MULTI-CHANNEL, MULTI-LEVEL FRAMEWORK FOR BEARING FAULT DIAGNOSIS IN ELECTRICAL MACHINES

İBRAHİM HALİL ÖZCAN

Ph.D. Thesis

Graduate School
Izmir University of Economics
İzmir
2021

MULTI-CHANNEL, MULTI-LEVEL FRAMEWORK FOR BEARING FAULT DIAGNOSIS IN ELECTRICAL MACHINES

İBRAHİM HALİL ÖZCAN

A Thesis Submitted To
The Graduate School of Izmir University of Economics
Ph.D. Program in Electrical and Electronics Engineering

Izmir
2021

ABSTRACT

MULTI-CHANNEL, MULTI-LEVEL FRAMEWORK FOR BEARING FAULT DIAGNOSIS IN ELECTRICAL MACHINES

Özcan, İbrahim Halil

Ph.D. Program in Electrical and Electronics Engineering

Advisor: Prof. Dr. Türker INCE

Co-Advisor: Prof. Dr. Levent EREN

August, 2021

Electric motors come with the advantage of stability, solidity and ease of use. Besides, they provide users with low operational and maintenance costs. Due to these major features, they have been extensively preferred in a variety of industrial applications. Malfunctioning of induction motors is a major concern due to its impact on industrial production. Induction machines widely use ball or rolling element bearings and the most common motor failures are due to these components. Detecting these potential bearing faults accurately in the earliest stage is critical to solve the problem at a lower cost by repairing and/or replacing relevant parts rather than facing inevitable hazards. In the literature, a variety of fault detection related research is concentrated on detecting binary {healthy, faulty} motor fault cases based on a single input. In this dissertation, a multi-channel, multi-level 1D-Convolutional Neural Network (CNN) framework is designed and utilized to process raw time-domain vibration signals to classify bearing faults at earlier levels with improved performance and hence can be used for predictive maintenance purposes. The proposed system employs ensemble of compact 1D CNNs each specialized for different type of fault to identify different fault types (i.e. {inner race, outer race, rolling element faults}) and achieve two-level (i.e. {early fault, advance fault}) fault detection corresponding to identified fault type, simultaneously. Additionally, sliding window technique is applied as a data preprocessing to achieve real-time implementation. The vibration signals as being the richest source of information for bearing defects are especially chosen for early detection of faults and the benchmark vibration dataset provided by the Center for Intelligent Maintenance Systems (IMS), University of Cincinnati, in collaboration with the National Aeronautics and Space Administration (NASA) is used for the experiments in this thesis to validate the performance of the proposed approach and compare the results with the traditional single-axis data-based fault detection methods.

Keywords: Multi-level bearing fault detection, multi-channel convolutional neural networks, ensemble classifier, sliding window method, predictive maintenance, intelligent fault detection system.

ÖZET

ELEKTRİK MAKİNELERİNDE RULMAN ARIZA TEŞHİSİ İÇİN ÇOK KANALLI, ÇOK SEVİYELİ ÇERÇEVE

Özcan, İbrahim Halil

Elektrik Elektronik Mühendisliği Doktora Programı

Tez Danışmanı: Prof. Dr. Türker İNCE

İkinci Tez Danışmanı: Prof. Dr. Levent EREN

Ağustos, 2021

Elektrik motorları kararlılık, sağlamlık ve kullanım kolaylığı avantajları ile birlikte gelmektedirler. Ayrıca kullanıcılara düşük işletme ve bakım maliyetleri sağlarlar. Bu önemli özelliklerinden dolayı çok çeşitli endüstriyel uygulamalarda yaygın olarak tercih edilmektedirler. Asenkron motorların arızalanması, endüstriyel üretim üzerindeki etkisi nedeniyle büyük bir endişe kaynağıdır. Asenkron makinelerde yaygın olarak bilyalı veya rulmanlı yataklar kullanılmaktadır ve en yaygın motor arızaları bu bileşenlerden kaynaklanmaktadır. Bu olası rulman arızalarının en erken aşamada doğru bir şekilde tespit edilmesi, kaçınılmaz tehlikelerle karşı karşıya kalmak yerine ilgili parçaları onararak ve/veya değiştirerek sorunu daha düşük maliyetle çözmek için kritik öneme sahiptir. Literatürde yer alan arıza tespiti ile ilgili pek çok araştırma, tek bir girişe dayalı ikili {sağlıklı, arızalı} motor arıza durumlarını tespit etmeye odaklanmıştır. Bu doktora tezinde, çok kanallı, çok seviyeli 1B-Evrişimli Sinir Ağ (CNN) yapısı, ham zaman alanlı titreşim sinyallerini işleyerek, rulman arızalarını daha erken seviyelerde geliştirilmiş bir performansla sınıflandırmak için tasarlanmış ve kullanılmıştır, ve dolayısıyla yapı kestirimci bakım amacıyla kullanılabilmektedir. Önerilen sistem, her biri farklı hata türleri için uzmanlaşmış kompakt 1B CNN'ler grubunu eş zamanlı olarak farklı hata türlerini ({iç bilezik arızası, dış bilezik arızası, yuvarlanan eleman arızası}, gibi) tanımlamak için kullanmakta ve tanımlanan arıza tipine ait iki seviyeli ({erken seviye arıza, gelişmiş seviye arıza}, gibi) arıza tespiti başarmaktadır. Ek olarak, gerçek zamanlı uygulamalar gerçekleştirebilmek için veri ön işleme olarak kayan pencere tekniği uygulanmaktadır. Rulman hataları için en zengin bilgi kaynağı olan titreşim sinyalleri, özellikle hataların erken tespiti için seçilmiştir ve Ulusal Havacılık ve Uzay Dairesi (NASA) ile işbirliği içindeki Cincinnati Üniversitesi, Akıllı Bakım Sistemleri Merkezi (IMS) tarafından sağlanan referans titreşim veri seti, önerilen yaklaşımın performansını doğrulamak ve sonuçları geleneksel tek eksenli veri tabanlı hata algılama yöntemleriyle karşılaştırmak için bu tezdeki deneylerde kullanılmaktadır.

Anahtar Kelimeler: Çok seviyeli rulman arıza tespiti, çok kanallı evrişimsel sinir ağları, toplu sınıflayıcı, kayar pencere metodu, kestirimsel bakım, akıllı hata tespit sistemi.

ACKNOWLEDGEMENTS

I would like to start by expressing my deepest gratitude to my advisor Prof. Dr. Türker İnce for encouraging me in perusing this field of research, sharing his profound knowledge, and providing his supervision along this long journey. I would also like to thank my co-advisor Prof. Dr. Levent Eren for his guidance in producing a successful article without overwhelming at a new field I am not very familiar with. They have always been very supportive and helpful to me. Undoubtedly, I would not have been able to complete this thesis without their support. As well as their guidance in the emergence of my doctoral thesis, their contribution to learning the intricacies and technical approaches of publishing an article, and their support in the process of turning my work into an article are invaluable.

I would also like to thank Prof. Dr. Murat Aşkar and Prof. Dr. Cüneyt Güzeliş. I deeply believe that with their valuable comments and constructive reviews, my work gained a much deeper meaning.

Also, I would like to thank my other jury member Prof. Dr. Mehmet Kuntalp for his valuable contributions and for not leaving me alone in this grueling PhD process by taking part in both my doctoral qualification and dissertation defense.

Having the chance to work with such valuable scientists is an indescribable experience for me. During this dissertation, I believe I have learned a lot, however it seems, this is the yet visible part of the iceberg, which drives me to continue diving more to learn. Thank you again and again to each and every one of you for opening a new horizon for me that I could not even imagine.

I am truly indebted a thank to Özer Can Devecioğlu for his cooperation, his positivism and also for the motivation phrases that he constantly used when I was tired and despaired as "you can do it”, “you passed the hard-part, it will be over”, “don’t worry, it's been already done”.

I want to also thank Taner Varanlı for his technical support whenever I need.

I dedicate this doctoral thesis to my “Full-House” family, starting from my parents Emine & Ali ÖZCAN. They made great sacrifices in my entire education duration. Most of the time, they have given up on themselves and brought me to this point. I guess, finally I will be proud of being able to give in return their efforts with the title of Dr. The core but my big family always made me feel safe and they always carried, supported, and encouraged me with their endless love. They are the reason why I can stand so strong.

I want to serve my last special thank (but it seems I need to thank more in the very near future) to my best friend and lovely wife, Yeşim KARKIN ÖZCAN, for all her unrequited love, patience and support she has given me under all circumstances. Please be strong my dear wife and don’t give up, because unfortunately I may make you experience more difficult and patience-requiring times in the future. Thanks in advance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CURRICULUM VITAE

İbrahim Halil ÖZCAN was born on 14.06.1985 in Şanlıurfa/Turkey. He graduated from Milli Piyango Anatolian High School, İZMİR, majored in science and mathematics, with high honours in 2004. He received his undergraduate diploma in 2010, at the Department of Physics, Izmir Institute of Technology in the city of İZMİR. He received his graduate diploma in 2014, at the Department of Electrical and Electronics Engineering, University of Ege in the city of İZMİR. He enrolled in the doctorate program in İzmir University of Economics Department of Electrical and Electronics Engineering in 2014.

During his doctoral study, his research interest deepened within the framework of machine learning, especially in convolutional neural networks. He used his related knowledge in the area of predictive maintenance for intelligent fault detection systems. He published and presented a conference proceeding titled “ Comparison of time-domain and time-scale data in bearing fault detection” in the IEEE Aegean Conference On Electrical Machines And Power Electronics & Optimization Of Electrical And Electronic Equipment Joint Conference (ACEMP-OPTIM), ISBN: 978-1-5386-7687-5, in 2019 and published an journal article titled “Enhanced bearing fault detection using multichannel, multilevel 1D CNN classifier” in the journal of Electrical Engineering, ISBN: 0948-7921, in 2021 with his supervisors.

CHAPTER 1: INTRODUCTION & PROBLEM STATEMENT

1.1 Introduction

In the developing world with the emerging technologies, many industrial needs have arisen. In recent years, a rapid progress has been observed in the industry in parallel with these needs. Electric motors play a critical role in the industrial aspect due to providing stable, solid and efficient solutions, and easy implementation compared to the traditional engines. Besides, they allow users longer operating life comes with low maintenance and low operational cost. These advantages make them preferable in a wide range of industrial applications from simple electro-mechanical implementations to complex aerospace applications. However, malfunctioning of induction motors is a major concern due to its impact on industrial production, workforce, and maintenance costs. In addition to various fault causes such as misalignment, lubricant failure or false and/or true brinelling, poor working conditions such as presence of corrosion and/or contamination, vibration, excessive load or overheating prevent the induction motors to be used in designed lifetime. These factors often result in working efficiency loss, costly maintenance, or the need to replace the entire component/motor. Not to face inevitable hazards and to maximize the longevity of the induction motor and benefit throughout its operating life, these failures must be identified in the incipient phase and the necessary repair and/or replacement of the relevant parts must be performed to save time and money.

1.2 Literature Review

There are generally three different techniques proposed in the literature for the problem of detecting and diagnosing electric motor faults. They can be listed as;

1. *Model-based,*
2. *Signal based,*
3. *Knowledge-based* techniques.

In the *Model-based* approach, firstly a model is developed, then the error is estimated regarding the developed model and finally to diagnose error, the examined outputs are compared with the estimated ones. *Signal based* techniques commonly process the acquired signals and use specific signal analysis techniques such as time-domain (Filippetti, Bellini and Capolino, 2013) and (Zhou, Habetler and Harley, 2008), frequency-domain (Kral, Habetler and Harley, 2004), (Schoen et al., 1995) and (Kliman et al., 1997) and time–frequency (Pons-Llinares et al., 2015), (Li, Wang, and Ismail, 2015), (Eren and Devaney, 2004) and (Yan, Gao and Chen, 2014) domain to monitor the system's health. Qualitative and quantitative methods form the basis of *Knowledge-based* techniques. Qualitative method is a symbolic intelligence-based approach. Under knowledge and predetermined rules, it uses human explicitly interpretable representations instead of numerical analysis. Contrary, quantitative methods utilize statistical-numeric data analysis (Gao, Cecati, and Ding, 2015) to interpret the condition of the motor with the help of unsupervised or supervised machine learning systems. The disadvantage of the knowledge-based method is the need for well-labeled data during training and testing for highly accurate fault detection.

Moreover, as a hybrid model, a statistical technique, principal component analysis (PCA) and dimensionality reduction function, Linear Discriminant Analysis (LDA) is used to reduce dimension and improve the optimized performance of fault detection (Zhao, Mio, and Liu, 2011). In addition, commonly used transform-based techniques such as discrete wavelet transform (DWT) and fast Fourier transform (FFT) can also be applied for signal analysis and fault detection purposes.

Recently, various deep learning (DL) models are introduced for diagnosing and detecting motor faults. DL methods have lots of advantages but may be the most critical contribution is learning of the features with a fully automated approach from raw data directly, as stated in (Zhang et al., 2017), (Ciresan et al., 2010), (Scherer, Muller and Behnke, 2010) and (Krizhevsky, Sutskever and Hinton, 2012). But, for good generalization performance, they require big and well-labeled data and proper training of the DL model. Besides, conventional pattern recognition approaches commonly have separate structures for either feature extraction or classification, which increases the computational complexity.

In the literature, many researchers have spent considerable time on utilizing and developing the aforementioned techniques to solve motor fault detection and classification problems. (Lou and Loparo, 2004) trained an adaptive neural-fuzzy inference system (ANFIS) to diagnose the localized bearing faults and classify into three different classes during the case of load variations. In the study of Matić et al.

(2012), the rotor condition regarding broken bar fault is classified with a support vector machine (SVM) classifier under various working environments with a wide load range. Yu et al. (2017) presents intrinsic mode functions that are computed by using vibrational data to reveal the fundamental statistical characteristics. The blending techniques, the empirical mode decomposition method and support margin local Fisher discriminant analysis (SM-LFDA) are used for features generation, extraction, and dimensionality reduction. Their approach was tested with two different horsepower engines in which the bearing failure occurred. The classification accuracies are obtained as 100% and 97.71%, respectively. In another study, complexity of rolling elements related vibration signals is measured with composite multiscale fuzzy entropy (CMFE). The measured signals are being input to the ensemble support vector machines (ESVMs) to extract hidden nonlinear features under variable speed and load as stated in (Zheng, Pan and Cheng, 2017). In Liu et al. (2017), short-time matching and SVM are fused efficiently for identifying roller element faults by applying raw vibration signals instead of not dealing with traditional sparse representation constraints. The research of Shen et al. (2013) depicts those faulty features can be extracted from distinct frequency-band characteristics. Also, they emphasized that an improved wavelet packet transform (IWPT) is efficient enough for fault detection from raw vibration data. Multiple signature processing structures detect fault comparably more accurately than single one as put forth in (Ayhan, Chow and Song, 2005). Bellini et al. (2001), Wang et al. (2015) and Ye, We and Sadeghian (2003) contributed to the literature with remarkable results by using steady-state and transient current analysis regarding the automatically detected spectral components in the case of broken rods in asynchronous motors.

The studies mentioned so far proclaimed and performed considerably accurate fault diagnosis and detection approaches. However, to analyze and process the applied data, designing manually crafted features takes relatively a long time.

1D Convolutional Neural Network (CNN) has drawn attention with its success in overcoming these adversities and providing many advantages such as its suitability for analysis of naturally 1D signals and instead of drowning in complex deep architectures, its compact architecture configuration with 1D convolutions results decrease in computational complexity, that makes it sufficient for practical real-time fault detection and monitoring problems. Besides, such FFT or DWT conversions are not required, so it becomes very convenient for processing raw data and real-time applications

1.3 Problem Statement

Electric motors widely use rolling (ball or roller) element bearings and mechanical bearing faults are the statistically most encountered fault type. Timely and accurate diagnosis and detection of the bearing faults is crucial for minimizing the economic and time loss in production lines, and this concept, which is also a crucial part of predictive maintenance, has been the main motivation of this dissertation.

The previous experience in the field of utilizing 1D CNN classifiers are presented in the past works such as classifying raw electrocardiogram (ECG) data in (Kiranyaz, Ince and Gabbouj, 2015), raw motor current and vibration data in Ince et al. (2016), (Eren, Ince and Kiranyaz, 2019), Ince (2019) and Eren (2017) and SAR Image data in Ahishali et al. (2019) which led to the motivation for this study.

Based on the extensive literature review, it was noticed that the encountered studies process only single sensor data, applied as input, at a time and their results are compatible with relatively smaller train/test data sets. Moreover, the studies that focus on early detection of faults are very scarce.

To our knowledge this dissertation approach is the first to be presented and, in this scope, a new multi-channel, multi-level framework for bearing fault detection compatible with real-time operation is introduced to the literature. The proposed multichannel 1D (CNN) classifier architecture is designed for simultaneously processing multiple channels of sensor data to achieve detection of multiple bearing fault severity levels (i.e., {early fault, advanced fault}) with enhanced performance.

For the implementation and test of the proposed model, the widely used IMS bearing vibration dataset is chosen to validate the simultaneous processing ability of multiple axis time domain raw vibration data. During the experiments, the real bearing defect data (Figure 1) (Arun, Lincon and Prabhakaran, 2019) are used which are collected in two independent axes [both *x*- and *y*-axis] with the aid of two accelerometers which are mounted on each bearing housing placed in the endurance test set-up. The network parameters (filter coefficients, weights, and biases) are optimized by the well-known Backpropagation (BP) algorithm. The results validate that the proposed architecture is a good candidate for simultaneously detecting different bearing faults at different severity levels compared to the traditional approaches.

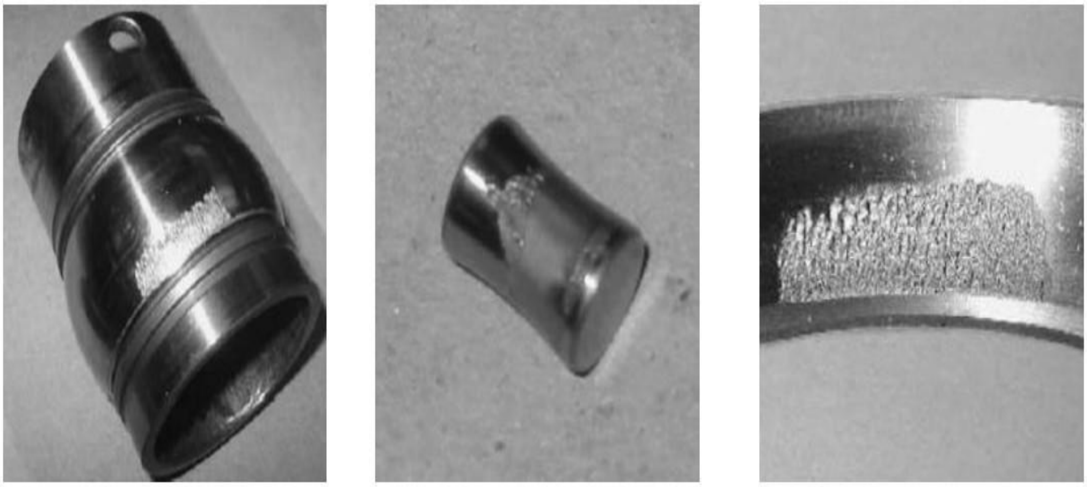


Figure 1. Real bearing faults developed in (a) inner race in third bearing (b) roller element in fourth bearing 4 (c) outer race in first bearing (Source: Arun, Lincon and Prabhakaran, 2019).

In the following section, briefly commercial bearing types are introduced. Also, mostly encountered fault causes and related bearing faults are given. Chapter 3 deeply covers the proposed 1D CNN classifier and detailed derivation of the Back Propagation (BP) on CNNs. Chapter 4 is related to the methodology utilized during the experiments and the proposed classifier implementation using the Keras (Python) library. In chapter 5 firstly, the information of the benchmark IMS bearing dataset used in the training and testing phases of the fault detection algorithm and then performance comparison of the single and multi-channel CNN classifier for real-time motor fault detection is presented. Finally, a comparison table of success in fault detection accuracies of other state of art techniques with our proposed classifier is given. Finally, chapter 6 summarizes the results obtained during this dissertation and mentions the planned future studies to increase the efficiency, success, and usage area of this study.

CHAPTER 2: BEARING FAULTS

In this chapter, firstly, commercially available bearing types and bearing related failure causes are introduced to enhance the understanding of the importance of fault detection. Secondly, fault types developed in induction motor bearings are explained. Finally, the induction motor bearing fault detection concept is discussed to better explain the motivation of this dissertation.

2.1. Bearing Types

A bearing is a mechanical element commonly used within machineries that enables linear or rotational movement of shaft, regulates the motion, and reduces the friction of moving parts and increases efficiency and speed. It constrains the relative movement to the desired movement and reduces the load placed on the part and the machine. There are different commercial bearing types according to the manufacturing structure as shown in Figure 2, and they can be classified as;

1. Plain bearings,
2. Magnetic bearings,
3. Fluid film bearings,
4. Rolling element bearings.

Plain bearings, also known as sliding bearings, are the most basic bearing structure designed without rolling elements and provide the relative rotational movement between the two machine elements with the sliding movement. While plain bearings offer high load-carrying capacity, low vibration, and low noise with high accuracy, due to the high sliding friction, they require more power consumption. Unease of installation limits interchangeability when disassembly and repair is required.

