Predictive Maintenance on Flexible Impeller Pumps Based on Acoustic Data

by

Ceren Çoker Turan

B.S., Electrical and Electronics Engineering, Istanbul University, 2017

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2021

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my research supervisor, Prof. Mehmet Akar for giving me the opportunity to do research and providing invaluable guidance throughout this research. It was a great privilege and honor to work and study under his guidance. My special thanks goes to Canberk Demircan, Oğuzhan Sevim, Ümit Develer, Yiğit Öksüz, and Dr. Özlem Feyza Erkan for the keen interest shown to complete this thesis successfully.

I am extremely grateful to my parents for their love, prayers, caring and sacrifices for educating and preparing me for my future. I am very much thankful to my husband for his love, understanding and continuing support to complete this research work. Besides, I express my thanks to my brother for his full support to my educational journey. Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

This study was funded by Scientific and Technological Research Council of Turkey (TUBITAK) 1505 project 5180025, in collaboration of Bogazici University and Eliar Company.

ABSTRACT

Predictive Maintenance on Flexible Impeller Pumps Based on Acoustic Data

The idea of smart factories has aroused with the emergence of Industry 4.0. A way to contribute to the construction of smart factories is to develop intelligent maintenance strategies which enable factories to reduce maintenance costs and keep them away from catastrophic damages. Predictive maintenance warns users regarding the need for maintenance of assets at a specific moment. In this thesis, predictive maintenance strategies for Flexible Impeller Pumps (FIP) are developed by using supervised and unsupervised learning methods.

After observation of real textile dye houses, an emulation setup has been designed to collect data from real use cases of FIPs. Data consisting of three conditions, healthy, looseness, and cavitation, have been collected via an acoustic sensor involving 3 microphones. Obtained time-series signals are converted to informative features which are utilized for training a supervised model, convolutional neural network (CNN). The trained model is able to generate a warning for a specific failure state. In addition to the supervised model, an unsupervised model has also been trained for anomaly detection of FIPs via extracted features of healthy data due to the applicability issues of unhealthy state data acquisition. The model can produce a warning when an anomaly is detected but it cannot specify the failure mode. Both models give satisfactory results for classifying failures and detecting anomaly situations of FIPs. Mel Frequency Cepstrum Coefficients (MFCC) have shown superiority among the features extracted to confirm human hearing based modeling can be applied successfully in maintenance methods.

ÖZET

Akustik Veriler ile Esnek Pervaneli Pompalarda Kestirimci Bakım

Endüstri 4.0 kavramının ortaya çıkmasıyla birlikte akıllı fabrika fikri önem kazanmıştır. Akıllı fabrikaların oluşumuna katkı sağlamanın bir yolu da akıllı bakım/onarım stratejileri geliştirmektir. Bu stratejiler fabrikaların bakım masraflarını düşürmesini ve geri dönülemez büyüklükteki sorunlardan kaçınmasını sağlar. Öngörücü bakım, parçaların bakım ihtiyaçları hakkında belirli bir zamanda kullanıcıyı uyarabilir. Bu tezde, gözetimli ve gözetimsiz yöntemler kullanılarak esnek pervaneli pompalar için öngörücü bakım stratejileri geliştirilmiştir.

Gerçek tekstil boyahanelerinin gözlemlenmesinden sonra, esnek pervaneli pompaların kullanım durumlarından veri toplamak için bir deney düzeneği tasarlanmıştır. Sağlıklı, gevşeklik ve kavitasyon olmak üzere üç koşuldan oluşan ses verileri, 3 mikrofon içeren bir akustik sensör aracılığıyla toplanmıştır. Elde edilen zaman serisi sinyalleri, bilgi içeren özelliklere dönüştürülmüştür ve bu özellikler gözetimli bir model olan Evrişimsel Sinir Ağlarını eğitmek için kullanılmıştır. Eğitilen model o anki hata durumunu bildirecek şekilde uyarı verir. Gözetimli modelin yanı sıra hatalı durumların toplanmasındaki uygulanabilirlik sorunlarından dolayı, esnek kanatlı pompanın anormal durum tespiti için sağlıklı durumdan çıkarılan özellikler kullanılarak gözetimsiz bir model de eğitilmiştir. Eğitilen model hata durumunun kaynağını tespit edemese de kullanıcıya anormal durum için uyarı vermektedir. İki model de pompanın hata durumlarının ve anormalliklerin tespit edilmesinde iyi sonuçlar vermiştir. Mel frekansı sepstral katsayılarının diğer özellikler arasında iyi sonuçlar vermesi insanın duyma özelliğinin bakım modellerinde başarılı bir şekilde modellenebilir olduğunu kanıtlamıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

c^i	i^{th} cluster
e_t	Error of t^{th} neuron
g_t	Gradient of a mini batch
$H_m(k)$	Amplitude of Mel filters
$J(w)$	Objective/error function
m_c	Sum of estimated weighted probabilities
m_t	Estimated mean of the gradients
$hatm_t$	Bias corrected mean of the gradients
$\mathcal{N}$	Multivariate gaussian distribution
O^i	Output of i^{th} layer
$p_{o,c}$	Predicted probability observation o of class c
$p(x)$	Probability distribution of x
$P_{x_m,M}(\omega_k)$	Periodogram of m^{th} window of $x(n)$
r_{ic}	Normalized estimated weighted probabilities
$\hat{S}_x^W(\omega_k)$	Estimated Power Spectrogram
v_t	Uncentered variance of gradients
$hatm_t$	Bias corrected variance of gradients
w	Learnable parameter
$w(n)$	Hamming window
W_c	GMM weights
W_i	Weight of i^{th} neuron
$\bar{x}$	Mean of x
$x_m(n)$	m^{th} window of $x(n)$
$x[n]$	Discrete time signal
$x(t)$	Time series signal
X_i	Input of i^{th} neuron
$X[k]$	DTFT of x
$X_w(a,b)$	Wavelet transformed signal

y_{actual}	Actual neuron output
$y_{o,c}$	Binary indicator of observation o of class c
$y_{predicted}$	Predicted neuron output
$y(t)$	High pass filtered time series signal
α	Filtering coefficient
β	Adam hyperparameter
Δf	Frequency deviation
Δt	Time deviation
η	Learning rate
μ_c	Updated mean of cluster c
μ_i	i^{th} cluster centroid
σ	Standard deviation
σ^2	Variance
Σ	Covariance matrix
Σ_c	Updated covariance matrix of cluster c
$\psi(t)$	Mother wavelet
$\omega(n)$	Window function

LIST OF ACRONYMS/ABBREVIATIONS

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
Adam	Adaptive Moment Estimation
ADC	Analog to Digital Converter
AFSA	Artificial Fish Swarm Algorithm
ANN	Artificial Neural Networks
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRA	Collaborative Recommendation Approach
CSWRU	Case Western Reserve University Bearing Data Center
DAE	Deep Autoencoder
DFT	Discrete Fourier Transform
DCTLN	Deep Convolutional Transfer Learning Network
DTFT	Discrete Time Fourier Transform
EM	Expectation Maximization
FC	Fully Connected
FT	Fourier Transform
FFT	Fast Fourier Transform
FIP	Flexible Impeller Pump
GMM	Gaussian Mixture Model
GPU	Graphical Processing Unit
HDMI	High-Definition Multimedia Interface
HPF	High Pass Filter
I2C	Inter Integrated Circuit
IMS	Intelligence Maintenance System
k-NN	k-Nearest Neighbor
MAE	Mean Absolute Error

MEMS	Micro Electro-Mechanical Systems
MF	Mean Frequency
MFCC	Mel-Frequency Cepstrum Coefficients
ML	Machine Learning
MLP	Multi-Layer Perceptron
MMD	Maximum Mean Discrepancy
MS	Mean Square
MS	Mel Spectrogram
MSE	Mean Squared Error
NASA	National Aeronautics and Space Administration
NLP	Natural Language Processing
PC	Personal Computer
PS	Pitch Shifting
PS	Power Spectrum
PSD	Power Spectral Density
ReLU	Rectifier Linear Unit
RGB	Red - Green - Blue
RMS	Root Mean Square
RPM	Revolutions per Minute
SD	Secure Digital
SGD	Stochastic Gradient Descent
SNR	Signal to Noise Ratio
SPI	Serial Peripheral Interface
SSDA-TL	Sparse Stacked Denoising Autoencoder with optimized Transfer Learning
SSH	Secure Shell Protocol
STFT	Short Time Fourier Transform
SVD	Single Value Decomposition
SVM	Support Vector Machine
TUBITAK	The Scientific and Technological Research Council of Turkey
UART	Universal Asynchronous Receiver Transmitter

USB	Universal Serial Bus
WiFi	Wireless Fidelity

1. INTRODUCTION

1.1. Motivation and Scope

Since Industry 4.0 concept has been introduced to the industry in 2011, Germany, it has gained a lot of popularity from both scientific and industrial fields. The concept of Industry 4.0 aims for an increase in productivity and sustainability of products by controlling production processes intelligently, thereby, it is urged to seek solutions so as to increase the quality of processes and products [1]. One of the most important aspects to increase efficiency in the industry is maintenance [2]. Maintenance usually requires machines to stop and therefore reduces the output; that is why companies try to reduce maintenance frequency and duration. On the other hand, it is possible to face catastrophic results due to failure propagation if the maintenance is not done in time with proper diagnostics. The propagation can start from a single part and affect the machine it belongs and/or interfaces and also the goods processed on the machines. Until the Industry 4.0 concept emerged, there had been two main approaches to dominate the industry: breakdown maintenance and planned maintenance [3]. While breakdown maintenance can be harmful due to aforementioned reasons, planned maintenance may result in unnecessary interruptions to the processes. In order to find an optimal maintenance timing, predictive maintenance concept is developed and gained speed with Industry 4.0. Predictive maintenance employs sensors and smart algorithms to detect failures before they happen or to warn users at a very early point to plan maintenance before a breakdown occurs [4].

Predictive maintenance research has gained speed in recent years thanks to the development of Industry 4.0 and the deep interest it has taken from both the academy and the industry. There has been a great deal of focus on prediction of failures of motors and motor parts as their diagnostics are more difficult and their breakdowns can cause great costs to the devices and products. Recently, predictive maintenance focuses mostly on bearing faults on motors with the employment of vibration, current, pressure,

temperature, and acoustic sensors [5]. However, there is a disproportionally low interest on Flexible Impeller Pumps (FIP) compared to their common usage in the industry. FIPs are commonly used in fields such as food processing, pharmaceutical, oenological, and textile finishing. Moreover, the research on them is mainly concentrated on using vibration sensors [6]. It is known that maintenance experts in the industry still use their hearing to detect failures of rotational devices; therefore, audio features of motors can be considered to contain valuable information about a device's health status.

In this thesis, it is aimed to create models that use acoustic signals from a flexible impeller pump in order to detect its failures. These models can contribute to the literature by focusing on FIPs and acoustic sensors for failure detection as both have taken insufficient interest. In order to achieve the thesis goal, an experimental setup is designed and built with contributions from Eliar Company and The Scientific and Technological Research Council of Turkey (TUBITAK) so that FIP sound recordings can be collected. With this setup, data from failure modes and the healthy state are recorded in various rotational speeds. Classification methods in the thesis can be divided into three main parts. First, time, frequency, and time – frequency features are extracted manually and fed into a Convolutional Neural Network (CNN) for failure classification. In addition to this, based on the fact that there can be deeper features embedded in the motor sounds, a deep learning algorithm is also created with raw data input as the second approach. These two approaches require data from both healthy and unhealthy states and can be considered as data hungry. However, in industrial applications, it is difficult to obtain data from unhealthy states as it requires stopping the manufacturing as well as it causes damage on the machines. Therefore, an anomaly detection, or one-class classification, algorithm that only requires healthy pump data is also proposed in the thesis. In this latter approach, unhealthy mode data are only used for verification purposes.

1.2. Industrial Pumps

Pumps are essential and common parts of machines used in various industries such as textile, food processing, cosmetics, chemical manufacturing, oil transfer areas, and so on. Fundamentally, they are divided into two families: centrifugal and positive displacement pumps. Both are mechanical devices used for liquid movement through rotational energy from impellers. However, positive displacement pumps are different from centrifugal pumps in a number of aspects. While positive displacement pumps are efficient at moving high viscosity liquids, efficiency of centrifugal pumps decreases as the viscosity is increased. Besides, centrifugal pumps cannot be used for dry suction, which means that they need positive inlet pressure to transfer the liquid. Finally, positive displacement pumps are more suitable in applications where proper dosing is desired since they produce the same flow at a given speed without needing discharge pressure [7]. Therefore, while centrifugal pumps are preferred in areas involving water supply and circulation, such as irrigation, displacement pumps are preferred in areas including biological fluids, foodstuffs, or emulsions.

Textile manufacturing industries use both types of pumps due to the necessity of transferring different types of fluids; however, this section and the rest of the thesis focus on flexible impeller pumps, a type of positive displacement pumps, as they lack sufficient interest in the literature compared to their usage. In Sections 1.2.1 and 1.2.2, features of flexible impeller pumps and their common failure modes will be explained in detail.

1.2.1. Flexible Impeller Pumps

Flexible Impeller Pump (FIP) was deigned by Arthur M. Briggs and patented in 1940 [8]. The pump is a kind of positive displacement pump based on the working mechanism of sucking liquids with vacuum and then transferring it round into its outlet. Essential parts of the pump are Pump Body, Impeller, Motor Flange, Shaft, Pump Cover, Seals, and O-Rings.

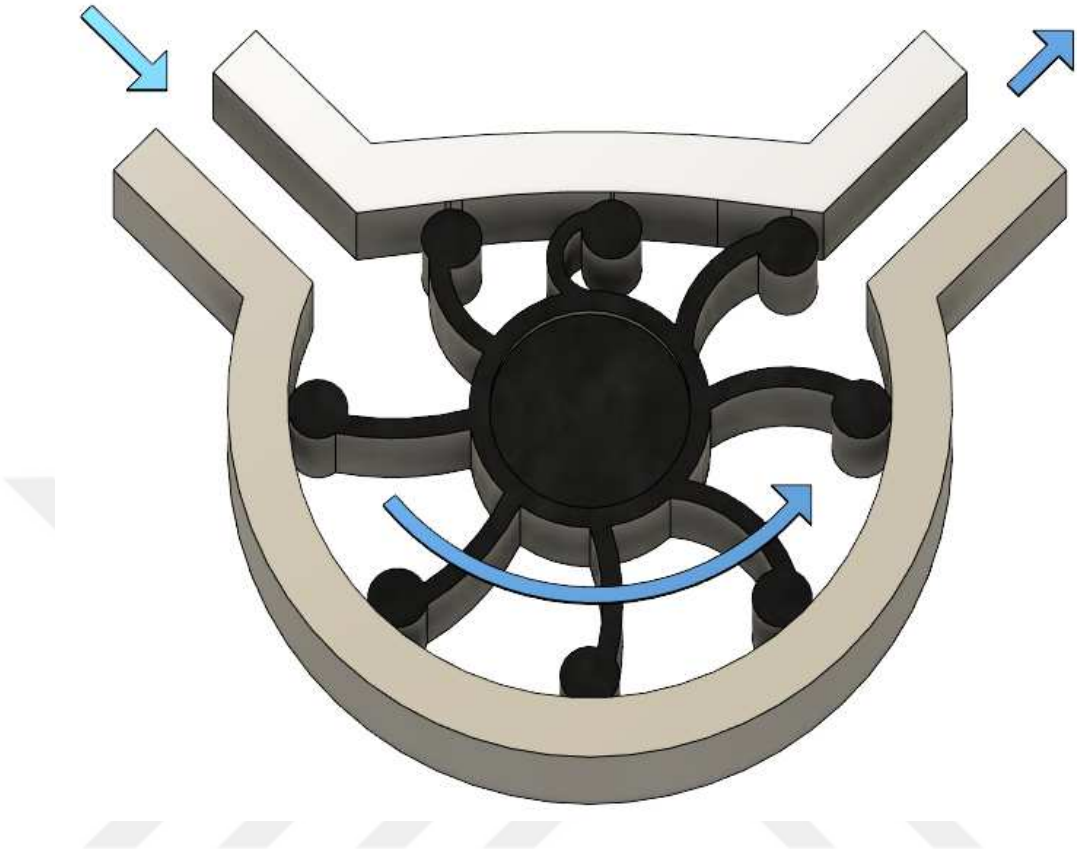

Figure 1.1. Liquid transfer in FIP.

The principle of operation of a FIP is as follows: a motor shaft enables impeller rotation in the pump body. When the motor shaft is rotating, the impeller, having flexible wings, forms a vacuum effect in the suction side and carries the liquid from the inlet port to the outlet port. After the liquid is discharged, the wings touch the eccentric interior walls of the body. Cross-section of liquid transfer in the pump is depicted in Figure 1.2.1. Liquids can also be pumped in the opposite direction with reverse rotation of the shaft.

1.2.2. Flexible Impeller Pump Failure Modes

Although flexible impeller pumps have several advantages owing to their insensitivity to liquid pressure changes and primer dry suction ability, some of their parts are susceptible to damage, especially at high-speed rotation. Relatively low cost impeller

is one of the most sensitive parts of the pump. High-speed rotation causes heat in the pump, so the flexible impeller, made of rubber, can be damaged and deformed. Factors that affect the expiration date of an impeller are listed below:

- High temperature
- Operational pressure
- Rotational speed
- Material features

The most common failure modes in FIPs are cavitation, looseness, and broken impeller [9]. Cavitation failure arises from low pressure due to unbalanced inlet and outlet liquids [10]. The low pressure leads to bubbles in the liquid due to evaporation, thereby, the bubbles blow up the impeller surface by hitting them in every rotation. On the other hand, looseness is an improper assembly failure [11]. It can induce liquid leakage or uncompensated conveying liquid owing to the unbalanced impeller. If failure modes can be detected in timely manner, the expiration and service dates of the pump can be extended.

One of the above failure modes may also trigger others. For example, when cavitation occurs, it can harm the impeller wings to a breaking point; on the other hand, the resulting vibration may also create looseness on the screws. Moreover, faults on the devices frequently cause damage on the processed materials. For example, erroneously dyed textile cannot be re-used or recovered. As a result, solving failure modes in a timely manner can reduce the maintenance and material costs by preventing failure propagation. Based on our consultations with experts in textile dye houses, common failures and their reasons are listed as:

(i) If there is not enough pump flow rate or the pump does not work, possible causes can be listed as:
   - Air on the pump suction or housing part
   - Leakage or hole on the pump receiver part

- Leakage on the gland
- Damaged shaft or coupling
- Congestion of the suction filter
- Broken impeller
- Broken seal

(ii) If there is a lot of vibration and noise, possible causes can be listed as:

- Misalignment of coupling
- Unbalanced motor bearing
- Foreign material in the pump body
- Slipped shaft
- Broken impeller
- Broken seal

Taking into account various causes that can lead to failure, two failure modes are chosen to be worked on: mechanical looseness and cavitation. Mechanical looseness can be caused by misalignment or imbalance of a motor, and cavitation is caused by pressure difference between the inlet and the outlet of the pump. In order to predict the failure modes of the pump setup under investigation, data are collected by means of acoustic sensors placed on the motor and they are analyzed by various learning algorithms and methods.

1.3. Literature Review

In this section, topics including rotating machine failure detection and sound classification models in the literature are investigated in order to discover the state of the art. Surveyed material is analyzed with respect to fields of application, data acquisition methods, features and methods employed, common problems in the field, and proposed solutions to attack them.

Since the Industry 4.0 concept has sped up the research on condition monitoring and predictive maintenance, these two fields have attracted attention from many

different fields. For example, Zhoa et al. proposed a fault diagnosis method for centrifugal pumps in order to detect a pump's state among five different modes: normal, bearing roller wearing, bearing inner wearing, bearing outer race wearing, and pump impeller wearing [12]. However, attention from the industrial machinery also focused on gearboxes and bearing failures as they are among the most common machine parts to be used and to fail. While Wang et al. worked on a mostly mathematical model that is applied to both gear and bearing failures [13] and Xia et al. diagnosed bearing and gearbox faults [14], Jing et al. created a CNN model to detect failures of gears only [15] and Zhang et al. created a graph based neural network to develop failure classification methods focusing on bearing failures [16]. Apart from industrial machinery, the methods for condition monitoring is also used in different fields such as wind turbine health monitoring [17] and composite beam failure detection [18].

As this thesis work is based on classification methods by means of acoustic sensors, there are many other research orientations that share a great deal with the methods to be used in the thesis. Similar algorithms and features are used in environmental sound classification and animal species detection. For example, Salamon et al. worked on environmental sound classification [19]. Since their dataset resembles flexible impeller pump sounds in many ways, their work brings contribution of importance to this thesis and related research. Moreover, Sprengel et al. won the International BirdCLEF 2016 Recognition Challenge [20] by identifying bird species from their sounds based on a CNN model [21] and Nanni et al. tested data augmentation techniques over animal audio classification by using Matlab audio augmentation library [22].

There are several sensor types used in the literature for condition monitoring of machines. The majority of the research, however, is focused on vibration sensors, followed by acoustic sensors in terms of usage frequency in the literature. Salah et al. worked on fault detection over compressor valves by employing an acoustic emission (AE) sensor [23]. They argue that AE sensors are more sensitive to higher-frequency waveform changes than vibration sensors. The data was acquired from their test rig with high frequency range and sampling rate acoustic sensor. Hasan et al. suggested

that using acoustic emission based signals is a suitable solution in order to detect low-frequency faults compared to using vibration sensors; therefore, they worked on fault diagnosis of bearings under various low rotational speeds [24]. They collected 1D acoustic signals from an induction motor rotating at 200, 250, and 300 rpm. Their model predicted the faults under various speeds with promising accuracy results. It is also suggested that increasing the number and/or type of sensors used is also beneficial for classification tasks. Xia et al. increased the number of vibration sensors that they used by placing multiple sensors on various places [14]. Their CNN model showed that using multiple sensors can increase accuracy compared to using a single sensor element. On the other hand, Lei et al. not only increased the number of sensors but also used different types of sensors in their work [17]. They implemented a classification algorithm by using data from a photoelectric switch sensor, an Eddy Current displacement sensor, and a piezoelectric acceleration sensor. Their results also support the idea that increased sensor elements can increase classification accuracy.

Apart from the sensor types, source of data also differs in the literature. Some works employ real operational devices on the field. Henze et al. used a real industrial device to obtain data and perform failure detection [25]. However, prevented them to obtain data from failure states as the operational machine on the field could not be failed on purpose. Their work consists of two stages, anomaly detection and classification. For the anomaly detection algorithm, they only need data from healthy state; however, for the classification they had to consult with experts to artificially create data for failure modes. Even though they faced data acquisition difficulties, their models worked with 98% accuracy. In order to overcome the limitation of working on a real life environment, several researches developed test rigs which emulate the real life applications. For instance, Salah et al. used a test rig to emulate compressor valve use cases [23] while Liu et al. developed an experimental setup simulating belt conveyors [26]. Another approach is to use pre-existing datasets, such as UrbanSound8k, which contain environmental sounds like drilling, car horn, and children playing sounds [27], and Case Western Reserve University Bearing Data Center (CSWRU) bearing dataset [28]. Salamon et al. [19] and Guo et al. [29] employed these

public datasets, respectively, for their research and became able to increase their effort on developing models and algorithms thanks to the spare work load created by getting rid of building a test rig and a data acquisition system.

