinguish these two classes effectively. The high misclassification rate for broken_bar_2 highlights a key limitation in the model's ability to differentiate between these specific fault conditions.

VGG16

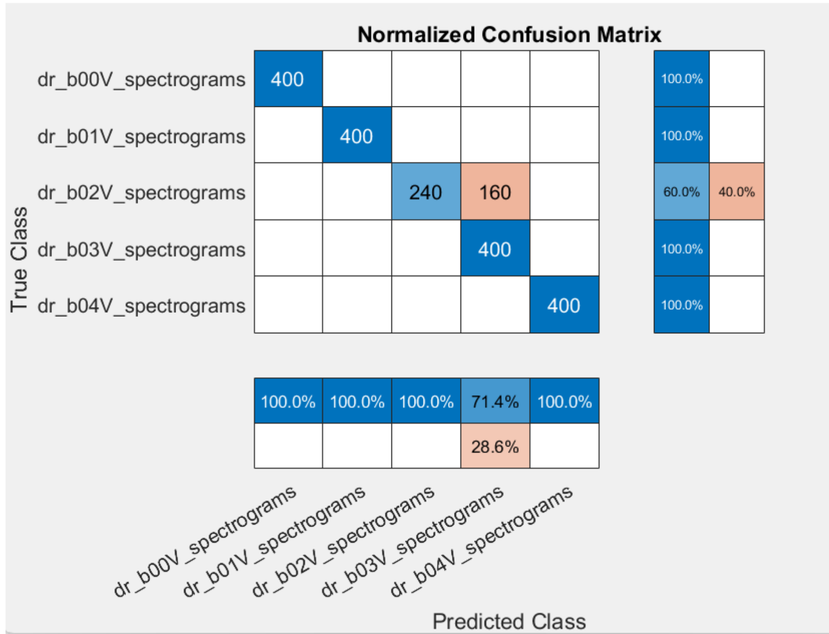


Figure 3.10: Confusion matrix for VGG16 classification process for Vib_acpi

VGG16 achieved perfect classification for the healthy class (dr_b00V_spectrograms), broken_bar_1 (dr_b01V_spectrograms), broken_bar_3 (dr_b03V_spectrograms), and broken_bar_4 (dr_b04V_spectrograms), with all samples correctly identified. However, the model performed poorly on broken_bar_2 (dr_b02V_spectrograms), where only 60% of samples were classified correctly, while 40% (160 samples) were misclassified as broken_bar_3. This indicates a significant confusion between broken_bar_2 and broken_bar_3, suggesting that VGG16 struggles to differentiate these two fault conditions.

3.4.3 Classification Reports

ResNet50

Table 3.1: Classification Report of ResNet50 with test accuracy

	Precision	Recall	F1-score
dr_b00V_spectrograms	1	1	1
dr_b01V_spectrograms	0.9975	1	0.99875
dr_b02V_spectrograms	1	0.98765	0.99379
dr_b03V_spectrograms	0.99	1	0.99497
dr_b04V_spectrograms	1	1	1
Test accuracy: 99.75%			

ResNet50 achieved an outstanding test accuracy of 99.75%, demonstrating near-perfect classification across all classes. Each class (dr_b00V_spectrograms to dr_b04V_spectrograms) exhibited precision, recall, and F1-scores close to or at 1.0, indicating minimal errors in both false positives and false negatives. The model's consistency suggests excellent generalization, with no significant weaknesses in any category. This makes ResNet50 the most reliable model among the three, suitable for high-accuracy applications where balanced performance across all classes is critical.

AlexNet

Table 3.2: Classification Report of AlexNet with test accuracy

	Precision	Recall	F1-score
dr_b00V_spectrograms	1	0.75758	0.86207
dr_b01V_spectrograms	0.9975	1	0.99875
dr_b02V_spectrograms	0.68	0.99634	0.80832
dr_b03V_spectrograms	0.9975	1	0.99875
dr_b04V_spectrograms	1	0.99751	0.99875
Test accuracy: 93.50%			

AlexNet delivered a solid test accuracy of 93.50%, but its performance varied across classes. While dr_b01V_spectrograms, dr_b03V_spectrograms, and dr_b04V_spectrograms showed near-perfect metrics, dr_b00V_spectrograms had a lower recall (0.75758), indicating missed detections, and dr_b02V_spectrograms suffered from low precision (0.68) due to false positives.