Magnetic bearings differ from traditional bearing types in using electromagnetic force to levitate rotating parts. There are commercially available magnetic bearings that are totally contact-free or partially in contact with traditional mechanical bearings. This feature allows a nearly frictionless environment that prevents mechanical wear, lowers vibration, reduces energy loss, and eliminates the need of lubrication. In the absence of abrasion, it has a theoretically unlimited lifetime.

The most distinguishing advantage is that; it can provide the desired highest speed due to not having maximum relative speed. However, magnetic bearings are costly compared to the traditional ones with heavy weight and relatively large volume.

In Fluid film bearings due to fluid pressure, the only contact between the stationary and rotating surfaces occurs when the system starts and stops. The surfaces are separated, and the load is supported by a pressurized liquid or gas lubricant thin film such as oil, water or air. Contactless structure eliminates sliding friction, resulting in less wear, less vibration and more expected runtime under proper conditions. However, power consumption is relatively high compared to ball bearings and they are more sensitive to temperature. For specific applications, fluid film bearings cannot operate under very low temperature unlike non-lubricated ball bearings, and they may become completely immobile under unexpected loss of supply pressure and show sudden catastrophic reactions.

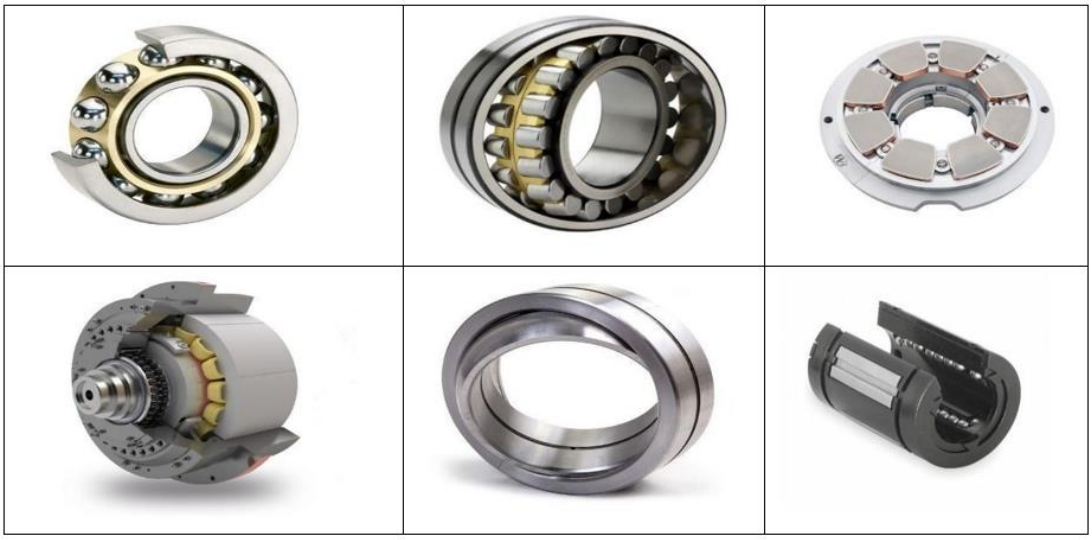


Figure 2. Types of bearing a) ball bearing, b) roller bearing, c) fluid bearing, d) magnetic bearing, e) plain bearing, f) linear bearing. (Source: Engineering Learn, 2021)

Rolling element bearings, also mostly known as rolling bearings, broadly take part in industrial mechanical applications in a range from basic electric motors to complex engines. Mainly, they are formed of four parts as: inner race, outer race, cage and rolling elements (Sidahmed and Dalpiaz, 2001). The fundamental purpose is supporting the shaft and transmitting axial and/or radial load with cylindrical rolling elements or balls, while reducing the rotational friction. For this purpose, rolling elements are placed between inner and outer races, so sliding motion is converted to rolling motion resulting in less friction. Rolling element bearings have two main types as: ball bearings and roller bearings. In ball bearings, which vary in bearing ring configuration as deep groove and angular contact types, spherical bearings are surrounded with a separator, and they are mounted between the inner and outer races, while in roller bearings, cylindrical, spherical, needle, barrel or tapered geometric shaped rollers are placed inside the cage as depicted in Figure 3.

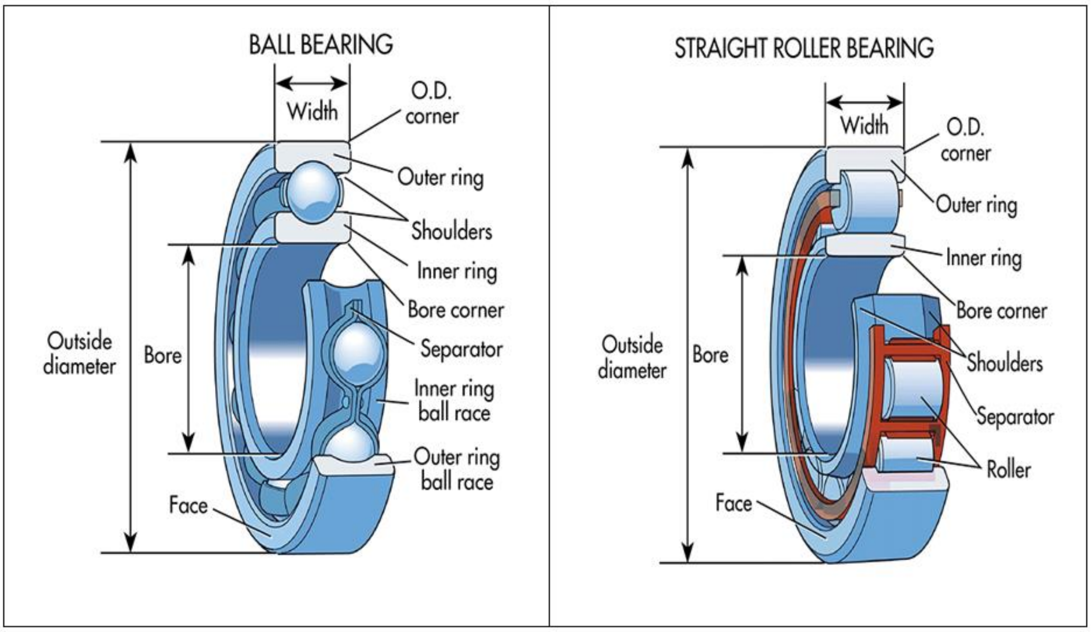


Figure 3. Structure of rolling-element bearing. a) ball bearing, b) roller bearing (Source: Machine Design, 2015).

Roller bearings can be designed in a wide range of shapes and sizes. This feature allows application specific customization. Figure 4 illustrates the common applications and related roller types that can be used.

Besides, diversifying the structure with flanges, different cage and roller types can further reduce friction, and using multiple bearing rows in versatile bearings can improve radial load capacity resulting in significant increase in performance to serve specific application requirements.

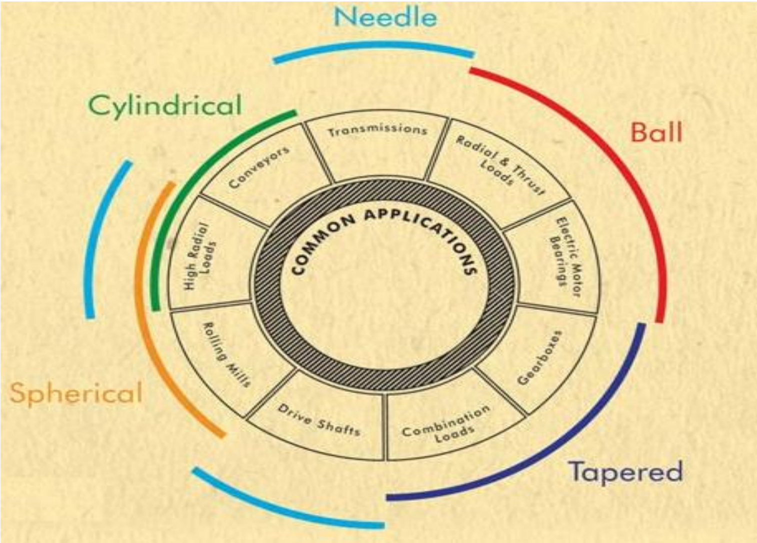


Figure 4. Application specific usage of rolling element bearings (Source: SKF Group, 2018).

Bearings are also classified according to applied force direction such as radial, thrust (axial), radial deep groove bearings (radial and axial). While axial or thrust bearings operate between discs, radial bearings operate between races.

All the mentioned rolling bearings have advantages and disadvantages over each other, and these features determine in which application they can be used more beneficially. However, in general they are superior to the second widely used bearing type, plain bearings, in the following features. Rolling bearings are smaller in size, durable, extensively standardized, mostly self-aligned, which lowers misalignment failures. Besides, they have ease of maintenance and interchangeability, less starting friction coefficient, they can operate more precisely with their narrower clearance gap, easily lubricate and moreover, they require less lubricant. Rolling element bearings can handle higher axial and/or radial loads and have predictable runtime under proper operating circumstances. But they are sensitive to shock loads, generally, not cost effective compared to plain bearings, they operate noisily, and pitting occurs during standstill.

Roller bearings can carry higher loads compared to the ball bearings due to the larger contact area, but they are not so feasible in high-speed applications. Roller bearings require less maintenance cost. However, they are more vulnerable to angular misalignment than ball bearings. Ball bearings are also superior in handling thrust and/or radial loads. Having a smaller contact area lowers the friction and they are the cheapest ones within the whole roller bearing types.

2.2. Fault Causes

In literature, twelve primary bearing failure causes are known, and they can be listed as; excessive load, overheating, false brinelling, true brinelling, normal fatigue failure, reverse loading, contamination, lubricant failure, corrosion, misalignment, loose fits and tight fits (Barden Precision Bearings, 2017). In some circumstances, identifying the main fault cause may be complicated since the fault can trigger another one. Corrosion, as an abrasive, causing rusting in a ball race, may cause wear, which can trigger preload loss or result in increased radial clearance. Also, wear debris may preclude greasiness, resulting in lubricant failure, and followed by overheating.

Apart from the failure causes given above, there are too many causes encountered and each fault may cause other known fault types. All causes will not be directly pointed out here since this thesis scope is beyond that and bearing fault detection from raw vibration data is the main concern. In the continuation of this part, the most common failure causes will be introduced (Harris, 2000) and (Xi, 2000).

Among the fault causes listed above, lubrication is one of the most well-known and leading fault modes (Figure 5) (SKF Group, 2017), which is crucial for benefiting bearings during their designed lifetime. The main purpose of lubricating is reducing wear and minimizing the friction coefficient by separating the contacting surfaces. Besides, it prevents corrosion, removes the friction generated heat. Not only the applied amount but also the applied improper lubricant amount with the improper method causes bearing faults. Less lubricant results in contamination, which leads to excessive wear, which causes overheat, and finally premature bearing failure evolves. Too much lubricant may result in seal failure and can also increase friction and hence heat in high-speed operations. That’s why lubricating at the proper frequency with the proper amount is crucial for bearing health.

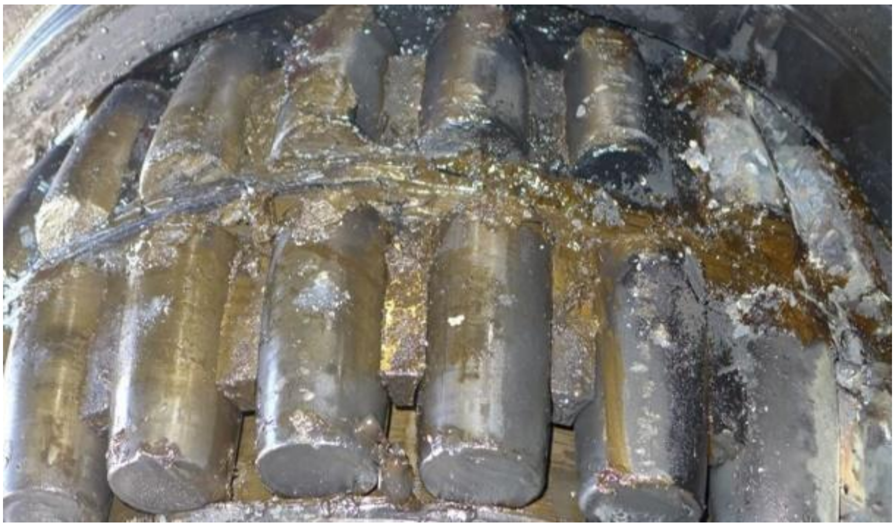
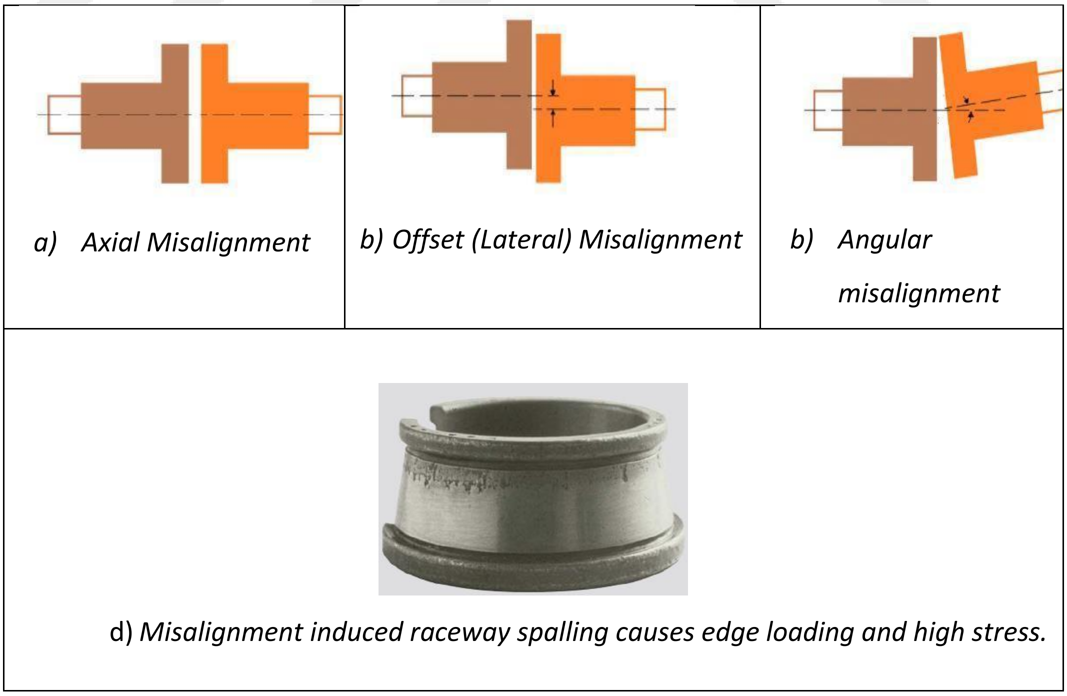


Figure 5. Typical real lubrication failure (Source: Micro-Lube, 2021).

Misalignment is another major fault mode, occurring due to the two housings needed to be coupled and are not properly aligned in the same center. Bent shafts, burrs, or dirt on shaft and/or housing shoulders, shaft threads that are not square with shaft seats and locking nuts with faces that are not square to the thread axis are the most prevalent causes of misalignment (Dannana, 2019).



a) Axial Misalignment

b) Offset (Lateral) Misalignment

b) Angular misalignment

d) *Misalignment induced raceway spalling causes edge loading and high stress.*

Figure 6. Three types of misalignments and real defects caused by misalignment. (Source: Dannana, 2019; SKF Group, 2017).

There are three types of misalignments (Figure 6) such as axial, offset (radial or parallel) and angular misalignment. Axial misalignment occurs when the axial shaft distance of the driving and driven elements varies. If the two shafts are not centered in the same vertical or horizontal line, offset misalignment emerges. Angular misalignment is the condition that the two support sides are not intersecting due to the angular difference relative to each other. Misalignment induces the excess load on bearings and/or seals which results in increased friction, hence heavy wear and temperature rise originates. Besides, more power consumption, high vibration and noise can be identified.

Corrosion is the wear of metal or metal alloys by oxidation or chemical effects. Under improper storage conditions, it takes place on the contact surfaces, resulting from the presence of atmospheric moisture sources such as water or acid and other corrosive foreign substances. When the lubricant film is diluted by these agents, viscosity is reduced, rust is formed and bearing surfaces are progressively corroded. Oil film deterioration causes erosion in which abrasive particles are created. These abrasive particles result in pitting on the roller surfaces. These all affect operating precision, early fatigue failure is triggered and bearing durability is decreased. Corrosion-origin real contact and inner ring fault picture is given in Figure 7.

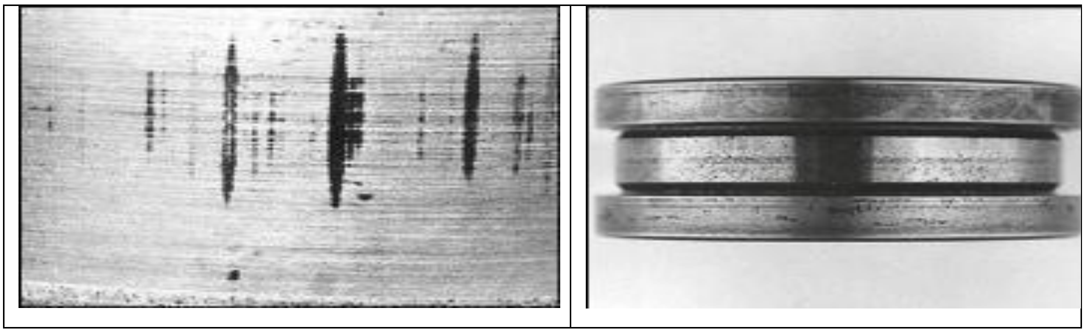


Figure 7. a) Raceway contact corrosion trace, b) Inner ring corrosion (Source: SKF Group, 2017).

Like corrosion, contamination is formed in the presence of many contaminants in the bearing lubricants or cleaning solutions such as dirt, dust, sand and/or abrasive grit. Moreover, dirt smeared with hands and/or tools and steel chips from the working area can get into the bearing through the ineffective seals. The rolling elements may cause over-roll of these contaminants, and lead to indentations creation in the inner and/or outer rings (Figure 8). Rigid contaminants may result in sharper-edged indentations. The dented surface, which also causes vibration, is under cyclic stress because of the over-roll. After a while, metal detaching from the raceway due to the surface fatigue, also called as spalling, begins and it continues until the bearing becomes unusable.

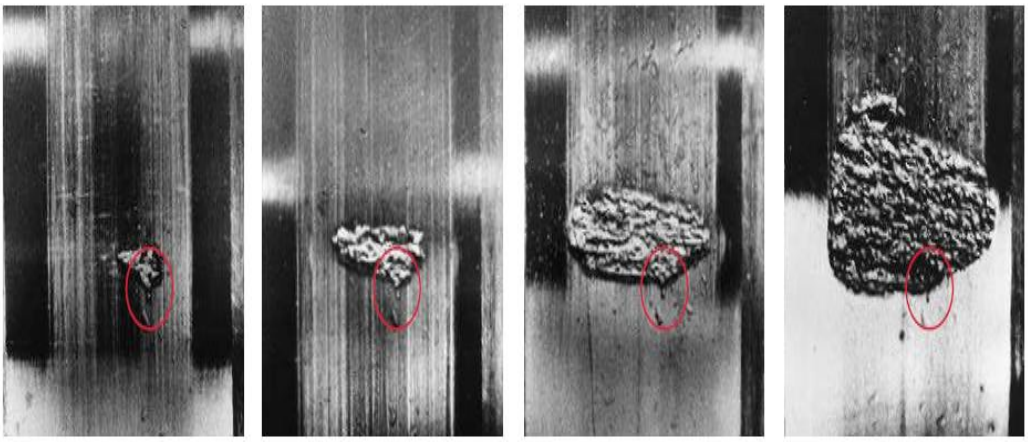


Figure 8. The progress of indentation due to the over-rolled rigid contaminant in the raceway of the roller and the prominence of indentation triggered spalling after a certain time (Source: SKF Group, 2017).

2.3. Bearing Fault Types and Induction Motor Bearing Fault Detection

Due to electric motors providing users with reliable, simple, and durable high precise and accurate solutions, besides low maintenance and operation cost, they are preferred in a variety of industrial production applications.

As mentioned in the previous section, induction motors can mostly be encountered with different types of faults that are critical for the operation, such as bearing faults, rotor bar breakage, shaft deflection, etc... (Aller, Lu and Habetler, 2008). Rotor electric faults, rotor mechanical faults, stator faults and power related faults can be listed in the classification of motor faults.