After a dataset is obtained for the task in hand, a decision is to be made about the way to use it. Researchers may use the raw data signal with none or minor preprocessing or hand-craft some features from the data and use these features for training, validation, testing, and application. Benefit of using raw data is eliminating the need for a field expert to rely on creating meaningful features; however, it mostly requires more processing power and more complex models to obtain hidden features in the data and use them for the classification. For example, Jia et al. diagnosed rotation machine faults without utilizing manual feature extraction techniques [30]. They trained an unsupervised autoencoder with raw spectral input. A deep neural network was developed and fine-tuned by labels in order to classify the health status. The authors suggested that the model trained by raw time signal instead of raw spectral signal could be applied to other machines as well. Jing et al. trained a CNN model with different configurations in order to find segmentation points in the vibration signals [15]. Along with the proper segmentation, the model was also trained by various input data types such as time data, frequency data, time-frequency data, time domain features, frequency domain features, and wavelet domain features. In conclusion, their final CNN model with segmented raw frequency data input gave the best accuracy over fault detection of gearboxes. On the other hand, many researchers obtained great results with manually extracted features. Nanni et al. developed CNN models for animal audio classification [22]. Their CNN models were trained by both raw signals and their visual representations as spectrograms with the help of data augmentation techniques. According to their results, spectrogram was shown to be the optimal input to classify animal sounds but they warned that the best augmentation protocol can vary depending on the nature of the audio.

With their ability to mimic human hearing, Mel Frequency Cepstral Coefficient (MFCC) features have also found usage in the literature. Mühlbauer et al. proposed an

automatic labelling method for acoustic signals to be used in anomaly detection [31]. They preprocessed the signals and then found a repeated cycle in the MFCC features between consecutive frames. They measured the MFCC feature deviation between cycles to detect unhealthy signals. Their methods succeeded in automatic labelling and deviation measurement, but the methods were found weak for the noisy signals. Liu et al. also based their research on MFCC features. They worked on fault detection of belt conveyors relying on acoustic signals [26]. In order to train their machine learning model, they extracted MFCC features from segmented acoustic signals. Their model demonstrated satisfactory accuracy results over their test rig with six fault conditions of belt conveyors.

Classification methods for failure mode detection have also great variance in the literature. A great number of researches, several of which are mentioned early in this section, shifted their work to deep learning methods. Convolutional neural networks, therefore, draw a great attention from the academy. Apart from CNNs, autoencoder models have also gained speed for feature extraction and classification tasks. For example, Sun et al. proposed a novel Sparse Stacked Denoising Autoencoder with optimized Transfer Learning (SSDA-TL) so as to detect bearing faults by using less labelled data [32]. They trained a stacked denoising autoencoder, an unsupervised learning method, by using the source data. Then, the trained encoder parameters were utilized in the classifier as hidden layer parameters. The classifier was trained by a small amount of labelled data. Afterwards, they measured the maximum mean discrepancy (MMD) between target and source data at each hidden layer output. These MMDs were added to the loss of the classification; thereby, the model was not trained for all target data. In conclusion, SSDA-TL model decreased the dependency on the labelled data and increased scalability. Haidong et al. also utilized a deep autoencoder model with correntropy loss function instead of mean squared error, and optimized the model by means of Artificial Fish Swarm Algorithm (AFSA) so as to diagnose rotation machinery faults [33]. Besides, they trained the model without using any signal processing feature extraction techniques. The change in error function provided the model with insensitivity against the complex and non-stationary background noises.

Their model gave good diagnosis results for both gearbox and electrical locomotive roller bearing.

Finally, there are researchers that stick to Machine Learning (ML) algorithms and get great results. For example, Durbhaka et al. implemented k-Nearest Neighbor (k-NN) clustering and Support Vector Machine (SVM) to the vibrational signals from wind turbines [34]. They combined their work with Collaborative Recommendation Approach (CRA) and got 93% accuracy on detecting wind turbine bearing faults. In their multi-sensor application, Paolanti et al. achieved 95% accuracy via regression and multi-class classification algorithms [35]. In order to improve the accuracy of their ML algorithms, they also added a decision forest algorithm on their result stage.

In the literature, one of the greatest problems in failure classification and all types of learning algorithms is data scarcity and data imbalance problems. In order to solve these problems, data augmentation techniques were used heavily. Salamon et al worked on environmental sound classification with UrbanSound8k by using a spectrogram CNN model that is supported by audio augmentation techniques [19]. Their work showed that pitch shifting, dynamic range compression, time-stretching, and addition of background noise have a good effect on learning accuracy in case of data scarcity. However, they also mentioned that the addition of background noise to noise-like signals, such as drilling or air-conditioners, suppressed actual features of the signal. Nanni et al. tested data augmentation techniques over animal audio classification [22]. By using Matlab audio augmentation library, they defined and implemented four augmentation protocols; standard image augmentation combining of rotating, reflecting, or translation of an image, standard signal augmentation including time stretching, pitch shifting, time shifting, volume changing or addition of random noise, spectrogram augmentation and signal augmentation.

Low generalization is also a somewhat data scarcity related issue in failure mode detection. Haidong et al. aimed to decrease the low generalization problem of dataset by utilizing 15 different Deep Autoencoders (DAE) which they trained with 15 different

activation functions in order to obtain various characteristic of vibration signals [36]. Outputs of each DAE were fed to softmax classifiers to decide on one of the failure modes. The accuracy is increased by eliminating some of DAE models when they were under the pre-determined threshold accuracy. The softmax probabilities of remaining models were weighted with their average accuracies and finally obtained the maximum of the probabilities as the failure mode.

Another common problem encountered in condition monitoring applications is noise because sensor elements are not ideal and use case scenarios mostly involve various disturbances. In order to solve the noise problem, different approaches have been employed. Zhang et al. worked on bearing fault diagnosis in noisy environments and different working loads [37]. They augmented their data by an overlapping window frame over the acquired signal. A CNN model that used raw vibration sensor data with small batches was proposed to overcome the noise. The first layer of the CNN had a large kernel with dropout such that it worked like a low pass filter for noise elimination. Their models outperformed other models such as SVM and Multi Layer Perceptron (MLP). On the other hand, O'Brien et al. investigated faults of the composite materials by means of acoustic signal data that was obtained from healthy and damaged composite beams [18]. In order to get rid of deteriorating background noise, Single Value Decomposition (SVD) method was utilized. They chose the most related eigenvalues and discarded the noise related ones. Their proposed model showed a satisfactory accuracy results on their experimental signals.

Vanishing gradient problem is an intrinsic drawback of Artificial Neural Networks (ANN) whose theoretical background will be explained in Section 3.3. In order to solve this problem, Guo et al. proposed an adaptive learning rate method for hierarchical adaptive deep convolutional networks for bearing fault diagnosis [29]. They employed a CNN network in order to extract feature patterns of faults from raw vibration signals. They utilized adaptive learning rate, that is, when the loss of the model decreases, the learning rate increases, and vice versa, so the model could be trained by the minimum vanishing gradient problem. Besides, the momentum term was added to the CNN

model so as to overcome the slow convergence and oscillation problems during the training. The model generated good results over Case Western Reserve University Bearing Data Center (CSWRU) [28] bearing dataset compared to deep convolutional neural networks without adaptive learning rate.

In the application of the condition monitoring based maintenance approaches, it is crucial to determine an optimal maintenance schedule based on the monitoring results. In order to increase the efficiency of the processes and the scheduling without risking the devices, fuzzy logic approach can be of a great solution. Sudarsanam applied fuzzy logic approach on compressor failures in petroleum refinery sites based on rider ring wearing and temperature [38]. The study aimed at predicting the life expectancy of the compressor at a four-level scale: Acceptable, Optimum, Caution, and Damage. Baban et al. implemented a similar method on grinding wheels of automated grinder lines [39]. They incorporated vibration and temperature data to obtain remaining life of the grinder in terms of the number of parts they would be able to process before they break down.

Finally, applicability of research to different fields or devices and even reproduction of experiments is a crucial problem in the literature. Transfer learning is created to solve this applicability problem. It can also be used for overcoming data scarcity problems. Guo et al. emphasized the scarcity of labelled data and non-scalability of the fault detection modes [40]. In order to solve these problems, a model named Deep Convolutional Transfer Learning Network (DCTLN) was proposed which consisted of two modules; condition recognition and domain adaptation. The condition recognition module enabled an automatic feature extraction with a deep 1D CNN and classification of the health condition with a Fully Connected (FC) network. In the domain adaptation module, they tried to minimize the distribution differences between the source and target domains by measuring the MMD. The DCTLN model was verified on different vibration signal datasets; CSWRU bearing dataset [28], Intelligent Maintenance System (IMS) bearing dataset, and railway locomotive bearing dataset; thereby, the model could be trained on labelled data from one domain and classify the condition

of a different domain. Zhang et al. worked on transfer learning for bearing fault diagnosis under various conditions such as different loads and speeds [41]. They separated data into two groups as source and target. The source data had more labelled data than the target data; whereas, it had less fault conditions than the target data. They trained a FC model with windowed source data. In order to use the trained weights for the target data, they randomly initialized new parameters since the number of fault conditions were not equal. The model was trained by the target data and the training time and accuracy results were higher than the model without transfer learning. The proposed model was verified by CSWRU vibrational bearing dataset and they got an improvement over different speed and load target data in terms of learning time and accuracy. Han et al. also adopted transfer learning under different speeds and fault conditions [42]. Their diagnosis model combined of CNN and classifier blocks. CNN with various number of layers were used to extract features from raw vibration data and a CNN with five layers was chosen due to its satisfactory result. In order to transfer the weights, they utilized three strategies. While the first strategy focused on the fine-tuning the classifier block with the target data, the second one relied on the fine-tuning of both blocks and the last one depended on tuning convolution and the top of the classification layer. They mentioned that the dataset size and similarity between source and target dataset impacted the choice of the strategies. Their models with different strategies and depths were compared over two datasets acquired from a gearbox and gave promising accuracy results on the target data.

1.4. Organization of the Thesis

The thesis is organized as follows:

- In Chapter 2, the experimental setup used in the thesis is explained in detail. This chapter starts with an explanation of a mechanical setup with a flexible impeller pump in order to emulate manufacturing processes in textile dye houses. The chapter continues with the electrical design of the setup where an acoustic sensor kit is selected and used with a microcomputer to collect FIP recordings.
- In Chapter 3, development of supervised models for failure mode detection is explained. At the beginning of the chapter, feature extraction and data augmentation methods are explained. Then, theoretical backgrounds of Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) are given. This chapter includes development of three different models for failure mode detection with three input types. The chapter is concluded with the experimental results of the model and their discussion.
- In Chapter 4, development of an unsupervised anomaly detection method is given. The chapter starts with theoretical backgrounds for the two most common methods for clustering and continues with their applications. The chapter includes two different inputs for the clustering method, Gaussian Mixture Model (GMM), their experimental results and concludes with a brief discussion.
- In Chapter 5, the thesis work is summarized and the results from two main methods are compared. The results are discussed in terms of their success and with their compliance to the thesis motivation. The chapter is finalized with future work proposals.

2. PROBLEM SETUP

In this chapter, an experimental setup, which emulates production process in textile dye houses is described. With the setup, two of the most common failures of flexible impeller pumps, looseness and cavitation, are focused on. In Section 2.1, mechanical design details of this setup are explained. In order to acquire data from the experimental setup, an acoustic sensor is chosen based on a set of critical features. Section 2.2.1 gives details about acoustic sensor background and considerations on sensor selection process. In Section 2.2.2, an electrical hardware including a single board computer which is used to develop software so as to control data acquisition and predict the condition of the motor is explained. The software development process is also explained in Section 2.2.3. Finally, in Section 2.3, methods that incorporate the experimental setup and form the basis of the thesis are explained briefly.

2.1. Mechanical Setup

Flexible impeller pumps are commonly used in textile dye houses. In order to simulate the actual application of the system on a textile dye house, an industrial manufacturing site is observed. An experimental model was produced by Eliar Company with the support of TUBITAK 1505 project funding. The simulation setup consists of a flexible impeller pump, a small liquid tank, valves, drive trains for transferring liquids, and an inverter running an electrical motor. The experimental setup is depicted in Figure 2.1.

Deterioration in FIPs is mostly derived from cavitation, which is caused by pressure difference between the inlet and the outlet of the pump. The low pressure at the inlet of the pump causes formation of bubbles in the pump body which generates shock waves over the rubber wings of the impeller. These shock waves generate loud audible sounds by which acoustic sensors can detect the failure. In order to collect audio signals from the cavitation mode, the input valve of the pump is closed until

some bubbles are generated due to low pressure caused by the vacuum at the inlet.

Another reason for breakdown of FIPs is wrong installation such as looseness, which may cause imbalance of suction for the pump or dripping. Unlike the cavitation sounds, the looseness mode does not generate major distinctive sounds, similar to the healthy condition. In order to acquire audio signals from the looseness mode, screws on the pump housing are slightly loosened. Along with signals of the cavitation and looseness conditions, data are also collected for the healthy condition so as to compare the failure modes.

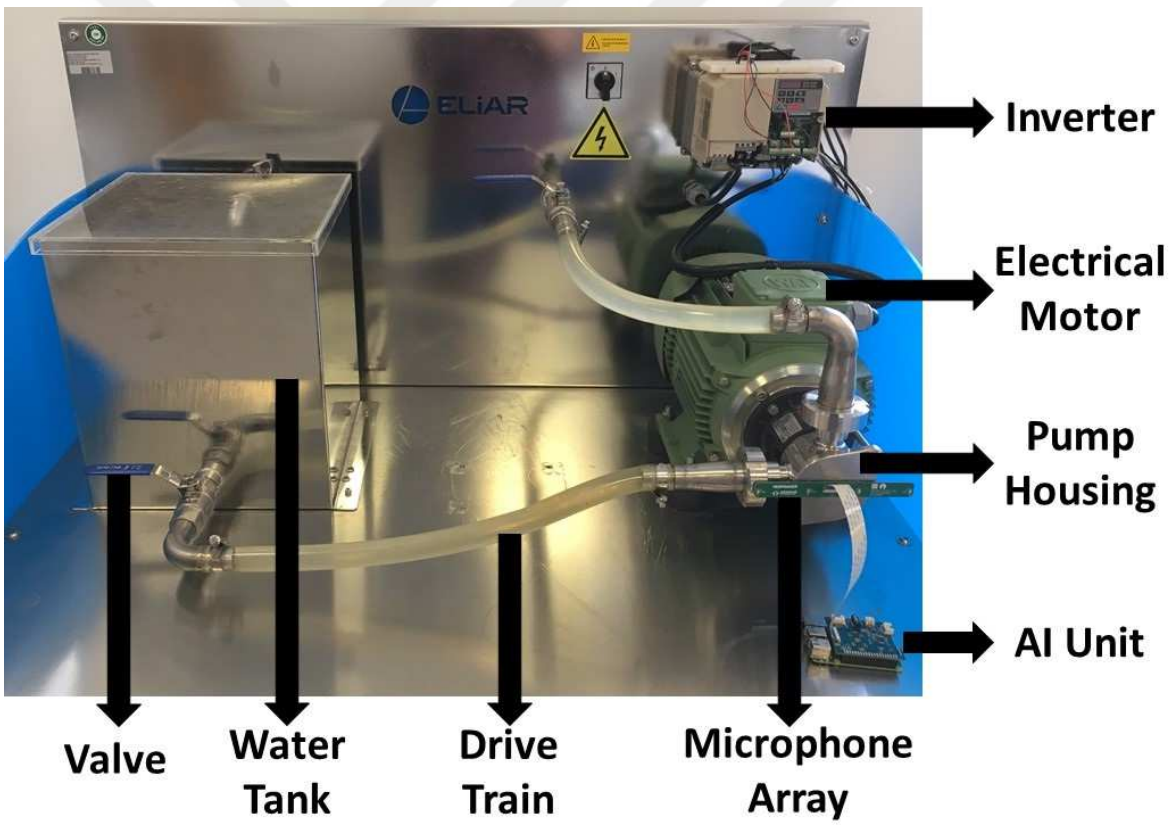

Figure 2.1. Experimental setup.

2.2. Hardware and Software Design

2.2.1. Sensor Selection

Sensors are measurement devices that convert physical quantities into electrical signals. Sensor output signals can be analog or digital depending on design of the sensor. Sensors have a wide range of usage in the industry. To illustrate, vibration sensors can be used for measurement of direction of a drone or fluid viscosity, and acoustic sensors can be used for communication in underwater, smart cities, or factories.

Each sensor has specific features that make them preferable in a specified area. Some of these features are sensitivity, measurement range, accuracy, Signal to Noise Ratio (SNR), and so on [43]. Sensitivity is an important feature to be considered before selecting a sensor. It indicates the ratio of change of the output that of input quantity. It can also be considered as a non-ideal transfer function in a model because the sensitivity can be affected by a wide range of external factors such as temperature, electrical noise, and material properties [44]. The minimum and maximum values that can be measured by a sensor is the input range such that a typical microphone can sense the 20 Hz to 20 kHz range. Another critical feature is accuracy, that is the difference between the expected and measured outputs. In order to compensate low accuracy, a calibration stage can be added for designs or applications. Finally, if the signal to noise ratio of a sensor is low, the measurement result may not be trusted since the noise signal is amplified as well as the output error.

Scope of this thesis embraces condition monitoring of FIPs by means of acoustic sensors. Acoustic sensors are devices. They convert mechanical energy from sounds to electrical signals. Microphones are a type of acoustic sensors that specialize in converting audible sound pressure to digital or analog electrical signals. When sound waves diffuse to the air, vibration of the microphone's thin and flexible diaphragm converts the energy from this vibration into an electrical signal. Ideally, this conversion process is linear and independent from the frequency within the sensor input range.

There are three types of acoustic waves which are differentiated from each other with their frequency range. These are defined as infrasonic waves below 20 Hz, audible waves between 20 Hz and 20 kHz, and ultrasonic waves above 20 kHz. While infrasonic and ultrasonic waves can be measured with specialized sensors, audible waves can be detected by using microphones. Therefore, it is substantial to determine the desired measurement frequency range in order to obtain proper recordings of the acoustic waves to be observed.

Sounds can diffuse through various media such as fluids like air or water or solids like metals or soil. Therefore, measurement of sounds takes a lot of attention from various areas such as medical applications to investigate fetus growth or geological applications to sense earthquakes. Another usage of acoustic sensors is condition-based monitoring for rotating machines. It is possible to detect failures of a motor with acoustic sensors since machines may generate different sounds depending on their status. Unlike vibration sensors, acoustic sensors are more sensitive to vibrations, but preprocessing of acoustic data is more complex. Although acoustic sensors provide advantages on real time fault diagnosis, the source signal may be attenuated until reaching the sensor. Thus, acoustic sensors should be as close to the source as possible so as to collect intelligible data.

Acoustic sensors used for condition-based monitoring must have high bandwidth and sensitivity. Industrial acoustic sensors are suitable for data collection, but their cost is too high for the type of dyehouse applications that we envision. Therefore, microphones, which can detect audible frequency range of humans from 20 Hz to 20 kHz, are used for data collection.

In order to record sounds of the flexible impeller pump, "ReSpeaker 4 - Mic Linear Array Kit for Raspberry Pi" is selected as the acoustic sensor. This commercial sensor kit consists of four omnidirectional Micro Electromechanical System (MEMS) microphones in a linear array placement. Along with the suitability of its sensor elements to our application, it also has built-in Analog to Digital Converters (ADC) and

direct compatibility with Raspberry Pi. These two additional features are of great importance to reduce the effort for analog and digital signal preprocessing steps. Acoustic and electrical characteristics of the sensor kit are given in Table 2.1 [45].

Table 2.1. Microphone specifications [45].

	Limits (nominal)	Unit	Condition
Directivity	Omni directional		
Sensitivity	-32	dB	@1kHz ref 1V/Pa
Operation voltage	1.5∼3.6	V	
Sensitivity loss across supply voltage	No change across the voltage range	dB	
Signal to noise ratio	66	dB	20 kHz bandwidth, A-weighted
THD	0.2	%	94dB SPL @1kHz S =Nom, Rload >2k
AOP	123	dB SPL	10% THD @1kHz S =Nom, Rload >2k
Out impedance	200	ohm	@1kHz
DC Output	0.7	V	
PSRR	70	dB	200mVpp sine wave @ 1 kHz, VDD = 1.8V
PSR	-100	dBV(A)	100 mVpp square wave @ 217 Hz, VDD = 1.8V A-weighted
Current consumption	150	μA	

2.2.2. Processing Unit

Raspberry Pi Model 3 is selected as the main processor of the thesis due to its low cost and portability. This microcomputer family is developed for educational purposes in the United Kingdom, in 2012, gaining popularity day by day from the industry as well. This is partly because of its compatibility with various third party hardware such as accelerometer kits, pressure sensors, display kits developed by commercial and open source communities. Nowadays, it is widely used in different applications from high school level education to huge research projects such as NASA's Open Source Rover [46]. Current last generation of the Raspberry Pi family when the thesis work started, Raspberry Pi 3 B+, was selected as the standalone computing block of the project.

Raspberry Pi 3 has a 1.4 GHz 64-bit quad-core processor with an integrated ARM-compatible Central Processing Unit (CPU), on-chip Graphical Processing Unit (GPU), 802.11n WiFi, Bluetooth, a micro SD port, HDMI, and USB boot capabilities [47]. A micro SD card is used as storage of an operating system and program memory. A customized Linux based operating system, Raspbian, is supported and supplied by Raspberry Pi's official website [48]. Besides, the board can be controlled remotely through SSH connection, which can be of importance in monitoring applications.

Another reason for why Raspberry Pi was chosen is its compatibility with peripheral sensors such as vibration, or acoustic sensors. The model has 40 interface pins with supported communication protocols such as I2C, SPI, UART, and so on. In addition, it enables user interface via HDMI without interrupting other protocols and applications. Thus, the sensor kit was selected by considering its compatibility with Raspberry Pi. The acoustic sensor with 4 microphones supports 8 input ad 8 output channels in the Raspbian system. The sensor kit's ADC board fits the 40 pins of Raspberry Pi thanks to its compatible design.

2.2.3. Software Development

The thesis work includes two parts of software development stages. The first is developed and run on Raspberry Pi that is explained in the previous section. This software is developed to obtain data from the sensor kit and send them to the computer where the second stage of software development is implemented. On a Personal Computer (PC), machine learning algorithms are developed and run by means of its powerful CPU and GPU, in contrast to Raspberry Pi. Both software are developed with Python.

Python is a programming language created by Guido van Rossum in 1991. There are two versions of it, known as Python 2 and Python 3; however, Python 2 had been supported until 2020 due to the release of Python 3, in 2008. The language is widely used by large entities such as Google, NASA and Facebook. The reasons why many of them prefer using Python are rich library ecosystem, easy readability, good visualization, and support by a large community. Besides, specialized machine learning and deep learning packages are available for Python. For example, Pytorch and TensorFlow are two of very well-known deep learning libraries that support the Python language. While Pytorch was released in 2018 by Facebook, TensorFlow was launched in 2017 by Google. In this thesis work, Python 3 language was chosen for the software development and both libraries were used for the development of neural networks.

2.3. Applied Methods

The experimental setup, described in Sections 2.1 and 2.2, in detail, is developed to facilitate this thesis work. The experimental setup is used for developing the models and methods, described in the rest of the thesis.

In Chapter 3, a supervised learning model based on Convolutional Neural Networks is developed to predict failure conditions of Flexible Impeller Pumps. In Section

3.1, feature extraction methods applied on the experimental data are described. These methods are implemented to obtain discriminative characteristics of the acoustic signals obtained from the setup. The investigated methods include Time Domain, Frequency Domain, Time-Frequency Domain and Wavelet Domain features respectively. Additionally, data augmentation techniques are also applied on the experimental setup dataset in order to increase the generalization performance of the models by enriching the data with the methods described in Section 3.2. This chapter continues with explanation of the theoretical background of Artificial Neural Networks and Convolutional Neural Networks in Sections 3.3 and 3.4, respectively. Implementation of these methods with various features extraction techniques are explained in Section 3.5, in detail.

In Chapter 4, an unsupervised learning method based on Gaussian Mixture Model is proposed. By this method, it is aimed to adapt the theoretical work developed on the experimental setup to the real-life scenarios by eliminating the need for obtaining broken device data. The Gaussian Mixture Model is a one-class classification method where only healthy signals from the experimental setup are used for training and clustering. The test signals are compared to the healthy clusters and the outliers are labelled as unhealthy. Theoretical basis of this model is explained in Sections 4.1 and 4.2. Implementation of the model with MFCC and PSD features are described in Sections 4.3.1 and 4.3.2, respectively.

