NATIONAL TECHNICAL UNIVERSITY OF ATHENS

POLYTECHNIC SCHOOL

DEPARTMENT OF NAVAL ARCHITECTURE & MARINE ENGINEERING

USING CONVOLUTIONAL NEURAL NETWORK (CNN) TO LEARN BY SENSOR SIGNAL CLASSIFICATION CRITICAL LOCAL AREAS IN A PHYSICAL FLEXIBLE SHAFT ROTOR SYSTEM AS AN EXAMPLE OF A COMPLICATED MECHANICAL SYSTEM

by

PANTELIS-PANAGIOTIS PAPAGEORGIOU

Mechanical Engineer, University of Thessaly, 2019

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Naval Engineering at the National Technical University of Athens

Athens

2022

NATIONAL TECHNICAL UNIVERSITY OF ATHENS

POLYTECHNIC SCHOOL

DEPARTMENT OF NAVAL ARCHITECTURE & MARINE ENGINEERING

USING CONVOLUTIONAL NEURAL NETWORK (CNN) TO LEARN BY SENSOR SIGNAL CLASSIFICATION CRITICAL LOCAL AREAS IN A PHYSICAL FLEXIBLE SHAFT ROTOR SYSTEM AS AN EXAMPLE OF A COMPLICATED MECHANICAL SYSTEM

by

PANTELIS-PANAGIOTIS PAPAGEORGIOU

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Naval Engineering at the National Technical University of Athens

Athens

2022

Certified by the members of the Thesis Committee:

First Examiner	Dr. Ioannis T. Georgiou
(Supervisor)	Professor, Department of Naval Architecture & Marine Engineering, National Technical University of Athens
Second Examiner	Dr. Konstantinos I. Spyrou
	Professor, Department of Naval Architecture & Marine Engineering, National Technical University of Athens
Third Examiner	Dr. Konstantinos N. Anyfantis
	Assistant Professor, Department of Naval Architecture & Marine Engineering, National Technical University of Athens

Acknowledgments

This project is accomplished in the scope of partial fulfillment of the requirements for the degree of Master in Naval Architecture & Marine Engineering of the National Technical University of Athens.

I would like to thank my thesis supervisor, Dr. Ioannis T. Georgiou, for allowing me the current Thesis to deal with such an interesting and promising topic. Furthermore, his expertise, valuable suggestions, comments, guidance, and patience were vital for the completion of this Thesis.

Furthermore, very special thanks go to Professor Konstantinos Spyrou and Assistant Professor Konstantino Anyfantis for accepting to be the referees of this work.

I would also like to express my great appreciation to the National Technical University of Athens (NTUA) to conduct experiments in the Lab Unit of Dynamics-Acoustics and Diagnostics of Complex Mechanical Systems, Division of Marine Engineering-School of Naval Architecture and Marine Engineering,

In addition, special recognition goes to my family for their continuous support and encouragement during this study. Last but not least, I would like to dedicate this work to my beloved grandparents were taught me not to give up, I lost from the pandemic Covid-19 in 2021.

Pantelis-Panagiotis Papageorgiou

BSc. Mechanical Engineering

USING CONVOLUTIONAL NEURAL NETWORK (CNN) TO LEARN BY SENSOR SIGNAL CLASSIFICATION CRITICAL LOCAL AREAS IN A PHYSICAL FLEXIBLE SHAFT ROTOR SYSTEM AS AN EXAMPLE OF A COMPLICATED MECHANICAL SYSTEM

Papageorgiou Pantelis-Panagiotis

Department of Naval Architecture & Marine Engineering, National Technical University of Athens

Supervisor: Ioannis T. Georgiou

Professor, Department of Naval Architecture & Marine Engineering, National Technical University of Athens

Abstract

Complex machinery contains critical areas, such as revolute joints and bearings, that are prone to damage initiation and growth. If not detected early, damage in complex local areas leads to premature failure. The complexity of an integrated system is a factor that limits developed classical methods from detecting early damage in complex local areas. A pure experimental data environment could provide solutions given the broad impact of machine learning. Here an interesting idea is introduced to support a machine-learning framework for damage detection in local critical areas. The vibration field developed in a local area surrounding a ball bearing support of a lab flexible shaft-rotor system was measured by a set of accelerometers to form a dataset environment. It was used as an experience for machine learning by a deep convolutional neural network adapted from the AlexNet Architecture. Our main result is casting a solid mechanics prediction problem into a classification problem and eventually computing a solution by a deep learning technique. Current technology innovations are improving computer speed, data storage media, and graphics processing units. These factors are turning machine learning techniques into state-of-the-art computation-and-prediction tools that can be automated to deal with large volumes of vibration data. Prediction-diagnosis of damage results in improved condition monitoring of complex mechanical systems and this in turn infers economic gains due to estimated low-cost maintenance. Classical condition monitoring techniques have serious limitations such as the inability in learning from datasets the dynamics properties onboard installed machinery units operating under varying environmental conditions.

Keywords: Vibration Signal Analysis, Sensor Classification, Deep Convolutional Neural Network.

Contents

List of Tables

List of Figures

List of Nomenclature

FFT	Fast Fourier Transformation
STFT	Short-Time Fourier Transformation
DFT	Discrete Fourier Transformation
IT	Information Technology
WT	Wavelet Transformation
ML	Machine Learning
AI	Artificial Intelligent
FEA	Finite Element Analysis
ANN	Artificial Neural Network
CNN	Convolution Neural Network
DCNN	Deep Convolution Neural Network
DCNNA	Deep Convolution Neural Network Architect
CMB	Ceramic Magnetic Bearing
RMS	Root Mean Square
PDF	Probability Density Function
CWT	Continuous Wavelet Transformation
SGB	Stochastic Gradient Descent
ReLU	Rectified Linear Units
RGB	Red Green Blue
CMYK	Cyan Magenta Yellow Key
SC	Scalogram
PNG	Portable network graphic
dpi	Dot per inches

Chapter 1. INTRODUCTION

1.1. Motivation

Rotating machines such as turbines, engines, propulsion plants, and energy conversion systems encounter in the worldwide industry. They consist of disks of various shapes, shafts with a circular cross-section that change shape depending on longitudinal position, and bearings situated at various positions to transmit power from one part to another. Their nonlinear coupling makes rotating machines quite a complicated system. During their operations, they become the main source of vibrations, hence they consist of areas prone to damage initiation and growth. A fighter plane in 1996 crashed due to a problem attributed to a bearing failure of the compressor region in one of the two engines [1, 2] before the estimated operating time for maintenance was reached. Past and new innovative ongoing research focus on vibration-based methods to pinpoint the cause root of failure. Vibration signal analysis is a process that monitors the levels and pattern of vibration signals within a component, machinery, and structure to detect abnormal vibrations and finally to evaluate the overall condition of the test subject.

In recent years, many researchers investigate various structures for analyzing, simplifying, and conceiving mathematical models capable to extract features. Georgiadis et al. [3] and Obuchowski et al. [4] calculated kurtosis and skewness of times series to extract features of ball bearing and gearbox correspondingly. Caesarendra et al. [5] extracted features to monitor conditions when a low-speed slew bearing degraded. The previous papers used basic and advanced methods based on the time and frequency domain of vibration signal analysis. Furthermore, Lin et al. [6] used only frequency-based methods to defect detection of rolling bearings. The most common frequency-based method is Fourier transform which is turned signals from the time domain into the frequency domain to extract undetected frequencies. One more advanced tool in vibration signal analysis is wavelet transformation. Rucka et al. [7] to detect failures applicate this tool to the experimental setup of a plexiglass cantilever beam and steel plate correspondingly. They produced plots representing the wavelet transformation across the length and width of test subjects.

Development of computers, such as increasing computational speed, storage of high volumes of data and so on, are motivations for using machine learning complex systems since they are produced nonlinear characteristics. Convolution neural networks in supervised learning are a powerful tool for image classification, which targets extracting features of the input dataset and then classifying them into possible categories. Hence, in current Thesis is investigated the performance of CNN instead of traditional vibration methods in extracting features and is focused on sensor signal classification of critical local areas of a complex system.

1.2. Introduction to Literature Review

It is well-known that the vibration dynamics of these complex systems provide exploitable information on their structural half and condition [8, 9]. Vibration monitoring systems are continuously being advanced due to mature sensors-and-actuators technologies as supported by data acquisition and communication technologies [10-14]. The tradeoff is the generation of large volumes of vibration data, hence bringing in the

issue of how the practicing engineer could extract anomaly features followed by learning. Dataset features such as min, max, mean values, root mean square, standard deviation, skewness, and kurtosis are extracted but practice reveals are not capable of describing reliably system health conditions [5], except for a few cases. Well-established as classical data processing techniques, both the Discrete Fourier (DFT) and Fast Fourier Transforms (FFT) come to barriers when it comes to detecting anomalies in non-stationary signals [15] as a result the Short-Time Fourier Transformation (STFT) and Winger-Ville distribution have aroused to enable a multi-resolution the in time-frequency domain. These methods have limitations in the presence of the well-known Heisenberg's uncertainty principle: an arbitrary signal cannot achieve high resolution in both domains [16-19]. These limitations have led to the discovery and development of the Wavelet Transform (WT) as a powerful data processing tool in the time-frequency domain. **Figure 1** shows the ascendancy of WT over STFT in the time-frequency plane. In WT the time interval is inversely proportional to the resolution of frequency information: low-frequency information pairs with long-time intervals whereas short-time-interval leads to high-frequency information. In STFT the window size of time-frequency is constant, as a result, time-and-frequency resolutions are inversely proportional. However, in the modern era of information technology (IT) accompanied by rapid advances, and where the norm now is the generation of large-volume datasets, these classical methods have very limited capabilities in the sense they cannot learn from datasets, so past experiences remain unexploited. The computation paradigm of machine learning (ML) bears great potential in machine health monitoring because experience from past data can lead to learning in the sense of how humans can become experts in performing some tasks.

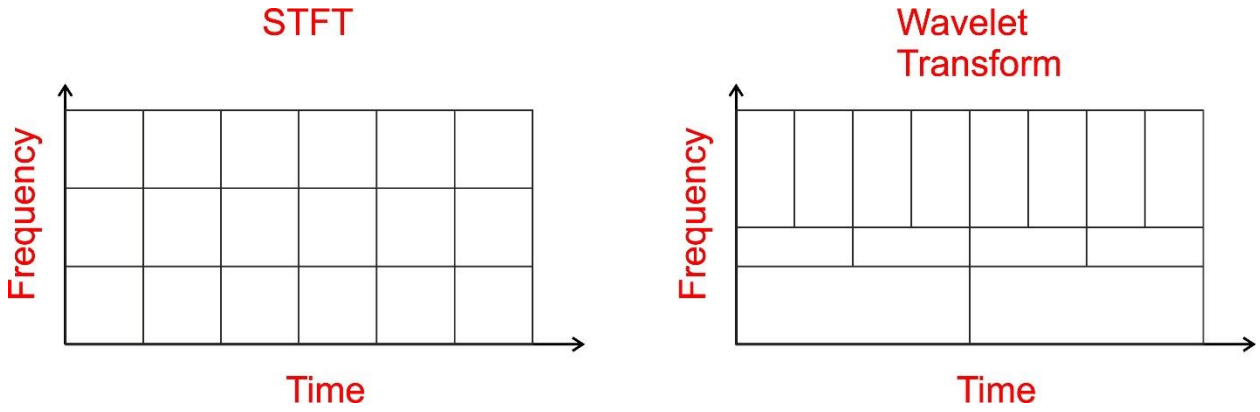


Figure 1: *Decomposition of Wavelet Transformation and Short-Time Fourier Transformation in time-frequency plane.*

In recent years, Artificial Intelligent (AI) via machine learning algorithms finds horizontal implementation in a variety of fields of science, such as Health, Materials, Finite Element Analysis (FEA), Chemistry, and so forth [20]. Machine learning is an algorithm with the ability to learn from datasets [21] by following supervised, semi-supervised, and unsupervised training techniques. Inspired by the way the human brain network of neurons processes information by its very specific and extremely efficient way of computation-not possible by the classical ways of computations in computers, an Artificial Neural network (ANN) [22] is a paradigm of machine learning algorithm with remarkable predictabilities. Deep Convolutional Neural Networks (DCNN) [22-25], Recurrent Neural

Networks (RNN) [26] and Long/Short Term Memory (LSTM) [27] are some well-researched subsets of ANN. A proven state-of-the-art algorithm, DCNN, mainly used in image processing, learns filters to extract features and classify them [25]. Industry applications reveal that the architectures of CNN algorithms AlexNet[23], GoogLeNet [28], and ResNet [29] are very promising in providing solutions to difficult problems. The great achievement of Krizhevsky et al. in ImageNet Large Scale Visual Recognition Challenger (ILSVRC 2010 & 2012), established DCNNA in the field of image classification [23].

1.3. Master Thesis Organization

The rest Thesis is organized as follows:

Chapter 2 contains the theoretical background needed to understand some basic principles of ball bearing and monitoring devices. In addition, this chapter represents basic information about the machine learning algorithm and describes in depth the expressions that the AlexNet algorithm occurs.

Chapter 3 describes the experimental setup and the conversion of the image data format to the input dataset and concludes with the AlexNet algorithm.