In some circumstances these failures evolve in time, but sometimes breakdown occurs suddenly without any warning, which results in high economic loss due to the interruption of the production line. Not to face inevitable hazards and maximize induction motor’s runtime, these failures should be identified in incipient phase to benefit from the machine along designed life by making necessary repair and/or replacement when it is required.

Electric Power Research Institute (EPRI) states that electric motors commonly employ rolling (ball or roller) element bearings, and these components trigger motor failures. Mechanical bearing faults are the most encountered fault type and have a 41% statistical occurrence possibility. Second highest occurrence rate is stated as electrical stator faults with 36%. The rest of the failure types emerged with 14%. Also, the other commonly referenced IEEE standard 493-1997 put forth the same fault type's statistical occurrences as 44%, 26%, 8% and 22%, respectively (IEEE Recommended Practice for the Design of Reliable Industrial and Commercial Power Systems, 1998) and (Allbrecht, Appiarius and McCoy, 1986).

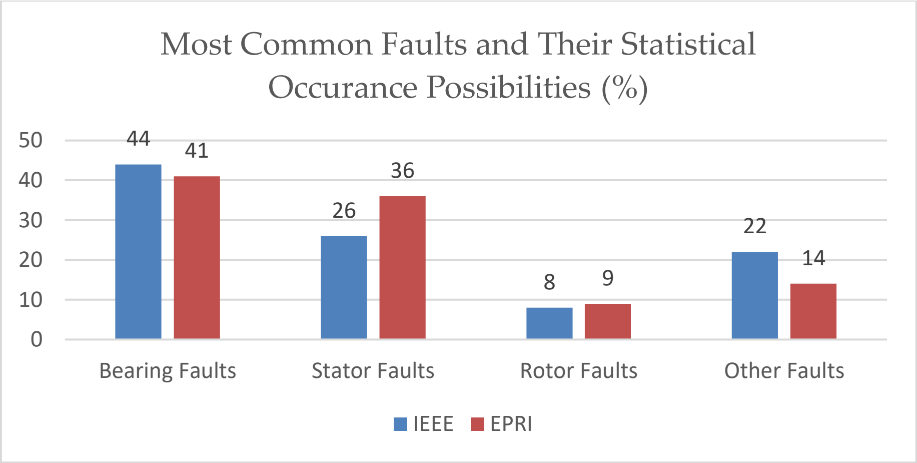


Figure 9. Study of EPRI and IEEE on induction machine failure occurrence possibilities.

Bearing faults generally begin to emerge in the narrow surface of the raceway or rolling element. While a rolling element passes over a defect, it causes a vibration impulse directly related to the defect size and load distribution. The resonance modes of the bearing and supporting structure are excited by this impulse as shown in Figure 10.

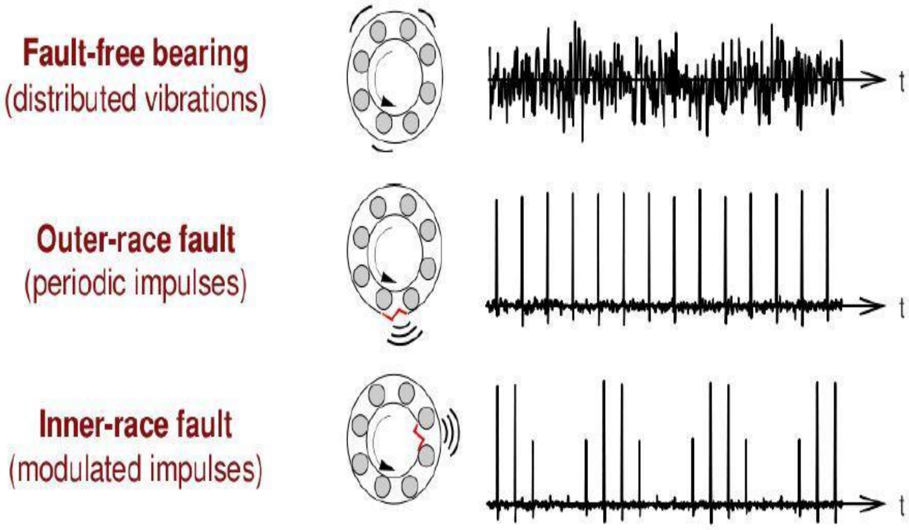


Figure 10. Vibration patterns of the ball bearing defects (Source: Ovacikli, 2011).

The vibration impulse is generally detected with a piezo-electric vibration accelerometer mounted to the bearing housing, and the digitized output voltage is created with an A/D converter. The increase in acceleration causes an increase in the measured voltage signal. Figure 11 depicts the phenomenon of an outer race fault occurring in the bearing and a condition monitoring system monitors the vibration signal generated by passing the roller over the fault. The signal is detected by an accelerometer and processed with a computer to give information regarding the health of the system (Klausen, 2019).

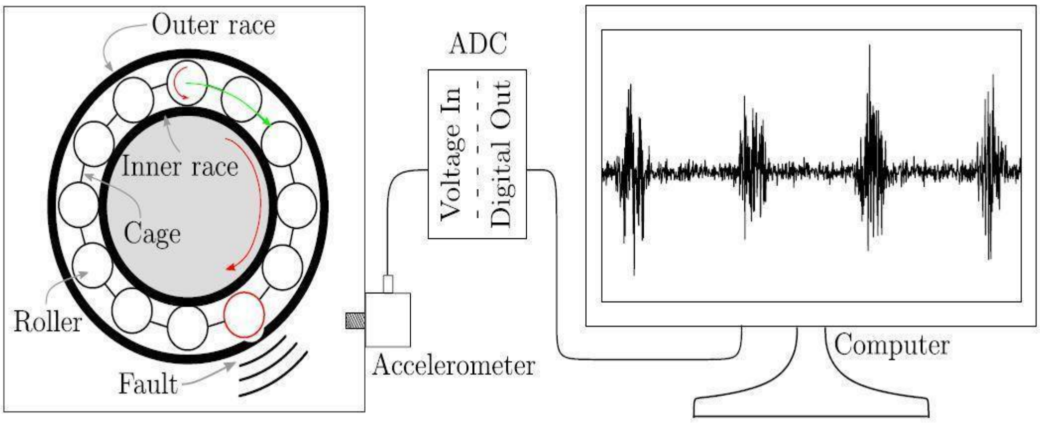


Figure 11. Regular ball bearing geometry. Ball bearings are a set of balls rotating inside a raceway (inner and outer ring) (Source: Klausen, 2019).

While the bearing fault is emerging, vibration is generated at the fault associated frequencies and different fault types can be identified with the help of these specific frequencies (Wowk, 1991).

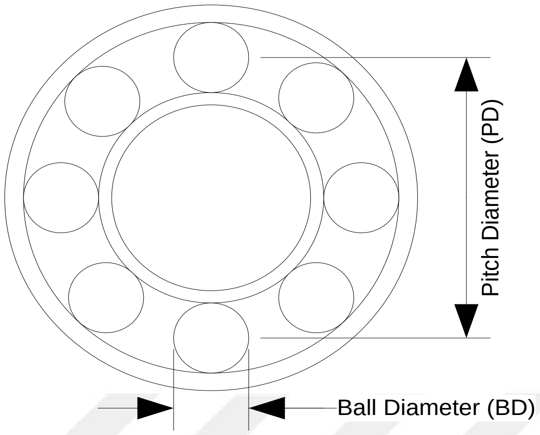


Figure 12. Ball bearing geometry (Source: Ince et al., 2016).

while the motor is operating, the cage, hence the inner and outer race, turns at a linear velocity. The fundamental cage defect, f_{CD} can be computed by using the linear velocity of the cage.

$$f_{CD} = \frac{V_c}{r_c} = \frac{V_i + V_o}{D_c} \qquad (1)$$

v_c, r_c, v_i, v_o, and D_c velocity of the cage, radius of the cage, velocity of the inner race, velocity of the outer race, and diameter of the cage respectively. Then,

$$f_{CD} = \frac{f_i r_i + f_o r_o}{PD} = \frac{1}{PD}\left(f_i \frac{PD - BDcoscos\,\theta}{2} + f_o \frac{PD + BDcoscos\,\theta}{2}\right) \qquad (2)$$

where f_i, f_o, r_i, r_o, PD, BD, and θ are frequency of the inner race, frequency of the outer race, radius of the inner race, radius of the outer race, pitch diameter, ball diameter and contact angle of the ball respectively. The housing is the stationary part of the motors, where the outer race is installed. The inner race and the shaft are mounted on each other; thus, they have the same angular speed. Based on this condition, the following assumption can be made,

$$f_o = 0 \; and \; f_i = f_{rm} \qquad (3)$$

where f_{rm} is the mechanical rotor speed in Hertz. Now, Eq. 2 can be written as

$$f_{CD} = \frac{1}{2} f_{rm}\left(1 - \frac{BD}{PD}\; cos\; cos\; \theta\right) \qquad (4)$$

The traverse rate of bearing balls over a defect point on the inner race determines the inner race defect frequency, f_{ID}. Each ball passes the defect point at the frequency difference of the cage and inner race. The frequency is directly proportional to the number of balls, *n*. The fundamental frequency of the inner race defect, then, can be computed as

$$f_{ID} = n|f_{CD} - f_i| \tag{5}$$

Equation 5 can be expanded as,

$$\begin{aligned} f_{ID} &= n\left|\frac{f_i r_i + f_o r_o}{PD} - f_i\right| \\ &= n\left|\frac{f_i\left(r_c - \frac{BDcoscos\,\theta}{2}\right) + f_o\left(r_c + \frac{BDcoscos\,\theta}{2}\right)}{PD} - f_i\right| \\ &= \frac{n}{2}\left|(f_i - f_o)\left(1 + \frac{BDcoscos\,\theta}{2}\right)\right| \end{aligned} \tag{6}$$

Substituting $f_0 = 0\ and\ f_i = f_{rm}$, f_{ID} becomes

$$f_{ID} = \frac{n}{2} f_{rm}\left(1 + \frac{BD}{PD}\ cos\,cos\,\theta\ \right) \tag{7}$$

The fundamental ball defect frequency, f_{BD} results from the rotation of the ball about its own axis through its center. The ball defect frequency can be formulated as

$$\begin{aligned} f_{BD} &= \left|(f_i - f_{CD})\frac{r_i}{r_b}\right| = \left|(f_o - f_{CD})\frac{r_o}{r_b}\right| \\ &= \frac{PD}{2BD}\left|(f_i - f_o)\left(1 - \frac{BD^2\theta}{PD^2}\right)\right| \end{aligned} \tag{8}$$

Substituting $f_0 = 0\ and\ f_i = f_{rm}$, f_{BD} becomes

$$f_{BD} = \frac{PD}{2BD} f_{rm}\left(1 - \frac{BD^2\theta}{PD^2}\right) \tag{9}$$

CHAPTER 3: NEURAL NETWORKS

3.1. Artificial Neural Networks

The concept of neural networks also called as Artificial Neural Networks (ANN) drew attention of many researchers over the past decades because of the ease of construction, availability of multiple training algorithms and requiring less formal statistical training. ANNs can learn highly nonlinear relationships and possible interactions between variables by optimizing its parameters through training (Tu, 1996). Contrary to the popular belief, neural networks are not exactly biological. They just resemble the biological architecture of neurons. The brain basically learns from the experiences and ANNs offer an alternative approach in terms of understanding and reflecting the human brain by using this approach. Throughout this idea, researchers try to simulate human-like performance in many fields such as financial, medical, industrial, data mining, sales & marketing, and science.

Neural network models can be considered in two main groups: hardware based in which neurons are represented with physical components and software based in which computer models are used. While working on either hardware or software based neural networks, a variety of topologies and learning algorithms are used. These algorithm series allow computer programs to perceive underlying relationships in data patterns through the human-like process to deal with generic problems related with machine learning and/or deep learning. Likewise the human brain neuron and belonging network functions, ANN mimics these topologies and different types of neural networks learn these algorithms with different approaches.

A neural network is formed of structured layer sets, with a set of nodes (neurons). Input layer is the first layer of the network, where the data are introduced, followed by hidden (mid-layers) layers. The layer number in hidden layers depends on the data and the application to implement. The final layer is called as the “output layer”.

There are two main topologies: Feedforward, and Feedback (or recurrent) (Figure 13). Feedforward networks are the one in which the informational flow is unidirectional-forward. The data first come to the input node then pass through the hidden layer, if there is any, finally pass to the output node. They contain no feedback loop or cycle and have fixed inputs and outputs. Conversely, feedback networks also known as recurrent neural networks (RNNs) are bi-directional which consist of a loop, meaning data can flow through the latter nodes to previous ones. Under these topologies, lots of network types are constructed according to the need of usage.

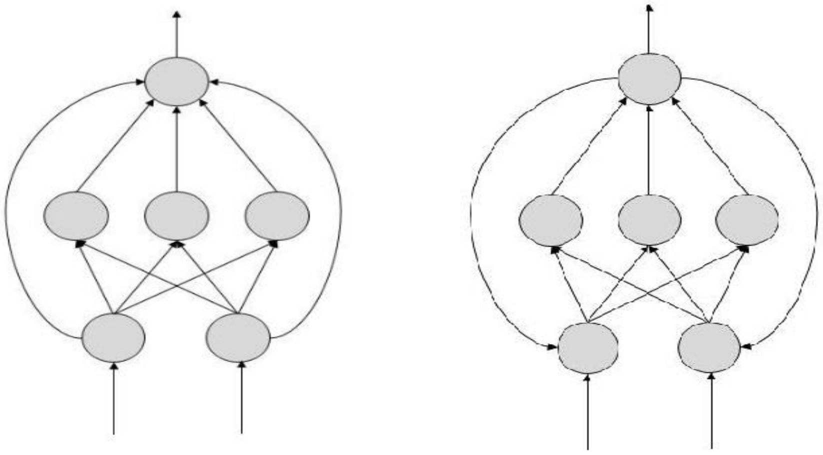


Figure 13. a) Feedforward topology, b) Feedback topology (Source: Tutorialspoint, 2021).

3.2. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs or ConvNets) are considered as one of the special and most developed versions of feedforward neural networks and inspired from biological nature. CNNs have very similar architecture with the animal visual cortex in terms of the connectivity pattern of neurons. They are hierarchical neural networks whose convolution layers alternate with subsampling layers and followed by dense (fully connected) layers (Wiesel, and Hubel, 1959) which are designed to operate over raw input data with minimal amount of preprocessing. The key difference from the traditional ANNs is, while conventional systems need both feature extraction and training classification, CNNs can use raw data with any sampling rate, meaning, there is no need for hand-crafted feature extraction or post-processing.

The convolutional layers of CNNs can learn to extract optimal features from data with proper training and the dense layers generate the output class vectors of each sample for classification problems. The backpropagation training algorithm with different learning methods is generally used to determine the best set of network parameters for a given problem. Researchers presented that even with the limited training data, CNNs can show superior performance with respect to the traditional counterparts (Hu et al., 1997), (Kiranyaz et al., 2013) and (Chazal et al., 2004).

Especially 2D CNNs have been used in a variety of applications and have been accepted as a standard approach in the computer vision area in recent years. They are commonly applied for deep learning tasks, such as image, video and pattern recognition, signal processing, control systems detection, segmentation, recommender systems and natural language processing, etc... In general, with the use of ANNs, many problems are solved in the aforementioned fields. Software simulations are the key factor and also performance is generally tried to be increased by more powerful computers, investigating ANN models capabilities or via new more efficient simulations.

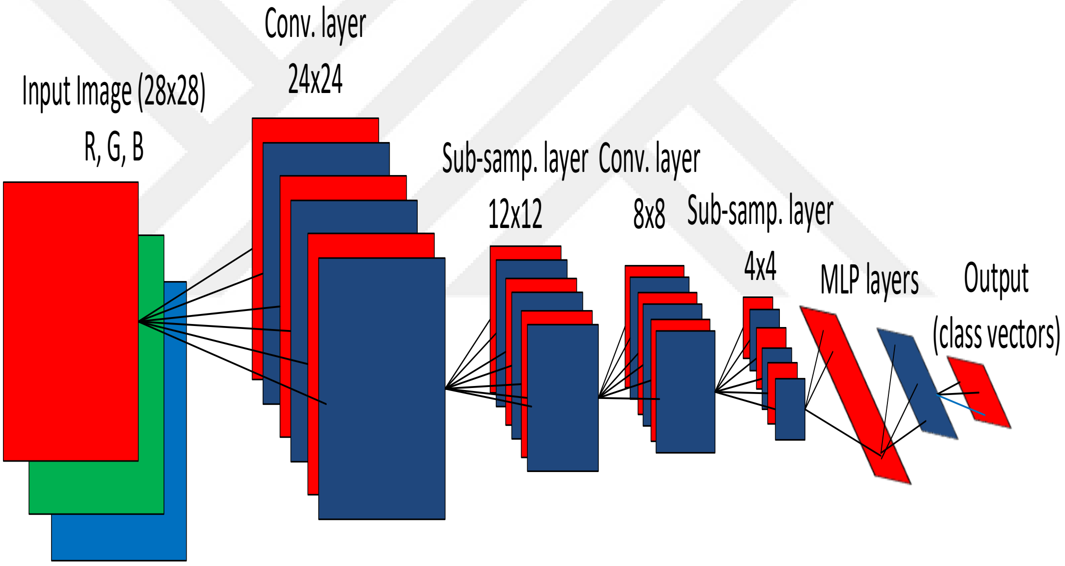


Figure 14. Sample 2D CNN configuration (Source: Ince et al., 2016).

A typical 2D CNN architecture with a 28x28 pixel (for this example) R, G, B image applied to the input layer with three neurons is depicted in Figure 14. A convolution layer is placed just after the input layer (the passive layer) and each convolution layer is followed by a subsampling layer in an alternating manner (Ince et al., 2016). While 2D feature maps, in this case (R, G, B) color channels, are propagating through convolutional and subsampling layers, they undergo filtering and decimating operations. Except for setting the two major CNN parameters beforehand, the kernel size and the sub-sampling factor (as 5 and 2 for the given illustration, Figure 14, respectively), 2D filter kernels do not require hand-crafted feature extraction process and/or parameter fixing, due to the backpropagation (BP) training optimization in the CNNs. And also, along the forward propagation, an adequate number of sub-sampling layers decimate the feature maps to a scalar (1D) at the output CNN layer, which is just before the fully connected layers that are identical to regular MLPs. It can be accomplished by setting the number of convolutional, sub-sampling and MLP layers to the input image dimensions or setting input image dimensions to the CNN configuration. In order to overcome this drawback, adaptive CNN implementation with certain adjustments regarding back-propagation training is presented in the following sections.

3.2.1. Adaptive 2D CNN Implementation

The main idea in designing the adaptive 2D CNN structure is making the CNN analogy easier to understand, and by simplifying CNNs, getting rid of the dependence on CNN parameters related with the image dimension applied to the input layer. The proposed topology has the advantage of working independent of the input layer dimension. The newly designed structure (Kiranyaz, Ince, and Gabbouj, 2015), as given in Figure 15, only consists of CNN layer rather than MLP layers and the extended (fused) hidden layer, so called the "CNN layer", neurons can process convolution and down-sampling operations. So, the adaptive CNN implementation is formed of an input layer, hidden CNN layers and the remaining dense layers and an output layer.

The interior structure of k^{th} neuron at layer l is illustrated in Figure 15. w_{ik}^{l-1} is the individual weight providing the connection between the k^{th} neuron at layer l *and* i^{th} neuron at layer *l-1,* and b_k^l is the bias of k^{th} neuron at layer l. The final output, s_k^l , is the subsampled form of the intermediate output $y_k^l = f(x_k^l)$. After convolving with their own kernels (weights matrix), the final output maps of the layer l are accumulated to form the input map of the layer *l+1,* as given in the Eq. 10.

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} conv2D(w_{ik}^{l-1}, s_i^{l-1}, 'NoZeroPad') \tag{10}$$

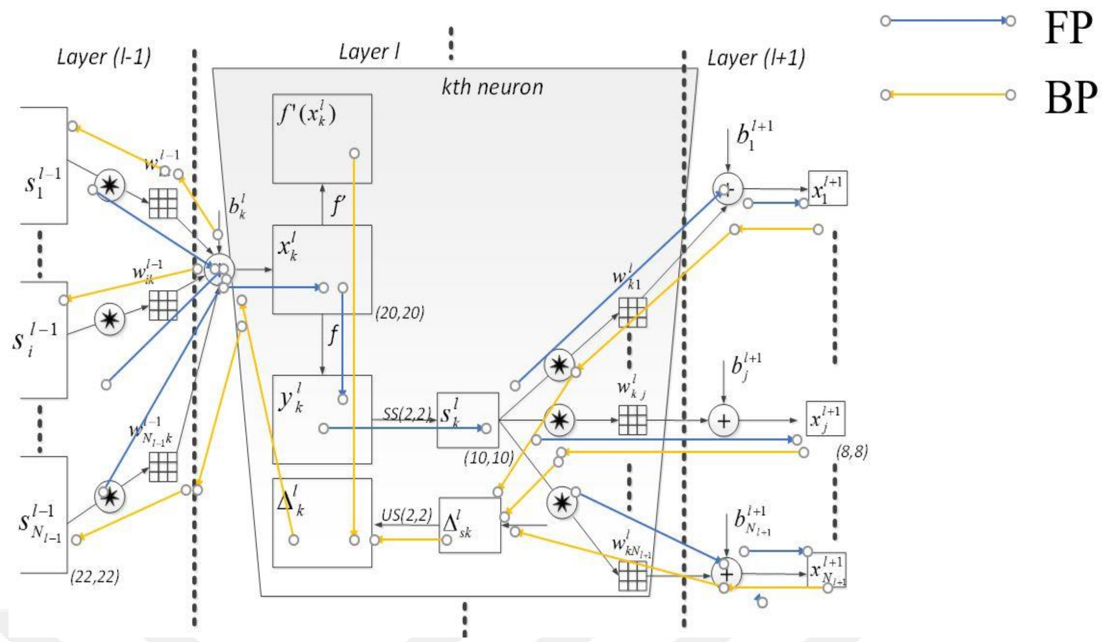


Figure 15. Adaptive 2D CNN Implementation (Source: Kiranyaz, Ince and Gabbouj, 2015).