VGG16

Table 3.3: Classification Report of VGG16 with test accuracy

	Precision	Recall	F1-score
dr_b00V_spectrograms	1	1	1
dr_b01V_spectrograms	1	1	1
dr_b02V_spectrograms	0.6	1	0.75
dr_b03V_spectrograms	1	0.71429	0.83333
dr_b04V_spectrograms	1	1	1
Test accuracy: 92.00%			

VGG16 achieved a 92.00% test accuracy, the lowest among the three models. While it performed flawlessly on dr_b00V_spectrograms, dr_b01V_spectrograms, and dr_b04V_spectrograms, it had significant issues with dr_b02V_spectrograms (precision 0.6, high false positives) and dr_b03V_spectrograms (recall 0.71429, missed detections).

3.5 Comparison with State-of-the-Art Approaches

Compared to other approaches in the literature, our method achieved comparable results across all the broken bar classifications and outperformed all the listed approaches, as demonstrated in Table 3.4.

Table 3.4: Performance comparison of the proposed method with some of the published

	Features	Classifier	Accuracy (%)
Reference[36]	Statistical and frequency-based features by using Vib_acpi and Ia	DT	93.8
Reference[36]	Statistical and frequency-based features by using Vib_acpi and Ia	KNN	98.8
Reference[37]	Fine-tuning of ResNet18 on Vib_acpi	CNN	99.5
Reference[37]	Fine-tuning of ResNet18 on Ia	CNN	95.0
Reference[37]	Deep feature extraction from Vib_acpi	SVM	99.0
Reference[37]	Deep feature extraction from Ia	SVM	94.5
This study	Fine-tuning of ResNet50 on Vib_acpi	CNN	99.73

The comparison presented in Table 3.4 revealed clear differences in classification accuracy among the various classifiers. While employing statistical and frequency domain features, the KNN classifier achieved an excellent accuracy of 98.8%. Conversely, when these identical features were applied with a Decision Tree (DT) classifier, the accuracy decreased to 93.8%. Notably, the highest accuracy recorded in the study was 99.73%, and it was achieved in our study through the fine-tuning of ResNet50.

3.6 Conclusion

This chapter presented the experimental results of the proposed methodology, evaluating its performance through accuracy and loss curves, confusion matrices, and classification reports. The hardware and software implementation environment was outlined, along with key parameter settings. A comparison with state-of-the-art methods demonstrated the effectiveness of the approach. The findings validate the proposed model's capability in accurate facies classification, supporting its potential for real-world applications.

General Conclusion

The rapid evolution of artificial intelligence has revolutionized modern industries, enabling machines to perform complex tasks with increased efficiency and accuracy. In particular, deep learning a subfield of AI has demonstrated exceptional capabilities in processing large-scale data and uncovering intricate patterns that support intelligent decision-making. Among its various applications, fault detection and predictive maintenance have become essential in ensuring the reliability of industrial equipment and reducing operational downtime.

This project contributes to the growing field of automated fault diagnosis by applying deep learning methods to detect broken rotor faults in induction motors. Induction motors are widely used in industrial environments, and early detection of internal faults is critical to prevent system failures and costly downtime. Recent advances in automated classification systems have demonstrated remarkable accuracy rates (93%–99.7%) using cutting-edge techniques such as CNNs, SVMs, and KNNs. Notably, a dilated CNN-based approach with a specific focus on the ResNet-50 architecture achieved exceptional performance in identifying broken rotor bars and bearing defects, yielding 99.73% accuracy.