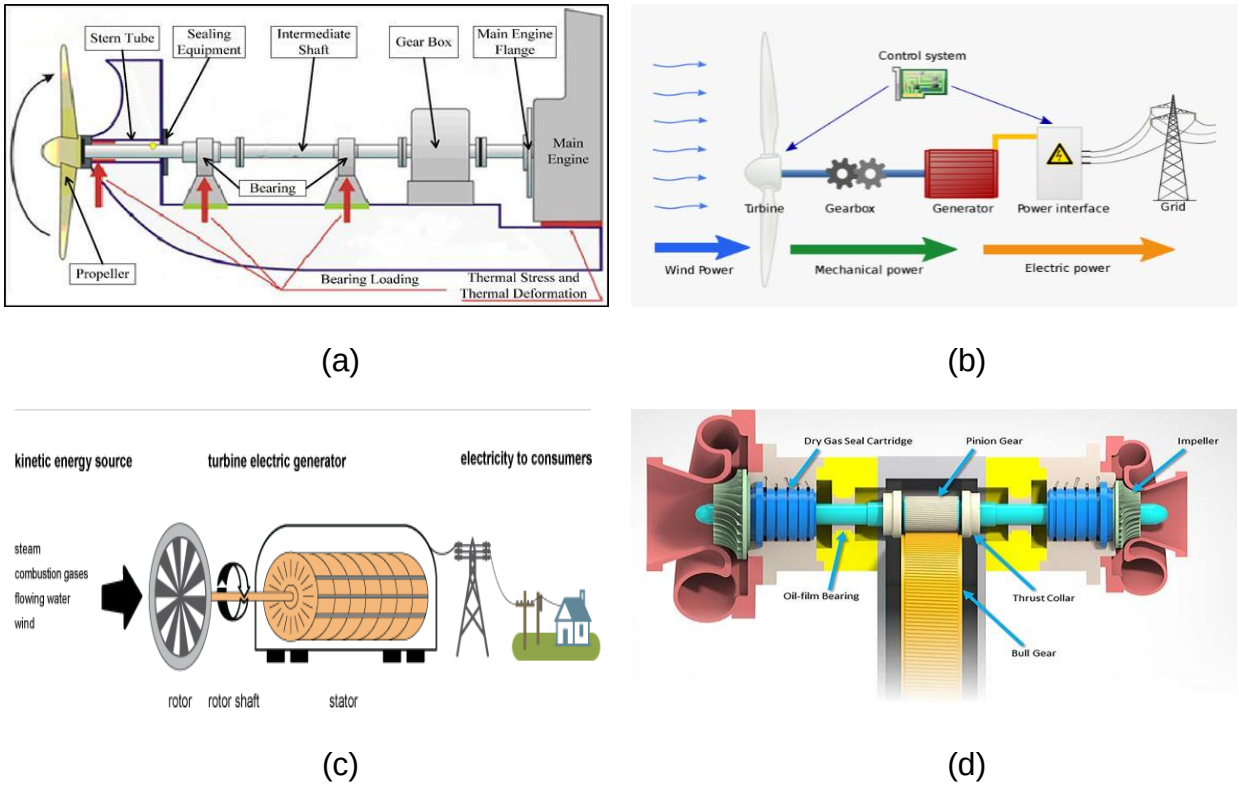
Follows Chapter 4 in which the results of the Thesis are represented and discussed.

Chapter 6 is mentioned the conclusion of the Thesis and is reported future research on how machine learning may give solutions to various mechanical problems.

Chapter 2. THEORETICAL BACKGROUND

2.1. Rotating Machines

Rotating systems in mechanic machines such as internal combustion engines, propulsion systems of ships, the gearbox of wind turbines, the power turbine generation and turbomachinery of hydroelectric power stations provide the backbone in numerous fields of the industry since they are used to transmit power from one part to another.



(a) (b)

(c) (d)

Figure 2: *(a) Propulsion system of ship, (b) Wind turbine, (c) Electric generator turbine, (d) turbomachinery.*

Operations like ship movement [30], rotation of blades of wind turbine [31] compression air in supercharger of the airplane, and rotation of camshaft of a vehicle are performed in critical areas and they are prone to damage resulting in harmful consequences. For example, an unexpected shutdown of a production line in a factory or the immobility of a ship leads to loss of money, as well as the endangerment of operation personnel. According to Tauqir et al., a fighter crashed due to damage to the ball bearing [1]. The defect was detected on the failure the of ceramic magnetic bearing (CMB) of the compressor region, even though it had been suspected 5 hours before the crash. Similar failures of ball bearings have been detected by Ejaz et al. [2] and Simon et al. [32] in the aero engine and turboshaft engine correspondingly.

Bearings are elements of rotating systems that depending on contact are classified into rolling-elements bearings and plain bearings. As depicted in **Figure 3** bearing consists of outer the ring, inner ring, metal shield, rolling elements (ball), cage, and metal

shield. According to the rolling element, this type of bearing is called a ball bearing. During their operation, the passage of rolling elements, natural vibrations of outer rings, geometrical imperfections, cage noise, and flaws noise are the main causes for producing vibration and noise. One more feature of ball bearings is produced nonlinear phenomena which are occurred by clearance and contact stiffness [8]. Due to the complexity of the system have been developed monitor devices such as sensors have to measure system dynamic conditions in the level of acceleration.

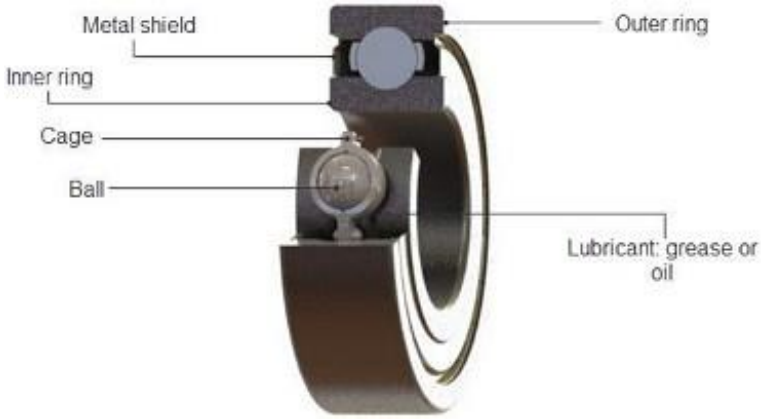


Figure 3: *Cutaway example of rolling-ball bearing*

2.2. Piezoelectric Sensor

A sensor is a device that detects variables of a physical system, such as displacement, temperature, elasticity, acceleration, and so on. According to their operation principles, they are categorized into classes [13]. In the field of vibration dynamics, the piezoelectric sensor has got wide implementation. Piezo is from the Greek word “πιέξειν” meaning to squeeze. In the piezoelectricity phenomenon, an electric charge is produced when a force is acting on the tested subject to deform it.

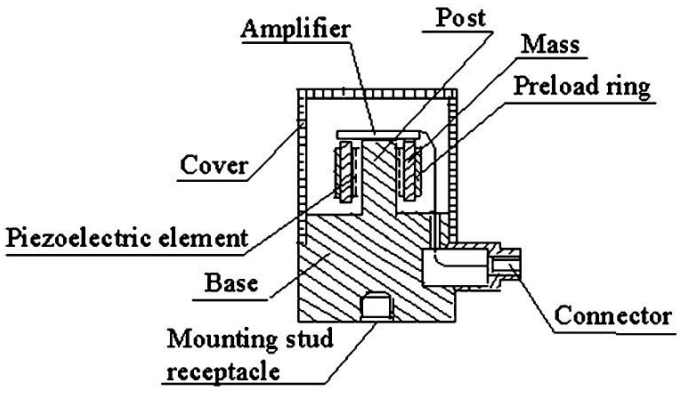


Figure 4: *Piezoelectric Sensor.*

Figure 4 presents a cross-section of piezoelectric sensors with the seismic mass and the piezoelectric material to constitute the main sensing elements. When motion is

subjected to the sensor casing an inertial force is generated and exerted on the seismic mass with the interacting coupled piezoelectric material. According to Second Newton's Law the generated force (N) is equal to the product of seismic mass (kg) and acceleration (m/s^2). The behavior of piezoelectric material can be approached as a capacitor. The applied stress (N/mm^2) and the generated force (N) are proportional to charge per unit area (C/ mm^2) and electrical charge (C) correspondingly. So, the output signal that is recorded via the connector is a voltage. The acceleration mass (a) is exposed to the same acceleration magnitude as the test object, over a wide frequency range, and so on the sensor measure test vibration dynamics [10-14].

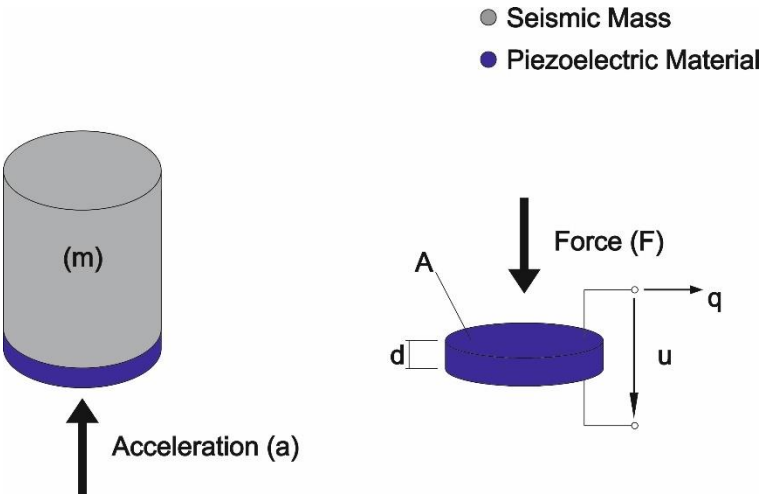


Figure 5: *Piezoelectric effect.*

Piezoelectric sensors are classified into 3 types regarding detecting system variables. These are force, pressure, and acceleration. Their cross-section is represented in **Figure 6**. In the first one, the wire connects to the electrodes of the crystal and the acting force deform the crystal. In the compression type between two back-to-back crystals, there is a thin metal membrane. The acting force of the mass is applied to the membrane and then it deforms the two crystals. In the last type of piezoelectric sensor, 4 crystals are mounted on rectangular posts and their deformation is happened immediately by the applied force of the seismic mass [12].

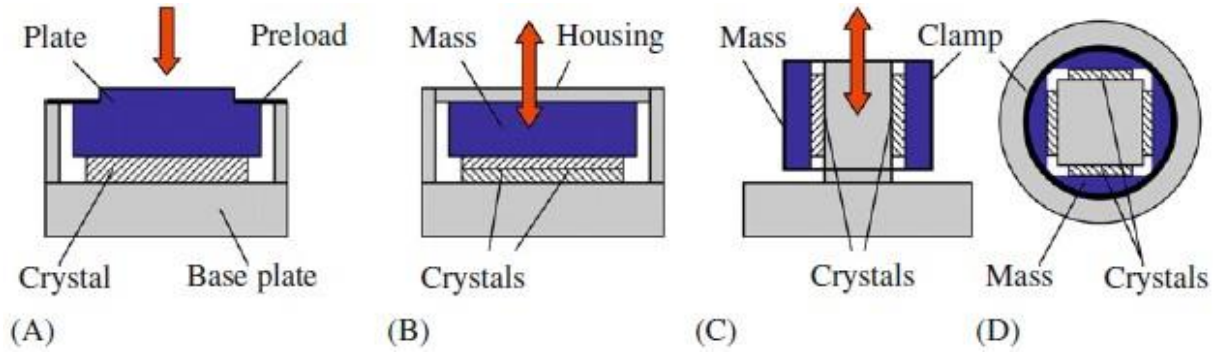


Figure 6: *Principle of piezoelectric sensors (A) Force sensor (longitudinal), (B) Compression type accelerometer, (C) Shear – type accelerometer, (D) top view of (C). [12]*

2.3. Vibration signal analysis

All the collected acceleration points from the sensors provide exploitable information about the system's dynamic condition. Vibration signal analysis as mentioned before is the process that uses the acceleration data to extract features such as health operation conditions, unbalance, misalignment, mechanical looseness, etc. Initially, researchers focused on a statistical analysis of signals in the time domain. Due to the limitation in monitoring the dynamic condition of the system, they drove the conversion of the signal in the frequency domain to extract features with great results.

2.3.1. Time-Domain

In the time-based domain extracting features is based on estimating the statical values of signals. The plot of signals in time order is called times-series. The most common statistical features that are used to identify the differences between time series are average, root means square (RMS), standard deviation, and variance. These basic values examine the probability density function and are tightly coupled with the system condition. Specifically, skewness identifies the asymmetry of PDF as well as kurtosis quantifies the peak value of PDF. The value of kurtosis and skewness for a signal of a health bearing in a normal distribution is approximately t and 0 correspondingly [33]. Other features such as entropy measure the randomness and uncertainty of data and provide noticeable information about systems' conditions.

Apart from the statistical features, have been developed non-dimensional features that based on the system mean and deviation values. These are the shape factor and crest factor which are affected by the system condition as well. The whole statistical-based features (whose formulas are represented in **Table 1**) may describe the dynamic condition of the bearing/system but they presented limitations in an advanced investigation that provides the following frequency domain.