There are notable differences between the adaptive 2D CNNs and regular MLPs due to the given features:

a) While MLPs use scalar multiplication, 2D convolutions with kernel filters are applied to 2D signals.

b) Kernel size limits and determines the pixels connectivity of the current layer input map's (synonyms of the scalar neurons of MLPs) to the pixels of previous layer neuron outputs,

c) All connections share the same weights, which reduces connection number to only the size of w_{ki}^{l}, instead of a fully connected MLP implementation with a significantly high number of 1-to-1 connections.

In adaptive CNN implementation, the output CNN layer's (the layer just before the first MLP layer) input map dimension is used to set the sub-sampling factor of the same layer, and that gives freedom in setting the number of hidden CNN layers to any number. According to the illustration in Figure 14, due to the input map dimension being given as 8x8, automatically, sub-sampling factors of the output CNN layer, (*(l+1)*th layer), is set to ssx = ssy = 8. These operations result in dimension decrease of input maps. Also, applying convolution without zero padding results in a gradual reduction in size of the input maps by (*Kx*-*1*, *Ky*-*1*), given the width, *Kx* and height, *Ky* of the kernel.

3.2.2. Back-Propagation (BP) Training of CNNs

The Back-Propagation (BP) is used for computing delta error by minimizing the contributions of network parameters at the output layer, hence, it initiates from the error at the output MLP layer and propagates through the inner layers. If input layer is defined as *l*=1, output layers as *l=L* and number of classes as N_L, the mean-squared error (MSE) in the output layer for the input vector *p*, E_p, can be expressed as follows:

$$E_p = MSE\left(t_i^p, [y_1^L, \dots, y_{N_L}^L]\right) = \sum_{i=1}^{N_L}\left(y_i^L - t_i^p\right)^2 \tag{11}$$

where, t_i^p is the input vector's target and $[y_1^L, \dots, y_{N_L}^L]$is the output vector.

3.2.2.1 Inter-BP among CNN layers: $\Delta s_k^l \xleftarrow{\sum} \Delta_\iota^{l+1}$

According to back propagation rule, when the output of the *k*th neuron at *l*th layer has contribution on *i*th neuron at *(l+1)*th layer with an individual weight (w_{ki}^l), due to contributing with same weights, the neuron's sensitivities of the error at *(l+1)*thlayer, also called as delta error *(*Δ_ι^{l+1}*)*, can be used in back-propagation to get the delta errors to the output sensitivity, Δ_k^l**,** of the neuron at the *l*th layer (Kiranyaz, Ince, and Gabbouj, 2015). This can be defined as in Eq. 12.

$$\frac{\partial E}{\partial s_k^l} = \Delta s_k^l \xleftarrow{\sum} \Delta_\iota^{l+1}, \forall i \in \{1, N_{l+1}\} \tag{12}$$

can be expressed explicitly as;

$$\frac{\partial E}{\partial s_k^l} = \Delta s_k^l = \sum_{i=1}^{N_{l+1}} \frac{\partial E}{\partial x_i^{l+1}} \frac{\partial x_i^{l+1}}{\partial s_k^l} = \sum_{i=1}^{N_{l+1}} \Delta_l^{l+1} \frac{\partial x_i^{l+1}}{\partial s_k^l} = \sum_{i=1}^{N_{l+1}} \Delta_l^{l+1}\, w_{ki}^l \tag{13}$$

for,

$$x_i^{l+1} = \ldots + s_k^l * w_{ki}^l + \ldots \tag{14}$$

Direct computation of derivatives from the convolution is not so easy. So, for better understanding, first we look at the contribution to the output from a single pixel, $s_k^l(m,n)$, over the (*l+1)*th layer *i*th input neuron's pixels, $x_i^{l+1}(m,n)$, with a 3x3 kernel:

$$\begin{aligned}
&x_i^{l+1}(m-1,n-1) = \dots + s_k^l(m,n).w_{ki}^l(2,2) + \dots \\
&x_i^{l+1}(m-1,n) = \dots + s_k^l(m,n).w_{ki}^l(2,1) + \dots \\
&\dots\dots \\
&x_i^{l+1}(m+1,n+1) = \dots + s_k^l(m,n).w_{ki}^l(0,0) + \dots
\end{aligned} \tag{15}$$

When the pixel in CNN structure is considered as synonyms of the scalar neurons of MLPs, delta of $s_k^l(m,n)$ can be written w.r.t to the BP formation as in Eq.16;

$$\frac{\partial E}{\partial s_k^l}(m,n) = \Delta s_k^l(m,n) == \sum_{i=1}^{N_{l+1}} \left(\sum_{r=-1}^{1} \sum_{t=-1}^{1} \Delta_\iota^{l+1}(m+r,n+t).w_{ki}^l(1-r,1-t) \right) \tag{16}$$

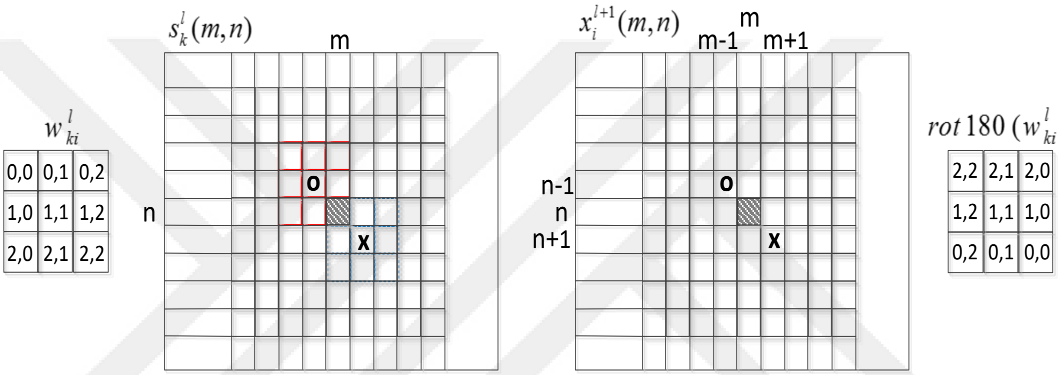


Figure 16. Contribution to the output from a single pixel, $s_k^l(m,n)$, on the next layer's input neuron's pixels, $x_i^{l+1}(m-1,n-1)$ and $x_i^{l+1}(m+1,n+1)$ (Source: Kiranyaz, Ince, and Gabbouj, 2015).

Generalization of Eq. 16 for all pixels of the Δs_k^l:

$$\Delta s_k^l = \sum_{i=1}^{N_{l+1}} conv2D\left(\Delta_\iota^{l+1}, rot180(w_{ki}^l), 'ZeroPad'\right) \tag{17}$$

$rot180\ (w_{ki}^l)$ is used for rotating the kernel to convolve, and $ZeroPad'$ refers *full* convolution with zero padding by (*Kx*-1, *Ky*-1) of the Δ_ι^{l+1} to provide an equal width and height size for Δs_k^l and Δ_ι^{l+1} with the s_k^l.

If the BP successfully operates from *(l+1)*th layer to the *l*th layer, then it can be carried on back-propagating up to the input delta. If $us_k^l = up(_{ssx,ssy}s_k^l)$ is zero order up-sampled map, delta error can be expressed as:

$$\Delta_k^l = \frac{\partial E}{\partial x_k^l} = \frac{\partial E}{\partial y_k^l}\frac{\partial y_k^l}{\partial x_k^l} = \frac{\partial E}{\partial us_k^l}\frac{\partial us_k^l}{\partial y_k^l} f'(x_k^l) = up(\Delta s_k^l)\beta f'(x_k^l) \tag{18}$$

β is used for $(ssx.ssy)^{-1}$, whereas pooling operation, averaging is applied to *ssx.ssy* number of pixels of y_k^l are to obtain s_k^l . If instead of averaging, max-pooling is used, then the given Eq. 18 should be reinterpreted according to the max-pooling approach.

While back-propagating from the first dense layer to the output CNN layer, as shown in Figure 17, CNN's output layer is linked to the 1st MLP layer. Due to this condition, the outputs of this layer CNN neurons, s_k^l, and hence, Δs_k^l are all becoming scalars. This recall is achieved by setting all the input map dimensions in that particular layer the same as the subsampling factors, *ssx*=*ssy*=8. Besides, CNN layer neurons related weights are also all scalar now, so scalar multiplication operations are achieved similar to MLP. In that condition, propagating through the MLP layer to CNN layer, BP operation is computed identical to the regular (scalar) MLP.

$$\frac{\partial E}{\partial s_k^l} = \Delta s_k^l = \sum_{i=1}^{N_{l+1}} \frac{\partial E}{\partial x_i^{l+1}} \frac{\partial x_i^{l+1}}{\partial s_k^l} = \sum_{i=1}^{N_{l+1}} \Delta_i^{l+1} w_{ki}^l \tag{19}$$

So, the intra-BP is used to obtain $\Delta_k^l \xleftarrow{BP} \Delta s_k^l$ becomes similar to Eq. 18, i.e.,

$$\Delta_k^l = \frac{\partial E}{\partial x_k^l} = up(\Delta s_k^l)\beta f'(x_k^l) \tag{20}$$

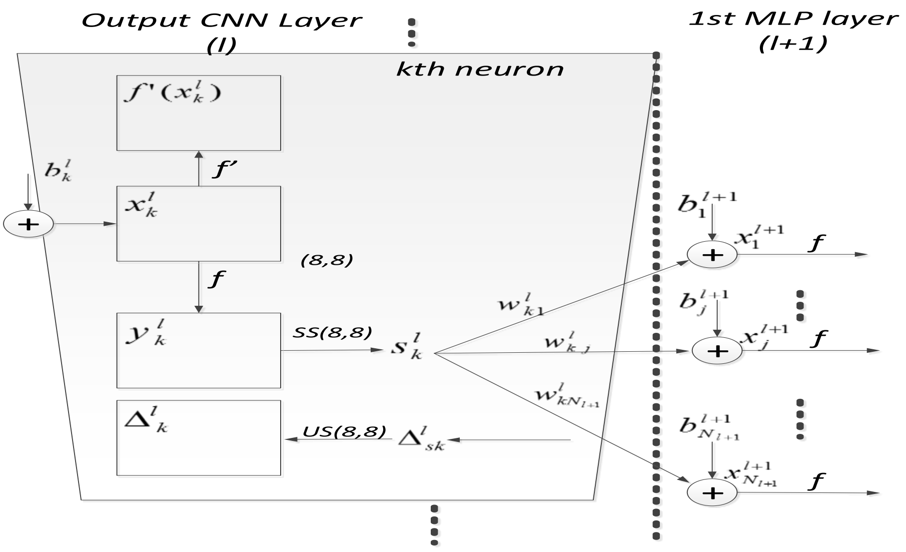


Figure 17. BP operations from the 1st dense layer to the output CNN layer (Source: Kiranyaz, Ince and Gabbouj, 2015).

Besides, the weight and bias sensitivities, similar to regular MLPs, can be expressed as given in Eq. 21,

$$\frac{\partial E}{\partial w_{kj}^{l}} = \frac{\partial E}{\partial x_{j}^{l+1}}\frac{\partial x_{j}^{l+1}}{\partial w_{kj}^{l}} = \Delta_{j}^{l+1} s_{k}^{l} \qquad and \qquad \frac{\partial E}{\partial b_{k}^{l+1}} = \frac{\partial E}{\partial x_{k}^{l+1}}\frac{\partial x_{k}^{l+1}}{\partial b_{k}^{l+1}} = \Delta_{k}^{l+1} \tag{21}$$

During the BP operation on dense layers, output error function is minimized using the gradient descent method, ($\Delta_k^l = \frac{\partial E}{\partial x_k^l}$), and the delta error of the i^{th} neuron at layer l+1, Δ_i^{l+1}, is processed to update the bias and all weights belongs to given neuron.

$$x_i^{l+1} = b_i^{l+1}+..+y_k^l w_{ki}^l+..\rightarrow \frac{\partial E}{\partial w_{ki}^l} = \frac{\partial E}{\partial x_i^{l+1}}\frac{\partial x_i^{l+1}}{\partial w_{ki}^l} = \frac{\partial E}{\partial x_i^{l+1}} y_k^l = y_k^l \Delta_i^{l+1}$$
$$and \ \frac{\partial E}{\partial b_i^{l+1}} = \frac{\partial E}{\partial x_l^{l+1}}\frac{\partial x_l^{l+1}}{\partial b_l^{l+1}} = \frac{\partial E}{\partial x_l^{l+1}} = \Delta_i^{l+1} \tag{22}$$

Weight and bias sensitivities of CNN layer neurons are obtained similarly as in MLPs. Input of the i^{th} neuron, x_i^{l+1}, at the next layer, l+1, is formed by the convolve operation of s_k^l (the current layer neuron output) and w_{ki}^l (the kernel, connecting the k^{th} neuron in the current layer to the i^{th} neuron) (see Figure 18).

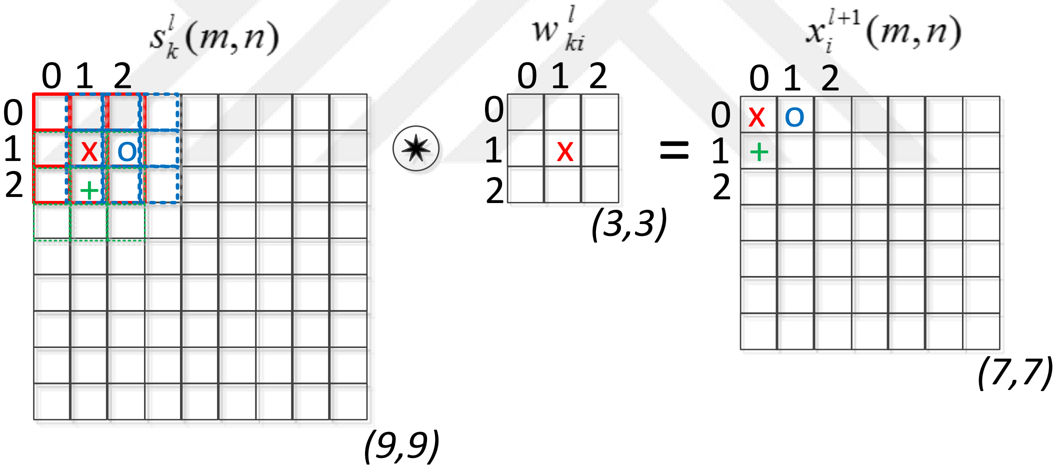


Figure 18. Input of the i^{th} neuron, x_i^{l+1}, at the next layer, l+1, is formed by the convolve operation of s_k^l (the current layer neuron output) and w_{ki}^l (Source: Kiranyaz, Ince and Gabbouj, 2015).

Analytically each kernel element contribution over the output can be written as:

$$
\begin{aligned}
&x_i^{l+1}(0,0) = ..+ w_{ki}^l(0,0)s_k^l(0,0) + w_{ki}^l(0,1)s_k^l(0,1) + w_{ki}^l(1,0)s_k^l(1,0) + ..\\
&x_i^{l+1}(0,1) = ..+ w_{ki}^l(0,0)s_k^l(0,1) + w_{ki}^l(0,1)s_k^l(0,2) + w_{ki}^l(1,0)s_k^l(1,1) + ..\\
&x_i^{l+1}(1,0) = ..+ w_{ki}^l(0,0)s_k^l(1,0) + w_{ki}^l(0,1)s_k^l(1,1) + w_{ki}^l(1,0)s_k^l(2,0) + ..\\
&...\\
&x_i^{l+1}(m,n) = ..+ w_{ki}^l(0,0)s_k^l(m,n) + w_{ki}^l(0,1)s_k^l(m,n+1) + w_{ki}^l(1,0)s_k^l(m+1,n) + ..\\
&\therefore x_i^{l+1}(m,n) = \sum_{r=-1}^{1}\sum_{t=-1}^{1} w_{ki}^l(r+1,t+1)\, s_k^l(m+r,n+t) + ..
\end{aligned}
\tag{23}
$$

Each neuron input, $x_i^{l+1}(m,n)$ are connected with the same weights that all connections share, so the derivative can be expressed as the summation of delta-output product from all pixels as in Eq. 24,

$$
\begin{aligned}
&\frac{\partial E}{\partial w_{ki}^l(r,t)} = \sum_m \sum_n \Delta_i^{l+1}(m,n) s_k^l(m+r,n+t)\\
&\Rightarrow \frac{\partial E}{\partial w_{ki}^l} = conv2D(s_k^l, \Delta_i^{l+1}, 'NoZeroPad')
\end{aligned}
\tag{24}
$$

Bias, b_k^l, sensitivity can also be expressed by accumulating the individual pixel sensitivities as given in Eq. 25 due to the same bias sharing in all pixels of the image.

$$
\frac{\partial E}{\partial b_k^l} = \sum_m \sum_n \frac{\partial E}{\partial x_k^l(m,n)} \frac{\partial x_k^l(m,n)}{\partial b_k^l} = \sum_m \sum_n \Delta_k^l(m,n)
\tag{25}
$$

3.2.3. Adaptive 1D CNNs and Back-Propagation Training

Based on the adaptive 2D CNNs and the BP training given in detail in the previous sections, adaptive 1D CNN and its BP training formulations will be derived next. During this dissertation, adaptive 1D CNN classifier processing raw motor current signals for fault detection is used to merge the feature extraction and learning stages into a single learning unit. Besides, the proposed architecture given in Figure 19 is capable of performing both convolution and sub-sampling operations with the hidden neurons of the fused convolution layers. Therefore, the 1D CNNs are formed of an input layer, hidden CNN layers, output layer and remaining MLP layers.

In the analytical formulations, the main structural difference between 2D and 1D versions is the array dimension. Instead of using 2D matrix manipulations *(conv2D),* 1D arrays (*conv1D)* that perform full convolution in one dimension are used at each neuron's inputs, outputs, sub-sampling (ss) and kernels (weights). Similarly, forward propagation (FP) is used to find output of each neuron at each layer, and back-propagation (BP) is used for computing delta error at the output layer, as seen in the Figure 19 below.

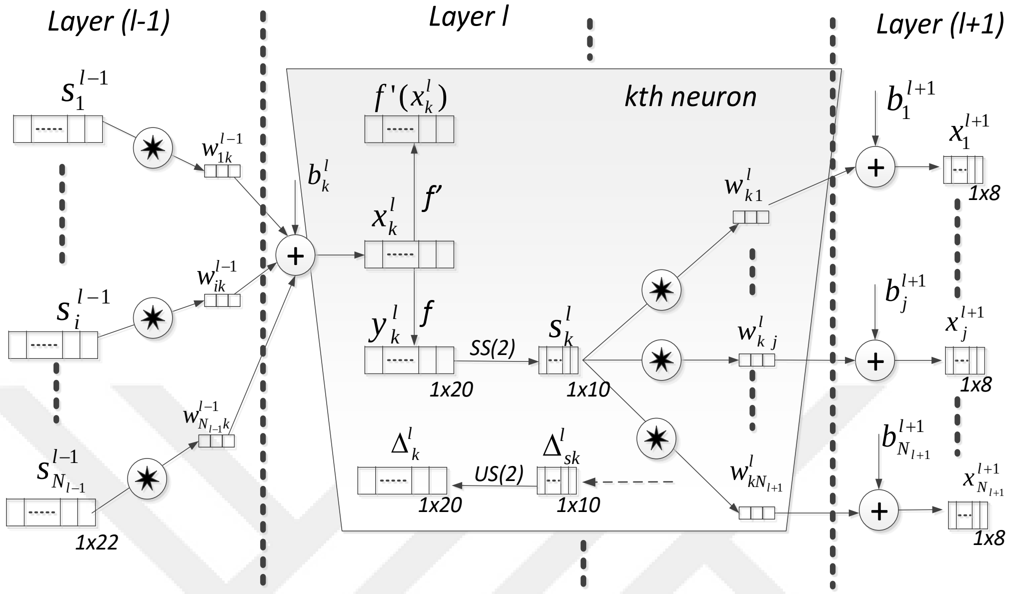


Figure 19. FP and BP operations in a 1D CNN neuron (Source: Kiranyaz, Ince and Gabbouj, 2015).