3. FAILURE MODE DETECTION WITH SUPERVISED LEARNING

After acquisition of data consisting of three flexible impeller pump conditions, healthy, cavitation, and looseness, from the experimental setup by means of 3 microphones, the data should be prepared for classification. This can be done by manual feature extraction. In Section 3.1, extraction techniques for these features are explained. Section 3.2, gives theoretical background about data augmentation techniques which are beneficial to increase accuracy by adding variations to the dataset. In this chapter, supervised learning is utilized to find a map between the features and the FIP conditions by means of labelled data. Depending on the conditions, supervised model learns connections from features to labels in terms of trainable neurons. In Section 3.3, intuitions about what a neuron is and how it learns are provided. Convolutional neural network models, a type of supervised model, are utilized for classification of the conditions. Theory of CNN is explained in Section 3.4. The models are trained by three types of different features and the results are given in Section 3.5, in detail. At the end of the chapter, comparison of the experimental results and remarks on the outputs are given.

3.1. Feature Extraction

The acoustic data used in this study are collected by using 3 microphones from a linear array kit mounted on a fixed position on the pump body. 8 different flexible impellers are used for the recordings. For each impeller, the pump is operated and its sounds were recorded twice for all 21 combinations of the following rotational speeds and pump conditions:

- 7 speed levels: 250, 300, 400, 500, 600, 700, and 800 rpm.
- 3 pump conditions: healthy, cavitation, and looseness.

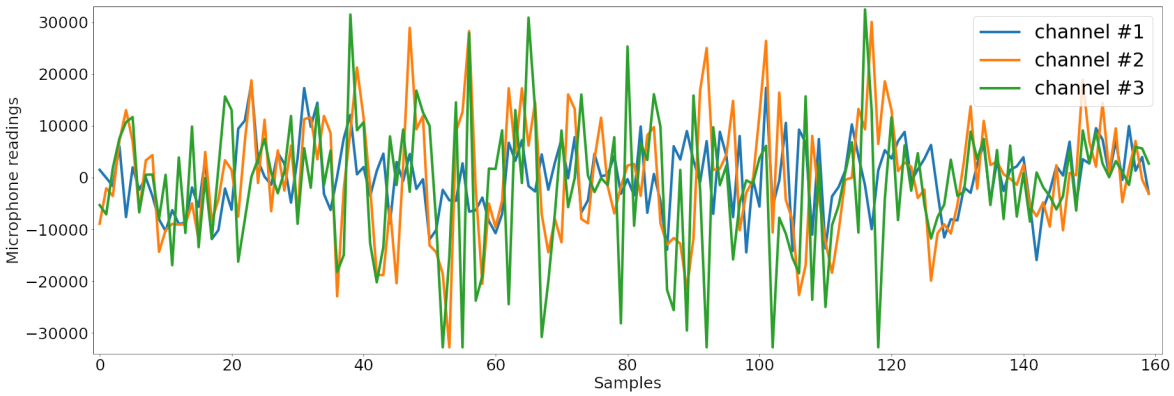

Figure 3.1. A random 10 ms interval from a healthy recording at 600 rpm.

Consequently, the dataset consists of 8×42 voice recordings where each recording is collected from 3 channels (3 microphones) for 3 minutes. Sampling rate for each microphone output is chosen as 16 kHz. Figure 3.1 shows a randomly chosen 10 ms interval from the recording of a healthy pump rotating at 600 rpm.

The microphone kit used for the recordings converts the analog readings into 16-bit signed integers. Therefore, the maximum and minimum values that can be read are 32767 and -32768. The readings beyond these limits are simply clipped.

Despite extraordinary developments in deep learning methods, many researchers still extract and use time-domain, frequency domain and time-frequency domain features relying on signal processing techniques. Since the traditional methods highly depend on feature extraction and feature selection, the acquired signal should be analyzed in terms of time and/or frequency features as explained in Section 1.3. Along with the traditional methods such as support vector machine (SVM), multi-layer perceptron (MLP), and k-nearest neighbor (k-NN) clustering, extracted features could also be beneficial for deep models when combined with unsupervised features. Thus, acoustic signals obtained from emulated production process of textile dye houses have been investigated with respect to time, frequency, time-frequency, and wavelet domain features in the following sections.

3.1.1. Time Domain Feature Extraction

In this section, the three fault modes, healthy, cavitation, and looseness, have been analyzed in terms of time domain statistical features, mean, variance, standard deviation, mean square (MS), root mean square (RMS), skewness, and kurtosis. The statistical features are calculated as

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \tag{3.1}$$

$$\sigma^{\mathbf{2}} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})^2 \tag{3.2}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}}_i)^2} \tag{3.3}$$

$$\text{MS} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i^2 \tag{3.4}$$

$$\text{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i^2} \tag{3.5}$$

$$\text{Skewness} = \frac{1}{N}\sum_{i=1}^{N}\frac{(\mathbf{x}_i - \bar{\mathbf{x}})^3}{\sigma^3} \tag{3.6}$$

$$\text{Kurtosis} = \frac{1}{N}\sum_{i=1}^{N}\frac{(\mathbf{x}_i - \bar{\mathbf{x}})^4}{\sigma^4} \tag{3.7}$$

where N refers to the total number of sample points in the signal x, x_i is i^{th} sample point of the signal, $\bar{x}$ indicates the mean of the signal, σ^2 is the variance, and σ represents the standard deviation.

Acoustic signals were acquired in 16 sets of recordings with double recordings for each of the 8 impellers. Each set consists of three failure modes in seven different rotational speeds. Each recording is taken from the three microphones on the sensor kit simultaneously. The average of signals from each microphone is taken as one recording. One set, which consists of 21 recordings, was analysed in terms of statistical features in order to find its discriminative features. Extracted feature values can be shown

separately for healthy, cavitation, and looseness modes in Tables 3.1, 3.2, and 3.3, respectively.

When the data in the tables are examined, it is seen that Kurtosis and Skewness are the only two promising features in terms of discrimination of the failure states [49]. During our experiments with classification models, using only Kurtosis and Skewness did not achieve considerable accuracy. In order to benefit from the discriminative ability of these two features, they are added to the other features employed in Section 3.5, in training Convolutional Neural Networks. However, their effect to the increase in accuracy was insufficient considering the processing power and time requirement they brought to the system. Therefore, in the final models, no time domain feature is used for failure mode classification.

Table 3.1. Time features of averaged signals in healthy condition.

Healthy							
RPM	**Mean**	**Variance**	**Standard Deviation**	**Mean Square**	**Root Mean Square**	**Skewness**	**Kurtosis**
250	**-7.58 E-5**	**1.15 E-3**	**3.35 E-2**	**1.15 E-3**	**3.35 E-2**	**-1.14 E-2**	**5.25 E-1**
300	**-7.44 E-5**	**1.81 E-3**	**4.18 E-2**	**1.81 E-3**	**4.18 E-2**	**8.84 E-3**	**2.53 E-1**
400	**-7.09 E-5**	**7.99 E-3**	**8.72 E-2**	**7.99 E-3**	**8.72 E-2**	**-4.37 E-2**	**6.04**
500	**4.34 E-6**	**1.73 E-2**	**1.3 E-1**	**1.73 E-2**	**1.3 E-1**	**-2.82 E-2**	**3.86**
600	**1.28 E-4**	**6.71 E-2**	**2.54 E-1**	**6.71 E-2**	**2.54 E-1**	**-1.72 E-2**	**1.97**
700	**-7.34 E-4**	**1.66 E-1**	**4.05 E-1**	**1.66 E-1**	**4.05 E-1**	**-2.04 E-3**	**1.44 E-1**
800	**-1.88 E-3**	**2.84 E-1**	**5.31 E-1**	**2.84 E-1**	**5.31 E-1**	**1.19 E-2**	**-6.18 E-1**

Table 3.2. Time features of averaged signals in cavitation condition.

Cavitation							
RPM	**Mean**	**Variance**	**Standard Deviation**	**Mean Square**	**Root Mean Square**	**Skewness**	**Kurtosis**
250	**-9.08 E-5**	**1.99 E-2**	**1.39 E-1**	**1.99 E-2**	**1.39 E-1**	**3.96 E-2**	**1.22 E-1**
300	**-6.21 E-5**	**5.31 E-2**	**2.28 E-1**	**5.31 E-2**	**2.28 E-1**	**6.1 E-3**	**4.56**
400	**-6.53 E-4**	**1.35 E-1**	**3.65 E-1**	**1.35 E-1**	**3.65 E-1**	**-2.49 E-3**	**1.23**
500	**2.38 E-3**	**3.68 E-1**	**6.04 E-1**	**3.68 E-1**	**6.04 E-1**	**-1.1 E-2**	**-8.78 E-1**
600	**2.13 E-4**	**4.9 E-1**	**6.98 E-1**	**4.9 E-1**	**6.98 E-1**	**1.52 E-3**	**-1.28**
700	**4.83 E-3**	**6.25 E-1**	**7.9 E-1**	**6.25 E-1**	**7.9 E-1**	**-1.13 E-2**	**-1.56**
800	**2.28 E-3**	**6.37 E-1**	**7.98 E-1**	**6.37 E-1**	**7.98 E-1**	**-3.8 E-3**	**-1.59**

Table 3.3. Time features of averaged signals in looseness condition.

Looseness							
RPM	**Mean**	**Variance**	**Standard Deviation**	**Mean Square**	**Root Mean Square**	**Skewness**	**Kurtosis**
250	**-7.44 E-5**	**1.69 E-3**	**4.06 E-2**	**1.69 E-3**	**4.06 E-2**	**3.2 E-2**	**3.66 E-1**
300	**-7.42 E-5**	**2.51 E-3**	**4.96 E-2**	**2.51 E-3**	**4.96 E-2**	**-1.59 E-2**	**1.87 E-1**
400	**-7.5 E-5**	**7.36 E-3**	**8.32 E-2**	**7.36 E-3**	**8.32 E-2**	**4.01 E-2**	**1.1 E-1**
500	**-7.18 E-5**	**1.27 E-2**	**1.12 E-1**	**1.27 E-2**	**1.12 E-1**	**-8.78 E-2**	**5.17 E-1**
600	**2.16 E-3**	**1.17 E-2**	**3.39 E-1**	**1.17 E-1**	**3.39 E-1**	**-3.03 E-2**	**9.38 E-1**
700	**-1.79 E-3**	**1.97 E-1**	**4.4 E-1**	**1.97 E-1**	**4.4 E-1**	**1.35 E-2**	**-1.14 E-1**
800	**1.05 E-3**	**3.97 E-1**	**6.25 E-1**	**3.97 E-1**	**6.25 E-1**	**-1.62 E-3**	**-1.06**

3.1.2. Frequency Domain Feature Extraction

Similar to time-domain features, frequency domain features can also be employed for detection of rotating machine failures. Each signal can be considered as a combination of cosine and/or sine waves with various frequencies and amplitudes. In order to extract periodicity features in frequency domain, each acquired time-domain acoustic signal is converted to frequency-domain by utilizing Discrete Time Fourier Transform:

$$\mathbf{X[k]} = \sum_{n=1}^{N-1} \mathbf{x[n]e^{-j\frac{2\pi}{N}kn}} \tag{3.8}$$

where N refers to the sample length of one period of the discrete time signal, $x[n]$, k is the frequency component index where $k \in [0, 1, ..., N-1]$.

In this thesis, mean frequency of power spectrum and its magnitude were extracted in a time segmented frame [50]. In other words, each of the three minutes of acoustic signals is divided into 1-second frames with 50% overlap. Each segment was converted to frequency domain via DTFT. Then, the power spectrum (PS) is calculated as:

$$\mathbf{PS} = \left| \sum_{k=1}^{N-1} \mathbf{x[k]} \right|^2 . \tag{3.9}$$

Time-domain, frequency-domain and power spectrum of two signals, collected in 300 rpm and 800 rpm from the healthy mode, are shown in Figure 3.2.

Mean frequency (MF) of the power spectrum is calculated as

$$\mathbf{MF} = \frac{\sum_{j=1}^{M} \mathbf{f_j PS_j}}{\sum_{j=1}^{M} \mathbf{PS_j}} \tag{3.10}$$

where f_j defines the frequency value of power spectrum of a signal at the frequency

bin j, PS_j refers to the power spectrum at the frequency bin j, and M represents the length of the frequency bin j.

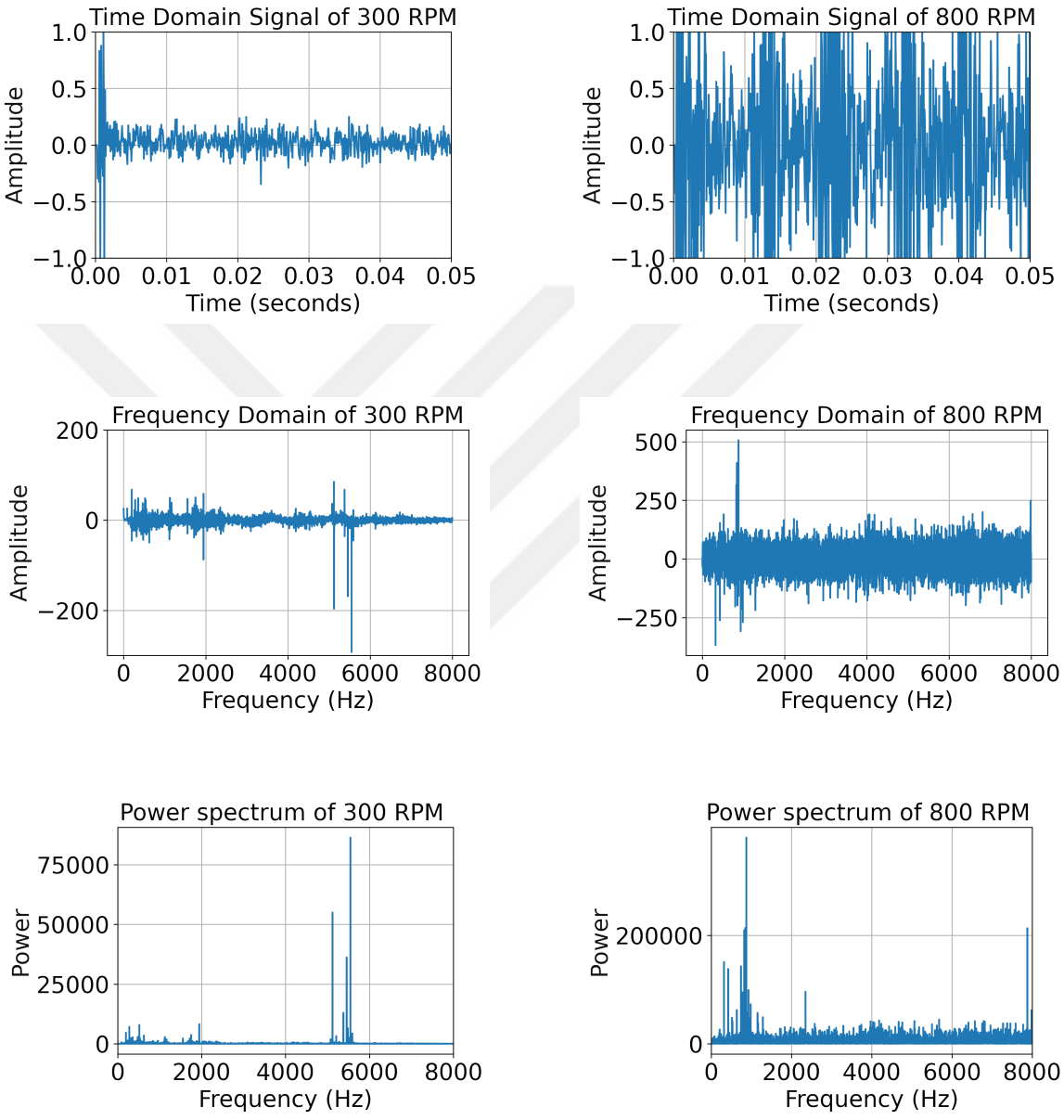

Figure 3.2. Time domain, frequency domain, and power spectrum representation of recordings at 300 rpm and 800 rpm.

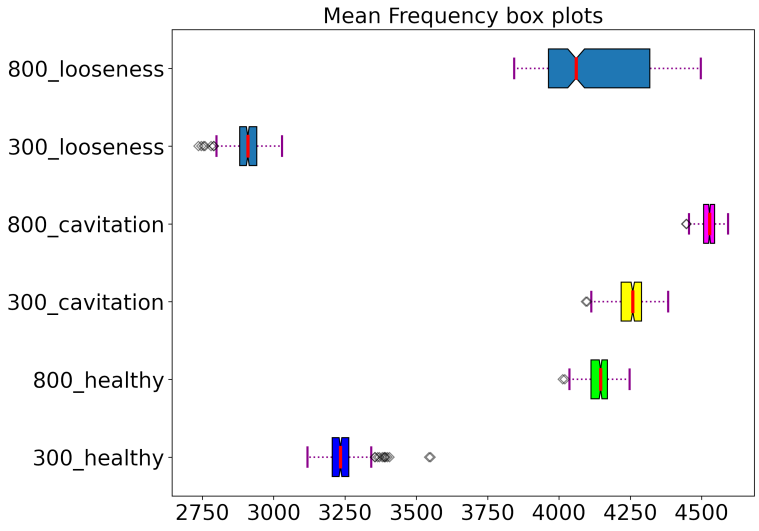

Figure 3.3. Mean frequencies in box plot.

In order to compare the failure modes, 300 rpm and 800 rpm recordings from all modes were chosen to investigate. For 1-second windows with 0.5-second overlapping frames, mean frequency of the signals were calculated and their box plots are shown in Figure 3.3. Besides, signal power for corresponding mean frequencies of selected frames were also depicted in Figure 3.4.

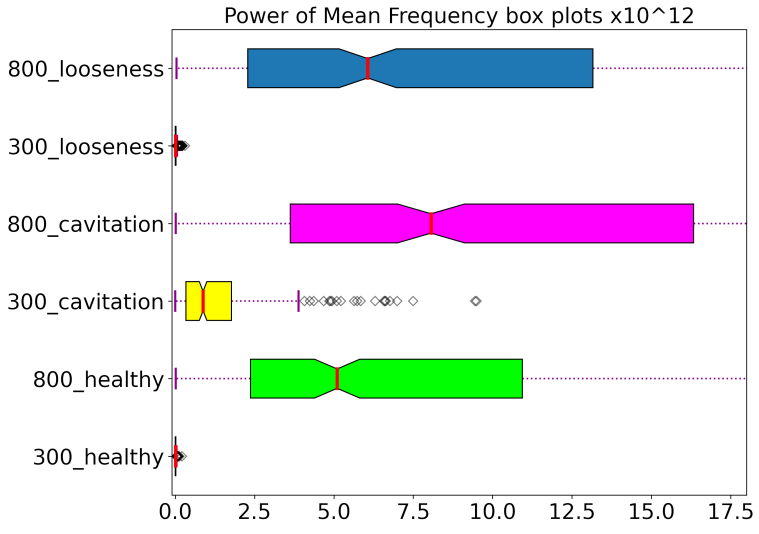

Figure 3.4. Power of mean frequencies in box plot.

In addition to power spectrum calculation, Welch's Method, an estimation operation of Power Spectral Density (PSD) of a signal, is employed. Time series data are divided in successive blocks with a predefined overlapping. The method calculates the average power spectrum of each block. This method reduces noise in the selected windows. The Welch's method calculates the estimated power spectral density using

$$\mathbf{x_m(n)} = \omega(n)x(n+mR) \tag{3.11}$$

$$\mathbf{P_{x_m,M}}(\omega_\mathbf{k}) = \frac{1}{M}\left|\mathrm{DTFT}(x_m)\right|^2 = \frac{1}{M}\left|\sum_{n=0}^{M-1} x_m(n)e^{-j\omega n}\right|^2 \tag{3.12}$$

$$\hat{\mathbf{S}}_\mathbf{x}^\mathbf{W}(\omega_\mathbf{k}) = \frac{1}{K}\sum_{m=0}^{K-1} P_{x_m,M}(\omega_k) \tag{3.13}$$

where $w(n)$, $n \in [0, M-1]$ shows the rectangular window, $m \in [0, K-1]$ denotes the number of windows in the discrete signal $x(n)$, R refers to the window overlapping size, K indicates the total number of available frames, $P_{x_m,M}(\omega_k)$ is the m^{th} window's periodogram, and $\hat{S}_x^W(\omega_k)$ is the average of periodograms through time, the estimated power spectrogram. The PSD and estimated PSD graphs of 300 rpm healthy mode signal are shown in Figure 3.5 and Figure 3.6.

Similarly, mean frequencies and powers calculated by the Welch's Method are also shown as a box plot in Figure 3.7 and Figure 3.8. The Welch's method is implemented and used to obtain Power Spectral Density as an input to the Anomaly Detection algorithm explained in Section 4.3.2.

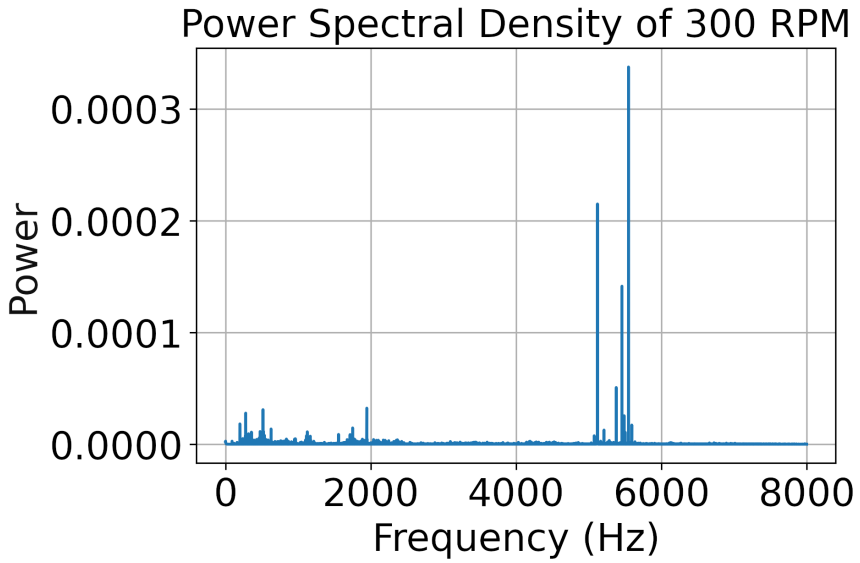

Figure 3.5. Power spectral density of healthy pump at 300 rpm.

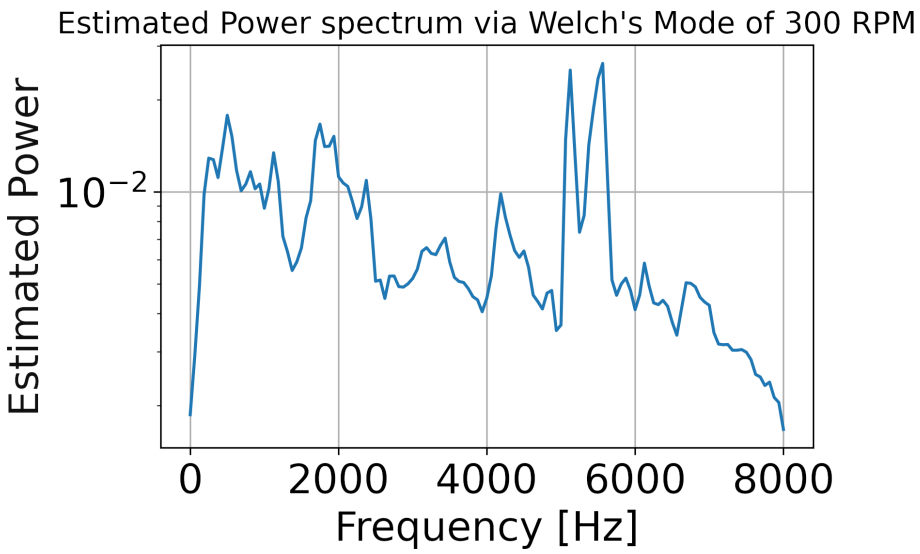

Figure 3.6. Estimated power spectrum via Welch mode of healthy pump at 300 rpm.