Feature Name	Formula
RMS	$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$
Variance	$Var = \frac{\sum_{i=1}^{N}(x_i - m)^2}{(N-1)\sigma^2}$
Skewness	$Sk = \frac{\sum_{i=1}^{N}(x_i - m)^3}{(N-1)\sigma^3}$
Kurtosis	$Var = \frac{\sum_{i=1}^{N}(x_i - m)^4}{(N-1)\sigma^4}$
Shape Factor	$SF = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert}$
Crest Factor	$CF = \frac{\max\lvert x_i\rvert}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}}$
Entropy	$e(p) = -\sum_{i=1}^{N} p(z_i)\log_2 p(z_i)$

Table 1: *Features formula.*

2.3.2. Frequency Domain

In the frequency domain, Fourier transformation has been mentioned as a great success. It was introduced by Joseph Furrier in 1822. He focused to solve the heat equation (or Fourier equation) to determine the solution of the Fourier series of partial differential equations in specific boundary conditions [19]. The solution is defined as,

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \quad (1)$$

The equation is a fundamental method in vibration signal analysis and machine health monitoring. He defined the transformation of a signal from the time-based domain f(t) to the frequency-based domain F(ω). Namely, in each frequency (ω) the integral measure the oscillations of function f. The F(ω) is called the frequency spectrum of a signal or waveform f(t) [15]. According to “Clarence” in chapter 10.1 mentioned that the great success of FT is owing to the conversion of time domain differential operations into algebraic operations [34]. Despite noticeable results, the Fourier transform seems to have limitations in further signal analysis. The two main reasons are the elimination to reflect in changes because it does not contain local information to represent the signal simultaneously in the time and frequency domain [19].

2.3.3. Time-Frequency Domain

The Fourier transformations constitute the base for new extensions, such as Short-Time Fourier Transformation and Wigner-Ville Distribution which have aroused to enable a multi-resolution in the time-frequency domain. Short-Time Fourier Transform, due to low computational complexity is used to analyze nonstationary signals as a pre-processing tool, which is defined as:

$$STFT(f,t) = \int_{t-T/2}^{t+T/2} w(t-\tau) x(\tau) e^{i2\pi f t} d\tau \quad (2)$$

As aforementioned, the most powerful tool of non-stationary signal with the capability of multi-resolution in both domains is wavelet transformation. It is about a relevant young method that appeared in the 20th century. The method is invented by Jean Morlet in 1982, according to Debnath [19]. This new mathematician tool finds wide application in the analysis of temporary phenomena, such as earthquakes, vibration signal analysis, pattern recognition [17], and so on. It is a wavelike function that oscillates in finite time. Wavelet is a set of family functions constructed from translations and dilations of a single function, called mother wavelet or (affine coherent states) ψ(t). It is defined as:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right), \quad a,b \in R,\ a \neq 0 \quad (3)$$

Term a and b denote respectively the scaling and translation parameters, and t is the time.

The continuous wavelet transformation (CWT) measures the local similarity of a signal and wavelet function using their inner product. Specifically, it compares the signal with shifted and compressed or stretched version of the mother wavelet. The CWT is defined by:

$$C(a,b) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{a}} \psi^* \left(\frac{t-b}{a} \right) \quad (4)$$

When the wavelet is being compressed (small scale a) the CWT is rapidly changing, thus resulting in fine details in signals of high frequency changing with time. On the other hand, the stretched wavelet extracts the low-frequency details of the signal. Hence there is a clear relationship between the scale and frequency of wavelet and signal respectively [15, 19, 34-36]. Despite these exploited features extracted by vibration signal analysis, the needs pushed them to implement new tools. Due to the great capabilities of computers that occur in storing and processing large volume datasets, humans have been led to use machine learning for monitoring.

2.4. Machine Learning

2.4.1. s Basics

Artificial intelligence is a well-known branch of computer science that the target to simulate or even recreate the capabilities of the human mind. Machine learning, a specific task of AI aims in training a machine how to learn. It is based on the human mind principle to tackle task T, perform measure P and then learn and improve from experience [21]. Due to the limitation of existing programs to solve difficult problems it has board implementation. The task is not defined as the whole project but only its process. An insightful example mentioned by Goodfellow et. al. is that the task to program a robot to be able to walk is walking [26]. The most common machine learning is classification (even though with missing points), regression, speech recognition, transcription, machine translation, structured output, anomaly detection, synthesis & sampling, imputation of missing values, denoising, density estimation, or probability mass function estimation. Those algorithms perform in phone and computer applications such as Facebook, Instagram, Google Maps, and so on. They use AI and even ML algorithms on data aggregated from user interaction to examine their behavior. Siri by Apple Inc., Google Now by Google, and Cortana by Microsoft use ML algorithms such as speech interpretation and recognition interfaces that make them intelligent personal assistants. Last but not least, another promising field of the market with the implementation of ML algorithm are self-driving vehicles [37].

Performance measures how well the machine learning algorithm performs on input. To evaluate it, design a quantitative measure. Because there are many types of tasks, measure P differs. For example, in classification and transcription, the most common measure is accuracy, which is the proportion of inputs for which the model predicts correct output. Another measure is the error rate, that range from 0-1, and in contrast to accuracy, it is the proportion of input for which the algorithm predicts an incorrect output. In density estimation, the previous performance metrics do not make sense. Indeed, the most proper

measure is a continuous-valued score for each input such as the average log-probability. In other cases, it is unachievable to measure the performance of the algorithm, consequently, an alternative criterion must be designed that corresponds to the inputs. All the measure types are used to improve the experience of the algorithm and finally to learn by its input. Based on experience, algorithms are classified into broad categories such as supervised, semi-supervised and unsupervised. The first one uses it for the task that is associated with a label or target and trains the algorithm to predict the class of each input data point. In an unsupervised algorithm, the data set contain various features, hence the algorithm is trained to predict in terms of the probability distribution of the input set.

As mentioned before the main implementation of the machine learning algorithm is to receive input files, process them via mathematical expressions, and finally predict an output. The evaluation of how well one algorithm performs is to give good predictions in unseen input files. This ability is called generalization. In a supervised machine learning algorithm, the pre-processing of categorization of input files in a train, validate, and test set is necessary.

A machine learning algorithm may be confused with optimization problems, in which one tries to maximize or eliminate an objective function. The algorithms of machine learning are separate from optimization problems because they want to eliminate the training and generalization (test) errors as well. The central challenge that makes an algorithm efficient is defined as:

- Make the training error small. (Underfitting)
- Make the gap between training and generalization errors small. (Overfitting).

One model is underfitting when it performs poorly on training data and it is unable to capture the relationship between the input examples and target values. One model is overfitting when it performs well in training data but does not perform well on validate data. It is because the algorithm is memorizing the data it has seen and is unable to generalize to unseen examples [38]. The behavior of algorithms is controlled by settings called hyperparameters. They cannot be adapted by the learning algorithm itself. They constitute convolution layers, neural networks hidden units, and so on which are analyzed, in-depth in the subsequence section.

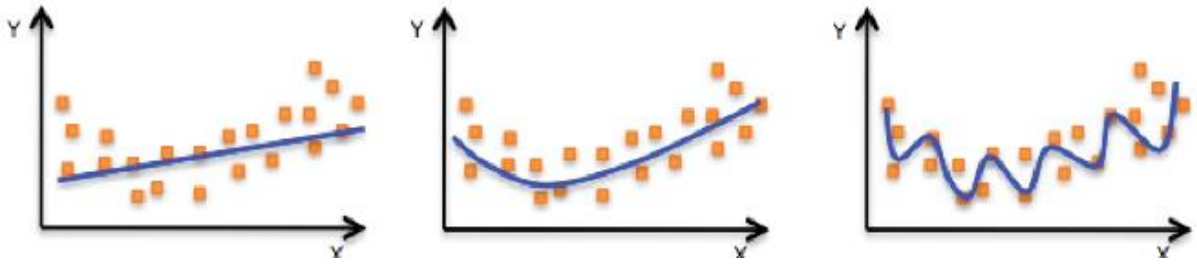


Figure 7: *Underfitting, Balance fitting, Overfitting.*

2.4.2. Deep Convolution Neural Network

Convolutional Neural Network (CNN) is a well-known deep learning algorithm formed architectures of basic computational, like biological neurons and nodal units. Inspired by neuroscientific principles such as the way human vision "sensors" are inputting information into the human brain, it performs as a state-of-art tool in image classification with a known grid-like topology [26]. Their main purpose is to approximate a function "f" that maps input data "x" to category data "y" by learning parameter "θ". Parameters "θ" are estimated by trial and error during the flow of data through the layers of the network. Because the information has no feedback connection the network is called feedforward. **Figure 8** depicts the architectural structure of a 2-D CNN. The algorithm is a series of mathematical expressions occurring in convolution layers with hidden layers and pooling layers, followed by a neural network composed of an input layer, a hidden layer, and the final layer which is called the output layer. A network is determined by chain functions in form $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$. The $f^{(1)}$ is called the first layer of the network, $f^{(2)}$ is called the second layer, and so on. The name "deep" arose from the overall length of these chain structures. Learning involves the minimization of a cost function. Minimization provides estimates for the learned parameters. The whole CNN can be thought of as a paradigm of computations of a certain character on input data to be matched to a fixed output. Below is described input data, convolution layers, hidden layers, pooling layers, neural layers, loss function, cost function, and finally represents how the algorithm gain experience via optimization.

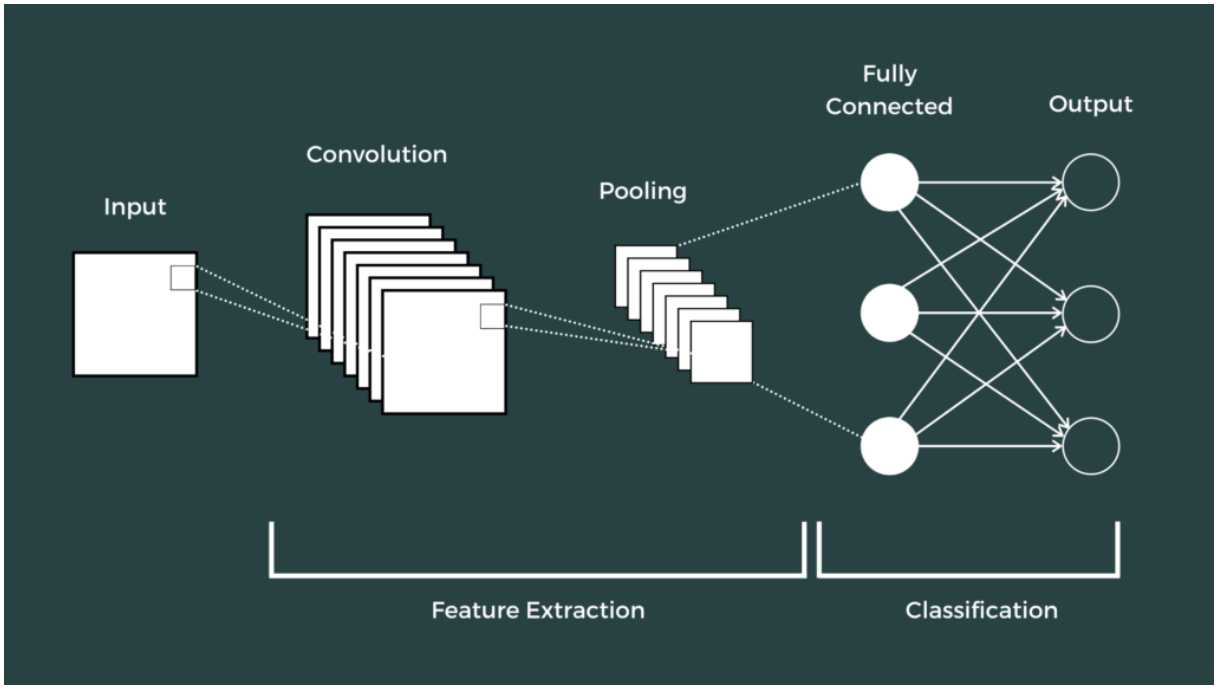


***Figure 8**: Basic architect of Deep Convolution Neural Network.*

2.4.2.1. Image format

Convolution neural networks as foregoing discussed achieved great success in processing data within a known grid-like topology such as an image [26]. An image is a composition of various pixels, and this in turn represents the color density of the current model. The most common model used worldwide to display an image is RGB (Red, Green, Blue), CMYK (Cyan, Magenta, Yellow, Key/black), and HSV (for Hue, Saturation, and Lightness). RGB, as it is used in the current Thesis, is an additive-based space, relating to trichromatic theory according to Allen et. al. [39]. The pixel ranges from 0 to 255, for example, a pixel displayed in the portal as 100% blue, in the RGB palette is expressed as [0,0,255]. **Figure 9** depicts the conversion of image format data to input array for CNN. A color image is mathematically expressed by a 3D array with a block size equal to 3 (RGB).

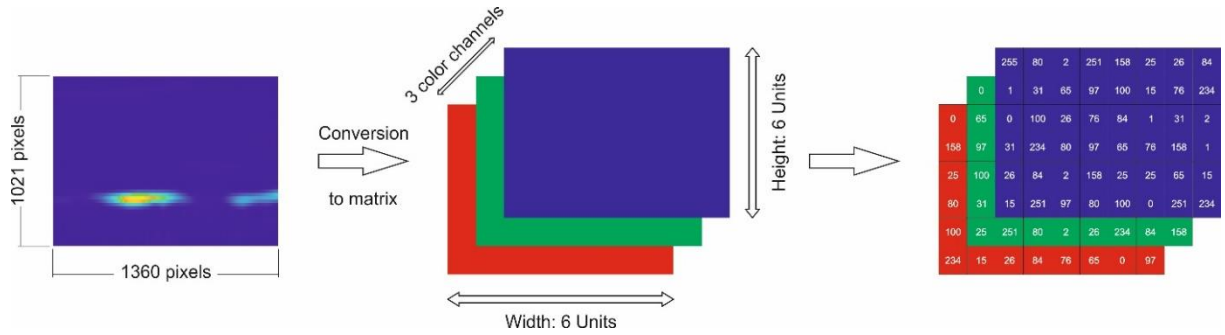


Figure 9: 1st layer of CNN (The array values are indicative).