Moreover, while convolving, instead of the lateral rotation (*rot180*) in 2D, reverse *(rev*(.)) function is used to reverse the array in 1D counterpart, however the fully connected MLP layers following the last CNN layer are similar to in 2D CNNs, so traditional BP formulation can be also used for 1D CNNs.

Forward propagation (FP) in 1D CNNs starts from the *i*th neuron at *(l-1)*th layer to the input of the *k*th neuron at *l*th layer, x_k^l , with a scalar bias, b_k^l, and the kernel filter, w_{ik}^{l-1}, from the the *i*th neuron at *(l-1)*th layer to the *k*th neuron at *l*th layer, can be expressed as in Eq. 26,

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} conv1D(w_{ik}^{l-1}, s_i^{l-1}) \qquad (26)$$

here, s_i^{l-1} refers to the output of the *i*th neuron at *(l-1)*th layer and then y_k^l, the intermediate output of the *k*th neuron at *l*th layer, can be expressed by the input, x_k^l , as follows:

$$y_k^l = f\left(x_k^l\right) \quad and \qquad s_k^l = y_k^l \downarrow ss \qquad (27)$$

where the k^{th} neuron output s_k^l is obtained from the intermediate output by a down-sampling operation with the factor, ss.

BP steps are similar to those previously expressed in the Eq. 11. Derivative of the cost function (mean-squared error) in the output layer for the input p, E_{p}, is computed to update network parameters based on the gradient descent can be expressed as follows:

$$E_p = MSE\left(t_i^p, \left[y_1^L, \ldots., y_{N_L}^L\right]\right) = \sum_{i=1}^{N_L} \left(y_i^L - t_i^p\right)^2 \tag{28}$$

For updating the parameters of the dense (MLP) layers, the regular (scalar) BP is simply performed as,

$$\frac{\partial E}{\partial s_k^l} = \Delta s_k^l = \sum_{i=1}^{N_{l+1}} \frac{\partial E}{\partial x_i^{l+1}} \frac{\partial x_i^{l+1}}{\partial s_k^l} = \sum_{i=1}^{N_{l+1}} \Delta_i^{l+1} w_{ki}^l \tag{29}$$

Then, similar to the 2D CNN case, the first BP is initiated from the *(l+1)*th layer, to the *l*th layer, which can then be back-propagated to the input delta, Δ_k^l. Given the zero order up-sampled map as, $us_k^l = up(s_k^l)$, The delta can be expressed as;

$$\Delta_k^l = \frac{\partial E}{\partial y_k^l} \frac{\partial y_k^l}{\partial x_k^l} = \frac{\partial E}{\partial us_k^l} \frac{\partial us_k^l}{\partial y_k^l} f'\left(x_k^l\right) = up\left(\Delta s_k^l\right) \beta f'(x_k^l) \tag{30}$$

where $\beta = (ss)^{-1}$ since each element of s_k^l was obtained by averaging ss consecutive elements of the intermediate output, y_k^l. The inter BP of the delta error ($\Delta s_k^l \xleftarrow{\Sigma} \Delta_\iota^{l+1}$) can be defined as,

$$\Delta s_k^l = \sum_{i=1}^{N_{l+1}} conv1Dz\left(\Delta_\iota^{l+1},\ rev(w_{ki}^l)\right) \tag{31}$$

where full convolution in 1D with K-1 zero padding between the corresponding weight parameters and the delta error is performed. Finally, the weight and bias sensitivities can be found from,

$$\frac{\partial E}{\partial w_{ki}^l} = conv1D\left(s_k^l, \Delta_i^{l+1}\right)\ and\quad \frac{\partial E}{\partial b_k^l} = \sum_n \Delta_k^l\,(n) \tag{32}$$

3.2.4. The Pseudo-Code of Back-Propagation Algorithm

The pseudo-code of the BP algorithm used for training the 1D CNN classifier can be summarized as follows (Ince et al., 2016):

```
1. Initialize the weights and biases randomly, U(-a, a)
2. For each BP iteration DO:
    a) For each training sample in the dataset, DO:
        i.   FP: Forward propagate input sample towards the output layer to find outputs of each neuron at each layer, y_i^l,∀i∈[1,N_l ]and∀l∈[1,L].
        ii.  BP: Compute delta error at the output layer and back-propagate it to first hidden layer to compute the delta errors, Δ_k^l,∀k∈[1,N_l ]and∀l∈[2,L-1]
        iii. PP: Post-process to compute the weight and bias sensitivities using Eq. 21.
        iv.  Update: Perform gradient descent updates on the weights and biases using the sensitivities scaled with the learning factor, ε:
```

$$w_{ik}^{l-1}(t+1) = w_{ik}^{l-1}(t) - \varepsilon \frac{\partial E}{\partial w_{ik}^{l-1}} b_k^l(t+1) = b_k^l(t) - \varepsilon \frac{\partial E}{\partial b_k^l} \tag{33}$$

CHAPTER 4: METHODOLOGY

In this chapter, the employed methodologies while constituting the proposed enhanced bearing fault diagnosis architecture and verifying the capability of multi-channel, multi-level 1D CNNs classifier which is processing benchmark raw time-domain bearing vibration data provided by IMS are presented.

4.1. Two-Channel, Single-Level 1D CNN Classifier

The proposed compact and efficient 1D classifier architecture is designed in two parallel and identical CNN structures in order to process the two independent data simultaneously. The identical adaptive 1D CNNs, used during the experimentations, are formed with an input layer (the passive layer), 3 hidden CNN layers (fused mid-layers in terms of convolution and down-sampling operations), an output layer and the remaining MLP layers. Input layer accepts the data, in our cases x or/and y axis are being input to the CNN classifier. Hidden layers, so called the “CNN layers” are synthesizing higher-level (abstraction) discriminative features with a large set of 1D filter kernels. Output layers of CNN are used for the classification as healthy state, inner race fault and rolling element fault. Back-propagation (BP) technique is used for computing delta error by minimizing the contributions of network parameters at the output layers. Further information and mathematical approach of BP is given in detail in Chapter 3.

The three-layered convolutional structure has 10 neurons at each layer. Filter sizes are set to 101, 26, 26, respectively and the number of neurons in a fully connected MLP layer is set to 20. Three neurons are placed at the output layer in order to classify the detected fault to healthy, inner race fault and outer race failure classes as given in Figure 20.

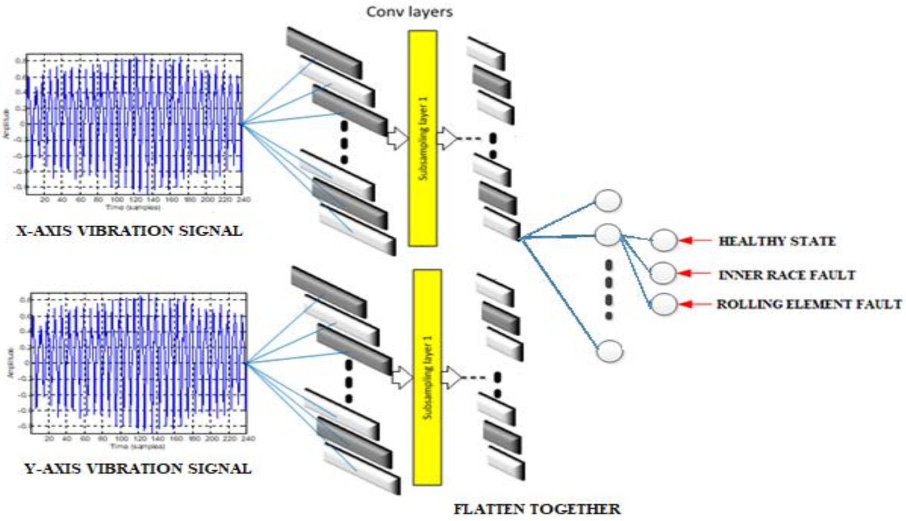


Figure. 20. Overview of proposed 1D architecture (Source: Ozcan et al., 2021).

As a transfer function, which is used to make it easier for the network to learn the complex pattern and maps the output to the desired scale with respect to the applied input, hyperbolic tangent (*tanh*) activation function is preferred in all CNN layers. It can be considered as the scaled activation sigmoid function, processes any real input value, and scales the output between -1 and 1 whereas in its sigmoid counterpart, the output is scaled between 0 and 1. Hyperbolic tangent function is a differentiable function**,** meaning, the slope of the tan*h* curve can be found at any two points.

In immediately followed subsampling layers, max-pooling is used for down-sampling. Pooling operation, in general manner, reduces the spatial size of applied input, results in reduction of number of parameters, hence the computational cost is reduced. During the convolution process of max pooling, in order to "accumulate" features from maps, the filter (kernel) extracts the maximum value as depicted in Figure 21.

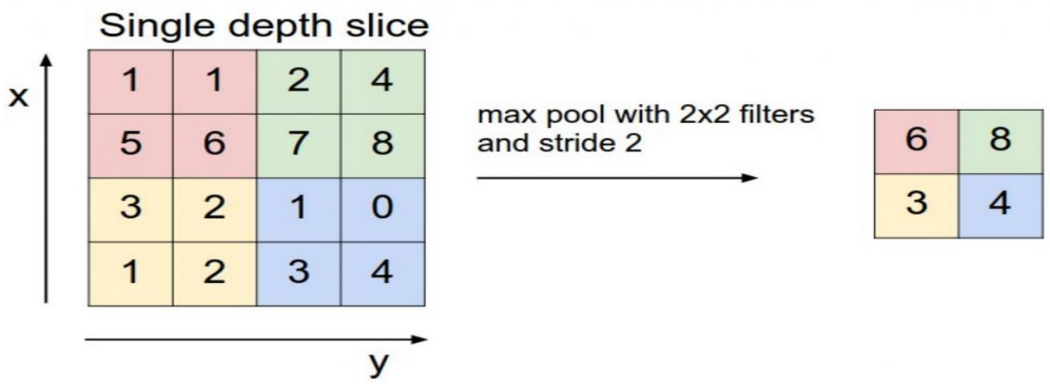


Figure 21. Max-pooling operation on a 4x4 input tensor with 2x2 filters with stride=2 (Source: dos Santos, 2019)

The system is also tested and verified by applying a tenfold cross-validation technique to put forth improvement in generalization, besides avoiding the overfitting problem during the training phase.

For optimization, a stochastic gradient descent (SGD) algorithm is chosen, which is an iterative optimizer that tries to locate the minimum of the function by following the negative gradient. Training and testing data are preprocessed independently for real-time operation.

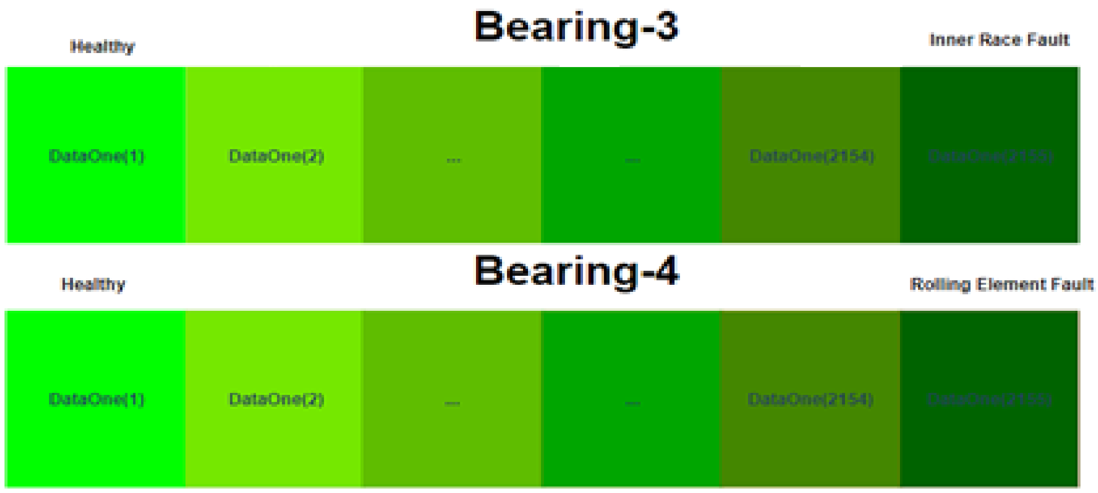


Figure 22. Dataset is sorted with respect to progression time from DataOne(1) to DataOne(2155). In this work, inner race and rolling elements faults from the bearings 3 and 4, respectively, are used for training and testing.

The system composed of four identical bearings placed on the test rig and each bearing contains recordings over accelerometers mounted on both *x*-axis and *y*-axis. 49680 min (34 days and 12 hour) recording duration is passed, and recorded data is written on 2155 text files and label as *DataOne*. *DataOne*(1) refers to the first recorded text file and *DataOne*(2155) to the last recorded one. As time passed, prominence of inner race fault occurred in bearing #3 and rolling element fault occurred in bearing #4.

As illustrated in Figure 22, the first 40 text files are considered healthy for both cases. After test setup operates nearly up to the end of the designed lifetime, initially healthy bearings show gradual deterioration and some fault types become more prominent in time. The dominance of inner race fault and rolling element faults are clearly observed at final 40 text files of bearing #3 and at final 40 files of bearing #4, respectively. Each text file consists of 20,480 points with sampling rate of 20 kHz but it is enough to identify faults when the input vibration signals of the CNN structure comprises 10,000 data points after it is transformed in Fourier Domain as shown in Figure 23.

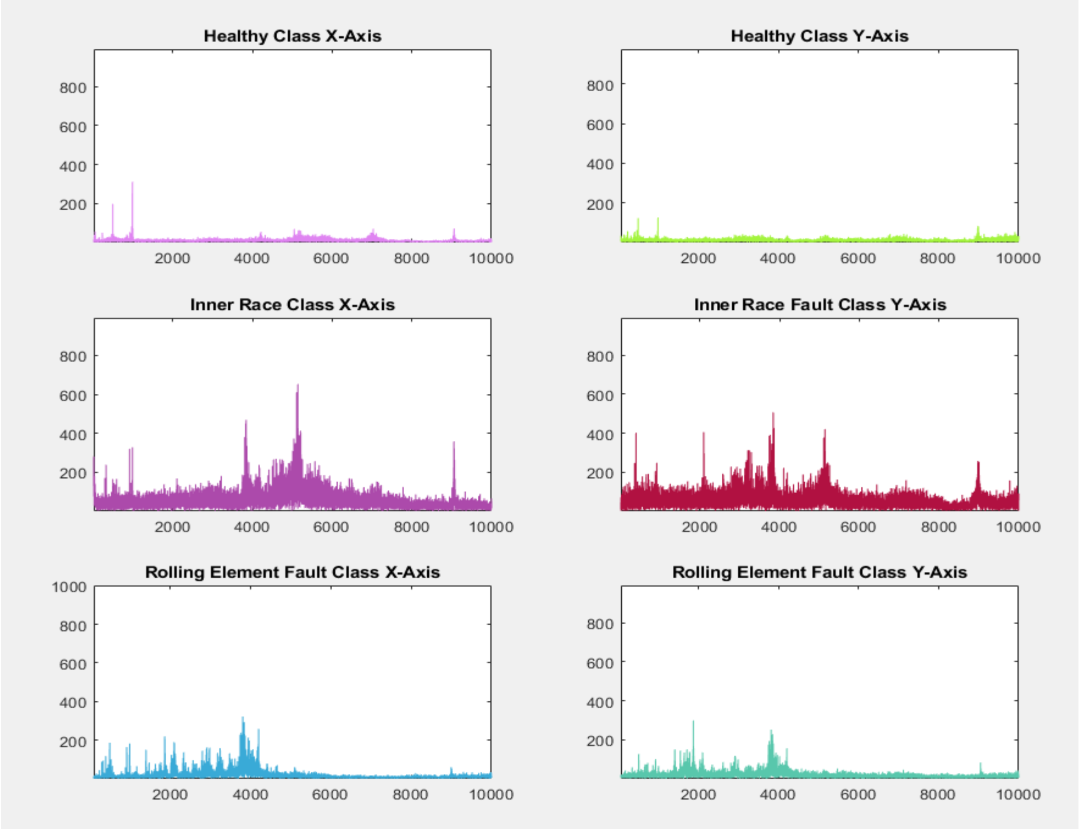


Figure 23. Corresponding vibration signals in Fourier Domain for healthy state and inner race fault cases of *x*-axis and *y*-axis from IMS Bearing Dataset.

4.2. Two-Channel, Two-Level 1D CNN Classifier

The main objective in using the multi-channel 1D CNN is to utilize available multi-channel vibration data and classify different types of motor bearing faults more accurately. When the proposed system is simultaneously trained with the two-channel data applied as an input to 1D CNN classifier, the network learns from both inputs (channels) and classifies output better than taking into consideration one channel at a time.

Additional to the two channel-single level 1D CNN classifier structure, identical two channel 1-D CNN classifiers are added to both inner race fault and rolling element fault classifier neuron. These newly added classifiers have six convolutional and two scalar MLP layers. Each layer of the six-convolutional layer structure has again ten neurons and but now filter size of 101, 26, 13, 9, 9, 9, respectively. All experiments are processed considering the suitability of the real-time condition monitoring of the two specific motor faults. The number of neurons in a fully connected MLP layer is set to 20, similar to the previous architecture. In this experiment, the output layer's neuron number is designed with two neurons to identify the severity levels of the corresponding faults as early and advanced level classes. The nonlinear activation function, *tanh*, is preferred again in all CNN layers and in immediately followed subsampling layers, max-pooling is used for down-sampling.

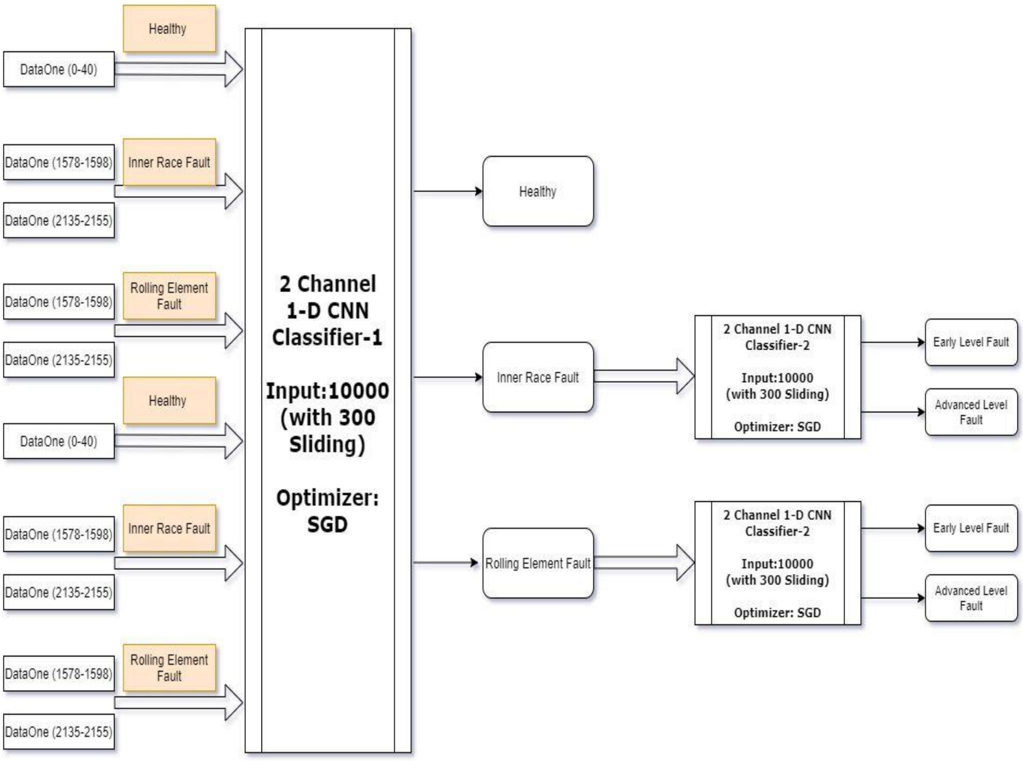


Figure 24. Main framework of the two-channel, two-level motor fault detection system (Source: Ozcan et al., 2021).

After the two-channel, single-level 1D CNN classifier showed considerably successful results in detection of healthy state, inner race fault and rolling element fault, the structure was developed by cascading with the aim of determining the two severity {early and advanced} levels of detected fault type, as the main topology clarified in Figure 24. The fundamental fault frequencies can be analytically calculated when the ball’s geometry is known. According to the detailed formulation given in Chapter 2 and determined fault frequencies are given in Table 3 Chapter 5, outer and inner race characteristic fault frequencies are known as 236 Hz and 297 Hz, respectively. When a fault emerges in bearings, there occurs a certain peak at a fault related frequency. When the fault level increases, harmonic peaks begin to prominent around these certain peaks, which can be used for classifying the fault levels. The data were labelled before being input to the two-channel, two-level 1D CNN classifier according to this deduced information from the frequency domain analysis of the inner race and rolling element faults (see Figure 25).