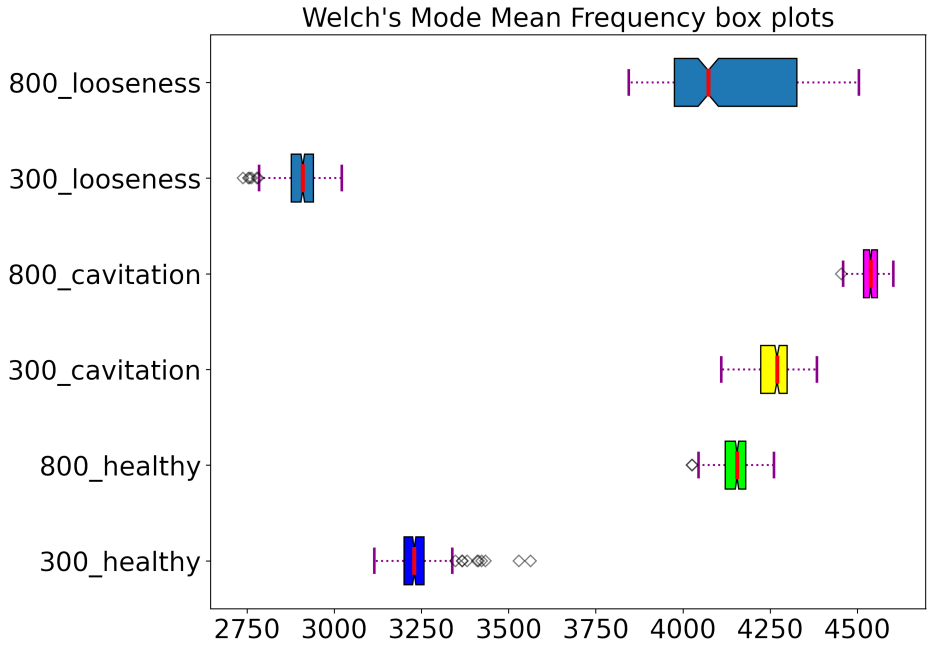

Figure 3.7. Welch mode mean frequencies in box plot.

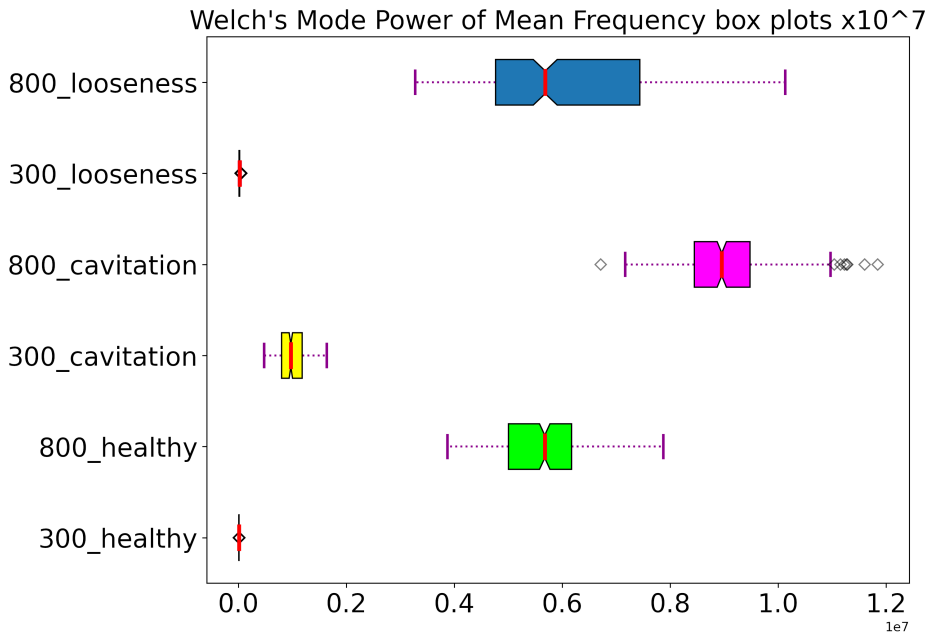

Figure 3.8. Welch mode power of mean frequencies in box plot.

3.1.3. Time-Frequency Domain Feature Extraction

,Mel-frequency cepstrum coefficients are widely utilized as features for classifying audio signals. Therefore, they are adopted for acoustic feature extraction from the pump sound recordings in this study. MFCC includes filters, which are linear below 1 kHz and logarithmic over 1 kHz, that are designed based on human hearing system. These filters imitate human hearing by the fact that humans can detect changes in sounds with a lower sensitivity as the frequency increases. Although MFCC is mostly used for speech recognition systems, it can also be used for condition monitoring [51]. MFCC is generally implemented by four main steps, whose block diagram is given in Figure 3.9.

Commonly, the first step of voice recognition systems is pre-filtering that is based on a high pass filter (HPF) which amplifies high-frequency regions. Since human voice signals intrinsically have low spectral power in the high frequencies, HPF amplifies this region to obtain a more uniform FFT signal such that their effect is not neglected [52].

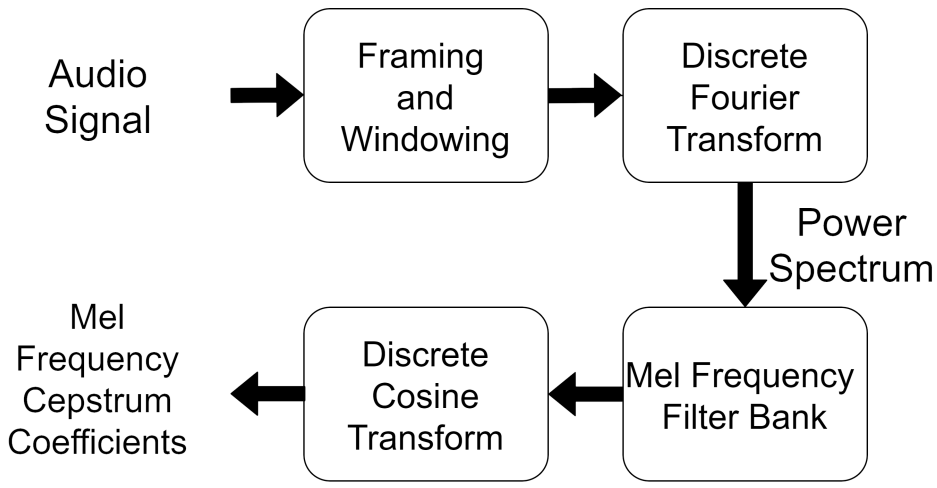

Figure 3.9. MFCC block diagram.

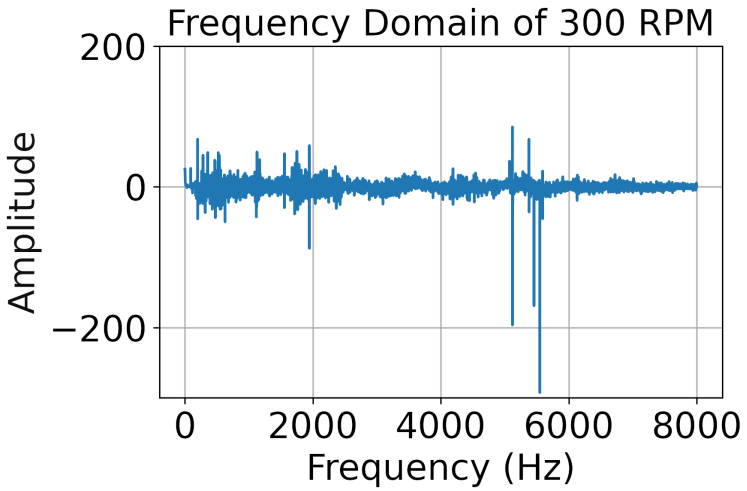

Figure 3.10. FFT plot of pump signal at 300 rpm.

However, when the audio signals of the pump are examined in the frequency domain, it is observed that the signal has high amplitude components in the high frequency range. FFT signals acquired from the pump via the acoustic sensors in 300 rpm are given in Figure 3.10. Thus, if the recordings were passed through a high pass filter, the low frequency range with relatively low amplitude would be attenuated further. As a result of this, high pass filtering step is not used for signal conditioning.

The acquired audio signal is non-stationary, therefore it should be windowed so as to obtain somewhat stationary sections. With the help of an optimal window size, the discrete fourier transform (DFT) can be obtained without losing the envelope of the signal. The window size is usually selected in a range from 15 ms to 50 ms depending on experimental results with various overlapping frame sizes. Each frame is multiplied with a Hamming window to preserve the continuity between the first and the last points within frames. The Hamming window equation is given as [53]

$$w(n) = 0.54 - 0.46\cos(2\pi n/N) \tag{3.14}$$

where n is the sample index, and N is the frame length. The resulting Hamming window plot is shown in Figure 3.11.

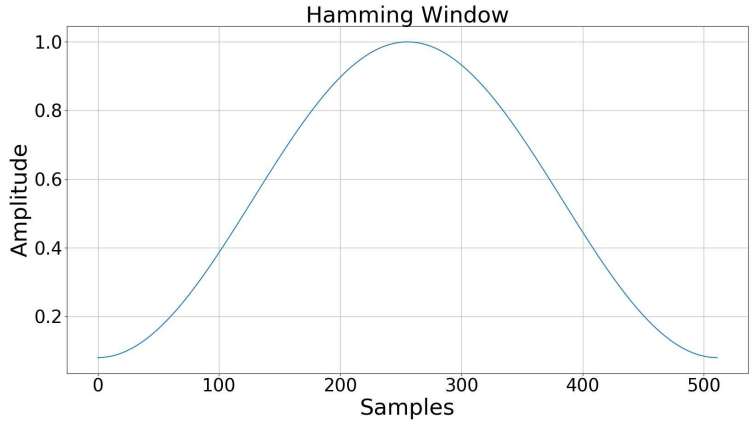

Figure 3.11. Hamming window.

In the next step, DFT is calculated for each frame and then the magnitude of the DFT signals is squared to obtain the power spectrum of the signals. The equation is given as

$$P = \frac{|FFT(x_i)|^2}{N} \tag{3.15}$$

where P denotes the power spectrum of the windowed signal; x^i is the i^{th} window in the audio signal, and N is the frame length.

The DFT signals are passed through a set of triangular band pass filters which are equally spaced by Mel frequencies. The Mel scale calculates the changes in the tone of a sound in a way similar to the human hearing system. The equation used to convert frequency scale to mel scale is

$$m = 2595 \log(1 + \frac{f}{700}) \tag{3.16}$$

where the frequency f is in Hertz and the mel scale m is in Mels.

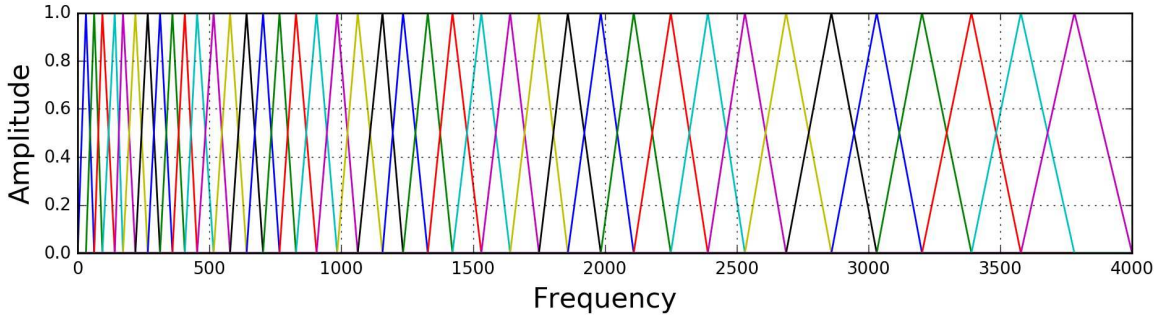

Figure 3.12. Mel filters.

Amplitude of each filter is calculated as

$$H_m(k) = \begin{cases} 0, & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)}, & f(m-1) \leq k < f(m) \\ 1, & k = f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)}, & f(m) < k \leq f(m+1) \\ 0, & k > f(m+1). \end{cases} \tag{3.17}$$

The number of filters used can change in a range from 10 to 100 depending on the user preferences. Obtained filters along with the Mel scale is shown in Figure 3.12. Afterwards, discrete cosine transform is applied on the band-pass filtered signal in order to collect the predetermined number of coefficients. Utilizing a large number of coefficients is not useful due to the fact that increased similarity among them causes less and less utility from the additional computational work. Generally, using 13 - 15 features is considered sufficient to represent the distinctive features [54].

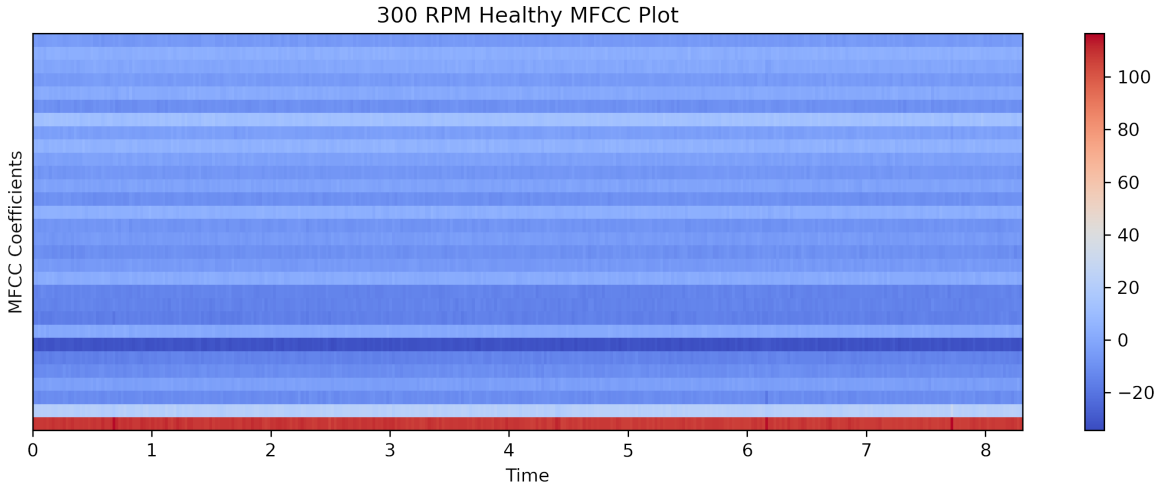

Figure 3.13. 300 rpm healthy MFCC coefficients.

MFCC coefficients obtained from healthy, cavitation, and looseness modes at 300 rpm are depicted in Figures 3.13, 3.14 and 3.15, respectively. These features are used as inputs of both supervised and unsupervised learning methods implemented Chapters 3 and 4, respectively.

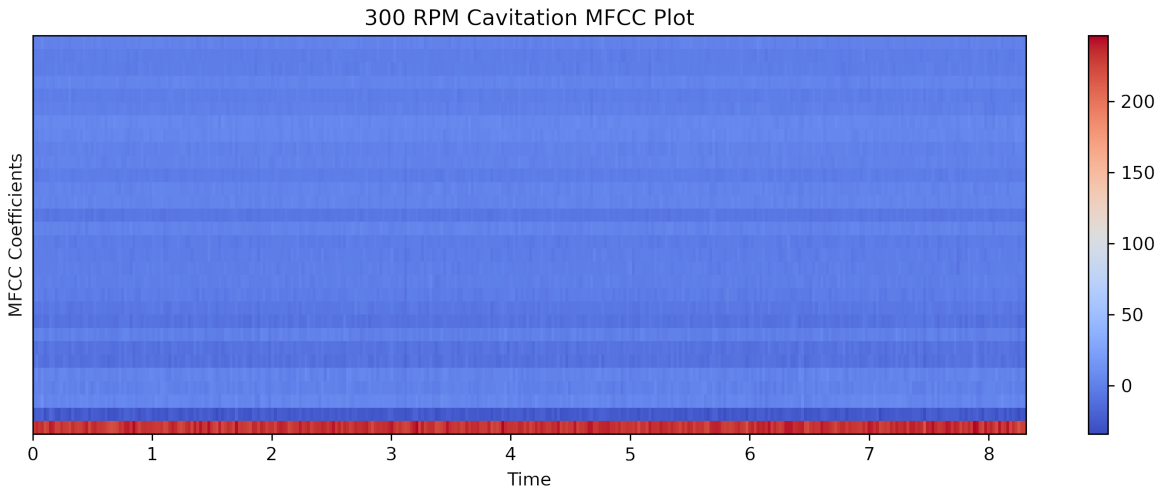

Figure 3.14. 300 rpm cavitation MFCC coefficients.

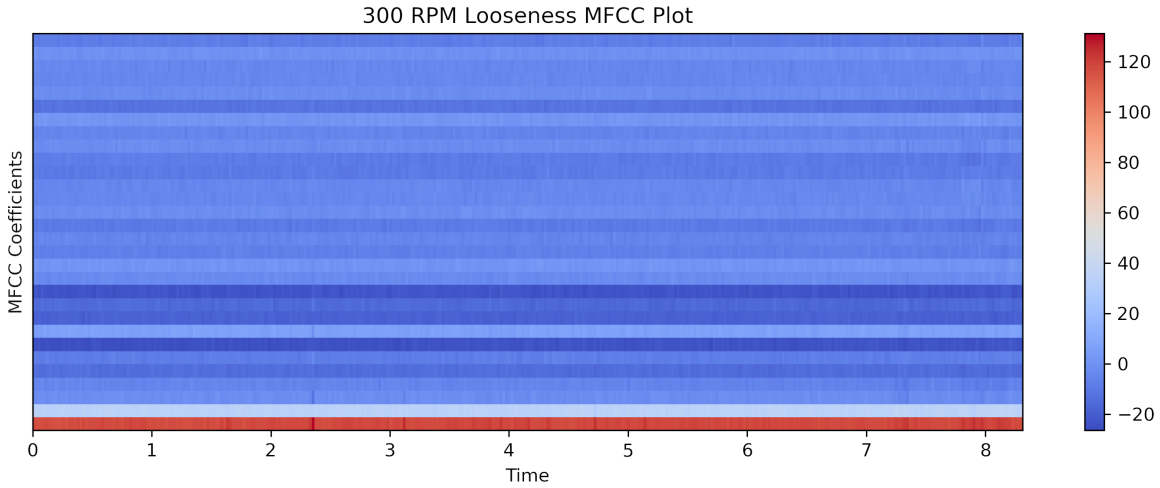

Figure 3.15. 300 rpm looseness MFCC coefficients.

A spectrogram is another time-frequency feature of an audio signal. It represents audio signals in a visually perceivable format, which shows the spectral energy changes over time. A spectrogram is structured as horizontal axis representing the time and vertical axis representing the frequency. Besides, the third dimension or color axis indicates the energy of the spectral signal at a particular time & frequency point.

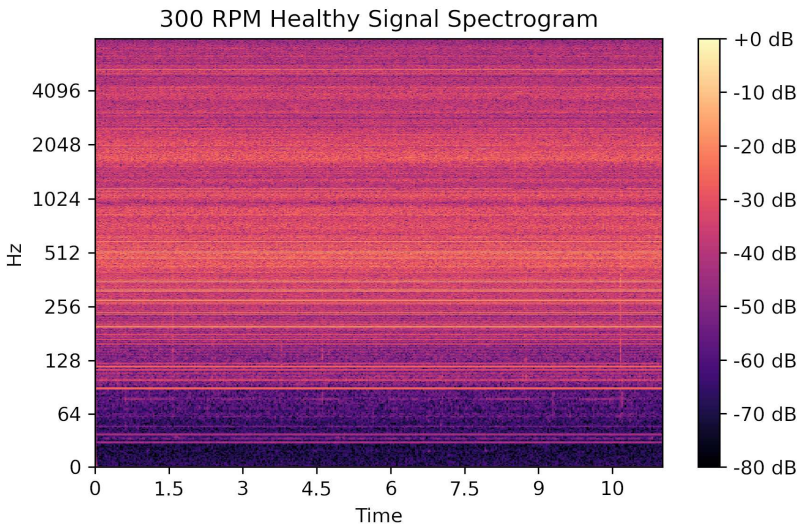

Figure 3.16. 300 rpm healthy signal spectrogram.

For the image representation, the signal is divided into predefined windows with overlap. Then, short-time Fourier transform is obtained for each window, and square of their magnitude represents the energy content of the signal. Taking logarithm of both y-axis and color axis provide the representation of energy in decibel form, and the obtained image is called a spectrogram. The spectrogram of healthy pump signal at 300 rpm with 50% overlapping 1-second windows is shown in Figure 3.16.

In addition to spectrogram, Mel-spectrogram is also used to convert audio signals into an image form. Similarly, the signal is separated into consecutive windows with optional overlapping and each window is converted to the frequency domain. Unlike the spectrogram, each window is filtered by Mel-filters which mimics the human ear features. Energy of each window corresponding to Mel-filters represented in decibel format is shown in color axis. The Mel-spectrogram of the 300 rpm healthy signal is shown in Figure 3.17. Mel spectrogram is used as the input of Spectrogram-CNN model discussed in Section 3.5.2.

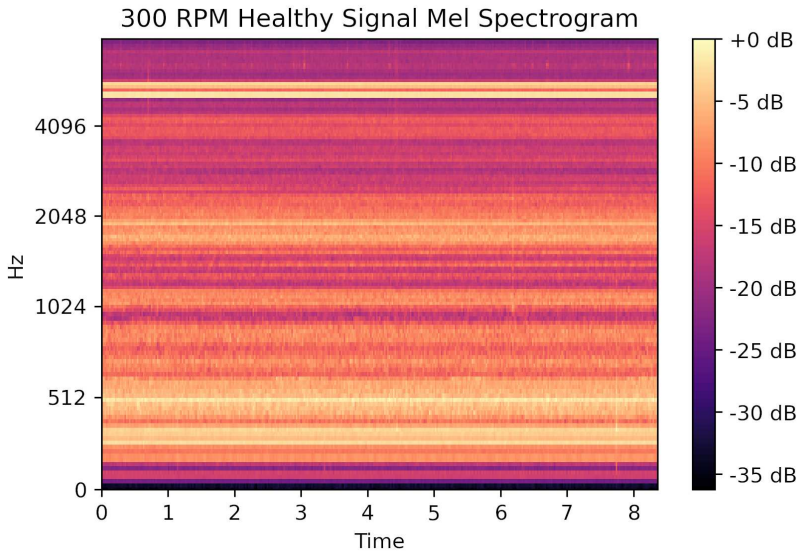

Figure 3.17. 300 rpm healthy signal Mel-spectrogram.

3.1.4. Wavelet Domain Feature Extraction

Although Fourier Transform (FT) decomposes time signals into their frequency components with corresponding amplitudes with the help of cosine and/or sine waves, it is not suitable to capture instant changes on the signal over time. In other words, the instant changes such as bursts in a signal are not periodic over time so the periodic components of the signal suppress the non-periodic events. Therefore, their frequency content may not be detected and represented by the FT. Signals acquired from the FIP are not stationary; thus, it is important to detect abrupt changes in the signal in order to find failure patterns. One solution is to adopt Short Time Fourier Transform (STFT) which is based on FT application to smaller windows of a signal. Even if short windowed signals enable us to capture high frequency features, having smaller windows prevent capturing low frequency features. This deficiency is based on the uncertainty principle in signal processing

$$\Delta t \Delta f \geq \frac{1}{4\pi} \tag{3.18}$$

where Δt and Δf represent the deviations of the signal in time and frequency domains, respectively. That is to say, the frequency and time resolution of the signal have negative correlation. When the window size is increased, the high frequency signals will be missed; otherwise, when the window size decreased, the low frequency signals will be lost.