The input image is convoluted against a fixed filter (kernel) according to the convolution math operation:

$$c(x)_k = (x * K)_k(i, j) + b_k = \sum_m \sum_n x(i+m, j+n) K_k(m,n) + b_k \qquad (5)$$

Where K_k is a set of filters which is also called the weight parameter, b_k provides the bias parameter, and c_k is called the feature map of the k^{th} layer in the set of convolutions. The motivation of convolution implements to extract features from images in a neural network is owing to three advantages. The first one is sparse interactions which are achieved by using a filter array smaller than the input and this in turn infers algorithms in better performance and simultaneously reduce its store capacity (see. **Figure 10**). For example, an image is composed of millions of pixels, but some features such as edges can be detected by using kernels that contain tens or hundreds of pixels.

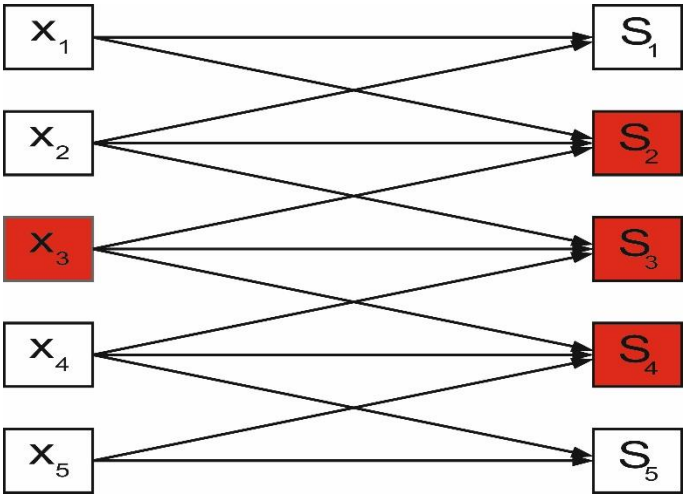


Figure 10: *When a convolution expression is occurred between kernel and input 3 are affected only three outputs.*

Another advantage of Convolution is the fact of parameter sharing. It means that in each layer the multiple of input image data occurred with the same filter array (see. eq. 8). It keeps store capacity at a low value and simultaneously does not reduce its statistical efficiency. The last leverage that makes the algorithm mark great success is equivariant representations. In math, this means that if an input change, the output change in the same way. For example, assume that an event appeared in dt time at time series. The same event appeared in output at dt. A disadvantage that occurred in the CNN algorithm is the fact that they cannot recognize image changes such as image rotation and image scale, because two rotated or scaled images provide unsimilar input arrays.

Between convolution and pooling, math expressions occurred in activation functions to detect nonlinear features. These layers are called hidden layers or detector stages. Even though in recent years has been developed many hidden layers, is an extremely active area for more research. The most common activation functions are logistic sigmoid, hyperbolic tangent, radial basis function, soft plus, hard tanh and rectified linear unit (ReLU)[40]. In each hidden layer of AlexNet [23] architect is used rectified linear unit function, which mathematically expression is given by:

$$\mathrm{Re}\,LU(c(x)) = \max(0, c(x)) \tag{6}$$

Eq. 7 outputs values only in the active units while it outputs zero across the rest domain. The derivative of active units is equal to 1. Thus, the gradient direction in the detector stage is free from second-order effects and this in turn infers great performance in CNN algorithms, increasing their resistance to forgetting how to perform in already training tasks [22, 26, 41]. The research team that developed the AlexNet algorithm observed that the implementation of the normalization function after the produced nonlinearity of the ReLU function increased generalizations [23]. The most common is Batch Normalization defined as follows:

$$x = \frac{x - \mu}{\sigma} \tag{7}$$

Where μ is the mean value of x and σ denotes the standard deviation.

The output of convolutions expressions often produced noise. Most modern image classification models, like AlexNet[23], and GoogLeNet[28] extract features by using max-pooling functions whereas combining the outputs of the ReLU function in a local array of features. Thus, the algorithm emphasizes task-related information [42]. **Figure 11** shows how an indicative input image array flows through one convolution layer. The input image pads with zero for the convolution expressions do not modify the size of the array.

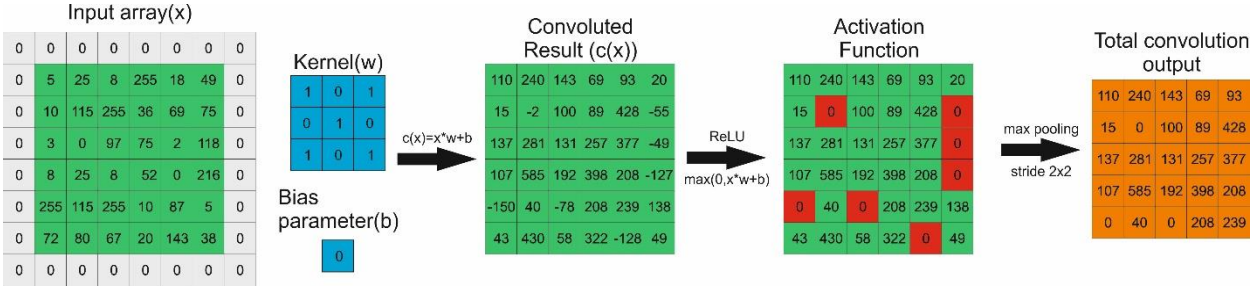


Figure 11: *Process of extract feature in single convolution layer.*

As the image data input flows through all convolution layers the feature map is flattened into a vector and is fed to the input layer of the neural network. Similarly with convolution operation, in each neural network, the function that occurred is defined as:

$$y_n = x * w_n + b_n \tag{8}$$

where w_n is a row vector that included the weight parameters, and b_n is the respective bias parameters of the n^{th} layer in the set of neural network expressions. Like the convolution model, the activation function ReLU is applied as a loss function in hidden layers. In addition, a commonly used function applied after the ReLU activation is called the dropout layer. According to Srivastava et al. [43], dropout computation is a technique that prevents an algorithm from overfitting and improves its performance on supervised learning tasks. The last layer of CNN is called the output layer and the common function applied is called the cost function. The choice of the cost function is crucial for the algorithm because it determines the representation of output units.

The last layer of the algorithm is called the output layer. As aforementioned, the main goal of the current paper is to predict the locations of the sensors whose signals are used as input wavelet images. This means that the output unit should be a probability distribution over the 8 sensor locations (classes). The function, used in several CNN architectures as well, that gives with great success this distribution is called the SoftMax function. It is defined as follows:

$$\text{SoftMax(y)} = \begin{bmatrix} p(y=1) \\ p(y=2) \\ \dots \\ p(y=n) \end{bmatrix} = \frac{\exp(y_i)}{\sum_{j=1}^{n} \exp(y_j)} \quad (9)$$

The fact that the CNN algorithm is associated with the human brain computation functions is owing to the SoftMax and in general the cost function's property. Since they calculate a probability distribution over "n" possible states their sum is equal to 1. Hence, when the values of 1 class are increased the rest values are decreased. This process is believed that occurs similarly in a human brain at the infinitesimal time [26].

Having described in depth the most crucial CNN functions that occurred in the classification problem of supervised learning, it is appropriate to discuss briefly how the learning process is taking place in the flow of information in the CNN algorithm. As mentioned by Krizhevsky et al. [23] the learning is based on the Stochastic Gradient Descent (SGD) computational process. It is an optimization algorithm that differs from the traditional optimizers because optimizes all the parameters *"θ",* allocated at different parts in the graphical schematic of the algorithm. Term parameters *"θ"* denote the weights and biases calculated per convolution layer, Eq. 1, and neural layer, Eq. 4. The SGD algorithm optimizes the error quantity defined by subtracting the quantity ***ε*g*** from the estimated old parameters (w, b), where "*g*" is the partial derivative of cost function SoftMax and "ε" provides the hyperparameter learning rate. A single running time of the algorithm, in which *"k=1, 2, …"* denotes the layer of convolution and neural network, is represented as follows:

Required="ε"

Required=parameters "θ"

Insert=Input image(multidimensional array)

Flow throught layers and then predict output

Estimate $\text{g}(\text{w}_k) = \frac{\partial}{\partial w_{k=1,\dots,m}} (SoftMax(y_k))$ and $\text{g}(\text{b}_k) = \frac{\partial}{\partial b_{k=1,\dots,m}} (SoftMax(y_k))$

Apply update: $w_k^{new} = w_k^{old} - \varepsilon * \text{g}(\text{w}_k)$ and $b_k^{new} = b_k^{old} - \varepsilon * \text{g}(\text{b}_k)$

Various CNN architectures developed for image classification use the above functions with success over a wide implementation of applications. Specific guidelines for the best approximation of function *"f"* have not been found as of yet. Therefore, the network architecture is adapted to the specifications of the case problem tackled.

Chapter 3. EXPERIMENT & METHODOLOGY

3.1. Experiment

The experiment focuses on collecting datasets using sensors distributed to cover a local area, considered critical. In complex mechanical systems, such an area hosts nonlinear dynamics with dissipation within crucial complex mechanical parts in machinery, ball bearings, for example, among others. **Figure 12** depicts the ball-bearing area of a flexible shaft bearing a large disk rotor at its center. While forced to rotational motion at constant angular velocity by an electric motor, vibration data are collected simultaneously using eight state-of-the-art piezoelectric accelerometers distributed properly to cover the local ball bearing. The vibration dynamics were sampled at 48 kHz and a resolution of 18 bits. The sensitivity and frequency range of piezoelectric accelerometers is about 1.00 mV/ (m/s^2) and 1 Hz-104 Hz respectively. Preprocessing reveals that the acquired signals are of very high quality regarding the level of signal-to-noise. A large dataset was collected with the main variable the value of the rotational speed. Several datasets were evaluated using advanced proper orthogonal decomposition tools to explore the reduced dynamics [44].

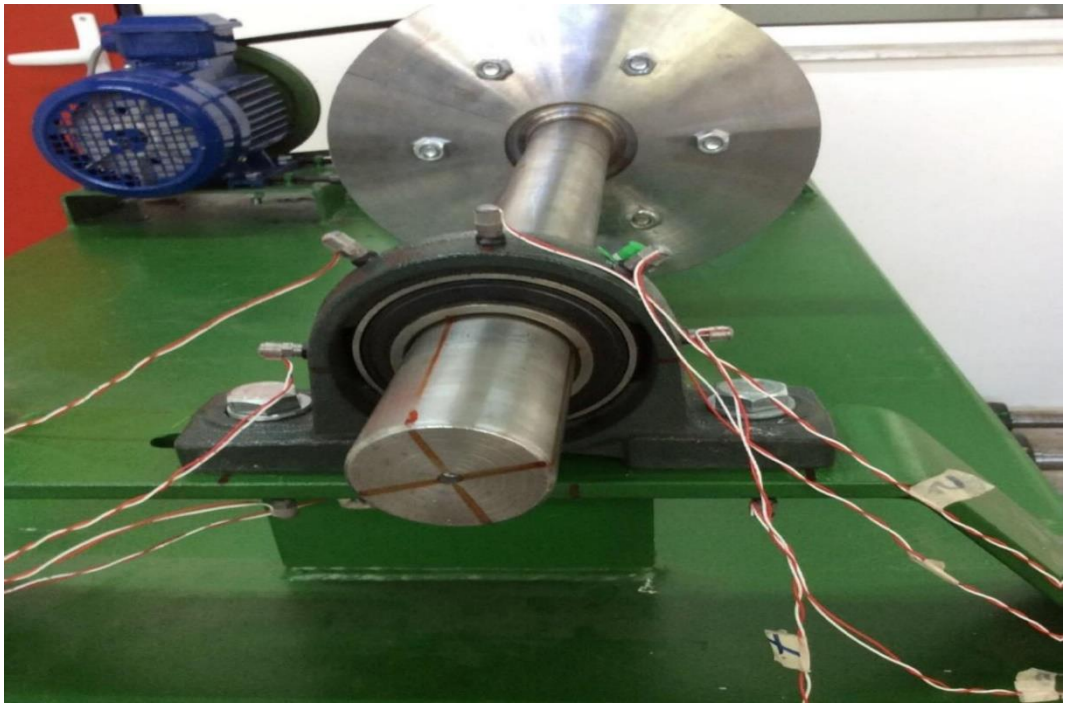


Figure 12: *Laboratory set-up of a flexible shaft-rotor system. Eight (8) sensors are arranged over a curve surrounding one of the local bearing areas. The arrangement is nearly symmetric about the vertical ball bearing axis of symmetry.*

For the involved computations and DCNNA predictions MATLAB [45] and Python [46], programming languages were used. In each software, libraries such as Wavelet Toolbox [47], Numpy [48], Tensorflow [49], Scikit-learn [50], Pandas [51], Matplotlib [52], OpenCV [53] were used.

3.2. Method

In foregoing sections citing that the current paper aims to explore the merits of how predictive could be a trained machine learning algorithm for predicting from datasets the location of the sensor that detected the acceleration signal. We address this very basic problem as a classification problem via the concept of learning from the dataset by a convolutional neural network. The classification set spans the space of locations of the sensors.