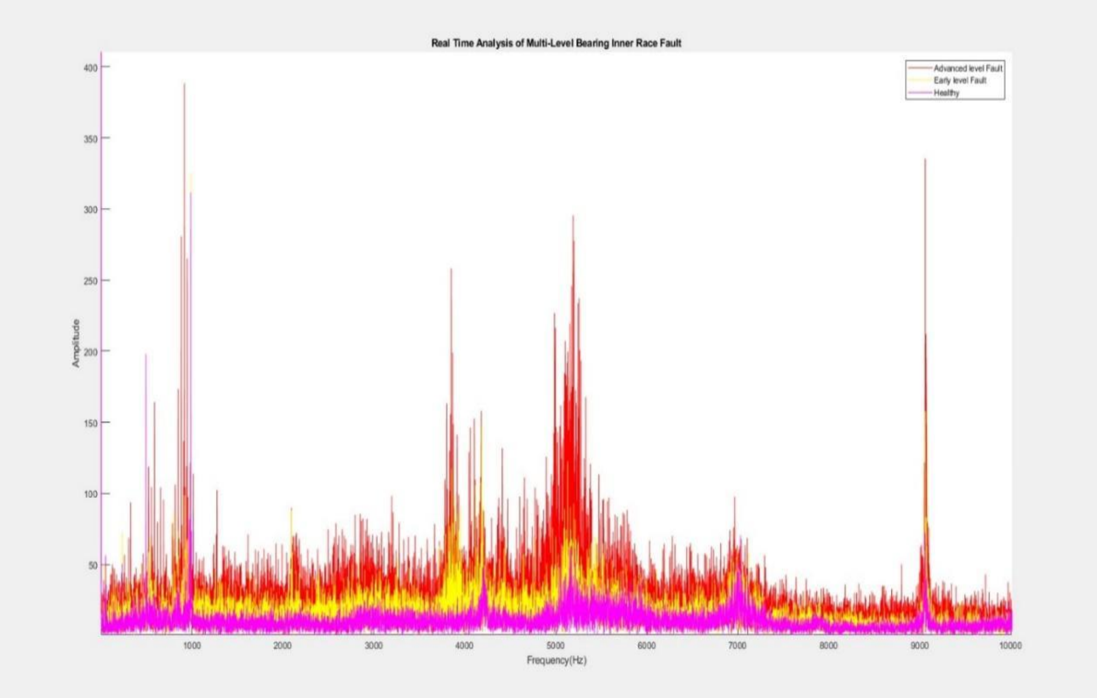


Figure 25. Frequency-domain analysis of inner race fault vibration signals.

The first 40 text files are accepted again as healthy, the text files between 1578 and 1598 are assumed as early fault related ones and finally, the files between 2135 and 2155 are considered as they are related with advanced fault level. These labeled data can be used in both train and test phases. In this structure an additional two neurons are placed at the output for detecting two new classes. The rest of the methodology is kept the same. The newly developed system is optimized under the same conditions with the same train/test ration and runs in 10 epochs similar to the first layer of the cascaded structure. The sliding window function is also applied.

4.2.1. Sliding-Window Technique

When a real time approach is considered, it requires a relatively long time to decide the class that fault belongs to. To avoid this shortcoming and in order to get more efficient results, a 10,000 data point-included structure with sliding 300 data points (at each time) is employed (as illustrated in Figure 26). Thanks to the proposed windowing method, the system works as if it includes 10,000 data points but it is computed as fast as it has 300 data points because in every case just 300 new data points is added to the structure. Moreover, this windowing approach allows us to work as we have 34 different structures instead of 2 structures that have 10,000 data points.

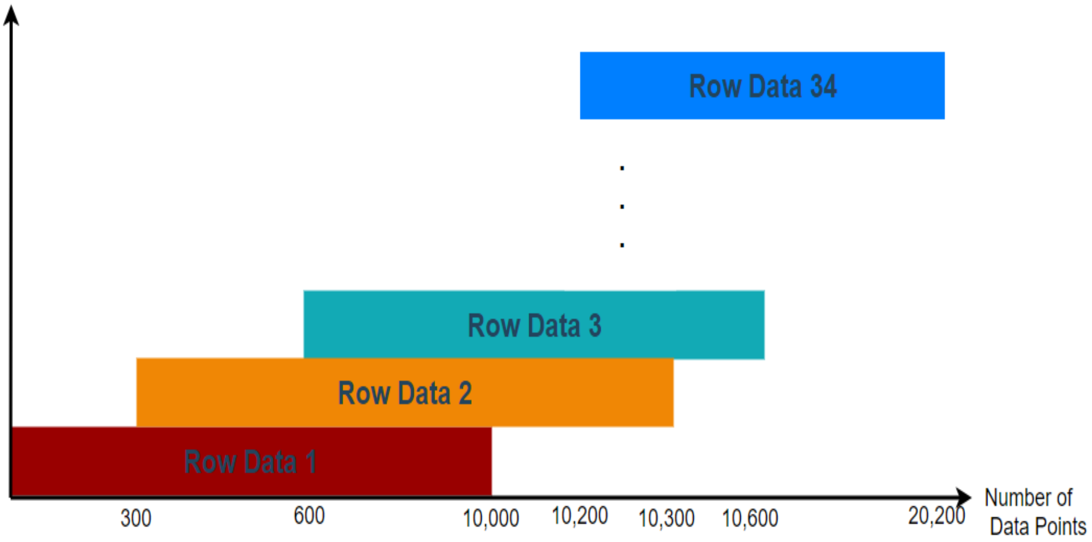


Figure 26. 10,000 points input and with sliding 300 data points.

Inner race and roller element faults can be seen as revealed along this progress in different bearings. Experiments show that proposed architecture achieved some promising results. The system is trained with 90% of the overall data using a ten-fold cross-validation method, which allows it to test the whole data. After data are splitted into train and test sets, they are normalized and standardized individually.

First, the system is trained with the data on *x*- and *y*-axis independently. Then the *x*- and *y*-axis are applied together. The overall confusion matrix and the sensitivities for a sample run for bearing fault diagnosis problems with respect to *x*- axis, *y*-axis and *x* & *y*-axis are presented in detail in Chapter 5.

The tests are executed with the SGD optimization algorithm. During the tests, default SGD parameters on Keras library (learning rate=0.01, momentum=0.0, decay=0.0, nesterov=False) with a batch size of 32 are used. The average classification accuracy over 10 trials is obtained as 88.4% when the system is trained with only *x*-axis and 83.6% with *y*-axis, respectively. However, when both channels are applied as input, the classification accuracy is significantly increasing to 100%. It can be clearly seen that the two-channel input case is superior to single channel input thanks to the proposed classifier architecture.

4.2.2. Real Time Software Implementation

The software-based experiment of the proposed 1D CNN classifier algorithm is tested and verified with Keras (Keras, 2018), one of the most popular libraries of the Python programming language. Keras is designed for rapid experimentation with deep neural networks implementation, which focuses on being user-friendly, and modular. It also has an extensible structure for different implementations. Keras can be considered as a wrapper that uses Theano, CNTK or Tensorflow as a backend. It uses the Python language and allows to define and train models in a very easy way, thanks to the many functional functions it consists of.

The Keras developers designed it as an open-source library so that anyone who wants to work on deep learning implementations can create their own scenarios instead of dealing with backend architecture. With a trial-and-error approach, without any prior deep knowledge beginners can learn how and which modifications will affect the models. Besides, the extensibility of Keras allows users to merge any kind of models to construct a complex structure to utilize for such classification, text generation, and speech recognition tasks. All NN models can operate smoothly on CPU vs GPU. Performing operations on the GPU ensures to save significant time.

It supports convolutional neural networks (CNN) for computer vision models and iterative neural networks (RNN) for continuous data. Forming a fresh structure only requires merging the individual modules (such as: NN layers, cost functions, optimizers, activation functions, etc…). These modules can be inserted easily just like inserting new classes or functions. Sequential model (linear model) and the functional API (complex model) are the two of the most used Keras libraries (for further information, see (Keras, 2018)).

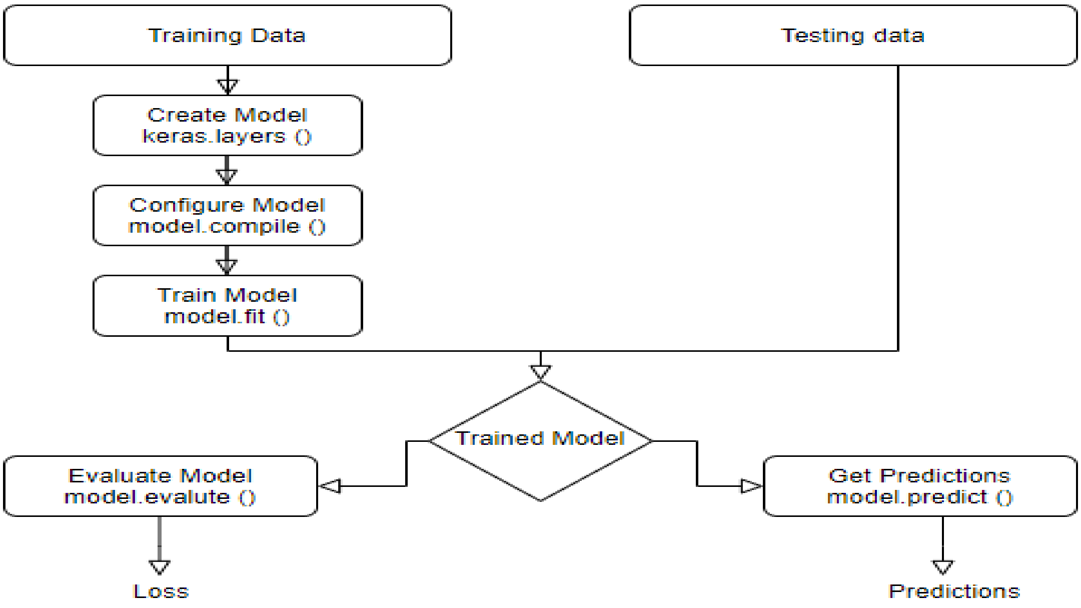


Figure 27. A typical Keras workflow.

4.2.3. Computation Complexity

All the tests run on a computer having Intel Core i7-8750H processor and NVIDIA GeForce 1050 Ti Graphics Processing Unit (GPU) on it. Compute Unified Device Architecture (CUDA) kernels, which is a platform presented by NVIDIA for parallel computing interface, process the train and test sets over 10-fold and 10 epochs with SGD optimizer. Under the same conditions, all tests are repeated with a CPU based computer to compare the performance in terms of computational time. While CPU runs average of 634.644 sec for training and average of 4.86 sec for testing of the single-channel CNN, these computation times are equal to 3.68 and 0.002 sec for GPU respectively. For two channel CNN classifiers, average run time of CPU is recorded as 931.8 for training and 7.02 for testing phase. Also, with GPU based computation, they drastically decreased to 5.18 and 0.014 sec, respectively. These results are also given in Table 8. There is a significant difference between GPU and CPU in computational times. For real time applications, GPU's performance is clearly in preferable condition.

Table 1. Comparison of computational times of single-channel and two-channel 1-D CNN classifier.

Time (sec)				
Trials	**single-channel CNN**		**two-channel CNN**	
	Train	**Test**	**Train**	**Test**
CPU	634.644	4.860	931.800	7.020
GPU	3.680	0.002	5.180	0.014

CHAPTER 5: EXPERIMENTAL EVALUATIONS

This chapter is concentrated on the experimental utilization of the proposed 1D CNN classifier on multi-channel implementation. Firstly, experimental set-up is given and the benchmark vibration dataset, which is used to validate the capability and feasibility of the proposed approach, is introduced. Secondly, data preprocessing of the sample vibration signal in terms of decimation, necessary filtering and normalization stages are explained. For better understanding, time- and frequency-domain depiction is given. Finally, the main scope of this dissertation, the proposed multi-channel, multi-level 1D CNN classifier architecture used for bearing fault detection and fault level identification simultaneously is presented. Also, the fault detection performance comparison of the single and two-channel for multiple-level in terms of commonly applied performance metrics are listed.

5.1. Experimental Setup

During this dissertation, the well-known bearing vibration benchmark dataset released by University of Cincinnati Intelligent Maintenance Systems (IMS) in 2007 (Lee et al., 2007) is used to validate the effectiveness and usefulness of the proposed 1D CNN classifier. Data are gathered with a Rexnord ZA-2115 four double row bearings used endurance test rig as illustrated in Figure 28. The shaft, on which four identical row bearings are mounted, and AC motor is coupled with a rub belt, so the rotation speed is fixed to 2000 RPM. While temperature and flow are regulated with a forced lubricated circulation system, bearings temperatures are measured with thermocouple sensors. A spring mechanism is used to apply 6000 lbs. radial load to the shaft. The data belonging to two axes (*x*- & *y*- axis) are collected from each bearing with the aid of two accelerometers (PCB model 253B33 High-sensitivity Quart ICP®), which are mounted on each bearing housing.

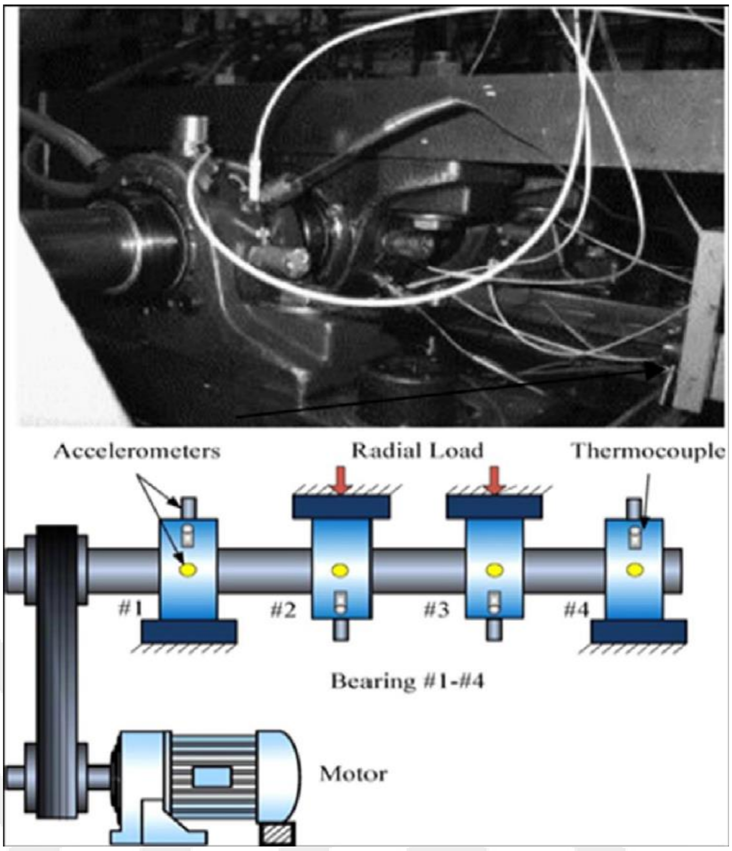


Figure 28. Bearing test rig and sensor placement illustration (Source: Lee et al., 2007).

All the data sampled at 20 kHz are collected with a data acquisition card (NI DAQ card 6062E) and recorded to the .txt files, each including 20480 data points. Each data file contains 8 columns in which data from two different axes (*x*- and *y*-axis) are recorded sequentially for four identical bearings. In example of, first and second data columns are related to data collected from *x*-axis and *y*-axis of bearing#1, respectively. Similarly, third and fourth data columns belongs to data collected from *x*-axis and *y*-axis of bearing#2. The rest is followed in the same order for bearing#3 and bearing#4. The first forty-three files are recorded at 5-minute intervals, while the next files are recorded at 10 minutes of data. The further dataset details are given in Table 2.

The bearings used in the experimental setup are manufactured to last over 100 million revolutions. The system was run as a test-to-failure experiment and at the end of the endurance test, inner race defect in the third bearing and roller element defect in the fourth bearing became prominent.

Table 2. Dataset Information (Source: Lee et al., 2007).

Number of files	Number of channels	Endurance duration	Duration of recorded signals	Announced damages at the end of the endurance
2156	8	49680 min 34 days 12h	36 min	Bearing 3: inner race Bearing 4: rolling element

Due to the scope of this dissertation, in order to identify the bearing fault types, the proposed algorithm is both applied to inner race defect related vibration data recorded onto the fifth and sixth column (also defined as channel), and to the rolling element fault related vibration data recorded onto the seventh and eighth column. The used vibration data are collected via the accelerometers mounted on third and fourth bearings, respectively. For the diagnosis, knowing the general information regarding the ball geometry and rotational speed is adequate to calculate the fundamental fault related vibration frequencies of rolling element bearings.

For the used bearing vibration benchmark dataset, the general information regarding the bearings is stated as follows; Rexnord ZA-2115 double row bearings are used. Each row has 16 rollers. Pitch diameter, roller diameter and tapered contact angle are 2.815 in., 0.331 in., and 15.17°, respectively. Rotational speed is fixed to 2000 RPM, and by using the given information, fault related frequencies are calculated as in Table 3.

Table 3. Analytically determined characteristic frequencies of the test rig.

Characteristic Frequencies	
Shaft Frequency	33.3 Hz
Ball Pass Frequency Outer race (BPFO)	236 Hz
Ball Pass Frequency Inner race (BPFI)	297 Hz
Ball Spin Frequency (BSF)	278 Hz (2x139 Hz)
Fundamental Train Frequency (FTF)	15 Hz

5.2. Data Preprocessing

The IMS bearing faults dataset, aforementioned comprehensively in the previous section, which contain two axes (*x*- and *y*-axis) vibration data, is being input to the specially designed two-channel 1D-CNN classifier framework (Figure 29) to detect inner race fault and rolling element fault evolved in third and fourth bearings, respectively. For training and testing phases, these two axes vibration signals are employed. Faster convergence and real time adaptability of the system are provided by individually z-score standardization and scaled linearization in the range of [− 1 1] of the time-domain vibration training and test sets. Standardization and normalization equations are presented in Eq. 34 & 35.

$$z = \frac{x - \mu}{\sigma} \tag{34}$$

$$X_{normalized} = \frac{X - ((X_{maximum} + X_{minimum})/2)}{(X_{maximum} - X_{minimum})/2} \tag{35}$$

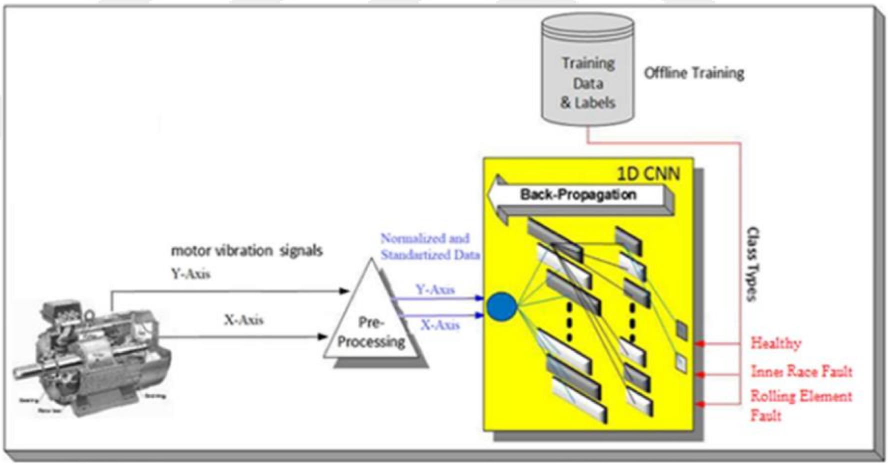


Figure 29. Proposed framework two-channel 1D CNN classifier-1 (Source: Ozcan et al., 2021).

The sample raw vibration signals and corresponding preprocessed vibration signals for healthy state, inner race fault, and rolling element fault for *x*- axis from IMS bearing dataset are illustrated in Figure 30.