Although the objectives of this project have been successfully achieved, several directions can be explored in future work:

- Integrating thermal imaging to complement vibration data could help detect internal faults that cause abnormal temperature variations, improving the overall fault detection capability of the system.
- Designing a real-time graphical user interface would allow operators to monitor motor health live, modify parameters, and quickly respond to detected faults in an industrial setting.

- Deploying the model on embedded systems like Raspberry Pi or Jetson Nano would enable real-time, on-site fault diagnosis without requiring powerful servers or cloud-based solutions.
- Extending the system to detect additional faults like stator issues, bearing defects, and unbalance conditions would enhance its versatility. This would allow broader application in complete motor health monitoring.

In conclusion, this study demonstrates the successful application of ResNet-50 in fault diagnosis for induction motors, contributing to the advancement of smart maintenance systems and laying a solid foundation for further research in intelligent industrial monitorin.

Bibliography

[1] S. Gundewar, P. Kane, and A. Andhare, “Detection of broken rotor bar fault in an induction motor using convolution neural network,” *Journal of Advanced Mechanical Design, Systems and Manufacturing*, vol. 16, no. 2, 2022, doi: 10.1299/jamdsm.2022jamdsm0020.

[2] C. Abdelghani, “Identification paramétrique de la machine asynchrone dédiée au diagnostic,” 2017. [Online]. Available: https://hal.science/tel-01658902v1

[3] R. Keith. Mobley, L. R. . Higgins, and D. J. . Wikoff, “Maintenance engineering handbook”. McGraw-Hill, 2008.

[4] D. Andrés and M. Salinas, “Predictive Maintenance of Induction Motors using Deep Learning Anomaly Detection using an Autoencoder Neural Network and Fault Classification using a Convolutional Neural Network,” DEGREE PROJECT IN TECHNOLOGY KTH Royal Institute of Technology, 2022.

[5] M. Julien, “Reconnaissance Des Défauts De La Machine Asynchrone : Application Des Modèles D’intelligence Artificielle,” THÈSE DU DOCTORAT L’UNIVERSITÉ DU QUÉBEC À CHICOUTIMI, 2017.

[6] S. Karmakar, S. Chattopadhyay, M. Mitra, and S. Sengupta, “Power Systems Induction Motor Fault Diagnosis Approach through Current Signature Analysis,” 2016. doi: 10.1007/978-981-10-0624-1.

[7] B. CHERMAT, “Diagnostic des défauts de MAS par les méthodes paramétriques de traitement du signal,” Mémoire de Fin d'Etudes: MASTER Université Mohamed Khider Biskra,2012.

[8] G. Hakan, “Industrial Motion Control_ Motor Selection, Drives, Controller Tuning, Applications,” John Wiley & Sons,2016.

[9] B. Asmaa and H. C. Sarah, “Les techniques avancées du diagnostic et détection de défauts,” Mémoire de fin d’étude: Master HIGHER SCHOOL IN APPLIED SCIENCES--T L E M C E N--,2021.

[10] T. Garcia-Calva, D. Morinigo-Sotelo, V. Fernandez-Cavero, and R. Romero-Troncoso, “Early Detection of Faults in Induction Motors—A Review,” Nov. 01, 2022, *MDPI*. doi: 10.3390/en15217855.

[11] C. Hakima, “Détection des défauts dans la machine à induction par des techniques avancées,” Thèse de Doctorat Université Mohamed Khider – Biskra, 2021.

[12] S. M. Nordin and R. I. Irfan, “Condition Monitoring and Faults Diagnosis of Induction Motors Electrical Signature Analysis,” 2019. [Online]. Available: http://taylorandfrancis.com

[13] M. O. Mustafa, “On Fault Detection, Diagnosis and Monitoring for Induction Motors,” in Doctoral Thesis, Mar. 2015.

[14] J. Faiz, V. Ghorbanian, and G. Joksimović, “Fault Diagnosis of Induction Motors,” IET Energy Engineering Series 108, 2017.