3.2.1. Cwt – Morse Wavelet

Through the programming language MATLAB and the wavelet toolbox, acceleration measurements are transformed over the frequency-time domain. The kind of wavelets that are used in MATLAB math software environment is continuous with Morse mother wavelet basis. The Fourier transform of the generalized Morse wavelet [54] is:

$$\psi_{\beta,\gamma}(\omega) = U(\omega)\alpha_{\beta,\gamma}\omega^{\beta}e^{-\omega^{\gamma}} \tag{10}$$

Parameter α β, γ is the normalizing constant, and U(ω) is the unit step function whereas ω is the frequency. Parameter γ characterizes the symmetry of the wavelet and b is a decay or compactness parameter.

3.2.2. Scalogram

To plot images that will be used as input files for DCNN, scalograms (SC) are computed in MATLAB. SC plots the collected signal and the scaled wavelet in a time plot correlation. The plot represents the percentage of energy for each coefficient, defined as:

$$S = |coefs * coefs| \tag{11}$$

$$SC = \frac{100S}{\sum_{i=1}^{n} S_i} \tag{12}$$

The wavelet transformation and scalogram are thus efficient tools in signal analysis because they could reveal some hidden features of the data not detected by another classical method [15]. **Figure 13** represents an original signal in the time domain and **Figure 14** shows the transformation of this signal in the frequency-time domain. In addition, **Figure 15** presents a 3D plot of the SC to clarify how the percentage of energy is represented.

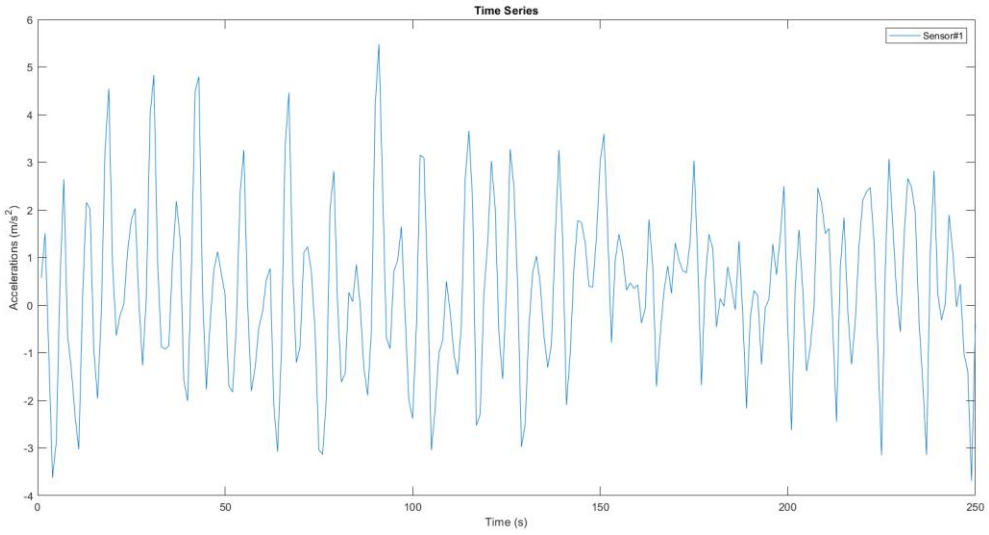


Figure 13: *Time Serie of 1st senor for 250 data points.*

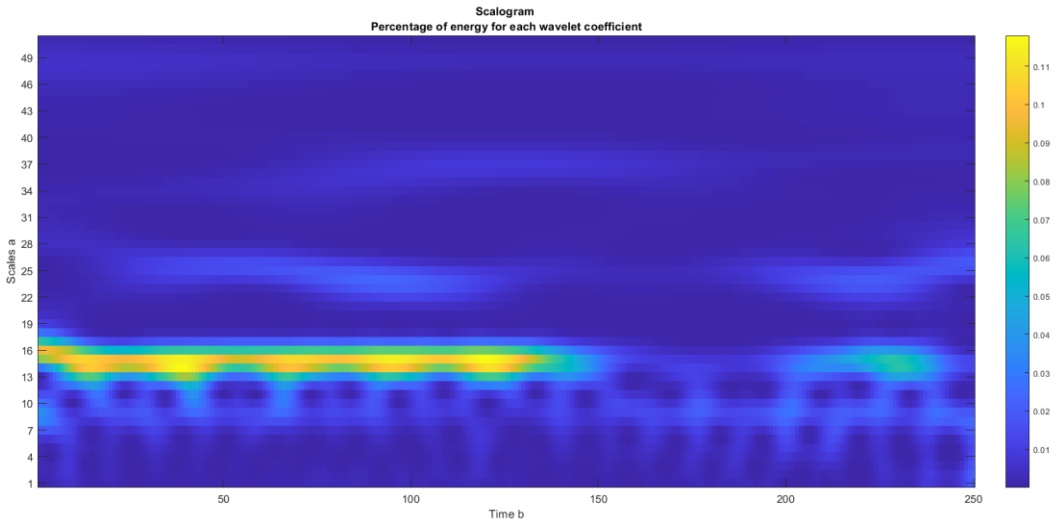


Figure 14: *2D scalogram plot of 1st sensor for 250 data points.*

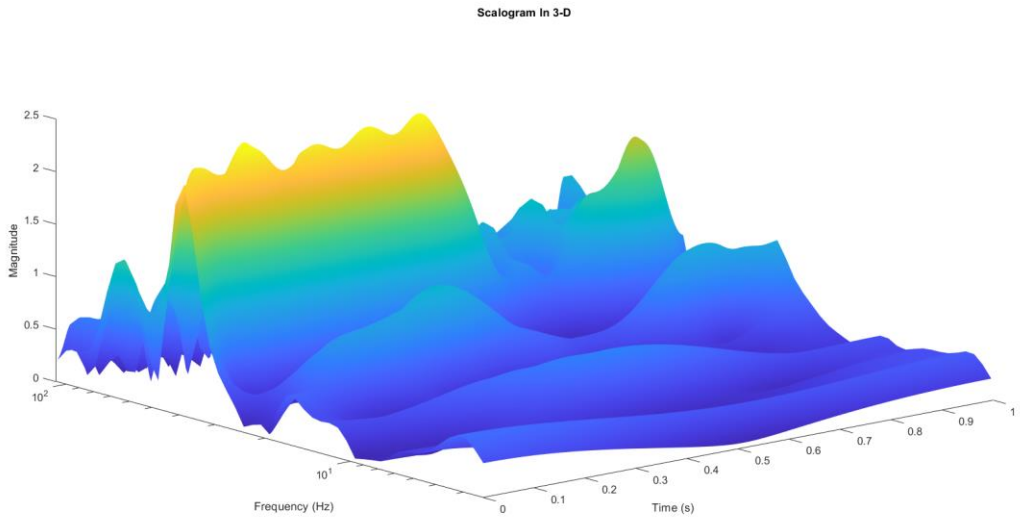


Figure 15: *Scalogram 3D plot of 1st sensor for 250 data points*

To use the image depicted in **Figure 14** as an input file into the AlexNet Architecture [23], we turned off all the marks such as axis, grid, label, and colormap to avoid redundant noise and finally saved it in portable graphic (png) format with 75 dpi (dots per inch) resolution [39].

3.2.3. Convolutional Neural Network - AlexNet

The computer facility running DCNN via the programming language python has the following specifications:

- Central Processing Unit (CPU): Intel® Core™ i7-11370H, (3.3GHz, 4 Cores)
- Graphic Processing Unit (GPU): Nvidia GeForce MX 450 (Core speed: 1395 MHz, Memory speed:10002 MHz, Max amount of memory: 2048 MB)
- Random Access (RAM): 16 GB – DDR4
- Operating System (OS): Windows 11 64-bit

As mentioned before, AlexNet architecture is used to extract features for the classification of sensor acceleration signals acquired in the local area of a ball bearing by a proper distribution of sensors, see **Figure 16**. This kind of DCNN has achieved remarkable performance in AI contests in comparison to existing and tested ML [26].

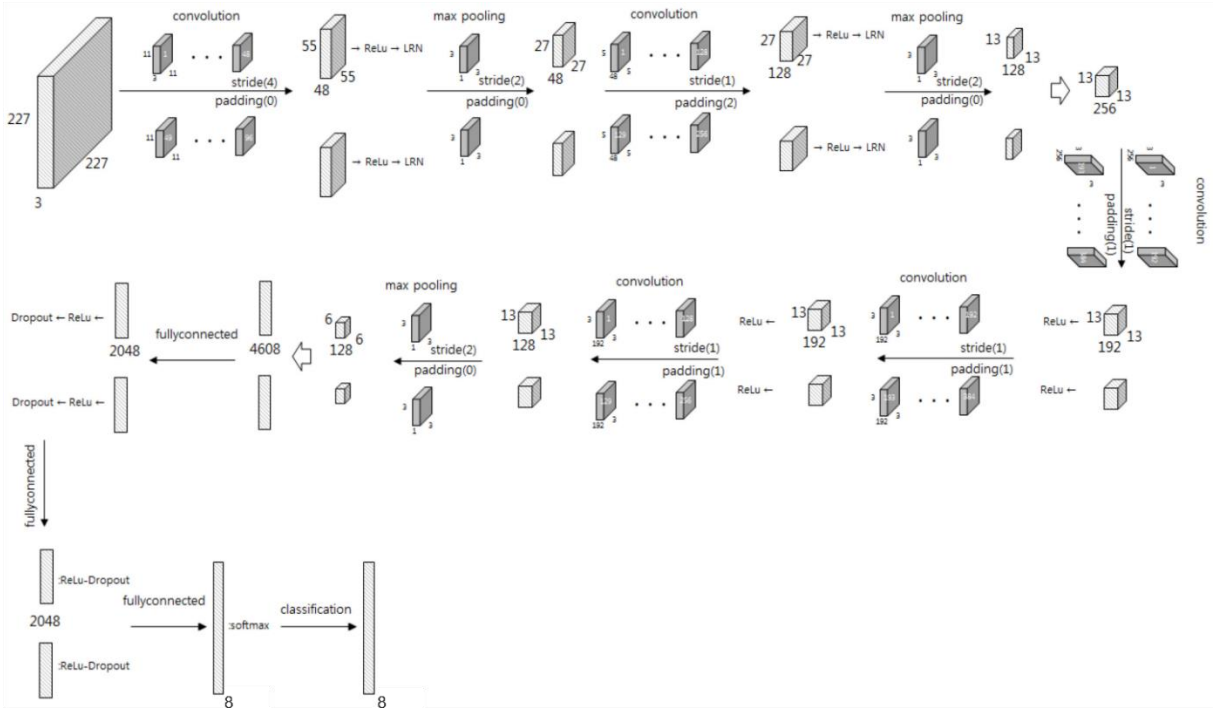


Figure 16: *The AlexNet Architecture [28]. The original architecture is designed to work in 2 parallel GPUs and in the last FC consisted of 1000 nodes. The representing architecture is adjustable to specific research needs.*

Algorithm DCNNA consists of five convolution layers, three pooling layers, two fully connected layers, and one output layer. Scalograms before entering as input files into the 1st convolution must be preprocessed. Specifically, the size of a scalogram in png format with 75 dpi resolution is 343x258 pixels. They are compressed in images with 227x227 pixels. The color model of the input image is RGB (red, green, blue), additive-based spaces, relating to trichromatic theory according to Allen et. all. [39] The input image thus is a 3D matrix of dimension 227x227x3. The intensity of each color ranges in value in the interval numeric [0,255] i.e., when a random pixel has the value [0,0,255] it is displayed as a 100% blue pixel in the computer display portal. Then the inputs passed in the 1st convolution layer with 96 filters of size 11x11x3 and a 4x4 stride. The output of this layer is 96 matrixes each of dimension 55x55. After that, the function ReLU acted on neurons and they normalized. The 2nd convolution layer takes as input all the processing neurons and filters it with a 256 kernel of size 5x5x48. The other 3 convolution layers are connected without intermediate pooling or normalization layers. They filter the input with 384 kernels of size 3x3x256, 384 kernels of size 3x3x192, and 256 kernels of size 3x3x192. The output of the convolution part is flattened and is inserted as a one-dimensional vector into the 1st fully connected (FC) layer. Between the two FC layers activation function, ReLU and Dropout with a rate of 0.5 are applied. Finally, the last FC layer maps the neurons into 8 nodes, as the sensor's classes, and a SoftMax function is applied to predict their percentage. **Figure 16** depicts the architecture investigated by Krizhevsky and his colleagues [22-25]. It is not the original one, since it has been adapted to the needs of the present research endeavor.

Chapter 4. **RESULTS & DISCUSSION**

Scalograms that consist of the algorithm's dataset is segmented into training, validation, and test datasets. Specifically, 70% of total inputs are used for training the algorithm and the remaining data are separated in equal numbers as validation and test data. The epochs and batch size of input data are 100 and 32 respectively. In each epoch, simultaneously is measured the loss and the accuracy of training and validation data. Training and validation loss indicates how well the model is fitting training and new data respectively. **Figure 17** depicts in the same plot how training and validation loss changes over time. This graphic indicates that the training error is low and the gap between the training and validate date tends to be low values after 50 repetitions. In machine learning these results indicate that the algorithm does not reveal underfitting and overfitting phenomena. These promising results are confirmed by **Figure 18**, showing how the accuracy of the algorithm is being increased in each epoch revealing that the algorithm can produce 93% correct predictive outputs.