In order to obtain both time and frequency features, wavelet transformation is considered as a suitable solution. Continuous wavelet transform is calculated as

$$X_w(a,b) = \frac{1}{|a|^{\frac{1}{2}}} \int_{-\infty}^{\infty} x(t)\psi\left(\frac{t-b}{a}\right) dt \tag{3.19}$$

where $x(t)$ is an input signal. ψ represents one of the mother wavelets (window functions) such as Gaussian, Mexican, or Morlet wavelets, which are shown in Figure 3.18. a indicates the scaling of the wavelet signal, and b is the amount of time shift. Each

scaled mother wavelet is convolved with the input signal to capture the specific frequency changes. While the larger scale helps to capture high frequency changes in the input signal, smaller scales provide detection of the low frequency changes. Furthermore, a right choice of the mother wavelet is based on the similarity between the signal and the wavelet function. Similar to the spectrogram, the energy changes over time in frequency domain can be shown in the particular scale factor, which is called as scaleogram or scalogram.

According to previous works of Demircan with the same experimental setup, wavelet domain features proven effective on vibration sensor data [55]. In the light of their work and according to our experiments with wavelet domain features on acoustic signals, the best results are achieved with the Shannon Wavelet, which is shown in Figure 3.19. This shows that Shannon Wavelet resembles the acquired signal from the FIP setup the best among others. However, our experiments with Shannon Wavelet did not yield sufficient accuracy and the experiments were stopped. Scaleogram of 300 rpm signal in all three modes from the 3 microphones are shown in Figures 3.20, 3.21, and 3.22.

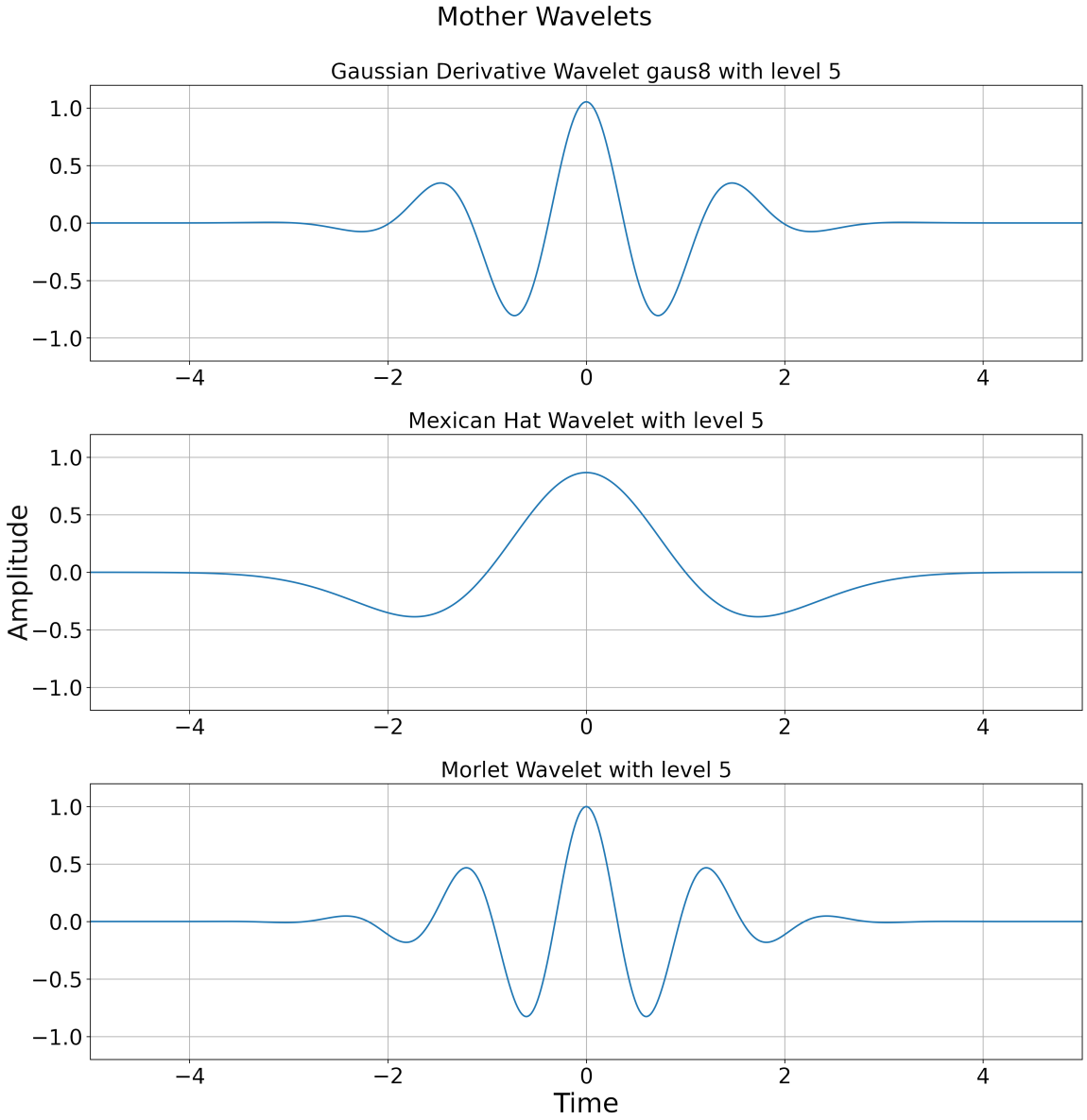

Figure 3.18. Mother wavelets.

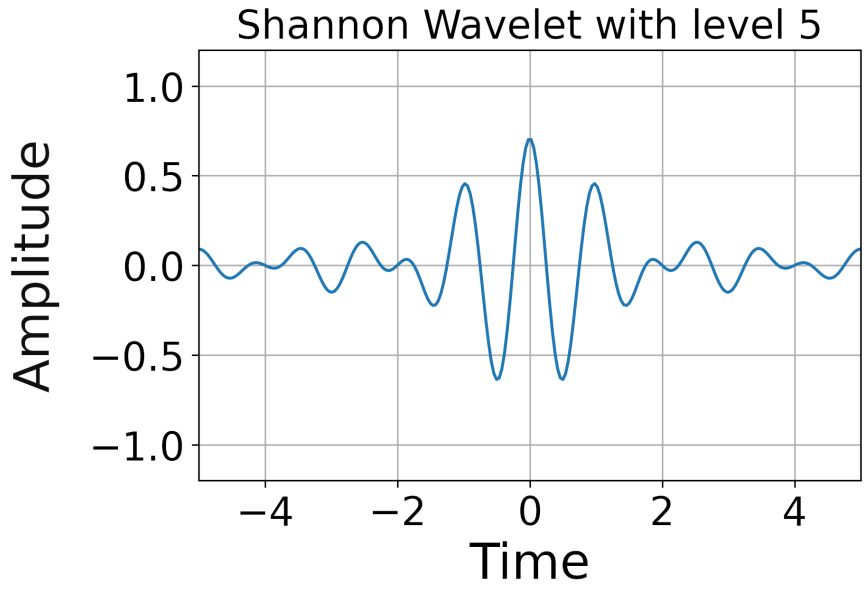

Figure 3.19. Shannon wavelet.

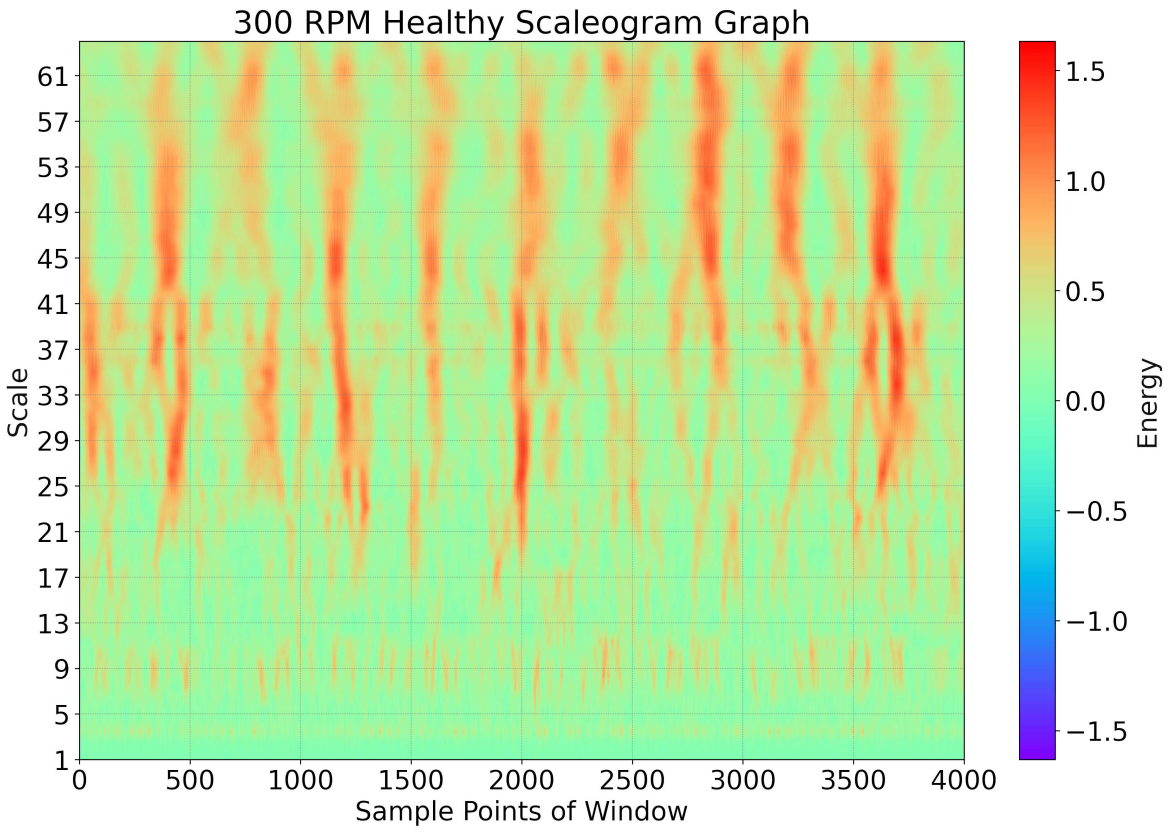

Figure 3.20. 300 rpm healthy scaleogram.

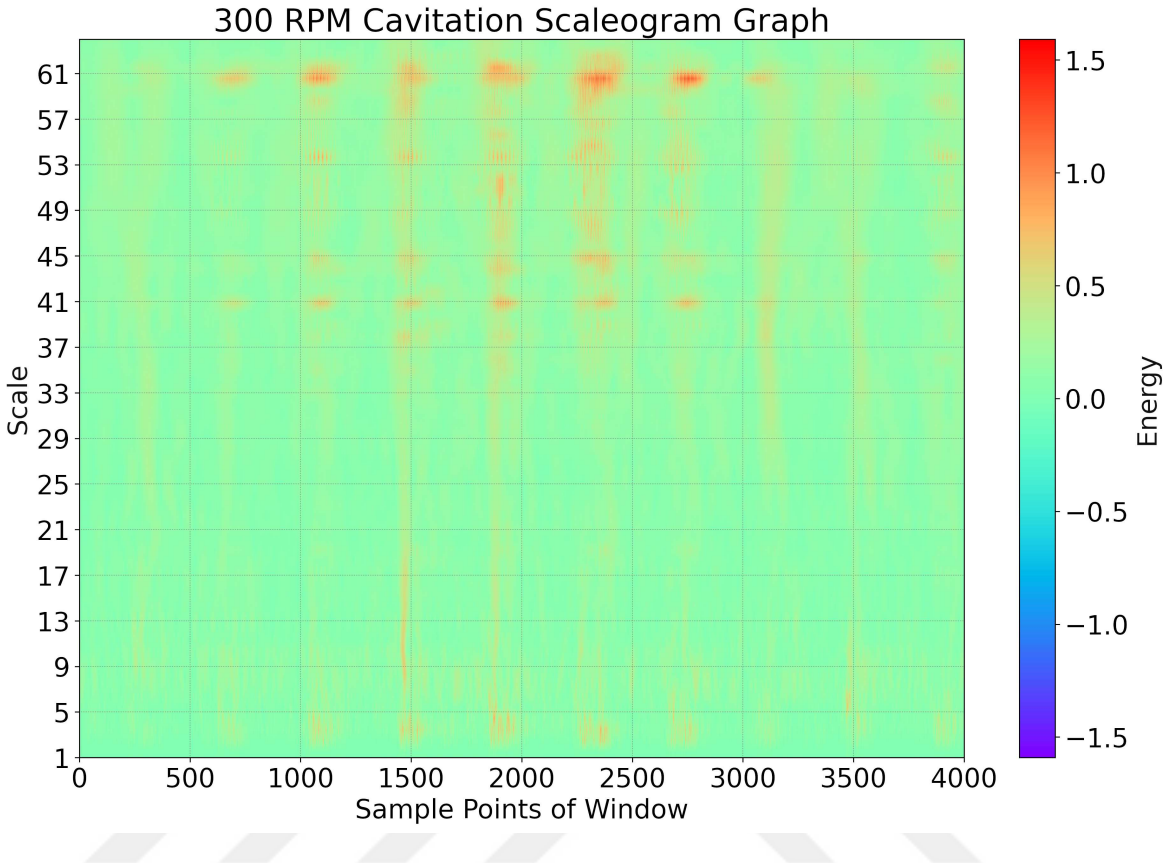

Figure 3.21. 300 rpm cavitation scaleogram.

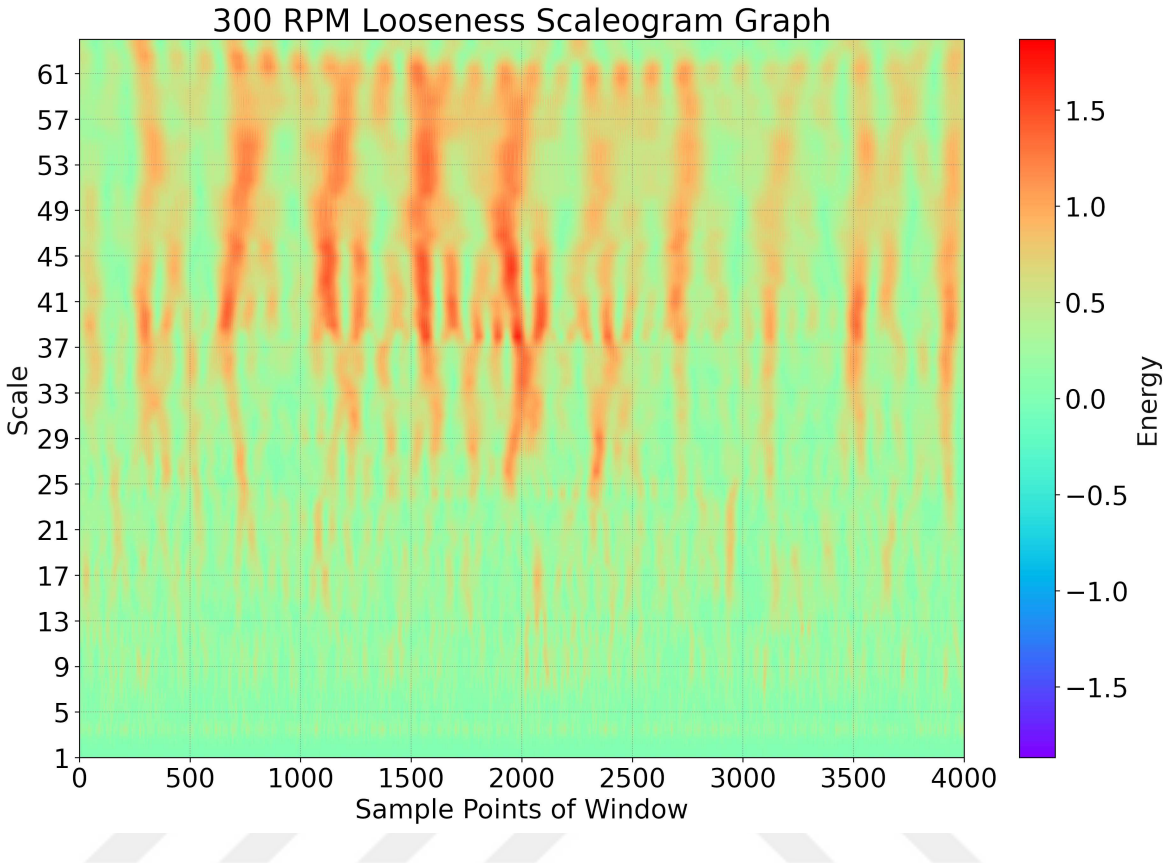

Figure 3.22. 300 rpm looseness scaleogram.

3.2. Data Augmentation

In case of labelled data scarcity or lack of a model's generalization performance, using data augmentation techniques could be an effective solution. The objective of augmentation is to deteriorate the data without changing its semantic meaning. To illustrate, rotating or mirroring an image may increase the learning capacity of image models. Furthermore, adding noise to a sound signal can enhance the performance of speech models. The most popular augmentation techniques for audio models are time stretching, pitch shifting, and background noise addition. Salamon et al. investigated the effects of audio data augmentation techniques over environmental sound classification via CNN models [19]. Their research showed that noise-like sounds such as air conditioner or drilling sounds might be affected negatively by noise addition due to suppression of the original signals. In the light of this, background noise addition is not in this thesis scope owing to the noise-like characteristics of FIP sounds.

Another method, time stretching, is based on slowing down or speeding up a signal. When the signal is slowed down, the amount of sample points in a frame with fixed duration will decrease or vice versa. In order to keep the number of sample points in a frame the same as before, due to the fixed input size of classification model, the slow signal will be cropped while the faster signal will be duplicated. Even if the slow audio looses some information, it keeps serving the generalization solution of the model. In Figure 3.23, 300 rpm, 1 second original and stretched signals, at double speed and half speed, respectively, are shown in the time domain.

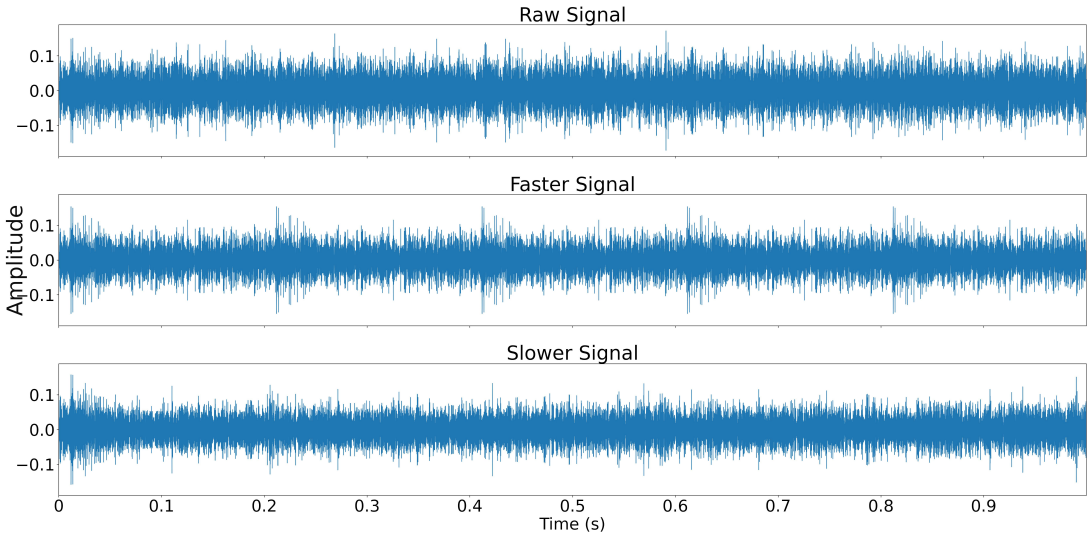

Figure 3.23. Time stretched signals.

Finally, pitch of a sound can be defined as a relative measure in frequency where there are predefined ranges that are used to group musical notes. A pitch consists of notes denoted by alphabetic letters from A to G. One group of notes is called an octave and difference between octaves is called a pitch. To illustrate, while a woman sings or speaks in a higher frequency level, i.e high pitch, men usually have a lower frequency level or low pitch voice. Pitch shifting is conveying a sound signal from one range to another with preserving the internal frequency relation of the signal. To preserve the relation among the frequency content, each octave is divided into bins. Figure 3.24 shows pitch shifting of a 300 rpm pump signal with different bin resolutions 12, 36, and 108, respectively.

During experiments on the effects of data augmentation over the FIP data, pitch shifting outperformed the other techniques. The effects of pitch shifting will be discussed in detail in Section 3.5.

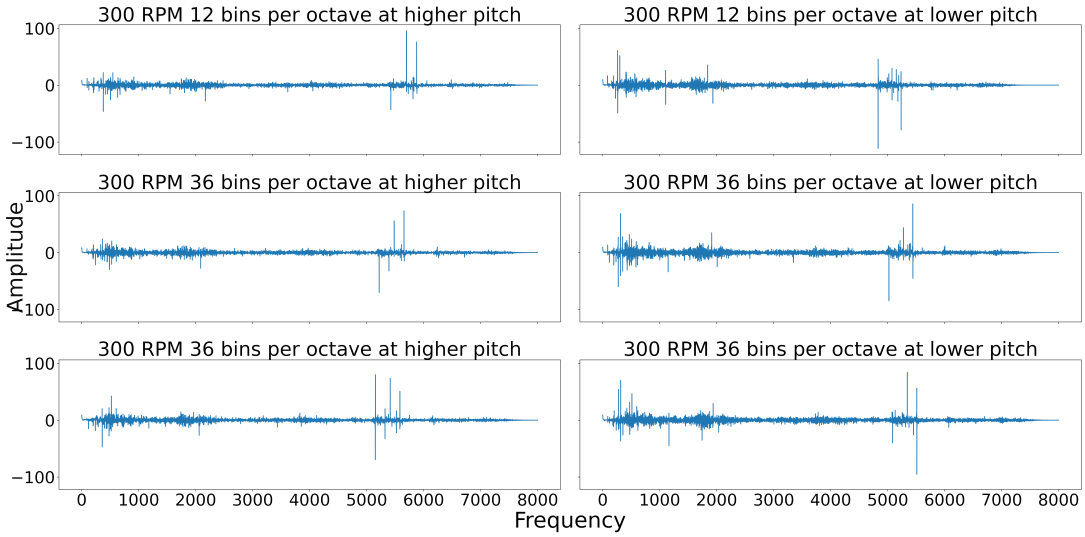

Figure 3.24. Pitch shifted signals.

3.3. Artificial Neural Networks

Before explaining an artificial neural network model, it is important to understand what a neuron is. A neuron in learning is designed by inspiration from the fundamental unit of human brain, also called as a neuron. Receiving, processing, and transmitting information are the main objectives of neurons. The human brain gathers information from the environment and its neurons decide which piece of information is more important to keep for long-term memory. A similar approach is used with the neurons in artificial models [56–59]. A neuron weights its input in order to learn what is important. The sum of the weighted set of information is called as a logit:

$$\sum_{i=0}^{n} W_i x_i \tag{3.20}$$

where W_i are the weights and x_i are the inputs of the i^{th} neuron. A single neuron is depicted in Figure 3.25. Afterwards, the logit is passed through an activation function to gain nonlinearity. Nonlinearity enables the neuron to learn complex mappings between the input and the output target sets. It is also possible to add bias to the logit

in some cases. Finally, the output of a neuron can be expressed as

$$\text{activation function}\left(\sum_{i=0}^{n} W_i x_i + b_i\right) \tag{3.21}$$

where b is the bias of the i^{th} neuron.

3.3.1. Activation Functions

Logit of a neuron gains nonlinearity by passing through a nonlinear function. In the literature, the function is usually referred as activation, squashing, or transfer function. The importance of nonlinearity in learning will be clearly explained in Section 3.3.3. In this section, the most common activation functions are shown with their mathematical backgrounds.