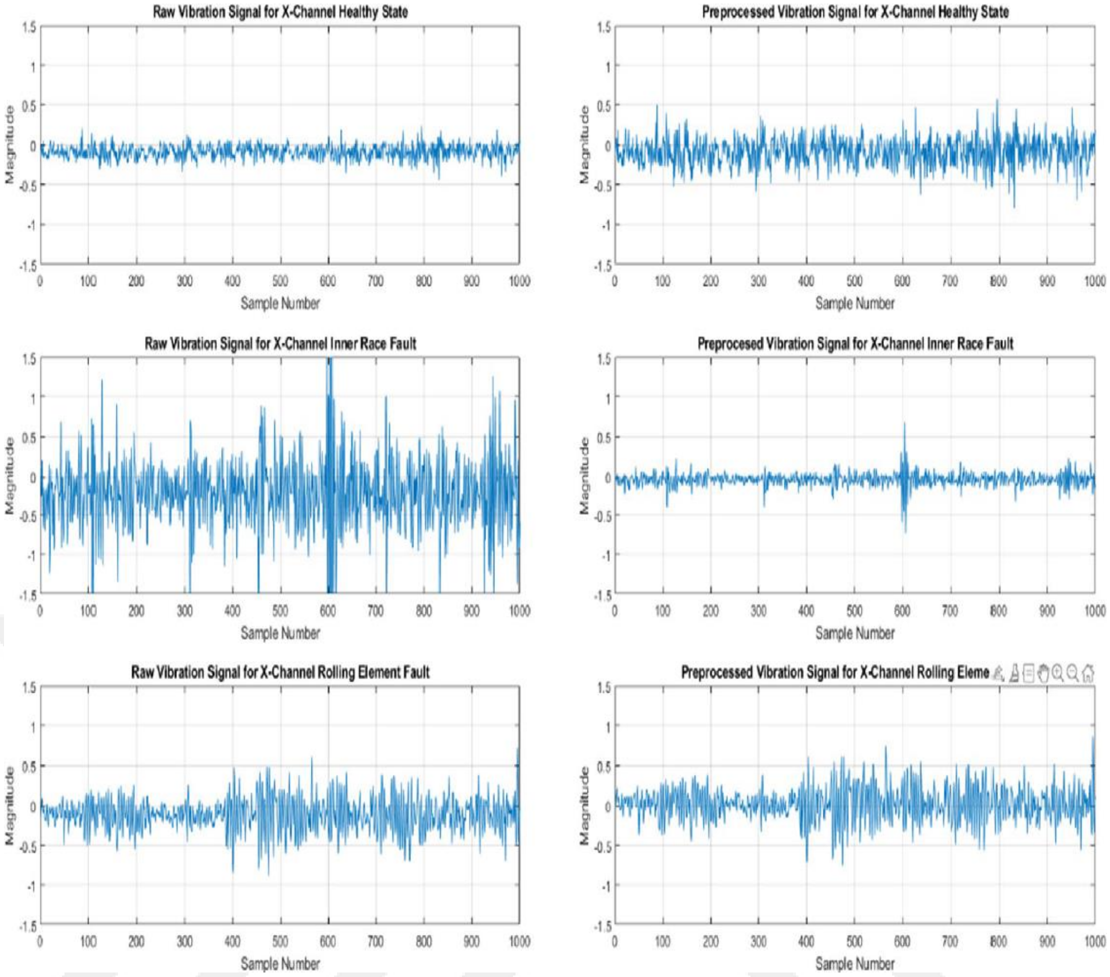


Figure 30. Sample raw vibration signals and corresponding preprocessed vibration signals for healthy state, inner race fault, and rolling element fault for *x*- axis from IMS bearing dataset (Source: Ozcan et al., 2021).

After test setup operates nearly up to the end of the designed lifetime, initially healthy bearings show gradual deterioration and some fault types become more prominent in time, if not intervened. Early and advanced stage faults can be obviously distinguished thanks to the benchmark bearing vibration signals analyzed in time- and frequency-domain as represented in Figure 31.

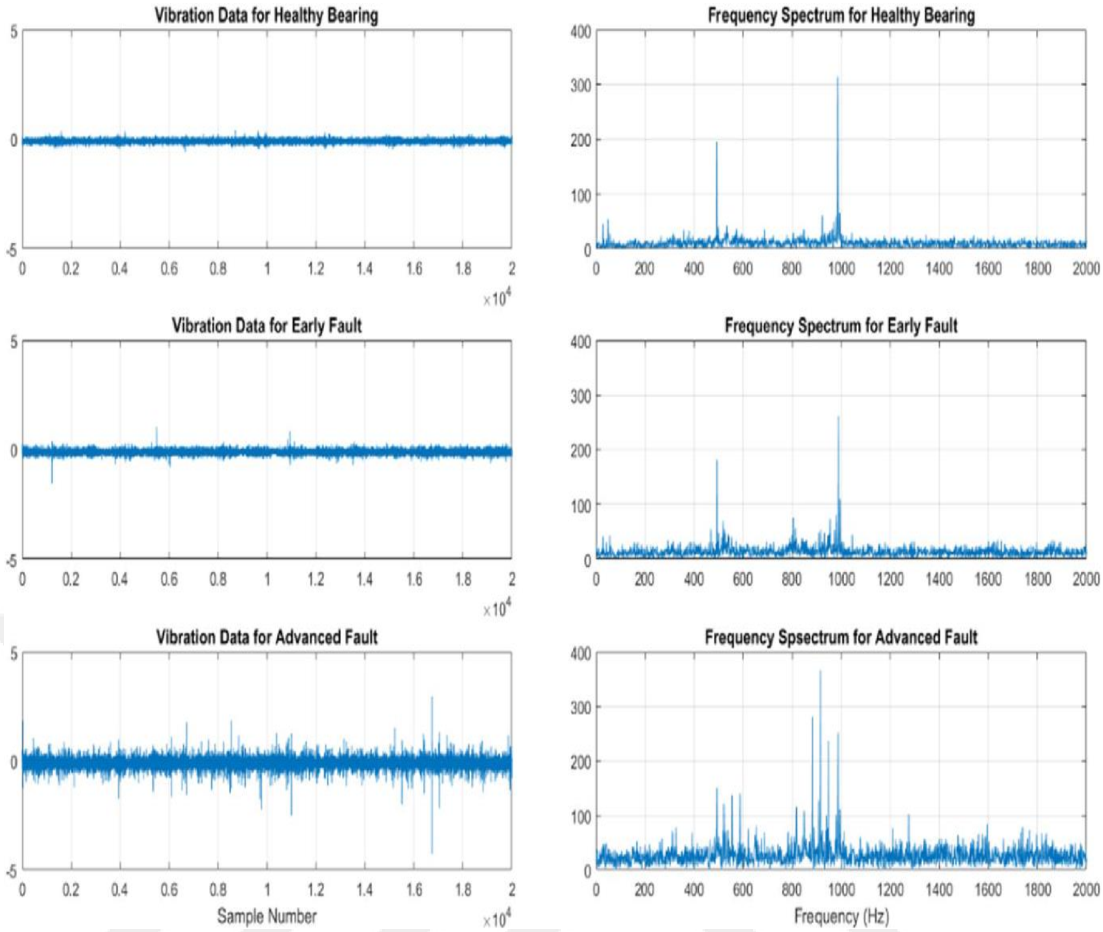


Figure 31. Time-domain (left) and frequency-domain (right) plots of inner race fault vibration signals (Source: Ozcan et al., 2021).

5.3. Two-Channel/Single-Level 1D CNN Classifier Training

Firstly, two-channel 1D-CNN architecture is designed with three hierarchical convolutional layers and two consecutive scaler MLP layers. The proposed architecture is duplicated and used for learning different types of bearing faults from both *x*- and *y*- axis vibration data, independently. This compact design provides our experiment with high computational efficiency with competitive performance. Implementation of the classifier is achieved with the aid of the programming language, Python, and its deep learning library, Keras which has become popular in recent years due to allowing fast and easy application development for researchers (Keras, 2018).

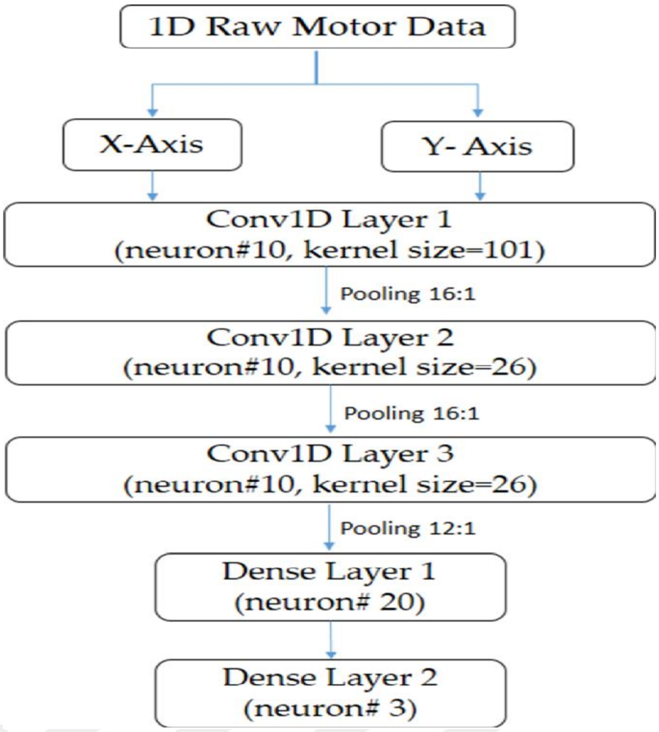


Figure 32. Overview of two-channel 1-D CNN classifier structure employing Conv1D (number of neurons, filter size) and dense (number of neurons) operations (Source: Ozcan et al., 2021).

After the architecture design phase, the two-channel data collected with an accelerometer mounted on the bearings are being input to the proposed classifier and they are processed with optimized (learned) filters of the hierarchical convolutional layers. Here, the main purpose is extracting the best possible features for the bearing fault classification and this process runs simultaneously on two parallel channels for both *x*- and *y*-axis. Training efficiency of this single machine learner model is provided by backpropagation (BP) iterations. As the inner design of 1D-CNN classifier is given on Figure 32, previous CNN layers' outputs are processed by each hidden CNN layer's learned filters to synthesize higher-level (abstraction) features. This compact and efficient design allows the proposed classifier to be used for real-time fault detections. Each layer of the three-convolutional layer structure has 10 neurons and filter size of 101, 26, 26, respectively. The number of neurons in a fully connected MLP layer is designed as 20. Finally, the output layer's neuron number is set to three neurons for the purpose of classifying the detected fault to healthy, inner race fault and outer race failure classes. The nonlinear activation function, *tanh*, is preferred in all CNN layers and in immediately followed subsampling layers, max-pooling is used for down-sampling. It is also tested and verified that applying a tenfold cross-validation technique puts forth improvement in generalization, besides avoiding the overfitting problem during the training phase. For optimization, as an iterative method, SGD is applied. Training and testing data are preprocessed independently for real-time operation.

5.4. Two-Channel/Two-Level 1D CNN Classifier Training

In the continuation of this dissertation study, fault detection phenomenon is tried to be enhanced in terms of severity levels. In this context, the architecture mentioned so far is improved by connecting two two-channel 1-D CNN architectures identifying faulty levels as early and advanced levels instead of just fault types. The enhanced and tested structure is as given in Figure 33.

Additional to the two channel-single level 1D CNN classifier structure, identical two channel 1-D CNN classifiers are added to both inner race fault and rolling element fault classifier neuron. These newly added classifiers have six convolutional and two scalar MLP layers. Each layer of the six-convolutional layer structure has again ten neurons and but now filter size of 101, 26, 13, 9, 9, 9, respectively. All experiments are processed considering the suitability of the real-time condition monitoring of the two specific motor faults. The number of neurons in a fully connected MLP layer is set to 20, similar to the previous architecture. In this experiment, the output layer’s neuron number is designed with two neurons to identify the severity levels of the corresponding faults as early and advanced level classes. The nonlinear activation function *tanh* is preferred again in all CNN layers and in immediately followed subsampling layers, max-pooling is used for down-sampling.

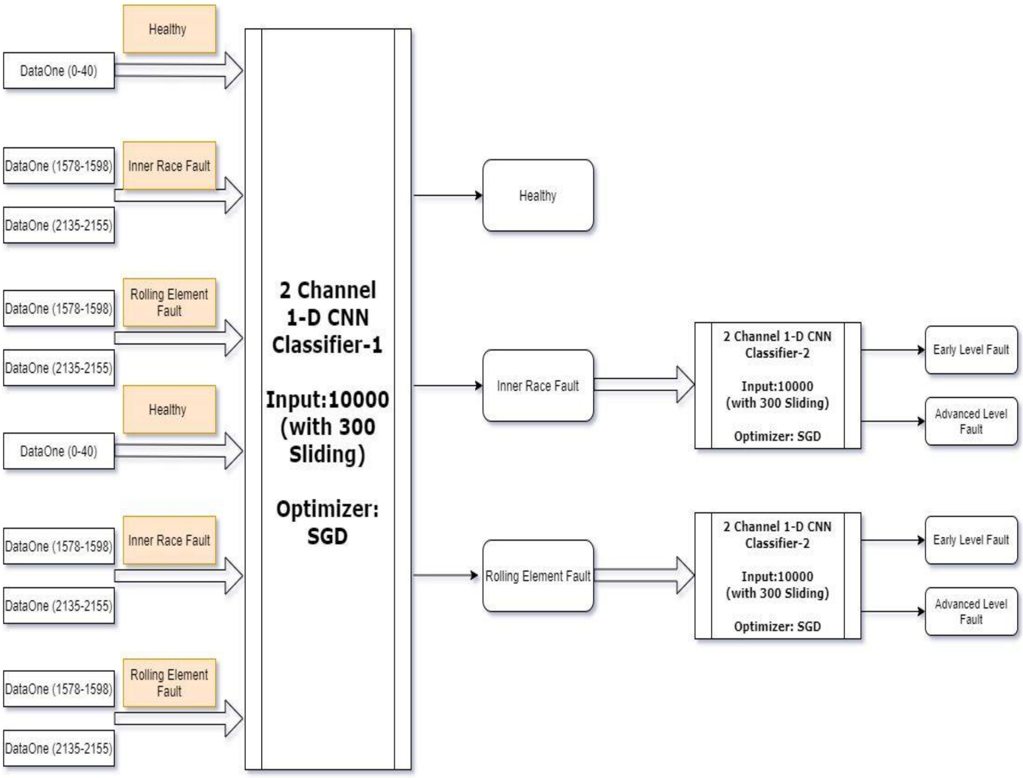


Figure 33. Main framework of the two-channel, two-level motor fault detection system (Source: Ozcan et al., 2021).

5.5. Experimental Results

During the training and testing phases of the proposed classifier, the benchmark IMS bearing dataset is used and while recording the data, the test-rig's rotational speed is fixed to 2000 RPM with a sampling rate of 20,000 samples per second. 600 sample data can be recorded per revolution.

While working on this experiment, to increase the efficiency of the classifier in training and also the testing phases, sliding window function technique is applied to each class. Windowing technique allows us to obtain 1400 data files with 10,000 (input) data points, almost obtained in sixteen revolutions.

Mainly in terms of compatibility with real-time classifications, all tests are achieved with the formerly proposed compact 1-D CNN architecture, designed with only three CNN and two MLP layers.

First, by keeping the training and testing conditions the same, the performance of a single-channel CNN classifier where only *x*- or *y*-axis vibration data is applied is compared with that of a two-channel CNN classifier that processes *x*- and *y*-axis vibration data simultaneously.

The experimental results put forth that by employing the proposed two-channel CNN structure, the overall detection accuracy is significantly improved with respect to the single-channel CNN classifier. The system is tested and verified by applying a tenfold cross-validation method for generalization improvement and avoiding overfitting problems. During the training phase, 90% of the overall data is used and each phase runs with 50 epochs. For optimization, as an iterative method, Stochastic Gradient (SGD) is applied to all experiments. The average classification accuracy over ten trials is obtained as 88.4% for *x*-axis only and 83.6% for *y*-axis only as given in Tables 4, 5. However, a two-channel 1D CNN classifier that processes *x*- and *y*-axis vibration data simultaneously achieves a significant increase in the classification accuracy, 100.00%, for classifying the healthy, inner race fault and rolling element fault classes. The confusion matrix for a two-channel classifier is given in Table 6.

The classification performance of the proposed framework (given in Table 5 and Table 8) is measured with the following five commonly used performance metrics in the literature as (Hu, Palreddy, and Tompkins,1997); Classification Accuracy (*Acc*), Recall or Sensitivity (*Sen*), Specificity (*Spe*), Precision or Positive Predictivity (*Ppr*), and F1-score (*F1*). These metrics are unique to each class, and they present the proposed classifier's ability to differentiate particular events from non-events. Accuracy is used to evaluate the performance of the entire system over all fault classes, however other metrics are class specific. *Accuracy* (Acc) is the ratio of the number of correctly classified patterns to the total number of patterns classified, Acc=(TP+TN)/(TP+FP+TN+FN); *Recall or Sensitivity* (Sen) is the rate of correctly classified events in all events, Sen=TP/(TP+FN); *Specificity* (Spe) is the rate of correctly classified nonevents in all nonevents, Spe=TN/(TN+FP); *Precision or Positive Predictivity* (Ppr) is the rate of correctly classified events in all detected events, Ppr = TP/(TP+FP) and *F*1-*score* (F1) is the harmonic mean (average) of the model’s Precision and Recall, F1=2*(Ppr*Sen)/(Ppr+Sen) where FN, FP, TN and TP are stands for false negatives, false positives, true negatives and true positives, respectively. *AUC* stands for "Area under the ROC (receiver operating characteristic curve) Curve" where ROC curve plots TP rate vs. FP rate at different classification thresholds and *AUC* measures the entire two-dimensional area underneath the entire ROC curve.

Table 4. Confusion matrix of a single-channel (*x*- or *y*-axis) bearing fault diagnosis problem.

Bearing Fault	*Classification Result*						
Ground Truth		*Healthy*		*Inner Race Fault*		*Rolling Element Fault*	
		***x*-axis**	***y*-axis**	***x*-axis**	***y*-axis**	***x*-axis**	***y*-axis**
	Healthy	1400	1400	0	0	0	0
	Inner Race Fault	3	10	1333	951	64	459
	Rolling Element Fault	0	0	420	240	980	1160

Table 5. Bearing fault detection performance of a single-channel 1D CNN classifier.

Bearing Fault	Fault Detection Performance											
	Acc		***Sen***		***Spe***		***Ppr***		***F1***		***AUC***	
	x-axis	*y*-axis	*x*-axis	*y*-axis	*x*-axis	*y*-axis	*x*-axis	*y*-axis	*x*-axis	*y*-axis	*x*-axis	*y*-axis
Healthy	88.4	83.6	100	100	100	100	99.8	99.3	99.9	99.7	100	100
Inner Race Fault	88.4	83.6	95.2	67.9	97.3	88.9	76.0	79.7	84.5	73.3	82.4	92.2
Rolling Element Fault	88.4	83.6	70.0	82.9	86.7	90.7	93.4	73.3	80.0	77.8	87.5	97.7

Table 6. Confusion matrix of two-channel (both *x*- and *y-axis* data) 1D CNN classifier for the same problem.

Bearing Fault	Classification Result			
Ground Truth		*Healthy*	*Inner Race Fault*	*Rolling Element Fault*
	Healthy	1400	0	0
	Inner Race Fault	0	1400	0
	Rolling Element Fault	0	0	1400

Table 7. Confusion matrix of two-channel, two-level 1D CNN classifier for the same problem.

	Classification Results					
Ground Truth		*Healthy*	*IR Early Level Fault*	*IR Advanced Level Fault*	*RE Early Level Fault*	*RE Advanced Level Fault*
	Healthy	1287	19	4	67	23
	Inner Race Fault (Early Level Fault)	155	521	24	0	0
	Inner Race Fault (Advanced Level Fault)	0	51	649	0	0
	Rolling Element Fault (Early Level Fault)	98	0	0	505	97
	Rolling Element Fault (Advanced Level Fault)	81	0	0	27	592

After reaching significantly successful results in classifying the healthy, inner race fault and rolling element fault classes by using two-channel 1D CNN classifier, the structure is depicted in Figure 33, a two-channel, two-level fault classifier is designed and tested for detection of the inner race and rolling element faults with severity levels as early fault level and advanced fault level. Average classification accuracy is computed as 84.64% over ten trials with the help of the confusion matrix (see Table 7). According to the fault detection performances, as given in Table 8, it can be clearly seen that sensitivity to early fault detection for inner race and rolling element faults are found to be comparably lower than advanced-level fault detection for the corresponding faults and healthy class.

Mean Squared Error (MSE) is used to calculate the error at the output of the classifier:

$$MSE = \frac{1}{n}\sum\nolimits_{i=1}^{n} (y_i - \tilde{y}_i)^2 \qquad (36)$$

where corresponding output and target vectors are y_i and $\tilde{y}_i$.

Test performance results are calculated by comparing the index of the predicted output's maximum value in row with known test output's index of maximum value in row.

Table 8. Confusion matrix of two-channel, two-level 1D CNN classifier for the same problem.

	Fault Detection Performance					
	Acc	***Sen***	***Spe***	***Ppr***	***F1***	***AUC***
Healthy	89.4	91.9	88.1	79.4	85.2	100
Inner Race Fault (Early Level Fault)	94.1	74.4	98.0	88.2	80.7	91.3
Inner Race Fault (Advanced Level Fault)	98.1	92.7	99.2	95.9	94.3	88.3
Rolling Element Fault (Early Level Fault)	93.1	72.1	97.3	84.3	77.7	86.4
Rolling Element Fault (Advanced Level Fault)	94.6	84.6	96.6	83.1	83.8	81.4

Table 9. Comparison of accuracy rates with different classification techniques using the same IMS bearing vibration dataset.