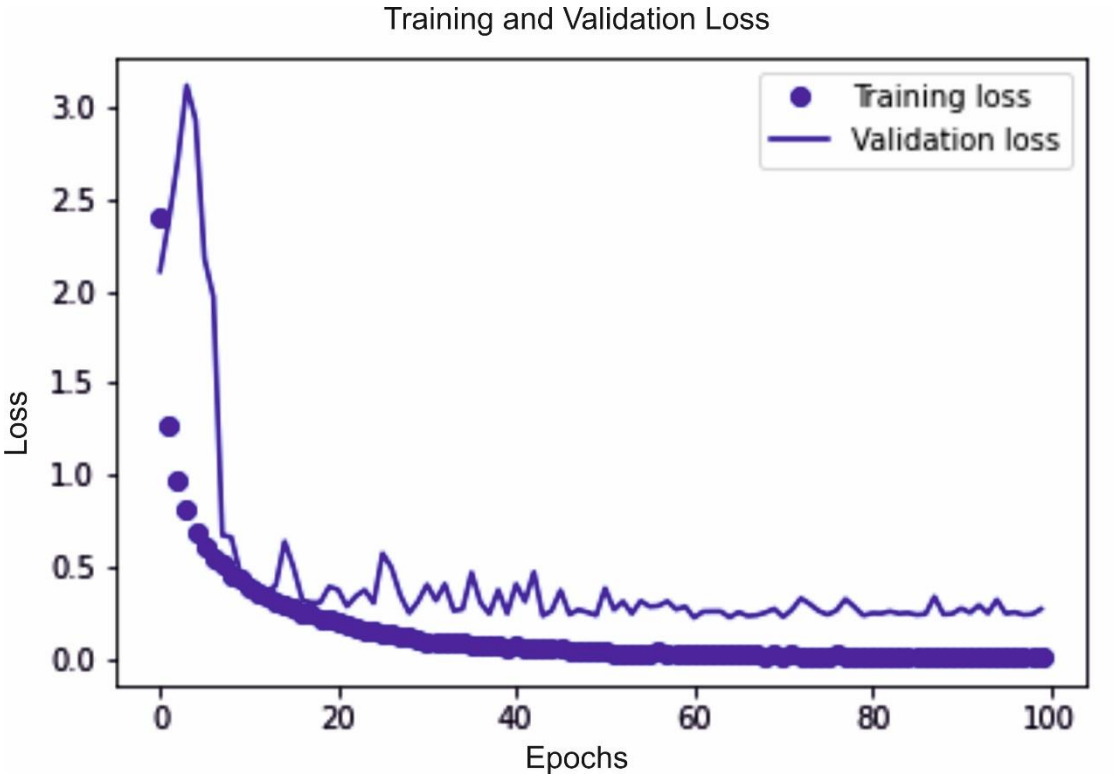


Figure 17: *Training and Validation Loss of DCNNA.*

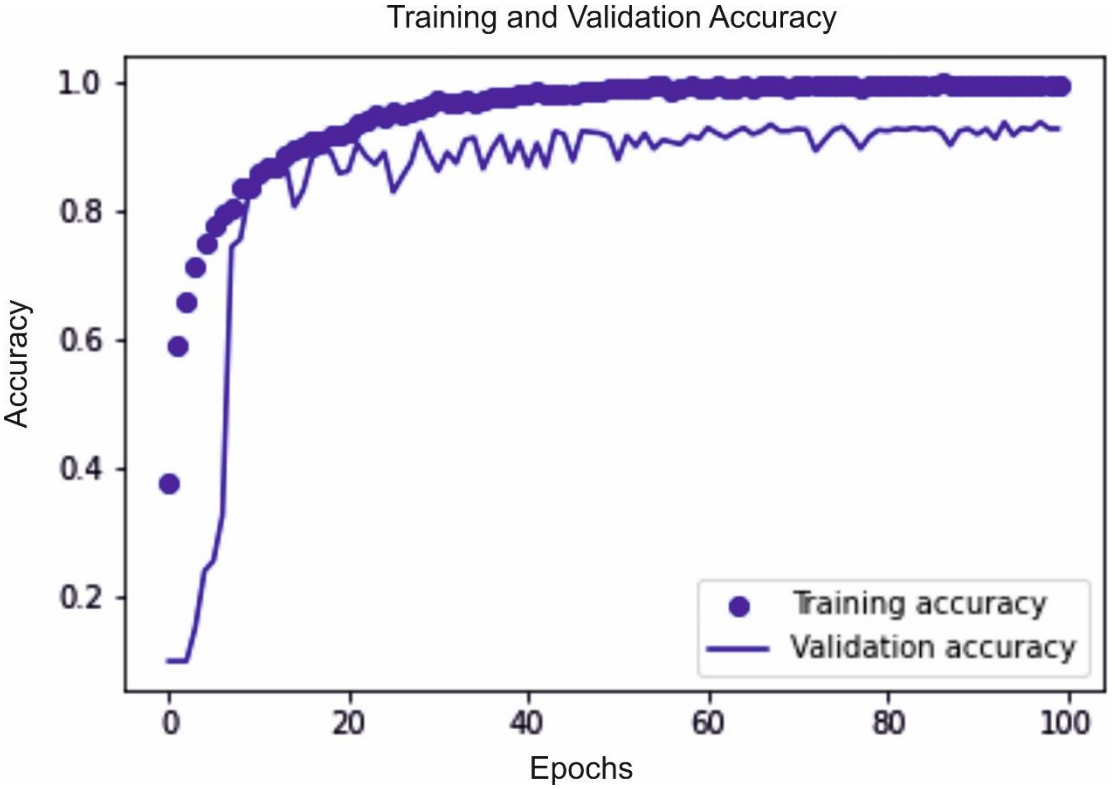


Figure 18: *Training and Validation Accuracy of DCNNA.*

To check out in more depth the resultant of the algorithm, its performance is measured over total unseen test datasets (576 samples). The performance metric is determined by the number of correct detectors and the true events that were detected and this is turn determined as the fraction of correct model detections and as a fraction of how true events were corrected which are called precision (p) and recall (r) correspondingly. In addition, another performance metric is estimated which turns precision and recall into one fβ-score [55]. It is given by:

$$\text{fβ-score} = (1+\beta^2)\frac{(p)*(r)}{(\beta^2)(p)+(r)},\ \beta=1 \tag{13}$$

Table 2 presents the performance metrics. Support is the random test dataset that is segmented by the total number of input images. The algorithm can predict the sensor location with a mean prediction score of 94%. His performance average value in predicting true events and f1 score is equal to 93%. In further analysis of the performance of each sensor, observed that the 3 intermediate sensors 4,5,6 which are located on the left side of the ball bearing have reduced prediction mean values. That may happen because the measured acceleration data points contain noise, or even the data sets are not sufficient for training.

Sensor Class	precision	recall	f1-score	support
1	100%	97%	99%	73
2	96%	100%	98%	69
3	96%	99%	97%	72
4	95%	76%	84%	70
5	90%	87%	88%	79
6	80%	95%	87%	73
7	97%	93%	95%	72
8	99%	99%	99%	68
weighted avg	94%	93%	93%	576

***Table 2**: Prediction Score of DCNNA*

Figure 19 is a plot of the algorithm performance in a form of a confusion matrix. A confusion matrix is a multidimensional array in a dimension equal to sensor classes, that is used to evaluate the prediction quality of an unseen dataset (test set) [52-54]. The matrix has been expressed via Python, so the sensor classes started from the value 0 which corresponds the sensor 1. Diagonal elements represent true predictions, while off-diagonal elements provide the mislabeled test images after classification. A colormap order is used to emphasize the class sensors that occur with the higher prediction score. The results of **Table 2** are represented in the confusion matrix, indicating that for sensor 4 the prediction of its location is correct 58 over 70 supper datasets. For sensor 2 the algorithm gives 69 times correct predictions over 69 support tests, this confirms the recall value of 100%.

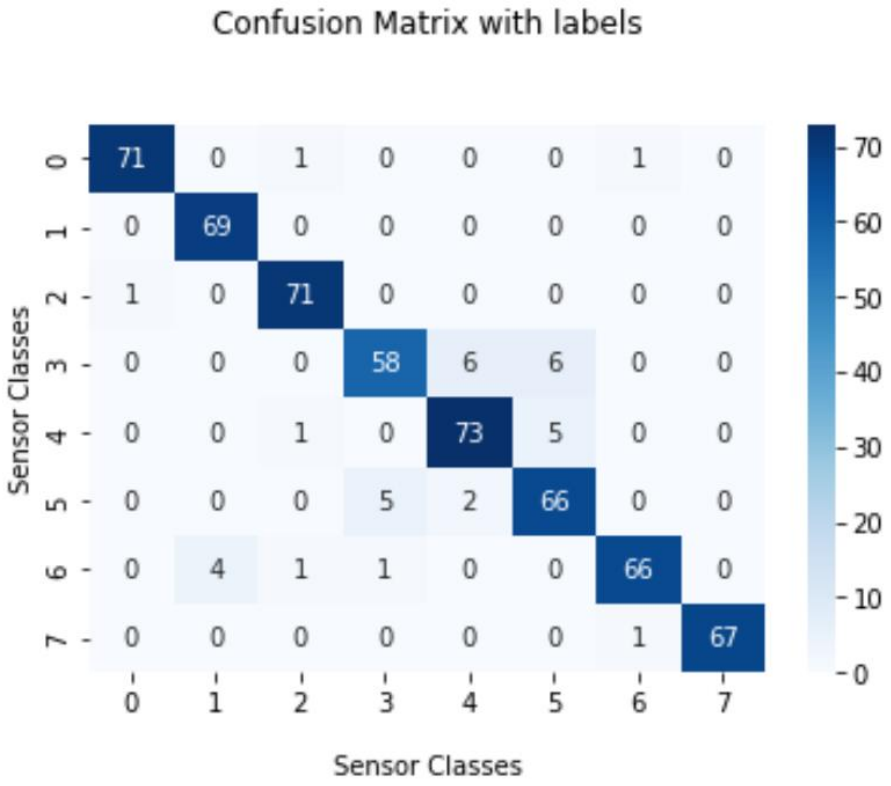


Figure 19: *Confusion Matrix of Algorithm.*

In respect of how the AlexNet CNN performs for the problem of identification of the eight sensors distributed in the local bearing region of the flexible shaft-rotor system, we wonder what would be the prediction of the algorithm if it uses new unseen scalograms. We perform 2 tests where the input scalograms are being defined as the average value of the signals of two neighboring sensors. This addresses to some extent the proposed problem. Thus, we used the trained algorithm to compute the prediction score, in percentage form, of every new scalogram of mean values over the 8 classes. To depict the whole prediction score, we summarize it by estimating the prediction score's mean value over sensor classes, see **Figure 20Figure 21**, **Figure 22**, **Figure 22**, and **Figure 23**. Is observed that in most cases at least one of the sensors that contributed to this linear combination receives a high value in the predicted distribution over the set of eight sequentially arranged sensors to cover the local bearing area. Specifically, sensor pairs such as 1-2, 2-3, 6-7, and 7-8 give remarkable results, since the two nonboring sensors have the highest values while the remaining sensors have quite lower values. From the standpoint of mechanics, this prediction as a distribution over the space formed by the sensors is quite interesting. Since the sensors are distributed in a local area of a continuum, they share common features of the dynamics. This is revealed by the algorithm prediction of the mean value of the signals stemming from two neighboring sensors.

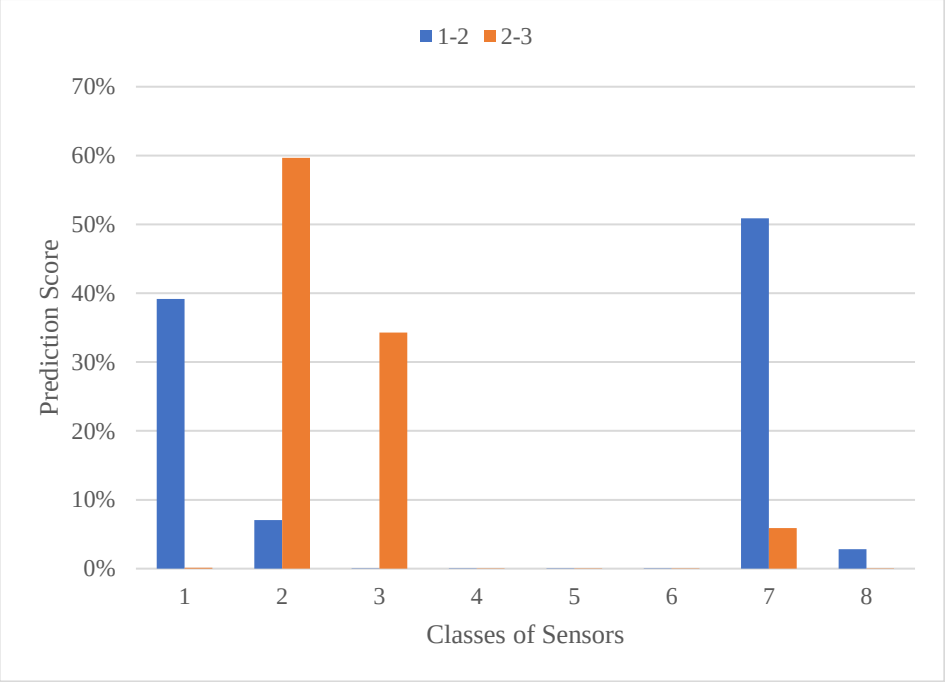


Figure 20: *Predicted distribution of mean-value signals of two neighboring sensors over the space of the eight sensors whose data were used to train a DCMM algorithm to solve the proposed problem. (S1, S2) & (S2, S3).*