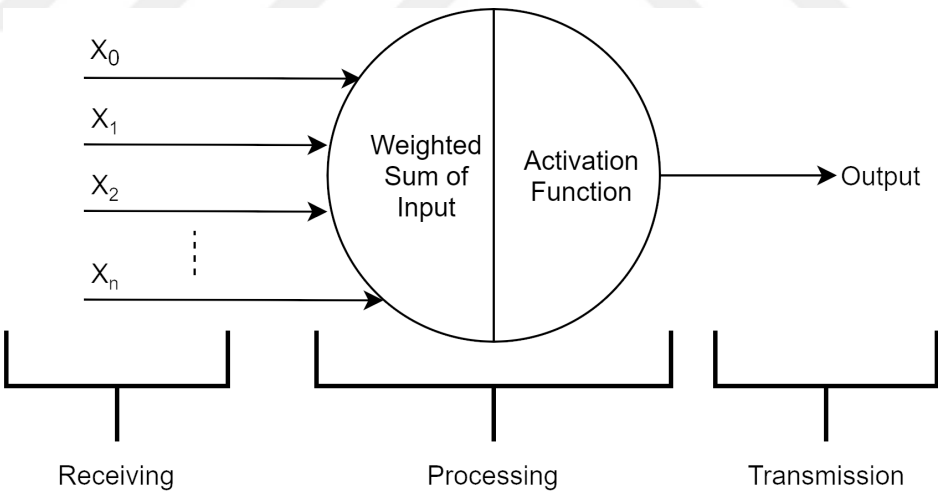

Figure 3.25. A simple neuron in artificial model.

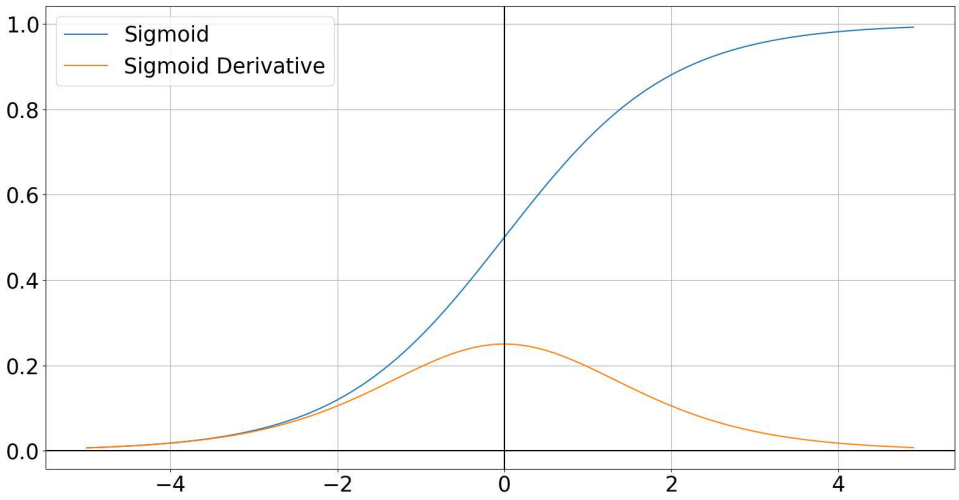

Figure 3.26. Sigmoid activation function and its derivative.

Sigmoid function is the most common activation function among others. It generates values in the range from 0 to 1 for any input. The sigmoid activation function and its derivative are depicted in Figure 3.26. When the input value goes to positive infinity, the output value converges to 1, and as the input goes to negative infinity, the output converges to 0. With sigmoid function, very large and very small input values are not learned due to close to zero derivative values at the extremes. This problem is also known as vanishing gradient. The function is generally used for classification tasks.

Different from the sigmoid function, tanh function, a.k.a. hyperbolic tangent function, produces values in the range from -1 to 1. Tanh function is symmetric around zero and its derivative gets higher than the derivative of the sigmoid function. The function is generally preferable over sigmoid function due to its zero centricity. The tanh activation function and its derivative are depicted in Figure 3.27.

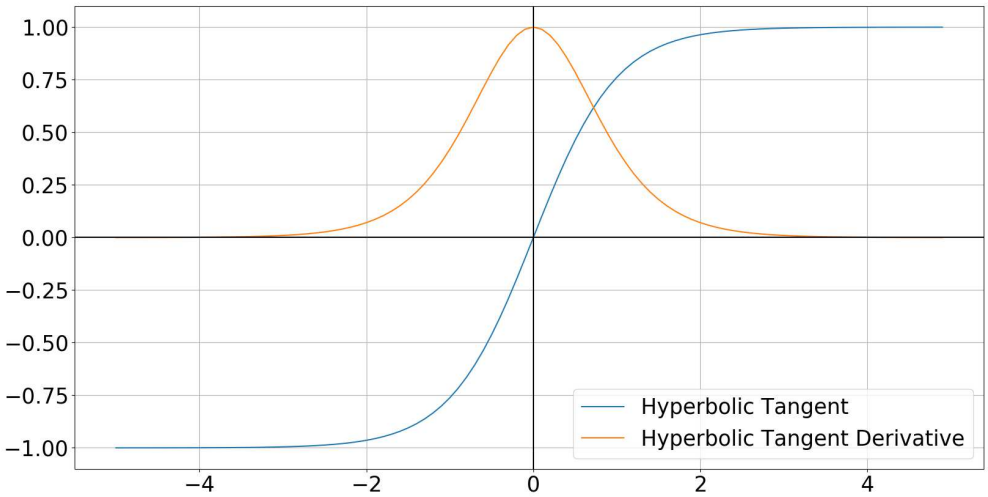

Figure 3.27. Tanh activation function and its derivative.

As a third common option, Rectifier Linear Unit (ReLU) function, which is linear in the positive input range but gives zero output for the negative inputs, is used. In other words, some neurons are not used entirely in the learning due to zero derivative in the negative side of the graph, which is known as the dead neuron problem. It is mostly used in convolutional neural networks with hidden layers. The ReLU activation function and its derivative are depicted in Figure 3.28. In order to solve the dead neuron problem, Leaky ReLU is designed as the improved version of ReLU. Leaky ReLU has a smaller linear transfer function for the negative side of the input range, so its derivative is not zero.

Finally, softmax function, which consists of multiple sigmoid functions, is used as an activation function. Unlike sigmoid, softmax generates a probability for each output class. Thus, it is generally used for multiclass classification tasks. The softmax function is calculated as

$$softmax(x)_i = \frac{exp(x_i)}{\sum_j exp(x_j))} \tag{3.22}$$

where x_i is the i^{th} input in the vector representation and j goes from 1 to the number of classes.

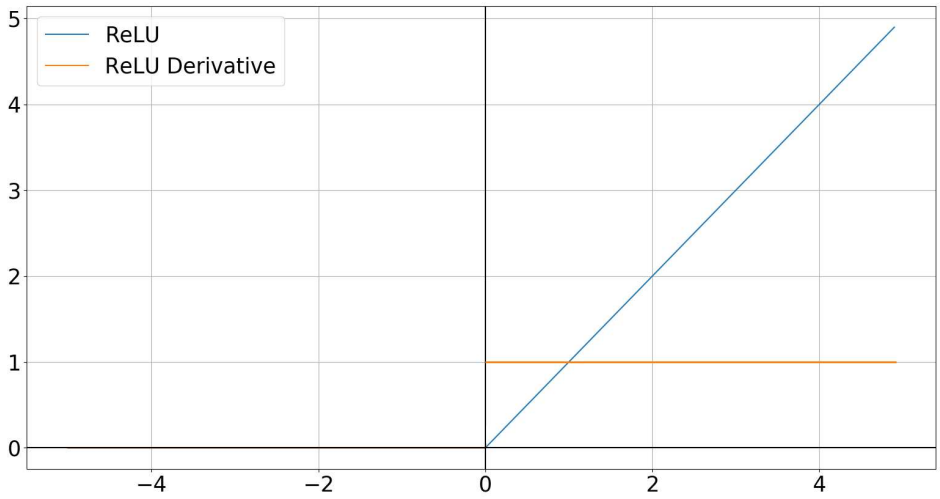

Figure 3.28. ReLU activation function and its derivative.

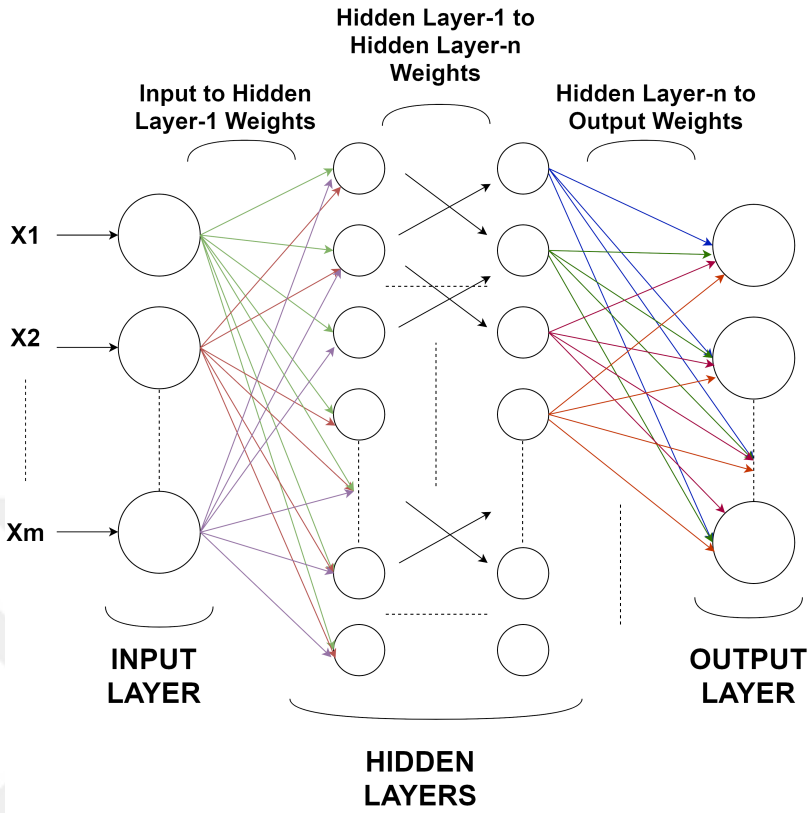

Figure 3.29. Deep feedforward network.

3.3.2. Deep Feedforward Network

Deep Feedforward Network or Fully Connected (FC) Network or Multilayer Perceptron Model (MLP) is one of the fundamental blocks of deep learning models. Deep learning is useful for many applications such as convolutional neural networks used for object recognition, natural language processing, classification problems, and so on. Whereas, a single neuron is not enough to perceive complex problems but the combination of neurons can tackle them. A deep feedforward network, consisting of input, hidden, and output layers including a number of neurons depending on the complexity of the problem, is shown in Figure 3.29.

The input in the vector form is represented by x, which is combined of $[x_1, x_2, x_3....x_m]^T \in R^{1\times m}$. It firstly is multiplied by a matrix consisting of the input

to hidden layer-1 weights with dimension $R^{m \times h^1}$, where m is the number of inputs and h^1 is the number of hidden layer-1 neurons. Then, the bias vector of layer-1 b^1 with dimension R^{1xh^1} is added. The result is passed through the selected activation functions. The same process continues to generate the output in the output layer. To sum up, a deep feedforward network includes many single neurons to solve complex problems such as image classification or Natural Language Processing (NLP). The first hidden layer output is calculated as

$$O^1 = f_{layer_1}\left(\begin{bmatrix} W_{11} & W_{21} & W_{31} & .. & W_{1m} \\ W_{21} & W_{22} & W_{23} & .. & W_{2m} \\ W_{31} & W_{32} & W_{33} & .. & W_{3m} \\ W_{41} & W_{42} & W_{43} & .. & W_{4m} \\ .. & .. & .. & .. & .. \\ W_{h_1 1} & W_{h_1 2} & W_{h_1 3} & .. & W_{h_1 m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ .. \\ x_m \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ .. \\ b_{h_1} \end{bmatrix}\right) \tag{3.23}$$

where f_{layer1} is an activation function. The closed form of the feedforward network is calculated as

$$O^1 = f^1(W^T x + b^1). \tag{3.24}$$

3.3.3. Backpropagation in Neural Networks

Backpropagation enables learning of network parameters such as weights and biases. The model learns via cost or error functions with supervised learning. Supervised learning has an insight into the output; therefore, an error value can be generated by comparing the desired output and the actual output of the model. Taking derivative of the error with respect to the weights indicates the amount of update for the weights. In other words, backpropagation is an optimization algorithm that minimizes the cost function by updating the weights using gradient descent rule. Some of the popular cost functions and optimization algorithms are explained in this section.

3.3.3.1. Error Functions. There are several cost functions used in the literature that can be modified or customized depending on the needs of the model. Thus, the three most popular cost functions are going to be explained in this section. The first well-known cost function is mean squared error (MSE). MSE is calculated as the mean of squared differences between the actual and the predicted outputs. It is generally used for regression problems. One of the fundamental drawbacks of it is the penalization of the weight. If the deviation from the actual output is high, the weight update will be drastic due to squaring. MSE is calculated as

$$MSE = \frac{1}{n}\sum_{t=1}^{n} e_t^2 \tag{3.25}$$

where n is the number of outputs and e_t, the error of the t^{th} output is

$$e_t = y_{actual} - y_{predicted}. \tag{3.26}$$

The second error function that is used in the literature is the mean absolute error (MAE). MAE calculates the average of the sum of absolute differences between the actual and predicted output. Similar to MSE, MAE is also not affected by the sign of the error. However, the high changes in the outputs do not induce huge problems since the equation does not include squaring. The MAE is calculated as

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |e_t|. \tag{3.27}$$

The last error function is the cross-entropy loss or negative log likelihood, which gives the probability of how much the real output is close to the actual output. Similar to the former error functions, the cross-entropy loss should be minimized as the predicted probability diverges from the actual label. It is mostly used for multiclass classification. For this thesis work, it is selected as the loss function in order to investigate the three FIP states with their probabilities. Cross Entropy Loss is calculated

as

$$\text{Cross Entropy Loss} = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{3.28}$$

where M is the number of classes, y is the binary indicator, and p is the predicted probability observation o of class c.

3.3.3.2. Optimization Algorithms. Optimization algorithms are utilized for training machine learning or deep learning models to minimize their error rates. Efficacy of an optimization algorithm is determined by its speed of convergence to a local or global optimum and its generalization capability. Stochastic Gradient Descent (SGD) and Adaptive Moment Estimation (Adam) are two popular optimization algorithms. Both algorithms are used for parameter optimization in models.

SGD algorithm sweeps from output to training set in order to update learnable parameters. SGD update rule is given as

$$w_t = w_{t-1} - \eta \nabla J(w_{t-1}) \tag{3.29}$$

where w is a learnable parameter, η represents learning rate of a model, $\nabla J(w)$ is the gradient of the objective/error function. Besides, whilew_t defines an updated new parameter, w_{t-1} refers the old parameter.

The Adam optimization algorithm is an extension to SGD, which computes individual learning rates for different parameters [60]. While SGD maintains a single learning rate η for all parameter updates, Adam maintains a learning rate for each parameter and updates them individually as the learning process continues. Adam also makes use of the average of mean and variance of the gradients as follows:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \end{aligned} \tag{3.30}$$

where m_t shows the estimated mean of the gradients and v_t represents the uncentered variance of the gradients; g is gradient of the current mini-batch; and β_1 and β_2 are hyper-parameters of the algorithm, which are mostly chosen as default values between 0.9 and 0.999.

Bias-corrected mean and variance estimates are calculated as

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1-\beta_2^t}. \end{aligned} \tag{3.31}$$

Then, they are used to update the parameters, which yields the Adam update rule given by

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \tag{3.32}$$

where w refers to a learnable parameter, η defines the learning rate, and ϵ is a hyper-parameter mostly given as 10^{-8}.

To summarize, optimization algorithms are utilized for finding proper values to update parameters by converging to a local minimum. Although there exists a lot of optimization techniques in the literature, SGD and Adam algorithms are explained in detail since they are employed for learning models in the thesis.

3.4. Theory of Convolutional Neural Networks

Convolutional Neural Networks (CNN) are among the most popular models in deep learning. CNN has been developed by an inspiration from visual cortex of human brain that responses the stimuli only in its receptive fields. It is mostly used for image classification, medical image analysis, natural language processing, object detection and so on. Along with image input, the model can also be used with audio input by

converting sounds to images by spectrogram or cochleogram. To illustrate, the model is used for classifying environmental sounds by converting audio signals to spectrograms which are image representations of the audio signals [19]. It can also be used in a hybrid model with a recurrent neural network in order to classify music genres [61]. Besides, the model can be used for acoustic event recognition by using cochleogram representation of the audio data, which is a finer version of the spectrogram [62]. In the thesis, a CNN model is used for motor failure classification depending on the image representation of audio data.

CNN model is similar to Fully Connected (FC) Networks such that both models consist of learnable weights and biases. On the other hand, CNN model does not need hand-crafted or traditional feature extraction techniques since it can extract intrinsic features of the data without prior knowledge or expertise. Besides, while each neuron connects to the output neuron in the FC network, the CNN model uses filters or kernels which requires less parameters to be learned than a FC network does. As a result of this feature, the model is less prone to overfitting problem compared to FC networks since the complexity of the network is decreased.

A general image based input to a CNN network has four main parameters: width, height, depth, and batch size. While width and height define the image size, depth of an image is the number of color channels in the image. To illustrate, while a greyscale image has a depth of 1, a Red - Green - Blue (RGB) image has a depth of 3. Batch size is the number of image inputs to the CNN network. In order to process the input to generate an output image, there should be filters called kernels with their sizes designed according to different needs for each layer. During filtering operation, there are two parameters to be considered: stride and padding. Stride specifies the amount of shift of the filter in each operation. Padding is an operation to re-shape the input image to increase performance at the image edges. At each step, padding and stride values specify a region and this region of the input image is multiplied with the filter by using matrix dot product. This operation is called as convolution for the CNN models but it differs from its actual meaning since the filter is not flipped [37]. The convolution,

stride and padding operations are depicted in Figure 3.30. There should be a separate filter for each channel; therefore, the depth of the input should be equal to the depth of the filter. The number of outputs can be changed based on the number of filters. For instance, if an RGB image is convolved with five 3D filters, where each dimension corresponds to a color channel, there will be five different outputs with depth of three. In Figure 3.30, 2D convolutional networks are depicted. Depending on the input shape, the output shape will be calculated as

$$in = [N, d_{in}, w_{in}, h_{in}] \tag{3.33}$$

$$f = [d_f, f_w, f_h] \tag{3.34}$$

$$w_{out} = \frac{w_{in} + 2 \times p - f_w}{s} + 1 \tag{3.35}$$

$$h_{out} = \frac{h_{in} + 2 \times p - f_h}{s} + 1 \tag{3.36}$$

$$out = [N, d_f, w_{out}, h_{out}] \tag{3.37}$$

where in refers to input image which is combined of N number of batches, d_{in} depth, w_{in} width, and h_{in} height; f describes the filter or kernel with d_f depth, f_w width, and f_h height; p indicates padding and s is stride; and w_{out} and h_{out} show width and height of the output after the input is passed through the filter.

After the CNN convolution operation, which is linear, the model gains nonlinearity with an activation function. Then, the output can pass through a pooling layer which helps reduce the size of the output and it does not use learnable parameters. The pooling layer can be max pooling or average pooling. In Figure 3.31, max pooling and average pooling operations are depicted.

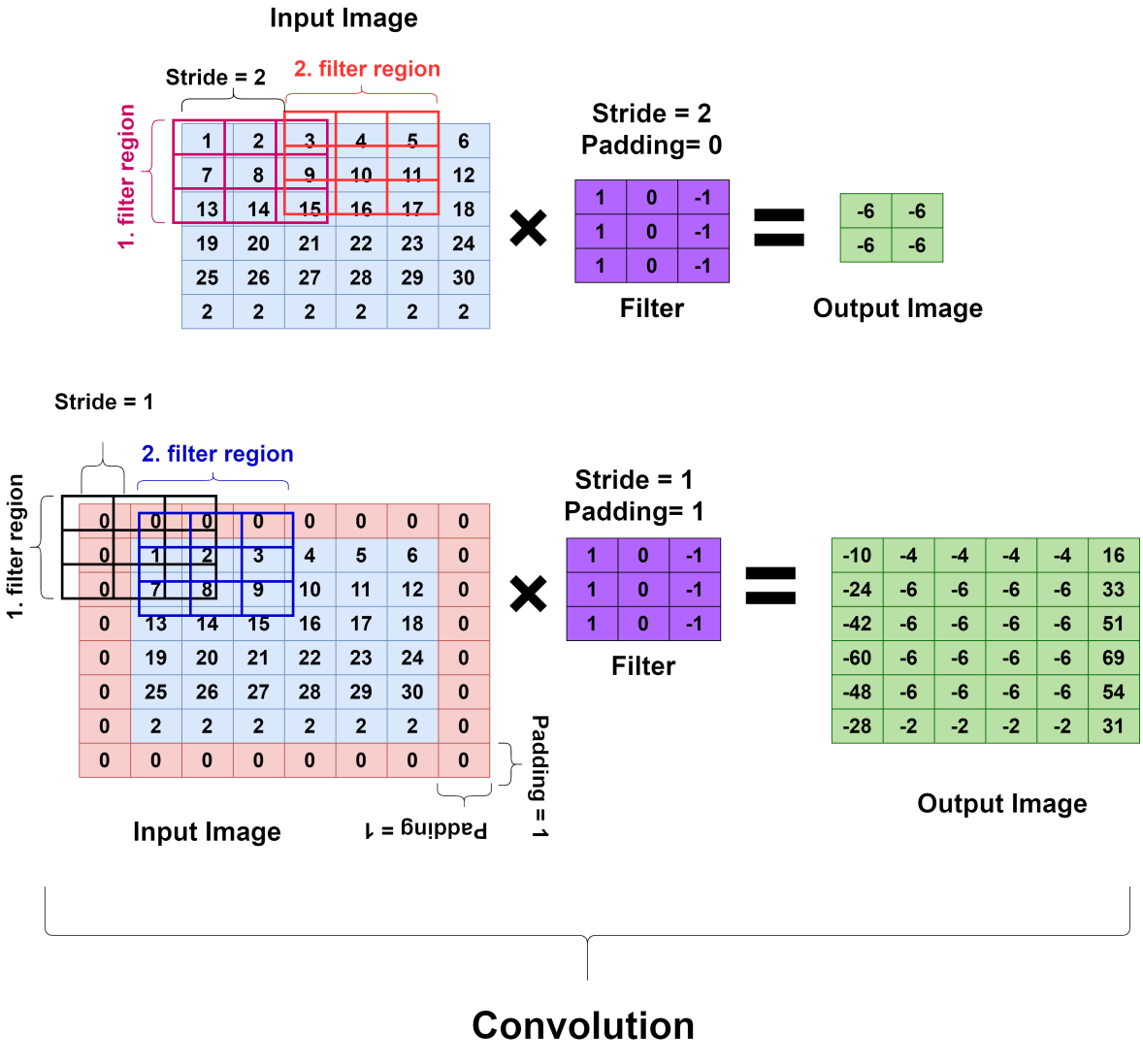

Figure 3.30. CNN Convolution operations.