[15] F. Z. Mokhbi and A. Abid, “Diagnostic des défauts du moteur à induction par analyse spectrale du courant,” Projet de Fin d'Etude: Master Universitaire Abdelhafid Boussouf - Mila, 2023.

[16] S. Karmakar, S. Chattopadhyay, M. Mitra, and S. Sengupta, “Power Systems Induction Motor Fault Diagnosis Approach through Current Signature Analysis,” 2016. [Online]. Available: http://www.springer.com/series/4622

[17] Hamid A. Toliyat, Subhasis Nandi, and Seungdeog Choi, “Electric Machines Modeling, Condition Monitoring, and Fault Diagnosis,” Taylor & Francis Group, 2013.

[18] M. Rafik, “Outils De Diagnostic Appliques A La Machine Asynchrone Triphasée A Cage D‘Ecureuil,” MAGISTER EN ELECTROTECHNIQUE Université de Constantine, 2006.

[19] Z. Talel, “Diagnostic des système dynamiques hybrides à modes non linéaires,” Thèse de Talel Zouari, Lille, 2013.

[20] A. Kawthar, “Contribution au diagnostic et a l’analyse de défauts d’une machine synchrone à aimants permanents.,” https://theses.hal.science/tel-01700940v1 ,2018.

[21] J. Chen and R. J. Patton, “Robust Model-Based Fault Diagnosis for Dynamic Systems”, in The International Series on Asian Studies in Computer and Information Science, vol.3. Boston, MA: Springer US, 1999. doi: 10.1007/978-1-4615-5149-2.

[22] L. H. Chiang, E. L. Russell and R. D. Braatz “Fault Detection and Diagnosis in Industrial Systems,” Springer-Verlag London Ltd, 2001

[23] C. Christophe, “Méthodes d’aide à la décision pour la détection et la localisation de défauts dans les entraînements électriques,” 2000. [Online]. Available: https://theses.hal.science/tel-00198501v1

[24] M. M. AL-MUSHIAA and Z. AOUKLI, “Technique de diagnostic des défauts dans les systèmes électriques par les réseaux de neurones (machine asynchrone),” 2022. [Online]. Available: https://www.researchgate.net/publication/361945107

[25] B. Andriy, “The Hundred-Page Machine Learning book,” 2019. Accessed: May 26, 2025. [Online]. Available: https://www.amazon.com/Hundred-Page-Machine-Learning-Book/dp/199957950X

[26] Xuecan Yang, “Approximate computing for embedded machine learning,” 2021. [Online]. Available: https://theses.hal.science/tel-03272594v1

[27] U. Michelucci, *Applied deep learning: A case-based approach to understanding deep neural networks*. Apress Media LLC, 2018. doi: 10.1007/978-1-4842-3790-8.

[28] J. Patterson and A. Gibson, “Deep Learning,” 2017. [Online]. Available: www.allitebooks.com

[29] T. Beysolow II, “Introduction to Deep Learning Using*” R*. Apress, 2017. doi: 10.1007/978-1-4842-2734-3.

[30] P. Kim, “MATLAB Deep Learning.” Apress, 2017. doi: 10.1007/978-1-4842-2845-6.

[31] M. Arif, W. Farooq, A. B. Saduf, A. Asif, and I. Khan, “Advances in Deep Learning.” doi: https://doi.org/10.1007/978-981-13-6794-6.

[32] J. Chahira, K. J. Mohammed, A. Limane, J. M. Abdelhai, and L. Supervisor, “Topic CNN Based Medical Image Classification,” ACADEMIC MASTER Thesis, 2022.

[33] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Dec. 2015, [Online]. Available: http://arxiv.org/abs/1512.03385

[34] M. Elgendy, “Deep Learning for Vision Systems,” Shelter Island, NY: Manning Publications,2020.

[35] S. Misra *et al.*, “Fault Detection in Induction Motor Using Time Domain and Spectral Imaging-Based Transfer Learning Approach on Vibration Data,” *Sensors*, vol. 22, no. 21, Nov. 2022, doi: 10.3390/s22218210.