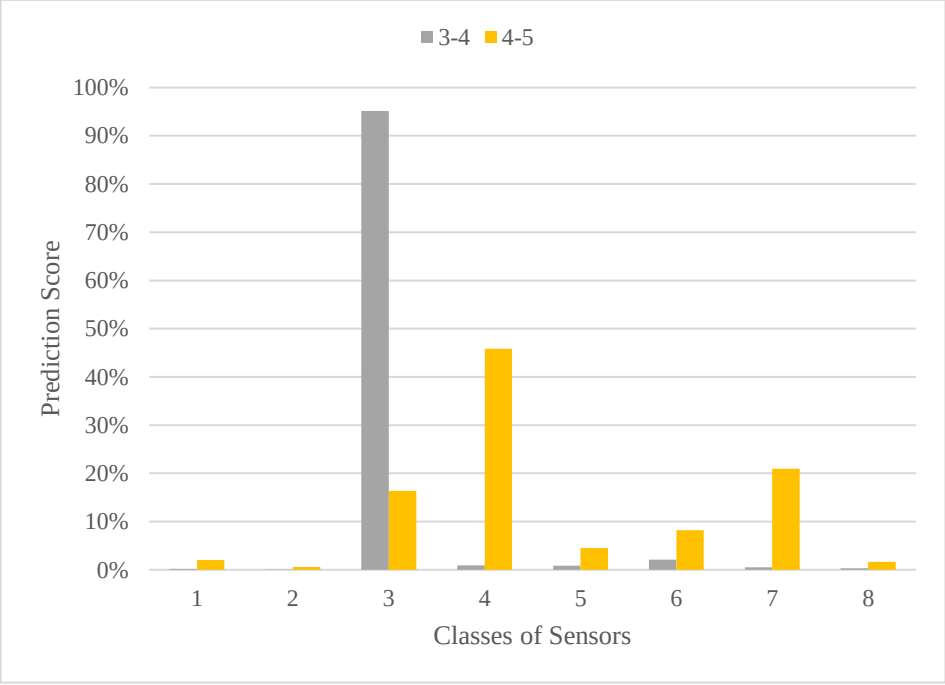


Figure 21: *Predicted distribution of mean-value signals of two neighboring sensors over the space of the eight sensors whose data were used to train a DCMM algorithm to solve the proposed problem. (S3, S4) & (S4, S5).*

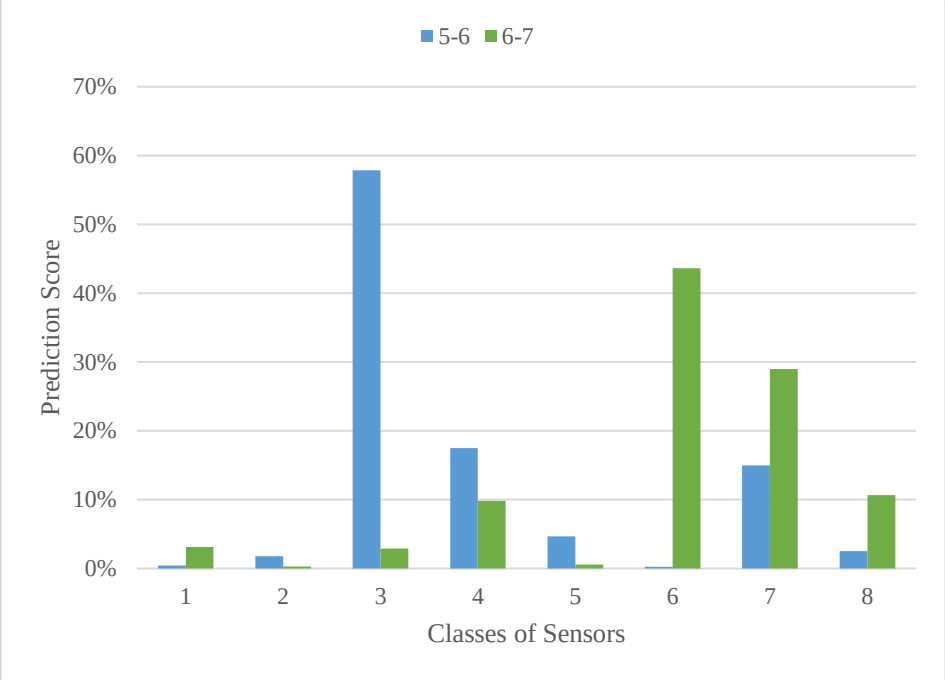


Figure 22: *Predicted distribution of mean-value signals of two neighboring sensors over the space of the eight sensors whose data were used to train a DCMM algorithm to solve the proposed problem. (S5, S6) & (S6, S7).*

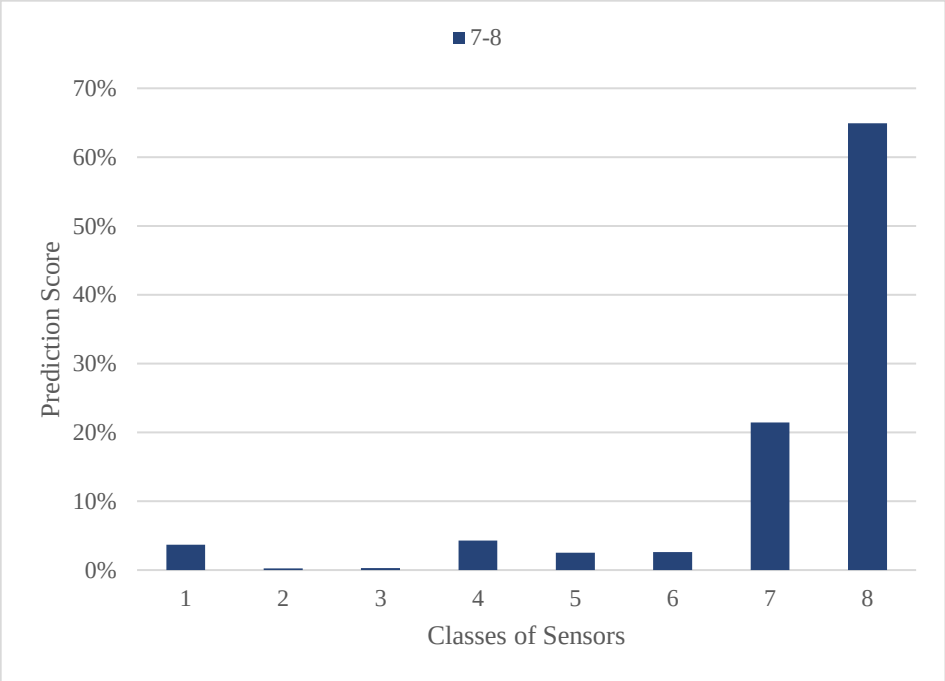


Figure 23: *Predicted distribution of mean-value signals of two neighboring sensors over the space of the eight sensors whose data were used to train a DCMM algorithm to solve the proposed problem. (S7, S8).*

For further checking of algorithm performance, we proceeded in another explorative test where the inputs scalogram are defined as those of generic linear combinations of the already measured acceleration at the specific eight points:

$$input_i^{(2)} = \frac{1}{3}s_i + \frac{2}{3}s_{i+1}, \quad i=1,2,..,7 \tag{14}$$

In this linear combination signal of the sensor, i+1 has a double contribution in a scalogram than that of the sensor. Similarly, with the previous experiment the results are quite optimistic (see. **Figure 24**, **Figure 25**, **Figure 26**, and **Figure 27**). In this case, sensor pairs such as 1-2, 2-3,4-5, and 6-7 give remarkable results, since the two nonboring sensors have the highest values while the remaining sensors have quite lower values.

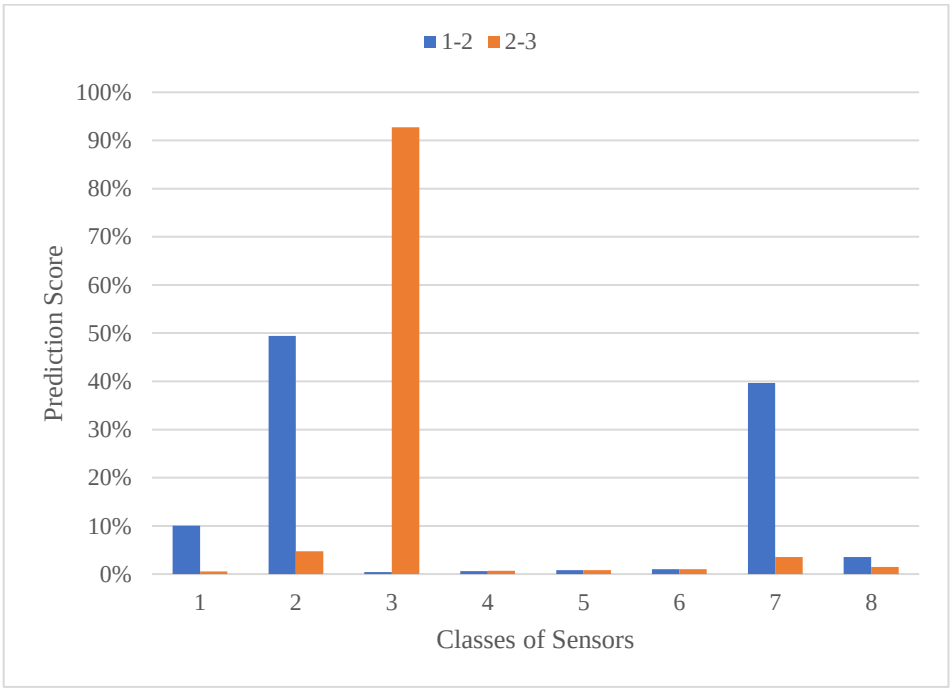


Figure 24: *Predicted distribution of linear combinations of sensors over the space of the eight sensors whose data were used to train a DCMM algorithm to solve the proposed problem. (S1, S2) & (S2, S3).*

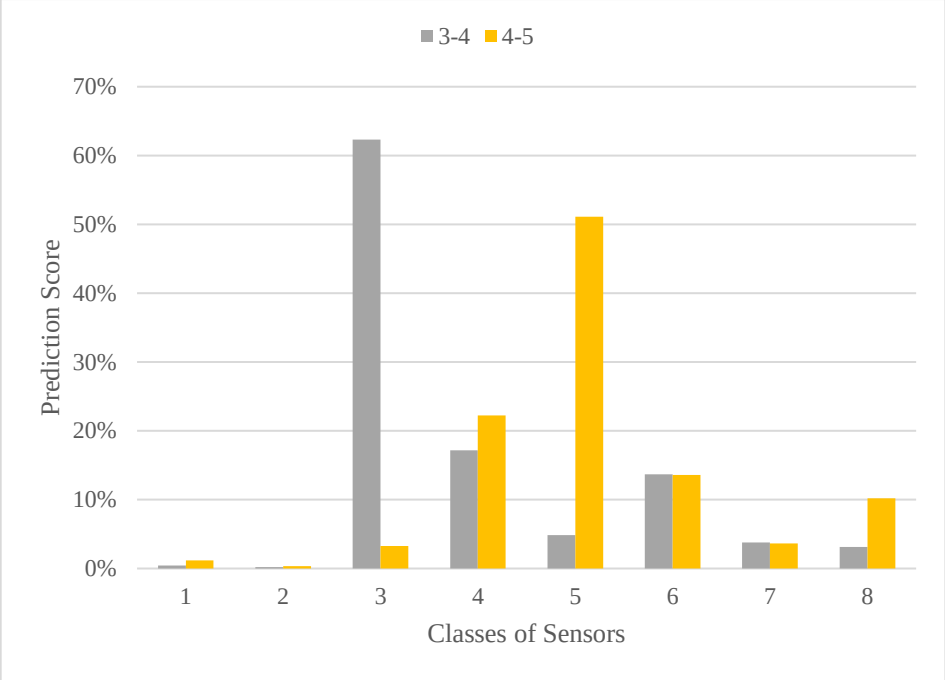


Figure 25: *Predicted distribution of linear combinations of sensors over the space of the eight sensors whose data were used to train a DCMM algorithm to solve the proposed problem. (S3, S4) & (S4, S5)*

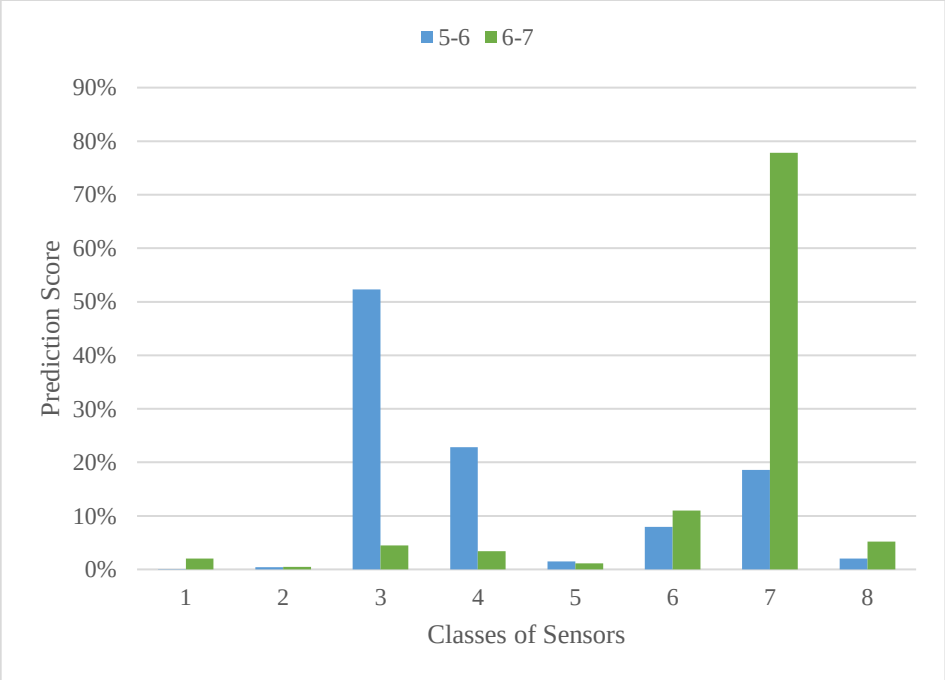


Figure 26: *Predicted distribution of linear combinations of sensors over the space of the eight sensors whose data were used to train a DCMM algorithm to solve the proposed problem. (S5, S6) & (S6, S7)*