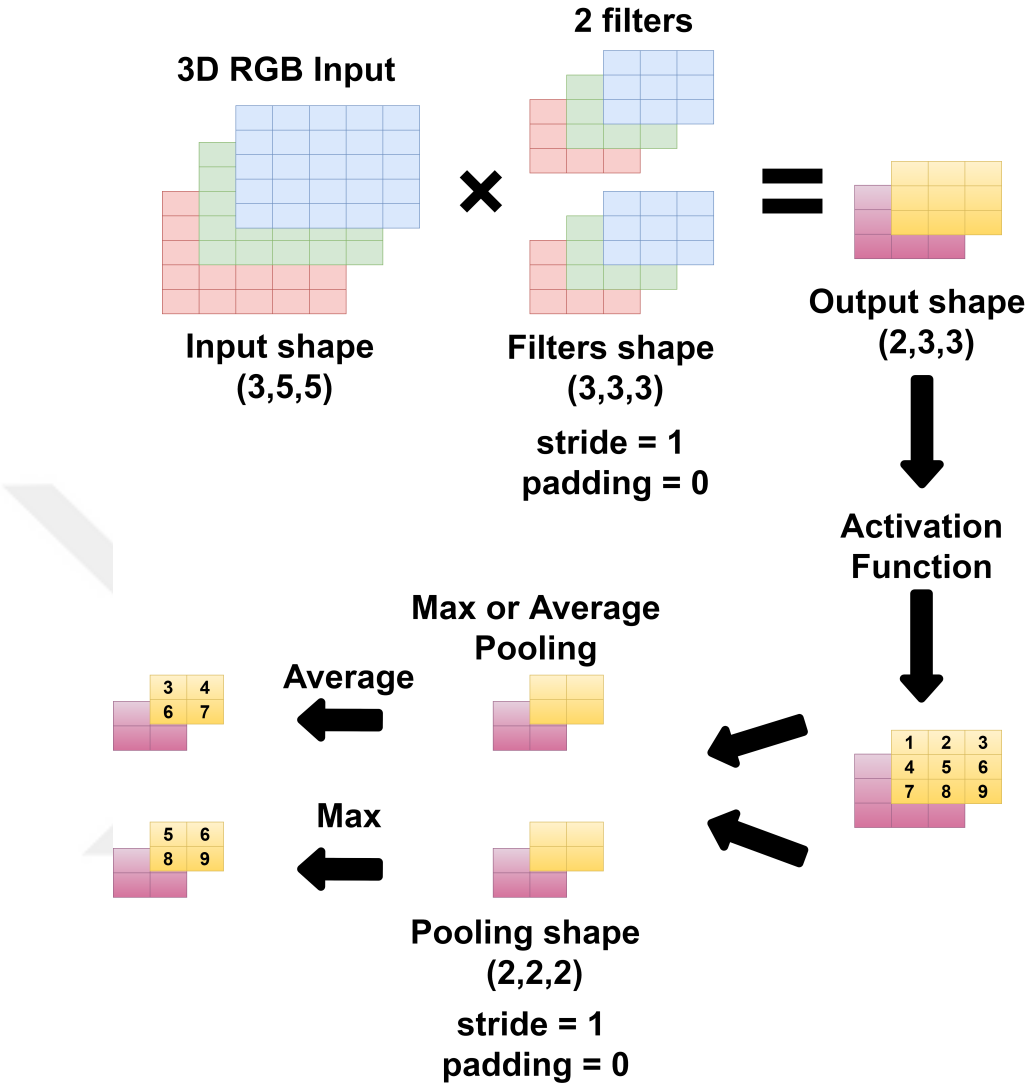

Figure 3.31. 3D Convolution with two filters and pooling operations.

As shown in Figure 3.31, the pooling layer changes the output size based on the output width and height calculations in (3.33) to (3.37). However, it does not affect the dimensions of the output. After the pooling layer, the CNN network can be connected to FC layers to classify the input.

Similar to the FC networks, the model learns weights and biases by using back-propogation. Briefly, the model predicts the outputs and compares the predictions against the actual labels for classification. Afterwards, the learnable parameters can

be updated according to the preferred optimization technique e.g., Stochastic Gradient Descent (SGD) or Adaptive Moment Estimation (Adam).

In the following section, applications of CNN to FIP failure mode classification problem are explained with experimental results.

3.5. Convolutional Neural Network Experiments

A CNN network is used for classification of flexible impeller pumps' failure modes, healthy, cavitation, and looseness. The model is trained by MFCC features, spectrogram, and raw data of the audio obtained from different impellers. A pitch shifting based data augmentation algorithm is also used in order to get more generalized models with higher accuracy. Furthermore, a majority voting scheme with a sequence of 10 predictions is applied to the test results by employing sequence of the predictions instead of a single prediction in order to increase testing accuracy. The trained model results are discussed in Section 3.6. The section also includes the effect of the majority voting sequence length on test accuracy.

As mentioned in Section 3.1, audio signals have been collected with 16 kHz sample rate from three microphones in a linear placement. Each version of data includes 3 minutes recordings of seven different rotational speeds, 250, 300, 400, 500, 600, 700, and 800 rpm. Overall dataset combines 16 different versions, where each version includes three failure modes called healthy, cavitation, and looseness. Each two consecutive dataset versions are recorded by using the same flexible impeller in different times. In order to have more reliable results, the models are tested in an 8-fold fashion such that the versions belonging to an impeller are used as test set and never used during training. For the reproducibility of the results, the versions that are used for training, validation and test purposes are given in Table 3.4 where $\mathcal{V}$ denotes the set that includes all available data versions from 1 to 16.

Table 3.4. Division of available data for different experiment scenarios.

	Training Set	**Validation Set**	**Test Set**
Fold-1	$\mathcal{V}$ \{1,2,3,16}	3,16	1,2
Fold-2	$\mathcal{V}$ \{3,4,5,16}	5,16	3,4
Fold-3	$\mathcal{V}$ \{5,6,7,16}	7,16	5,6
Fold-4	$\mathcal{V}$ \{7,8,9,16}	9,16	7,8
Fold-5	$\mathcal{V}$ \{9,10,11,1}	11,1	9,10
Fold-6	$\mathcal{V}$ \{11,12,13,1}	13,1	11,12
Fold-7	$\mathcal{V}$ \{13,14,15,1}	15,1	13,14
Fold-8	$\mathcal{V}$ \{15,16,14,1}	14,1	15,16

3.5.1. Convolutional Neural Network with MFCC Features

In this section, a model that is only trained by MFCC features discussed in Section 3.1.3 is implemented due to the fact that human hearing is a common method in failure detection and MFCC is based on mimicking human ear. In the model, the raw data sampling window is chosen as 16000 points from 16 kHz audio data which means 1 s windows. The windows are overlapped with 25% overlap ratio and 32 MFCC features are extracted from each window. The features from each microphone are stacked row by row so the input shape is a 3×32 matrix in one channel. The model consists of one CNN block, a max-pooling layer and 2 FC blocks. A dropout layer with a rate of 0.2 is added in order to overcome the overfitting issue. Besides, a small filter size for the model is used since the MFCC features reduce the noise and they do not need additional noise filter with large filter shapes in CNN block. The model is optimized by Adam optimization and early stopping is also utilized in order to prevent the overfitting problem. The MFCC-CNN model is depicted in Figure 3.32. The model is trained by 128 batches of input and the error is measured by cross-entropy loss. Additionally, the same model is also trained and tested with enriched dataset by implementation of pitch shifting, which is an augmentation model for audio signals that is explained in Section 3.2. The pitch shifting factor is selected as -1, -2, 1, and 2. The model outputs

are shown in Table 3.5, where the effect of pitch shifting (PS) is added for comparison.

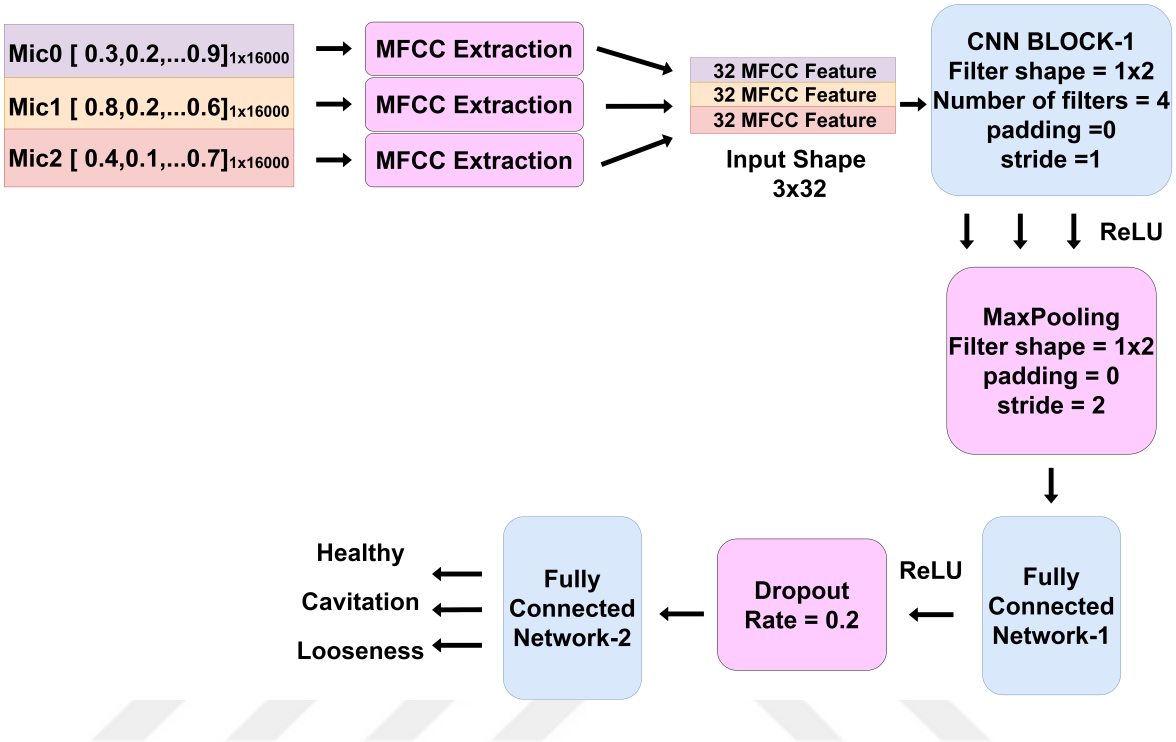

Figure 3.32. MFCC input with CNN model.

Table 3.5. MFCC CNN based fault classifiers.

	Train Accuracy	**Validation Accuracy**	**Test Accuracy**	**Majority Voting Accuracy**
MFCC	99.90 (0.04)	94.52 (2.83)	94.46 (2.42)	95.47 (2.50)
MFCC+PS	99.72 (0.14)	93.89 (4.06)	94.37 (2.70)	94.92 (3.37)

According to Table 3.5, training the CNN model with MFCC features is better than augmented MFCC model (MFCC + PS) and it is also seen that majority voting scheme increases the test accuracy for both cases.

3.5.2. Convolutional Neural Network with Spectrogram Input

In the literature, a significant proportion of the audio signal classification tasks are done by CNN with spectrogram, image representation of audio signals [22]. It shows how the frequency content of a signal changes over time. In order to obtain the spectrogram, the signal is separated into frames and shifted with an overlap. Each frame's Fourier Transform is then displayed with respect to their time intervals. In this thesis, a Mel Spectrogram discussed in Section 3.1.3 is implemented as a feature extraction method for the CNN model.

After experimenting with CNN models, the final version of the model is combined of two convolution layers with a ReLu activation function, a maxpooling layer, and two fully connected layers. The overall model is shown in Figure 3.33.

Each audio recording is broken into 16000 samples frames with 4000 samples shifting. Afterwards, in order to convert the frames to Mel Spectrogram (MS), a window of 3000 samples is used to obtain 28 Mel Coefficients and shifted by 1000 samples. As a result of this, each frame generates a 14×28 matrix in one channel. Images obtained by the three microphones are merged so as to produce 3 channel input data. However,

due to unsatisfactory results with Mel Spectrogram input, the data is enriched by pitch shifting. The pitch shifted data are appended to 20 seconds of raw data for conversion to Mel Spectrogram. The pitch shifting factor is selected as -1, -2, 1, and 2. The model results with and without data augmentation are shown in Table 3.6.

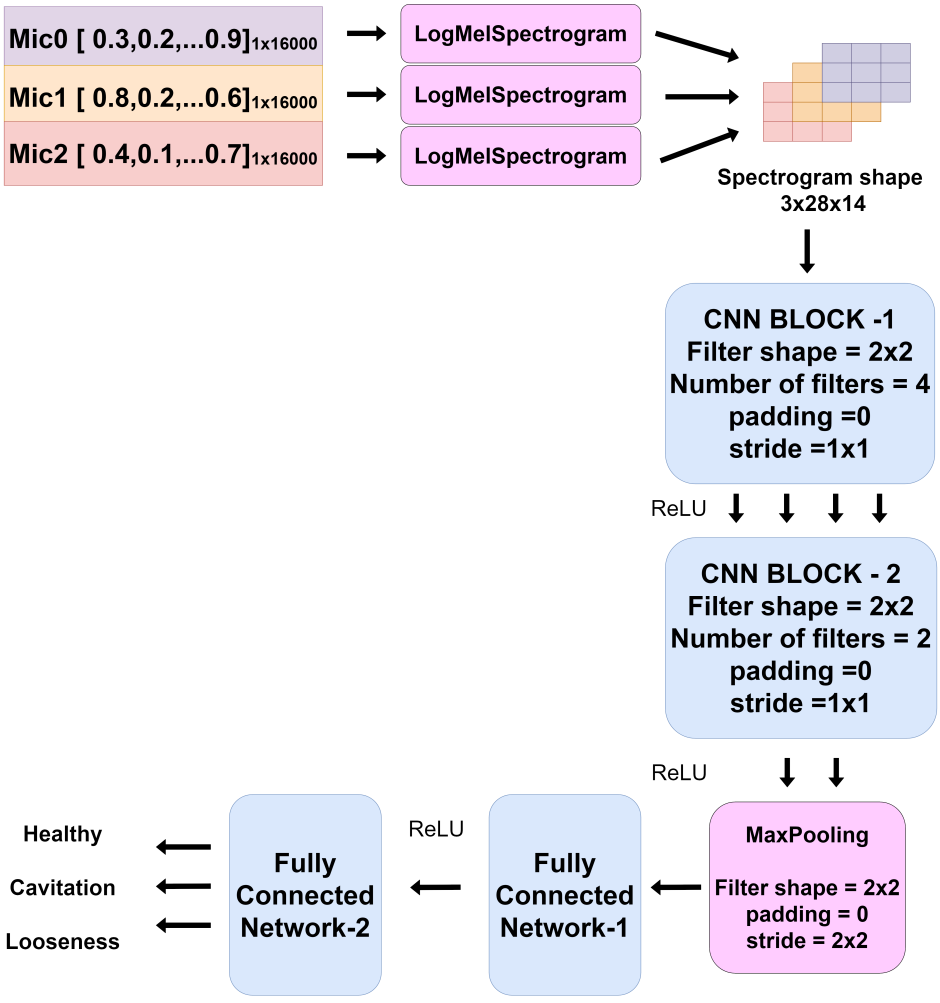

Figure 3.33. Spectrogram input with CNN model.

Table 3.6. CNN-based fault classifiers with spectrogram input.

	Train Accuracy	**Validation Accuracy**	**Test Accuracy**	**Majority Voting Accuracy**
MS	99.89 (0.08)	90.56 (3.08)	91.63 (8.16)	92.21 (10.76)
MS + PS	99.89 (0.06)	89.70 (3.94)	93.25 (6.27)	93.29 (7.26)

To summarize, the CNN model with spectrogram needs more complexity to extract the features of the signal. Besides, the data should also be enriched by a signal augmentation technique in order to overcome the overfitting issue. Moreover, small frame size of the signal affects the learning performance in a negative way.

3.5.3. Convolutional Neural Network with Raw Input

The layout of the raw data CNN model has also been determined after a large number of experiments. The window is selected as 2000 data points from $16kHz$ sampled audio data and it is shifted by 500 points which means 25% overlap of the consecutive windows. The serialized three channel, windowed, time series data are used as input for a one dimensional convolutional network. The model consists of one convolutional layer, a max pooling layer, and two fully connected layers. The convolutional layer and the first FC layer passes through a rectified linear unit (ReLU) to obtain nonlinearity. The overall model is depicted in Figure 3.34. CNN block filter shape is chosen as 1×64 and max pooling layer shape is selected as 1×32 in order to filter noise in raw-time series input. The model is optimized by Adam optimization and early stopping is utilized in order to prevent the overfitting problem. The epoch limit is given as 50 but the model stops learning after the validation loss is increased 5 times. The model is trained by 1024 batches of input without needing any prior knowledge about time-series signal.

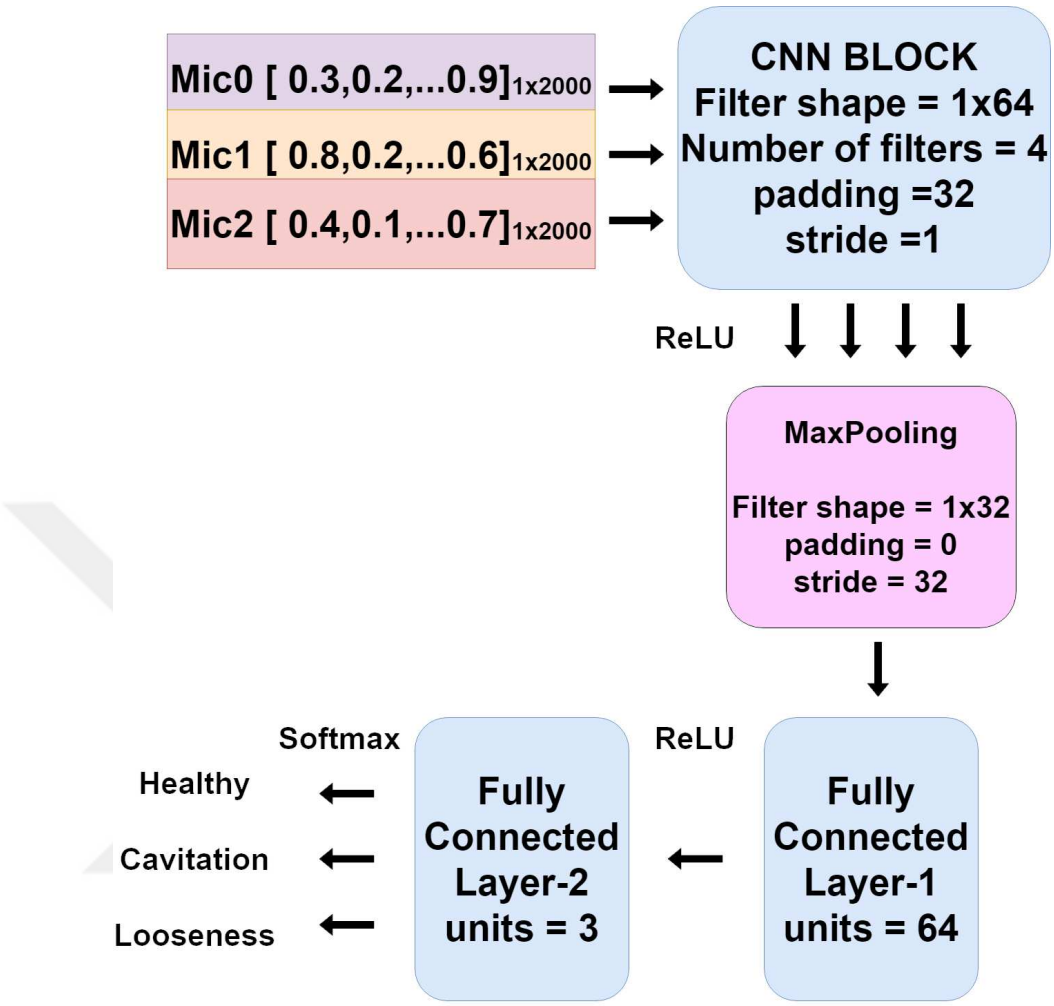

Figure 3.34. Raw audio input with CNN model.

The model's accuracy results for training, validation, test, and majority voting are given in Table 3.7 for raw data (RD) with comparison to the augmented model with pitch shifting (PS). The table shows the mean and standard deviation of 8-fold training results. According to the table, although the raw series training gives 91.864 accuracy in test results, the augmented and majority voting enhancements in the model lead to higher accuracy results. Thus, the dataset should be augmented by pitch shifting to achieve higher accuracy. Besides, the majority voting scheme enhances the accuracy due to accepting large series of predictions instead of individual ones.

Table 3.7. CNN-based fault classifiers with raw audio data input.

	Train Accuracy	**Validation Accuracy**	**Test Accuracy**	**Majority Voting Accuracy**
RD	98.91 (0.28)	91.08 (3.18)	91.86 (3.78)	95.01 (2.64)
RD + PS	98.63 (0.38)	91.81 (3.18)	92.39 (4.57)	95.13 (4.72)

3.6. Concluding Remarks

According to the accuracy results of MFCC-CNN model, Spectrogram-CNN model, and Raw-CNN model shown in Tables 3.5, 3.6, and 3.7, respectively, all models achieve test accuracy over 90% and the majority voting scheme mostly increases the accuracy in 8-fold training. However, standard deviation increases except for the Spectrogram-CNN model. While the raw-CNN model is trained by only 2000 points raw signals, MFCC-CNN and Spectrogram-CNN models need larger number of data points, in this case, 16000. Although the Raw-CNN model has a larger filter in CNN blocks to get rid of noise in the raw audio signal, the others use a small filter sizes since their feature extraction block handles the noise reduction. The Spectrogram-CNN model has a more complex model than the others due to the depth of the spectrogram.

In conclusion, the MFCC-CNN model shows more satisfactory results compared to the Spectrogram-CNN model or Raw-CNN model. The MFCC-CNN model accuracy results are given in Table 3.5 for 8-fold training and the model test accuracy is over 94%. The majority voting scheme slightly increases the accuracy. This result complies with the idea that human expertise by listening the motor sounds can be implemented on deep learning methods. As the MFCC features mimic human hearing, they outperformed the alternative methods considered in audio based classification.

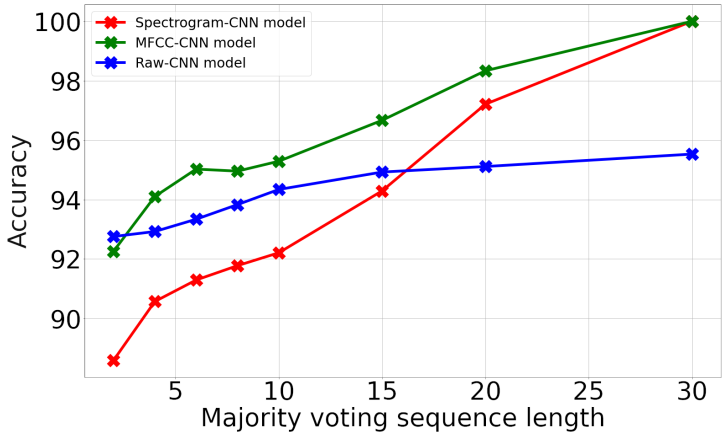

Figure 3.35. Majority voting accuracy with changing sequence length.

In order to increase the test accuracy, a majority voting scheme is applied on the predictions where a set of 10 predictions are converted into the mode of the set. As the number of voters increases, the test accuracy increases as well. However, the increase in the voters requires more and more predictions with less and less increase in the accuracy. Therefore, using 10 voters is selected as the optimal stopping point. Figure 3.35 shows the relation between the test accuracy and the number of voters on the same set of predictions.

Finally, when the models are tested with old data from the same setup, which are collected one month earlier than the dataset used in the thesis, it is observed that the test accuracy drops to orders of 85%. This observation indicates that in order to implement the models, data from all conditions must be collected on the field in a close time window of the prediction. However, collecting data from failure conditions on an operational machine environment requires artificial malfunction states. This scenario reduces the quality of the failure mode data as well as contradicts the initial goal of predictive maintenance, which is increasing the uptime of the machinery. Therefore, a method that does not require failure mode data is of crucial importance to the implementation of the models on real life applications. In Chapter 4, this problem is addressed by an anomaly detection algorithm.

4. ANOMALY DETECTION WITH UNSUPERVISED LEARNING

Supervised learning requires labels of data so as to extract meanings from features but acquisition of data from unhealthy conditions is troublesome since they may need or cause a break in the system; that is, the collection of labelled data might be highly costly. Therefore, a training method that only uses data acquired from the healthy condition is required to overcome the difficulties with collecting failure states' data. Unsupervised learning is a proper solution to handle this problem because it depends on finding patterns from unlabelled data such that it seeks the commonalities inside the data. Unsupervised learning can be employed for clustering, representation learning, and density estimation. In order to detect anomalies in the audio recordings of the FIP, Gaussian Mixture Model (GMM), which is one of clustering models in ML is adopted. Theoretical backgrounds of unsupervised models K-means Clustering and Gaussian Mixture Model are given in Sections 4.1 and 4.2, respectively. In this chapter, Gaussian Mixture model is used for anomaly detection of FIPs and is trained by two different features separately. Anomaly detection results are given under Section 4.3. At the end of the chapter, results are discussed with concluding remarks.