[36] MathWorks, “Broken Rotor Fault Detection in AC Induction Motors Using Vibration and Electrical Signals,” https://www.mathworks.com/help/predmaint/ug/broken-rotor-fault-detection-in-ac-induction-motors-using-vibration-and-electrical-signals.html. Accessed: April. 20, 2025. [Online]. Available: https://www.mathworks.com/help/predmaint/ug/broken-rotor-fault-detection-in-ac-induction-motors-using-vibration-and-electrical-signals.html

[37] F. DİŞLİ, M. GEDİKPINAR, and A. SENGUR, “Deep Transfer Learning-Based Broken Rotor Fault Diagnosis For Induction Motors,” Turkish Journal of Science and Technology, vol. 18, no. 1, pp. 275–290, Mar. 2023, doi: 10.55525/tjst.1261887.

Abstract

Induction motors, vital in industrial systems, are prone to electrical faults like broken rotor bars. Early detection is crucial to prevent failures. This study uses an IEEE dataset of motor signals to classify faults under varying loads via time-frequency analysis (STFT spectrograms) and deep learning. Three pre-trained CNNs (ResNet50, AlexNet, and VGG16) were tested, with ResNet50 achieving the highest accuracy (99.73%). Evaluated using accuracy/loss curves, confusion matrices, and precision/recall metrics, ResNet50 proved most robust for early fault detection, ensuring reliability across operational conditions.

Keywords: Induction motor; Broken rotor bars; Deep learning; Convolutional Neural Network; time-frequency analysis; Short-Time Fourier Transform; early fault detection.

Résumé

Les moteurs à induction, essentiels dans les systèmes industriels, sont sujets à des défaillances électriques telles que la rupture des barres du rotor. Une détection précoce est cruciale pour prévenir les pannes. Cette étude utilise un ensemble de données IEEE sur les signaux des moteurs pour classer les défaillances sous différentes charges à l'aide d'une analyse temps-fréquence (spectrogrammes STFT) et de l'apprentissage profond. Trois CNN pré-entraînés (ResNet50, AlexNet et VGG16) ont été testés, ResNet50 obtenant la plus grande précision (99,73 %). Évalué à l'aide de courbes de précision/perte, de matrices de confusion et de mesures de précision/rappel, ResNet50 s'est révélé le plus robuste pour la détection précoce des pannes, garantissant la fiabilité dans toutes les conditions de fonctionnement.

Mots-clés : Moteur à induction ; Barres de rotor cassées ; Apprentissage profond ; Réseau neuronal convolutif ; Analyse temps-fréquence ; Transformée de Fourier à court terme ; Détection précoce des défauts.

الملخص

المحركات الحثية، التي تعتبر حيوية في الأنظمة الصناعية، معرضة لأعطال كهربائية مثل كسر قضبان الدوار. الاكتشاف المبكر أمر بالغ الأهمية لمنع الأعطال. تستخدم هذه الدراسة مجموعة بيانات IEEE لإشارات المحركات لتصنيف الأعطال تحت أحمال متغيرة عبر تحليل التردد الزمني) مخططات (STFT والتعلم العميق. تم اختبار ثلاثة شبكات CNN مدربة مسبقًا (ResNet50 و AlexNet وVGG16) ، وحققت ResNet50 أعلى دقة (99.73٪). تم تقييمها باستخدام منحنيات الدقة/الخسارة ومصفوفات الارتباك ومقاييس الدقة/الاسترجاع، وأثبتت ResNet50 أنها الأكثر قوة في الكشف المبكر عن الأعطال، مما يضمن الموثوقية في جميع ظروف التشغيل.

الكلمات المفتاحية: محرك حثي؛ قضبان دوارة مكسورة؛ التعلم العميق؛ الشبكة العصبية التلافيفية؛ تحليل التردد الزمني؛ تحويل فورييه قصير المدى؛ الكشف المبكر عن الأعطال.