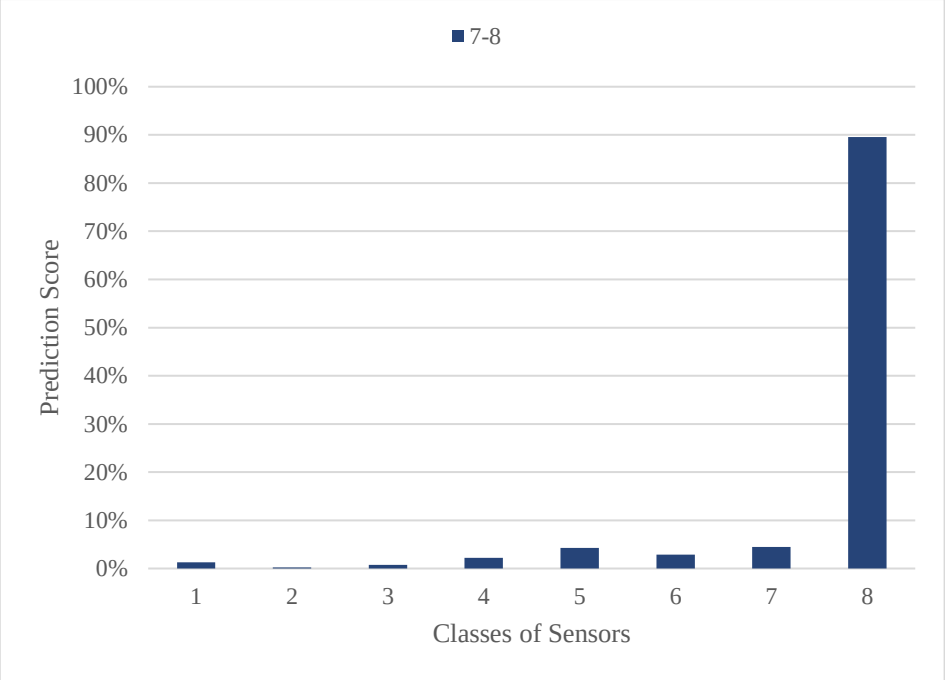


Figure 27: *Predicted distribution of linear combinations of sensors over the space of the eight sensors whose data were used to train a DCMM algorithm to solve the proposed problem. (S7, S8)*

Both experiments based on the mechanic's principle, reveal the properties of the system's dynamic condition and make CNN a promising tool for extracting features by measuring acceleration data. As a result, their implementation in system monitoring for further analysis may drive in finding new aspects that reduce systems health.

Chapter 5. CONCLUSION – SUGGESTIONS FOR FURTHER STUDY

A novel classification of vibration signals is successfully performed by training a tested deep neural network with measurements of vibration dynamics at the level of acceleration. The fact that the traditional vibration method analysis is used only for extracting features and is unable to learn from the experience inspired us to use convolution neural networks because they overcome these limitations. Thus, training the AlexNet algorithm on measuring acceleration data of eight sensors which are arranged over a curved surrounding one of the local bearing areas of a flexible shaft rotor systems, turned CNN into a detection tool of sensors position. Specifically, the collected acceleration data share dynamic features of each signal and they turned into images (scalograms) to train the AlexNet algorithm. Algorithm evaluation ranges from 93-94% for unseen datasets, and this in turn infers further experiments.

The main motivation for further research is the fact that sensors share the common feature of dynamics since they are distributed in a local area of a continuum system. Hence, to evaluate the principle of the dynamic system produced two datasets. In the first one scalograms depicts dynamic features of the mean value of 2 consecutively signals while in the second dataset the contribution of one signal is double from the other. Plotting the mean value of probability distribution of each signal over 8 possible sensor positions observed that the detection tool gives promising results in the detection of the local area. The prediction as a distribution over the space of sensors is quite interesting from a physics point of view: since the sensors are distributed in a local area of a continuum, they share common dynamics and this revealed the prediction from where the mean value of the signals is coming.

This quite promising research effort inspires further studies in the field of data-driven vibration dynamics with the powerful computation-and-prediction tools paved by Artificial Neural Networks as evidenced in the research community in computational science and real-world big data problems across major industries. Further research aimed to explore fundamental machine learning-based methods is being under process. The main goal is the implementation of the CNN algorithm as a detection tool, to find the distance of the collected signal from a reference point. So, in laboratory conditions, an experiment of shaft-bearing was set up to simulate an intermediate shaft of the propulsion system of the ship. Eight sensors were arranged along the length of the shaft at equal distances from one another and then was conducted a tap test by using a modal hammer. Data sets were collected to train the CNN algorithm so that it can detect the local location of each sensor. After training, the algorithm will be tested in unseen datasets, which have been collected by repeating the experiment and adjusting the sensors in the intermediate positions of the previous positions. Some initial results are quite promising and they will be represented hopefully shortly.

REFERENCES

1. Salam, I., et al., *An air crash due to fatigue failure of a ball bearing*, in *Failure Analysis Case Studies II*, D.R.H. Jones, Editor. 2001, Pergamon: Oxford. p. 415-423.
2. Ejaz, N., I. Salam, and A. Tauqir, *Failure analysis of an aero engine ball bearing.* Journal of Failure Analysis and Prevention, 2006. **6**(6): p. 25-31.
3. Georgiadis, A., X. Gong, and N. Meier, *Vibration analysis based on the spectrum kurtosis for adjustment and monitoring of ball bearing radial clearance.* MATEC Web of Conferences, 2018. **211**: p. 06006.
4. Obuchowski, J., R. Zimroz, and A. Wyłomańska, *Blind equalization using combined skewness–kurtosis criterion for gearbox vibration enhancement.* Measurement, 2016. **88**: p. 34-44.
5. Caesarendra, W. and T. Tjahjowidodo, *A Review of Feature Extraction Methods in Vibration-Based Condition Monitoring and Its Application for Degradation Trend Estimation of Low-Speed Slew Bearing.* Machines, 2017. **5**(4).
6. Lin, H.-C., et al., *Bearing vibration detection and analysis using enhanced fast Fourier transform algorithm.* Advances in Mechanical Engineering, 2016. **8**(10): p. 1687814016675080.
7. Rucka, M. and K. Wilde, *Application of continuous wavelet transform in vibration-based damage detection method for beams and plates.* Journal of Sound and Vibration, 2006. **297**(3): p. 536-550.
8. Yamamoto, T. and Y. Ishida, *Linear and Nonlinear Rotordynamics: A Modern Treatment with Applications.* Applied Mechanics Reviews, 2002. **55**(3): p. B45-B46.
9. Genta, G., *Dynamics of rotating systems.* 2005.
10. Burgemeister, J. and J. Wagner, *Piezoelectric Accelerometers*. 2021.
11. Levinzon, F., *Piezoelectric Accelerometers with Integral Electronics*. 2015. 169.
12. Regtien, P.L.P. and E. Dertien, *Sensors for Mechatronics*. 2012.
13. Sinclair, R.I., *Sensors and Transducers*. 2001.
14. Steinem, C. and A. Janshoff, *Piezoelectric Sensors*. Springer Series on Chemical Sensors and Biosensors. 2007.
15. Al-Badour, F., M. Sunar, and L. Cheded, *Vibration analysis of rotating machinery using time–frequency analysis and wavelet techniques.* Mechanical Systems and Signal Processing, 2011. **25**(6): p. 2083-2101.
16. Brouzas, S. and T.I. Georgiou, *Spatial Localization of Air Inclusions in Carbon Fiber T-Beam, by use of Wavelet Entropy Time Series from Hammer Tap Test*, in *Proceedings of the ASME 2021 International Mechanical Engineering Congress and Exposition*. 2021.
17. Tang, Y.Y., *Wavelet Theory and Its Application to Pattern Recognition*. 2009: World Scientific Publishing Co., Inc.

18. Walnut, F.D., *An Introduction to Wavelet Analysis*. 2014: Birkhäuser Basel.

19. Debnath, L., *Wavelet Transforms & Their Applications*. 2002.

20. Shubhendu, S. and J.F. Vijay. *Applicability of Artificial Intelligence in Different Fields of Life*. 2013.

21. Mitchell, T.M., *Machine Learning*. 1997: McGraw-Hill, Inc.

22. Jiao, J., et al., *A comprehensive review on convolutional neural network in machine fault diagnosis.* Neurocomputing, 2020. **417**: p. 36-63.

23. Krizhevsky, A., I. Sutskever, and G. Hinton, *ImageNet classification with deep convolutional neural networks*, in *NIPS*. 2012.

24. Gu J., et al., *Recent advances in convolutional neural networks.* Pattern Recognition, 2018. **77**: p. 354-377.

25. Byeon, Y.-H., S.-B. Pan, and K.-C. Kwak, *Intelligent Deep Models Based on Scalograms of Electrocardiogram Signals for Biometrics.* 2019. **19**(4): p. 935.

26. Goodfellow, I., Y. Bengio, and A. Courville, *Deep Learning*. 2016: The MIT Press.

27. Cowan, N., *What are the differences between long-term, short-term, and working memory?* Progress in brain research, 2008. **169**: p. 323-338.

28. Szegedy, C., et al., *Going deeper with convolutions*. 2015. 1-9.

29. He, K., et al. *Deep Residual Learning for Image Recognition*. in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

30. Molland, A.F., *The Maritime Engineering Reference Book: A Guide to Ship Design, Construction and Operation*. 2008: Butterworth-Heinemann.

31. Mathew, S. and G.S. Philip, *Wind Turbines*. 2012. p. 93-111.

32. John, S.K., et al., *Investigation of Bearing Failure in a Turbo Shaft Engine.* Journal of Failure Analysis and Prevention, 2020. **20**(1): p. 34-39.

33. Eftekharnejad, B., et al., *The application of spectral kurtosis on Acoustic Emission and vibrations from a defective bearing.* Mechanical Systems and Signal Processing, 2011. **25**(1): p. 266-284.

34. Silva, C.W.d., *Vibration and Shock Handbook*. 2005, Boca Raton.

35. Aguiar-Conraria, L. and M.J. Soares, *The Continuous Wavelet Transform: Moving Beyond Uni- and Bivariate Analysis.* Journal of Economic Surveys, 2014. **28**(2): p. 344-375.

36. Leise, T.L., *Wavelet-based analysis of circadian behavioral rhythms.* Methods Enzymol, 2015. **551**: p. 95-119.

37. Mohammed, M., M. Khan, and E. Bashier, *Machine Learning: Algorithms and Applications*. 2016.

38. van der Aalst, W.M.P., et al., *Process mining: a two-step approach to balance between underfitting and overfitting.* Software & Systems Modeling, 2008. **9**(1): p. 87.

39. E., A. and T. S., *The Manual of Photography*. 2017: Focal Press.

40. Berner, J., et al., *The Modern Mathematics of Deep Learning*. 2021.

41. Maas, A.L. *Rectifier Nonlinearities Improve Neural Network Acoustic Models*. 2013.

42. Boureau, Y.L., J. Ponce, and Y. Lecun, *A Theoretical Analysis of Feature Pooling in Visual Recognition*. 2010. 111-118.

43. Srivastava, N., et al., *Dropout: a simple way to prevent neural networks from overfitting.* 2014. **15**(1 %J J. Mach. Learn. Res.): p. 1929–1958.

44. Georgiou, I.T., *A single pair-of-sensors technique for geometry consistent sensing of acceleration vector fields in beam structures: damage detection and dissipation estimation by POD modes.* Meccanica, 2015. **50**(5): p. 1303-1330.

45. MATLAB. 2021, The MathWorks Inc.: Natick, Massachusetts.

46. Rossum, G.V. and F.L. Drake, *Python 3 Reference Manual*. 2009: CreateSpace.

47. Michel, M., et al., *Wavelet Toolbox™ 4*. 2007.

48. Harris, C.R., et al., *Array programming with NumPy.* Nature, 2020. **585**(7825): p. 357-362.

49. Abadi, M., et al., *TensorFlow: a system for large-scale machine learning*, in *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*. 2016, USENIX Association: Savannah, GA, USA. p. 265–283.

50. Buitinck, L., et al., *API design for machine learning software: Experiences from the scikit-learn project.* API Design for Machine Learning Software: Experiences from the Scikit-learn Project, 2013.

51. team, T.p.d., *Pandas*. 2020, Zenodo.

52. Hunter, J.D., *Matplotlib: A 2D Graphics Environment.* Computing in Science & Engineering, 2007. **9**(3): p. 90-95.

53. Bradski, G., *The OpenCV Library.* Dr. Dobb's Journal of Software Tools, 2000.

54. Lilly, J. and S. Olhede, *Higher-Order Properties of Analytic Wavelets.* IEEE Transactions on Signal Processing, 2009. **57**: p. 146-160.

55. Lawrence, N.D., et al. *Precision-Recall-Gain Curves: PR Analysis Done Right*. 2015.