Classifiers	**Features**	**Classifier inputs**	**Single-level fault detection accuracy**	**Two-level fault detection accuracy**
2-channel, 2-level 1-D CNN Classifier	-	*x- and y-axis raw vibration data*	*100%*	***84.64%***
Compact 1-D CNN	-	*single axis*	93.90%	-
CRNN	-	*x- and y-axis data concatenated*	97.13%	-
DNN	-	*single axis*	94.40%	-
SFS-DLSVM	*WPT and EMD*	*merged heterogeneous features input to DLSVM classifier*	99.10%	-

Finally, the other state-of-the-art works in the literature, that use the same bearing vibration benchmark dataset, are searched and the accuracy comparison with our work is listed as in Table 9. However, fault detection with two-channel/two-level 1D CNN classifiers is not encountered in any of these studies.

CHAPTER 6: CONCLUSION

Electric motors are used in a wide range of industrial applications from simple electro-mechanical implementations to aerospace applications. Malfunctioning of induction motors is a major concern due to its impact on industrial production.

In some circumstances these failures evolve in time, however occasionally the motor suddenly stops functioning without any warning, which results in high economic loss due to the interruption of the production line. Not to face inevitable hazards and to maximize the longevity of the induction motor and benefit throughout its operating life, these failures must be identified in the incipient phase and the necessary repair and/or replacement operations must be performed to save time and money.

Electric motors commonly employ rolling (ball or roller) element bearings and mechanical bearing faults are the statistically most encountered fault type. The main motivation for this dissertation arose from the importance of timely diagnosis and detection of the bearing faults to minimizing the economic and time loss. At the time of significant advances in sensor-based Internet-of-the-Things (IoT) technologies, development and advances in the signal processing and data analytics technologies are needed and equally important.

In this scope, a new bearing fault diagnosis system compatible with real-time classifications is introduced to the literature. The proposed multichannel 1D CNN classifier architecture is designed on the purpose of utilizing multiple channels of sensor data simultaneously to achieve enhanced bearing fault detection performance. For the implementation and test of the proposed structure, the widely used IMS bearing vibration dataset is preferred to validate the processing ability of multiple axis vibration data simultaneously. The real bearing defect data used during the experiment is collected in two independent axes (*x*- & *y*-axis) with the aid of two accelerometers which are mounted on each bearing housing placed in the endurance test set-up.

These motor vibration data gathered from multiple independent axes are applied as input information to the multichannel 1D CNN classifier to classify inner race and rolling element faults simultaneously. Proposed compact 1-D CNN architecture is designed with only three CNN and two MLP layers to be compatible with real-time classifications and in all tests, sliding window function technique is applied to increase the efficiency of the classifier in training and the testing phases. During the training phase, 90% of the overall data is used and each phase runs with 50 epochs. The system is tested and verified by applying a tenfold cross-validation method for generalization improvement and avoiding overfitting problems. Stochastic Gradient (SGD) is applied to all experiments as an optimizer.

The experimental results put forth that by employing the proposed two-channel CNN structure, the overall detection accuracy is significantly improved with reasonable computational time. One of the attractive advantages of the proposed classifier is that 1D CNNs can learn highly distinctive features from multiple channels of raw data, meaning, there is no need for any manual feature extraction and/or parameter design.

It is also obvious from the experimental results that, multichannel CNN classifier is superior to the standard single-channel CNN classifier. While the 1D CNN classifier is trained with data collected from *x*-axis, 88.40% of classification accuracy is achieved. It is obtained as 83.60%, when the vibration data collected from the y-axis is used. On the other hand, the inner race and rolling element fault detection accuracy is reached to 100.00 % when the multichannel CNN classifier processes the *x*- and *y*-axis data simultaneously. The proposed classifier can efficiently utilize the independent vibration information collected at perpendicular directions to improve bearing fault detection performance.

After reaching significantly successful results in classifying the healthy, inner race fault and rolling element fault classes by using two-channel 1D CNN classifier, a two-channel, two-level fault classifier is designed and detected inner race and rolling element faults' severity levels (i.e., {healthy, early fault, advanced fault}) are aimed to be identified with the same benchmark IMS bearing dataset to increase the contribution of this study. Over ten trials, average classification accuracy is obtained as 84.64%.

As a future work, the proposed multi-channel multilevel 1D CNN fault classifier architecture is planned to be tested using larger real datasets including data from variable speed applications. Moreover, the possibility of providing more classes (healthy, early fault, medium fault, and advanced fault) about fault levels will be explored by using three-axis accelerometers in data collection.

REFERENCES

Ahishali, M., Kiranyaz, S., Ince, T. and Gabbouj, M. (2019) *Dual and Single Polarized SAR Image Classification Using Compact Convolutional Neural Networks*, Remote Sens., Vol. 11(11), pp.1340-1363.

Albrecht, P. F., Appiarius, J. C. and McCoy, R. M. (1986) *Assessment of the Reliability of Motors in Utility Applications-Updated*, IEEE transactions on Energy Conversion, Vol. 1(1), pp. 39-46.

Arun, P., Lincon, S., and Prabhakaran, N. (2019) *An Automated method for the analysis of bearing vibration based on spectrogram pattern matching*, Journal of Applied Research and Technology, 17(2).

Ayhan, B., Chow, M. and Song, M. (2005) *Multiple signature processing-based fault detection schemes for broken rotor bar in induction motors*, IEEE Trans. Energy Convers., Vol. 20, pp. 336–343.

Ballal, M.S., Khan, Z.J., Suryawanshi, H.M. and Sonolikar, R.L. (2007) *Adaptive Neural Fuzzy Inference System for the Detection of Inter-Turn Insulation and Bearing Wear Faults in Induction Motor*, IEEE Trans. Ind. Electron., Vol. 54, pp. 250–258.

Barden Precision Bearings. (2017). *Bearing Failure: Causes and Cures* [Online]. Available at: https://www.schaeffler.com/remotemedien/media/_shared_media/08_media_library/01_publications/barden/brochure_2/downloads_24/barden_bearing_failures_us_en.pdf (Accessed: 30 May 2020)

Bellini, A., Filippetti, F., Franceshini, G. and Tassoni, C. (2001) *Quantitative evaluation motor broken bars by means of electrical signature analysis*, IEEE Trans. Ind. Appl., Vol. 37, pp. 1248–1254.

Bin, G.F., Gao, J.J., Li, X.J. and Dhillon, B.S. (2012) *Early fault diagnosis of rotating machinery based on wavelet packets–empirical mode decomposition feature extraction and neural network*, Mech. Syst. Signal Process, Vol. 27, pp. 696–711.

Chauvin, Y. and Rumelhart, D. E. (1995) *Back Propagation: Theory, Architectures, and Applications*, 1st Edition, London: Lawrence Erlbaum Associates Publishers.

Ciresan, D.C., Meier, U., Gambardella, L.M. and Schmidhuber, J. (2010) *Deep big simple neural nets for handwritten digit recognition*, Neural Comput., Vol. 22, pp. 3207–3220.

Dai, X. and Gao, Z. (2013) *From Model, Signal to Knowledge: A Data-Driven Perspective of Fault Detection and Diagnosis*, IEEE Trans. Ind. Informat., Vol. 9, pp. 2226–2238.

Dannana, S. (2019) *What is a shafts misalignment? |Causes|Effects|* [Online]. Available at: https://extrudesign.com/different-shafts-misalignment/ (Accessed: 1 May 2021).

de Chazal, P., O'Dwyer, M. and Reilly, R. B. (2004) *Automatic classification of heartbeats using ECG morphology and heartbeat interval features*, IEEE Trans. Biomed. Eng., Vol. 51(7), pp. 1196–1206.

dos Santos, L. A. (2019). Artificial Intelligence [Online]. Available at: https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/pooling_layer (Accessed: 19 December 2019).

[Engineering Learn]. (2021). Types of Bearings [Web-based visual]. Available at: https://engineeringlearn.com/types-of-bearings/ (Accessed: 1 May 2021)

Eren, L. (2017) *Bearing Fault Detection by One-Dimensional Convolutional Neural Networks*, Mathematical Problems in Engineering, Vol. 2017, pp. 1-9.

Eren, L. and Devaney, M.J. (2004) *Bearing Damage Detection via Wavelet Packet Decomposition of the Stator Current*, IEEE Trans. Instrum. Meas., Vol. 53, pp. 431–436.

Eren, L., Cekic, Y. and Devaney, M. (2009) *Broken rotor bar detection via wavelet packet decomposition of motor current*, International Review of Electrical Engineering, Vol. 4, pp. 844-850.

Eren, L., Ince, T. and Kiranyaz, S. (2019) *A Generic Intelligent Bearing Fault Diagnosis System Using Compact Adaptive 1D CNN Classifier*, Journal of Signal Processing Systems, Vol. 91 (2), pp.179-189.

Farabet, C., Poulet, C., Han, J. and LeCun, Y. (2009) *An FPGA-based Processor for Convolutional Networks*, In Proceedings of the International Conference on Field Programmable Logic and Applications, International Conference on Field Programmable Logic and Applications 2019. Prague. 31 August-2 September 2009.

Filippetti, F., Bellini, A. and Capolino, G.A. (2013) *Condition monitoring and diagnosis of rotor faults in induction machines: State of art and future perspectives*, In Proceedings of the IEEE WEMDCD, IEEE Workshop on Electrical Machines Design, Control and Diagnosis (WEMDCD) 2013. Paris. 11-12 March 2013.

Gao, Z., Cecati, C. and Ding, S.X. (2015) *A Survey of Fault Diagnosis and Fault-TolerantTechniques—Part I: Fault Diagnosis with Model-Based and Signal-Based Approaches,* IEEE Trans. Ind. Electron, Vol. 62, pp. 3757–3767.

Grubic, S., Aller, J.M., Lu, B. and Habetler T.G. (2008) *A survey on testing and monitoring methods for stator insulation systems of low-voltage induction machines focusing on turn insulation problems,* IEEE Ind Electron., Vol. 55, pp. 4127–4136.

Harris T.A., (2000), *Rolling bearing analysis*. New York: John Wiley.

Hu, Y. H., Palreddy, S. and Tompkins, W. J. (1997) *A patient-adaptable ECG beat classifier using a mixture of experts approach,* IEEE Transactions on Biomedical Engineering, Vol. 44(9), pp. 891-900.

IEEE Gold Book (1998) *IEEE Recommended Practice for the Design of Reliable Industrial and Commercial Power Systems,* IEEE Std. 493-1997, pp. 1-464, Appendix H.

Ince, T. (2019) *Real-time broken rotor bar fault detection and classification by shallow 1D convolutional neural networks,* Electrical Engineering, Vol. 101(2), pp. 599-608.

Ince, T., Kiranyaz, S., Eren, L., Askar, M. and Gabbouj, M. (2016) *Real-Time Motor Fault Detection by 1D Convolutional Neural Networks,* IEEE Trans. Ind. Electron. 63, pp. 7067-7075.

[Keras] Deep Learning Library [Online]. Available at: https://keras.io/.

Kiranyaz, S., Ince, T. and Gabbouj, M. (2013) *Multi-Dimensional Particle Swarm Optimization for Machine Learning and Pattern Recognition*. New York, NY, USA: Springer.

Kiranyaz, S., Ince, T. and Gabbouj, M. (2015) *Real-Time Patient-Specific ECG Classification by 1D Convolutional Neural Networks,* IEEE Trans. Biomed. Eng., Vol. 63, pp. 664-674.

Kim, K. and Parlos, A.G. (2002) *Induction motor fault diagnosis based on neuropredictors and wavelet signal processing,* IEEE/ASME Trans. Mechatron., Vol. 7, pp. 201–219.

Khorram A. and Khalooei M. (2019) *Intelligent Bearing Fault Diagnosis with Convolutional Long-Short-Term-Memory Recurrent Neural Network.*

Klausen, A. (2019). *Condition Monitoring of Rolling Element Bearings During Low and Variable Speed Conditions.* Doctoral Thesis. University of Agder.

Kliman, G.B., Premerlani, W.J., Yazici, B., Koegl, R.A. and Mazereeuw, J. (1997) *Sensorless online motor diagnostics,* IEEE Comput. Appl. Pow. Vol. 10, pp.39–43.

Konar, P. and Chattopadhyay, P. (2011) *Bearing fault detection of induction motor using wavelet and support vector machines (SVMs)*, Appl. Soft Comput., Vol. 11, pp. 4203–4211.

Kowalski, C.T. and Kowalska, T.O. (2003) *Neural network application for induction motor faults diagnosis*, Math. Comput. Simulat., Vol. 63, 435-448.

Kral, C., Habetler, T.G. and Harley, R.G. (2004) *Detection of Mechanical Imbalances of Induction Machines without Spectral Analysis of Time Domain Signals*, IEEE Trans. Ind. Appl., Vol. 40, pp.1101–1106.

Krizhevsky, A., Sutskever, I. and Hinton, G. (2012) *Imagenet classification with deep convolutional neural networks*, In Proceedings of the Advances in Neural Information Processing Systems (NIPS), pp. 1097-1105.

[KSB SE & Co]. (2021). Rolling element bearing [Web-based visual]. Available at: https://www.ksb.com/centrifugal-pump-lexicon/rolling-element-bearing/191386/ (Accessed: 1 May 2019)

Lee, J., Qiu, H., Yu, G. and Lin, J. (2007) Rexnord Technical Services, IMS, University of Cincinnati. Bearing Data Set, NASA Ames Prognostics Data Repository. NASA Ames Research Center: Moffett Field, CA, USA. Available online: https://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/#bearing (Accessed: 15 Mar 2017).

Li, B., Chow, M-Y, Tipsuwan, Y. and Hung, J.C. (2000) *Neural Network Based Motor Rolling Bearing Fault Diagnosis*, IEEE Trans. Ind. Electron., Vol. 47, pp. 1060-1069.

Li, D.Z., Wang, W. and Ismail, F. (2015) *An enhanced bispectrum technique with auxiliary frequency injection for induction motor health condition monitoring*, IEEE Trans. Instrum. Meas., Vol. 67, pp. 2279-2287.

Liu, R., Meng, G., Yang, B., Sun, C. and Chen, X. (2017) *Dislocated time series convolutional neural architecture: An intelligent fault diagnosis approach for electric machine*, IEEE Trans. Ind. Inform., Vol. 13(3), pp. 1310-1320.

Lou, X. and Loparo, K. A. (2004) *Bearing fault diagnosis based on wavelet transform and fuzzy inference*, Mechanical Systems and Signal Processing, 18(5), 1077-1095.

Matić, D., Kulić, F., Pineda-Sánchez, M. and Kamenko, I. (2012) *Support vector machine classifier for diagnosis in electrical machines: application to broken bar*, Expert Syst. Applications, Vol. 39 (10), pp. 8681-8689.

Mao, W., Wang, L. and Feng, N. (2019) *A New Fault Diagnosis Method of Bearings Based on Structural Feature Selection*, Electronics, Vol. 8, p. 1406.

[Machine Design]. (2015). What's the Difference Between Bearings? [Web-based visual]. Available at: https://www.machinedesign.com/learning-resources/whats-the-difference-between/article/21831901/whats-the-difference-between-bearings

[Micro-Lube]. (2021). A Brief Introduction to Lubrication Failure [Web-based visual]. Available at: https://www.micro-lube.com/blog/post/a-brief-introduction-to-lubrication-failure/

Ordaz-Moreno, A., Romero-Troncoso, R.D., Rivera-Guillen, J.R., Vite-Frias, J.A. and Garcia-Perez, A. (2008) *Automatic Online Diagnosis Algorithm for Broken-Bar Detection on Induction Motors Based on Discrete Wavelet Transform for FPGA Implementation*, IEEE Trans. Ind. Electron, Vol. 55, pp. 2193-2202.

Ovacikli, A.K. (2011). *Condition Monitoring of Ball Bearings Using Vibration Analysis*. Master's Thesis. Lulea University of Technology.

Ozcan, I.H., Devecioglu, O.C., Ince, T., Eren, L. and Askar, M. (2021) *Enhanced bearing fault detection using multichannel, multilevel 1D CNN classifier*, Electrical Engineering, pp. 1-13.

Pons-Llinares, J., Antonino-Daviu, J.A., Riera-Guasp, M., Lee, S. B., Kang, T.J. and Yang, C. (2015) *Advanced induction motor rotor fault diagnosis via continuous and discrete time–frequency tools*, IEEE Trans. Ind. Electron, Vol. 62, pp. 1791 – 1802.

Qian, N. (1999) *On the momentum term in gradient descent learning algorithms*, Neural Networks, Vol. 12, pp. 145–151.

Randall, R. B. and Antoni, J. (2011) *Rolling element bearing diagnostics-A tutorial*, Mech. Syst. Signal Process., Vol. 25, no. 2, pp. 485–520.

Scherer, D., Muller, A. and Behnke, S. (2010) *Evaluation of pooling operations in convolutional architectures for object recognition*. 20th International Conference on Artificial Neural Networks (ICANN). Thessaloniki, Greece. 12 September 2010.

Schoen, R.R., Habetler, T.G., Kamran, F. and Bartheld, R.G. (1995) *Motor bearing damage detection using stator current monitoring*, IEEE Trans. Ind. Appl., Vol. 31, pp. 1274-1279.

Shen, C., Wang, D., Kong, F. and Tse, P.W. (2013) *Fault diagnosis of rotating machinery based on the statistical parameters of wavelet packet paving and a generic support vector regressive classifier*, Measurement: Journal of the International Measurement Confederation, Vol. 46 (4), pp. 1551-1564.

Shuai, J., Shen, C. and Zhu, Z. (2017) *Adaptive Morphological Feature Extraction and Support Vector Regressive Classification for Bearing Fault Diagnosis*, Int. J. Rotating, Vol. 2017, pp. 1-10.

Sidahmed, M., and Dalpiaz, Q. (2001) *Encyclopedia of Vibration.* 1st Edition. Cambridge: Academic Press, pp. 1184-1193.

[SKF Group]. (2017). Bearing Failures and Their Causes [Web-based visual]. Available at: https://www.skf.com/binaries/pub12/Images/0901d1968064c148-Bearing-failures---14219_2-EN_tcm_12-297619.pdf (Accessed: 10 May 2020)

[SKF Group]. (2018). Types of Bearings [Web-based visual]. Available at: http://www.skf.com (Accessed: 10 May 2020)

Tu, J.V. (1996) *Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes,* Journal of Clinical Epidemiology, Vol. 49(11), pp. 1225-1231.

Tutorialspoint (2021) *Artificial Intelligence - Neural Networks* [Online]. Available at: https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_neural_networks.html (Accessed: 3 June 2021).

Vakharia, V., Gupta, V.K. and Kankar, P.K. (2014) *A multiscale permutation entropy-based approach to select wavelet for fault diagnosis of ball bearings,* J. Vib. Control, Vol. 21, pp. 3123-3131.

Wang, X., Zheng, Y., Zhao, Z. and Wang, J. (2015) *Bearing fault diagnosis based on statistical locally, linear embedding,* Sensors, Vol. 15, pp.16225–16247.

Wiesel D.H. and Hubel, T.N. (1959) *Receptive fields of single neurones in the cat's striate cortex,* J. Physiology, Vol. 148, pp. 574–591.

Wowk, V. (1991) *Machinery Vibration, Measurement and Analysis*, New York, NY, USA: McGraw-Hill.

Xi, F., Sun, Q. and Krishnappa, G. (2000) 'Bearing Diagnostics Based on Pattern Recognition of Statistical Parameters', *Journal of Vibration and Control*, Vol. 6(3), pp. 375–392.

Yaqub, M.F., Gondal, I. and Kamruzzaman, J. (2012) *Inchoate fault detection framework: Adaptive selection of wavelet nodes and cumulant orders,* IEEE Trans. Instrum. Meas., Vol. 61, pp. 685-695.

Yan, R., Gao, R.X. and Chen, X. (2014) *Wavelets for fault diagnosis of rotary machines: A review with applications,* Signal Process, Vol. 96, pp. 1–15.

Ye, Z., Wu, B. and Sadeghian, A. (2003) *Current signature analysis of induction motor mechanical faults by wavelet packet decomposition,* IEEE Trans. Ind. Electron., Vol. 50, pp. 1217–1227.

Yu, X., Dong, F., Ding, E., Wu, S. and Fan, C. (2017) *Rolling bearing fault diagnosis using modified LFDA and EMD with sensitive feature selection,* IEEE Access, Vol. 6, pp. 3715–3730.

Zhang, R., Peng, Z., Wu, L., Yao, B. and Guan, Y. (2017) *Fault Diagnosis from Raw Sensor Data Using Deep Neural Networks Considering Temporal Coherence*, Sensors, Vol. 17(3), pp. 549-566.

Zhao, N., Mio, W. and Liu, X. (2011) *A hybrid PCA-LDA model for dimension reduction*, The International Joint Conference on Neural Networks 2011. San Jose, CA. 31 July 2011- 5 August 2011.

Zheng, J., Pan, H. and Cheng, J. (2017) *Rolling bearing fault detection and diagnosis based on composite multiscale fuzzy entropy and ensemble support vector machines,* Mechanical System and Signal Processing, Vol. 85, pp. 746-759.

Zhou, W., Habetler, T. and Harley, R. (2008) *Bearing fault detection via stator current noise cancellation and statistical control*, IEEE Trans. Ind. Electron, Vol.55, pp. 4260-4269.