4.1. K-Means Clustering

K-means is a type of clustering algorithm, which assigns data points to two or more clusters in an unsupervised manner. The algorithm of K-means clustering can be described as follows:

Suppose we have data x^i, which consists of a set of samples $x^i = x_1, x_2, \ldots, x_m$. Then,

- Choose the number of clusters to obtain, then initialize cluster centroids randomly. μ_i consists of k cluster centroids, $\mu_i = \{\mu_1, \mu_2, \ldots, \mu_k\}$.

After the initialization step, K-means algorithm includes 2 iterative steps:

- Compute the Euclidean distance between cluster centroids and data points. Afterwards, assign each data point to the closest cluster:

$$c^i = \underset{j}{\operatorname{argmin}} \|x^i - \mu_i\|^2 \tag{4.1}$$

  where j = 1, 2, . . . , k.

- Recalculate cluster centroids with their assigned data points:

$$\mu_{ij} = \frac{\sum_{i=1}^{m} I\{c^i = j\} x^i}{\sum_{i=1}^{m} I\{c^i = j\}} \tag{4.2}$$

  where $I\{c^i = j\}$ is the indicator function.

The iterative steps, (4.1) and (4.2), are repeated until clustering centroids stop moving. K-means is not a probabilistic model so it uses hard assignments which means each data point belongs a cluster with 100% probability.

4.2. Gasussian Mixture Model

Gaussian Mixture Model (GMM) assumes that the data consists of K Gaussian distributions in order to separate N data points in R^D into K clusters. While in the univariate case, each Gaussian has a mean and variance, in the multivariate case, each Gaussian has a mean and covariance. Gaussian distribution can be calculated for the multivariate case as follows:

$$\mathcal{N}(\vec{x} \mid \vec{\mu}, \Sigma) = \frac{1}{(2\pi)^{D/2} \mid \Sigma \mid^{1/2}} exp\{-\frac{1}{2}(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})\} \tag{4.3}$$

where x is a point in D-dimensional space from the dataset $x^i = x_1, x_2, \ldots, x_N$ so $\vec{x} \in R^D$; Σ represents covariance matrix of size $D \times D$; and $\vec{\mu}$ is the mean of features in N data points so $\vec{\mu}$ is a $D \times 1$ vector.

In order to initialize K Gaussian distributions, mean, covariance, and weights of each cluster are chosen randomly. The weights (W_c) determine the amplitudes of the Gaussian distributions. Probability distribution of the data can be calculated as

$$p(x) = \sum_{c=1}^{K} W_c \mathcal{N}(x; \mu_c, \Sigma_c) \tag{4.4}$$

where W_c represents the probability of how likely a data point belongs to a cluster c, which can be expressed as

$$p(z = c) = W_c \tag{4.5}$$

where z is a latent (hidden/unobserved) variable. Then the probability of a data point in a given cluster c is given as

$$p(x \mid z = c) = \mathcal{N}(x; \mu_c, \Sigma_c). \tag{4.6}$$

where W_c is restricted such that $\sum_{c=1}^{K} W_c = 1$ so that the total probability of all data points belonging to any of the distributions adds up to 1.

When we use maximum likelihood estimation for μ and Σ parameters in multivariate Gaussian model, we can obtain

$$\frac{\partial log\mathcal{N}(\vec{x}; \vec{\mu}, \Sigma)}{\partial \vec{\mu}} = 0, \quad \vec{\mu} = \frac{1}{N} \sum_{i=1}^{N} x^i \tag{4.7}$$

$$\frac{\partial log\mathcal{N}(\vec{x}; \vec{\mu}, \Sigma)}{\partial \Sigma} = 0, \quad \Sigma = \frac{1}{N} \sum_{i=1}^{N} (x^i - \vec{\mu})^T (x^i - \vec{\mu}). \tag{4.8}$$

After initializing μ, Σ, and W_c of clusters, expectation maximization (EM) algorithm is used for optimization of the parameters. EM algorithm consists of two steps known as expectation step and maximization step, respectively. In the expecta-

tion step, the probability of each data belonging to a given cluster c is calculated and normalized with all probabilities

$$r_{ic} = \frac{W_c \mathcal{N}(x^i; \mu_c, \Sigma_c)}{\sum_{c\prime}^{K} W_{c\prime} \mathcal{N}(x^i; \mu_{c\prime}, \Sigma_{c\prime})} \tag{4.9}$$

where $i = 1, 2, \ldots, N$, $c = 1, 2, \ldots, K$; r_{ic} is a normalized version of estimated weighted probabilities of data x^i in the cluster c; and $c\prime$ defines all clusters.

After calculating all expectations for each data point, expectation matrix can be derived as

$$r_{ic} = \begin{bmatrix} r_{11} & r_{12} & \ldots & r_{1K} \\ r_{21} & r_{22} & \ldots & r_{2K} \\ \vdots & & \ddots & \\ r_{N1} & r_{N2} & \ldots & r_{NK} \end{bmatrix}.$$

In the next step, Maximization uses assigned probability of r_{ic} in order to update parameters

$$m_c = \sum_{i=1}^{N} r_{ic} \tag{4.10}$$

$$W_c = \frac{m_c}{N} \tag{4.11}$$

$$\mu_c = \frac{1}{m_c} \sum_{i=1}^{N} r_{ic} x^i \tag{4.12}$$

$$\Sigma_c = \frac{1}{m_c} \sum_{i=1}^{N} r_{ic} (x^i - \mu_c)^T (x^i - \mu_c) \tag{4.13}$$

where m_c is the sum of all estimated weighted probabilities for all data in cluster c; and m_c is normalized by the size of dataset N. Updated mean (4.12) and covariance (4.13) are calculated in terms of m_c. Note that Σ_c uses the updated version of μ_c. EM steps are iterated until convergence occurs. In order to measure the convergence of

GMM model, log-likelihood of all data with new parameters are calculated as

$$\log P(\vec{X}) = \sum_{i=1}^{N} \log \left[\sum_{c=1}^{K} W_c \mathcal{N}(x; \mu_c, \Sigma_c) \right] \tag{4.14}$$

where $\vec{X}$ represents all data with D-dimension, $\vec{X} \in R^{N \times D}$. If there is not an increase in the log-likelihood by a predefined threshold value, the EM algorithm is stopped and K clusters will be available with boundaries for data.

While K-means clustering draws circular boundaries for data due to Euclidean distance measurement, GMM can draw elliptical boundaries thanks to the covariance matrix. However, since GMM uses the EM model, it is more computationally consuming than K-means. Besides, GMM converges to one of the local maxima points more quickly. In order to deal with this, Gaussian mixtures can be used with K-means so as to update cluster centroids.

In the thesis, GMM is used for detecting anomalies in the audio data of FIP. It is enhanced with K-means algorithm to update cluster centroids. In order to train the GMM, only healthy impeller data from a pump rotating with all selected rotational speed values are utilized. Half of the healthy data are employed as test data and the data from the unhealthy conditions, cavitation and looseness, are only used for test purposes. After training the GMM, the model learns boundaries for healthy data points. Data points for testing are considered as anomalies if they are out of these boundaries. Furthermore, a threshold value can also be added to the boundaries so as to improve results.

4.3. Gaussian Mixture Model Experiments

A suitable feature extraction technique is a very crucial step for training the GMM as it is for the majority of clustering models. Obtaining features with high representation of the data can provide favourable cluster boundaries to separate anomalies. For this purpose, MFCC and power spectrum density features are utilized for training. In

the following sections, GMM training and its results are shown for both features.

4.3.1. GMM result with MFCC features

The first step of machine learning is generally based on feature extraction. Since an anomaly can be detected by a maintenance expert by hearing, MFCC features are used for GMM training due to their capability of mimicking human hearing features. 32 MFCC features have been extracted from each averaged 5 seconds long audio samples from 3 microphones. Since the audio data are recorded with the sampling rate of 16 kHz, the corresponding length of each sample window is taken as 80000 samples and each window has 50% overlap. Then, the length of each segment for MFCC extraction is taken as 16000 samples (1 second) where two consecutive segments have 50% overlap. For the training, MFCC values are extracted from half of 42 minutes healthy mode dataset. The other half and 84 minutes unhealthy mode datasets are employed in the test phase.

The GMM has been initialized with K-Means algorithm and trained with 8 different impellers separately with different number of clusters, K. Calculating the mean of obtained accuracy values for selected cluster counts from 8 impellers yields the average accuracy depicted in the vertical axis of Figure 4.1. As it can be seen in this figure, mixture of three Gaussians can achieve over 95% accuracy for all conditions. When the number of Gaussians in the mixture is higher than 3, the healthy data accuracy starts to decrease. To sum up, the GMM with three Gaussians can draw representative boundaries for a healthy dataset and it can also detect the anomalies with an accuracy of over 98% and the accuracy values of healthy, cavitation and looseness modes are measured as 99%, 96%, and 99%, respectively.

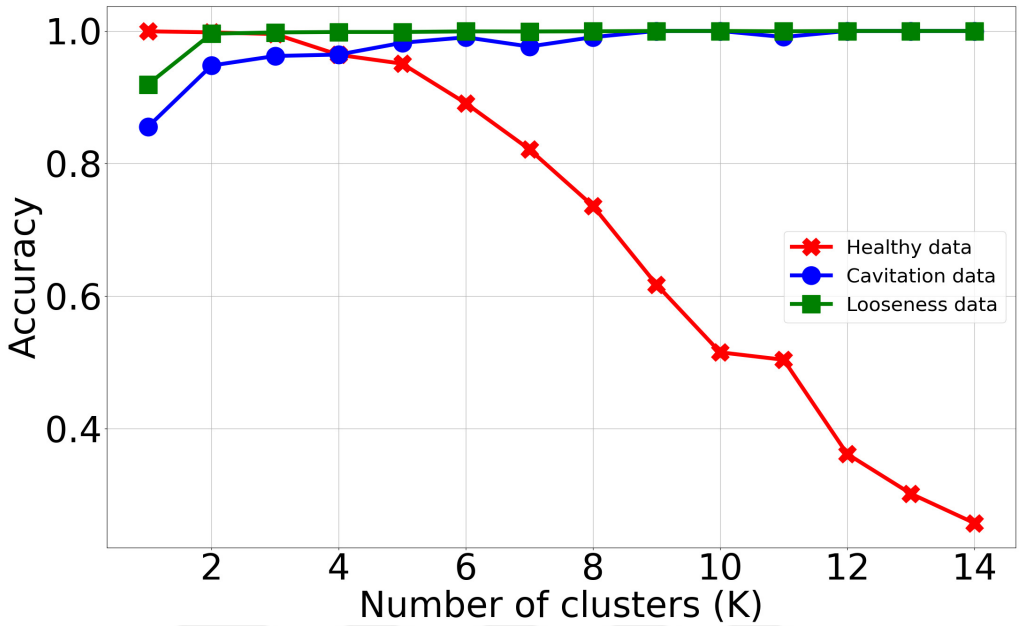

Figure 4.1. Results of MFCC+GMM approach.

4.3.2. GMM result with Welch estimated PSD features

Power Spectral Density (PSD) is another feature extraction method described in detail in Section 3.1.2 for audio signals. A signal can be decomposed into sine and/or cosine signals of different frequencies by using Fourier Analysis. Power spectral density, or power spectrum, gives the power distribution of a signal over frequency intervals. Mathematically, PSD is equal to the square of frequency components of a signal and integration over power spectrum results in the variance of a signal. Welch method is an estimation method for finding the PSD, which is based on splitting a given audio signal into windows with an overlap in time-domain and then computing the average of periodograms from each segment. Thus, the variance of the periodogram is proportional to the square of the values of the spectrum itself. If a large window size is chosen, averaged power spectrum will be noisy, that is, the variance will be high. On the other hand, even if the variance of a signal is low with small window size, the

resolution of the signal will be poor. Therefore, selecting a proper window and overlap size is substantial to extract meaningful information from a signal. In addition, using Welch method makes features robust against noise and non-stationary signals.

In the experiments, 5 seconds long audio samples are used. Afterwards, each sample is divided into 1024 sample long segments with 50% overlap. With these parameters, PSD estimate of any given sequence is calculated for 512 equally sized frequency bins from 0 Hz to 8 kHz. In order to reduce the number of features and make the detection system more robust against frequency shifts, 16 bins of power densities are averaged to obtain a single value. This process eventually leads to 32 power density values to be used as features of a single audio sample.

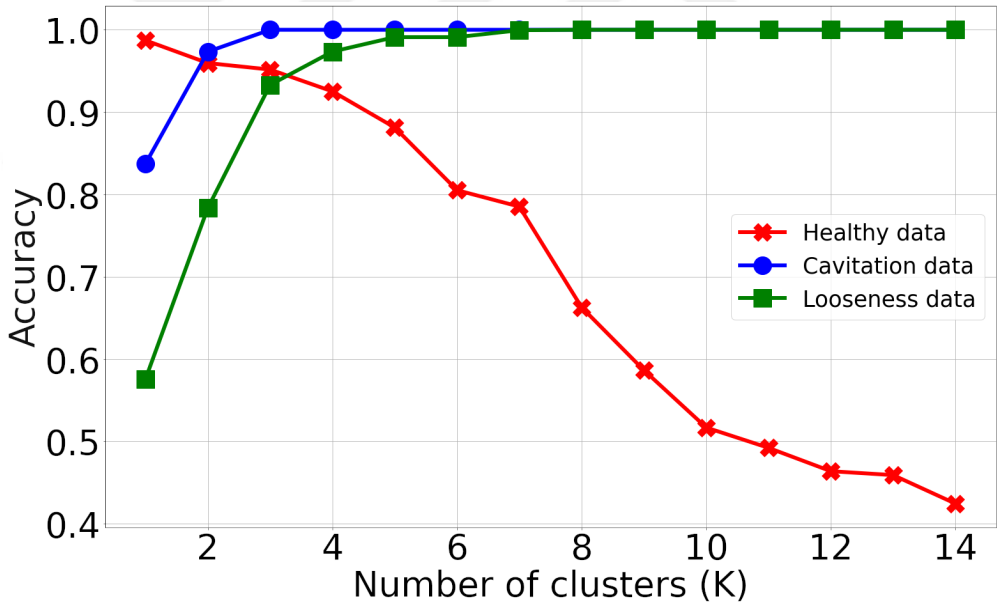

Figure 4.2. Results of PSD+GMM approach.

GMMs that are initialized with K-means algorithm are trained with only half of the healthy sets for each impeller as in MFCC-GMM training. The model is trained with different estimations for number of Gaussians in the mixture so as to fit the healthy dataset. The model accuracy is measured with the existing labels in the dataset. The change in testing accuracy with respect to number of clusters (K) is shown in Figure 4.2. The accuracy corresponding to K values is an average of all test accuracies for eight different impeller data. According to Figure 4.2, K = 3 can be chosen as the optimal point. When the number of clusters is chosen as K = 3, the accuracy values of healthy, cavitation and looseness modes are measured as 95%, 100%, and 93%, respectively.

4.4. Concluding Remarks

GMM depends on updating the initial randomly assigned parameters, which are mean, covariance, and weights of a number of Gaussians, with respect to the Expectation and Maximization algorithms until the parameters converge to their local minima points. In this chapter, two anomaly detection schemes are proposed based on MFCC features and estimated power spectrum density. The first GMM model is trained by 32 MFCC features while the other is trained by 32 estimated PSD values for each 5 seconds healthy FIP audio recordings. The maximum number of components in both GMMs are chosen as starting from K = 1, which corresponds to a single multivariate Gaussian model, to K = 14. After training both models with healthy data, the models are tested with the unhealthy dataset. Since the labels of unhealthy dataset exist, the accuracy of each unhealthy conditions looseness and cavitation could be calculated. The two GMMs give the best accuracy at K = 3. Intuitively, this may be considered as each cluster corresponds to a failure mode. However, the clustering process does not include unhealthy data; therefore, each of the three clusters represents data points from the healthy mode, and the unhealthy data points are expected to deviate from these clusters.

As a result of the experiments, the test accuracy with MFCC values is higher than that of estimated PSD values. Thus, MFCC features can represent distinctive

features of the FIP audio signals since it uses Mel filters, which can copy the human hearing specifics. The theory that a maintenance expert can detect anomalies by only listening to an audio of a FIP is supported by GMM with MFCC features.

5. CONCLUSION

With the emergence of Industry 4.0 in Germany, the concepts of efficiency and sustainability of manufacturing processes by means of intelligent controlling has gained attention from both the industry and the researchers. One way to contribute to the concept is to develop maintenance approaches. In the past, maintenance mainly relied on two strategies known as breakdown and planned maintenance. While breakdown maintenance depends on changing part of a process after damage occurs, planned maintenance is based on controlling a process in time intervals. However, both methods have drawbacks. While the former can cause catastrophic failures, the latter simply reduces throughput due to unnecessary maintenance losses. Nowadays, thanks to Industry 4.0 and smart technologies, predictive maintenance is being utilized to control processes more efficiently. Predictive maintenance depends on controlling a process when a warning is raised by condition monitoring systems. The objective of the thesis has been to provide predictive maintenance solutions for pumps, specifically FIPs, since their presence has been limited in the literature.

An experimental setup was designed after the examination of a real textile manufacturing process. Detailed information about the experimental setup can be found in Chapter 2. Briefly, the emulation setup consists of a small tank, drive trains, vanes, an inverter, an electrical motor, and a FIP. Data were acquired via an acoustic sensor with a decision based on the fact that a maintenance expert could detect a failure only by listening the pump sounds. Based on the ability of human hearing, acoustic data were collected from the FIP via 3 microphones. Acquired data were analyzed in terms of time, frequency, and time-frequency domains in order to extract valuable features. The detailed analysis results are given in Section 3.1. Supervised and unsupervised models were trained by extracted features and the models provided satisfactory solutions for different approaches in predictive maintenance.

Supervised learning is based on the learning relations between input and given outputs by means of neurons. The basics of neural networks are introduced in Section 3.3. In the thesis, a convolutional neural network was used for the classification of the FIP conditions called healthy, cavitation, and looseness. The CNN model was trained by MFCC features, which mimic human hearing, spectrograms, which are the visual representation of audio signals, and raw inputs, separately. Constructed models are given under Section 3.5. Based on the experimental results, the CNN model with MFCC features gave the highest accuracy results without pitch shifting which is an augmentation technique in audio signals. Although raw signals gave good accuracy results with inclusion of pitch shifting and majority voting scheme, its standard deviation was worse than MFCC without pitch shifting and with majority voting scheme. Thus, MFCC features can be considered to keep distinctive features that the CNN model can classify. The idea that an expert's ability to detect a failure only by listening can be modelled with machine learning methods was supported by the proposed model and experimental results.

Even though the supervised model CNN generates good accuracy results with 3 different inputs, the models need data from unhealthy conditions to classify them. However, the acquisition of unhealthy state data needs a broken system so a model only requires data from healthy condition is necessary to obtain a more practical application for the industry. Thus, unsupervised learning methods called K-means and Gaussian Mixture models are studied in Chapter 4. GMM with K-means Clustering enhancement is chosen as the anomaly detection model and it was trained only by healthy data so as to detect unhealthy conditions when a signal distinguish from the healthy data structure. The experimental results of GMM with MFCC and PSD features are given in Section 4.3. Based on the experiments, GMM with MFCC inputs gave the best test accuracy results.

Table 5.1. Comparison of accuracy results of the models.

Models	**Accuracy (%)**
Anomaly detection with MFCC	98
Anomaly detection with PSD	96
CNN with Majority MFCC	95.47
CNN with Majority RD + PS	95.13
CNN with Majority RD	95.01
CNN with Majority MFCC + PS	94.92
CNN with Majority MS + PS	93.29
CNN with MS + PS	93.25

Depending on supervised and unsupervised learning experiments with audio signals, both models gave satisfactory results with MFCC values, thus the ability of hearing can be used for predictive maintenance of FIPs with high accuracy. The overall accuracy results are given in Table 5.1. According to the table, anomaly detection with the GMM model gave the best accuracy results with MFCC features from the acoustic data. Although supervised learning can output specific failure state to ease diagnostics and maintenance, data acquisition for classification models is costly. On the other hand, anomaly detection with only healthy data can give a warning without specifying an error condition with a higher accuracy. Both models show that audio signals have great deal of information about the FIP states and MFCC features is the most successful method to represent this information among all methods examined in this thesis.

This thesis can be basis of further research on rotational machine failure detection by means of audio signals and MFCC features. This can be expanded to a wide range of applications from centrifugal pump error detection to wind turbine health monitoring. Thanks to MFCC features, having an efficient feature extraction method with great representation of the audio data, future works on audio based classification problems can focus on developing better models with higher accuracy and more diverse error

cases for the supervised cases. Moreover, the data used in this thesis can be used for other FIP setups in different environments by means of transfer learning methods. Therefore, one can obtain cavitation and looseness data for their system by only using their healthy FIP recordings. Finally, the results from the predictive models can be used as data source of maintenance scheduling studies based on fuzzy logic approaches. These studies may also lead to efficient maintenance planning as the result of fuzzy logic methods may be able to estimate the life expectancy of the devices instead of stating their conditions.

REFERENCES

1. Schwab, K., *The Fourth Industrial Revolution*, Currency, 2017.

2. Zonta, T., C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade and G. P. Li, "Predictive Maintenance in the Industry 4.0: A Systematic Literature Review", *Computers & Industrial Engineering*, Vol. 150, p. 106889, 2020.

3. Budai, G., R. Dekker and R. P. Nicolai, "Maintenance and Production: A Review of Planning Models", *Complex System Maintenance Handbook*, pp. 321–344, 2008.

4. Motaghare, O., A. S. Pillai and K. Ramachandran, "Predictive Maintenance Architecture", *2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pp. 1–4, 2018.

5. Orhan, S., N. Aktürk and V. Celik, "Vibration Monitoring for Defect Diagnosis of Rolling Element Bearings as a Predictive Maintenance Tool: Comprehensive Case Studies", *Ndt & E International*, Vol. 39, No. 4, pp. 293–298, 2006.

6. Scheffer, C. and P. Girdhar, *Practical Machinery Vibration Analysis and Predictive Maintenance*, Elsevier, 2004.

7. Villareal, S., J. J. Pop, A. Hoefel and K. D. Harms, "Adaptive Pump Control for Positive Displacement Pump Failure Modes", US Patent 8,757,986, Jun. 24 2014.

8. Briggs, A. M., "Rotary Pump", US Patent 2,189,356, Feb. 6 1940.

9. "SPX FIP - Series Flexible Impeller Pumps, Instruction Manual", `https://www.spxflow.com/assets/pdf/JPM_IM_FIP-US.pdf`, accessed in May 2021.

10. Brennen, C. E., *Cavitation and Bubble Dynamics*, Cambridge University Press,

2014.

11. Geitner, F. K. and H. P. Bloch, "Chapter 5 - Vibration Analysis", F. K. Geitner and H. P. Bloch (Editors), *Machinery Failure Analysis and Troubleshooting (Fourth Edition)*, pp. 391–478, Butterworth-Heinemann, Oxford, fourth edition edn., 2012.

12. Zhao, W., Z. Wang, C. Lu, J. Ma and L. Li, "Fault Diagnosis for Centrifugal Pumps Using Deep Learning and Softmax Regression", *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, pp. 165–169, 2016.

13. Wang, Y., Z. Wei and J. Yang, "Feature Trend Extraction and Adaptive Density Peaks Search for Intelligent Fault Diagnosis of Machines", *IEEE Transactions on Industrial Informatics*, Vol. 15, No. 1, pp. 105–115, 2018.

14. Xia, M., T. Li, L. Xu, L. Liu and C. W. De Silva, "Fault Diagnosis for Rotating Machinery Using Multiple Sensors and Convolutional Neural Networks", *IEEE/ASME Transactions on Mechatronics*, Vol. 23, No. 1, pp. 101–110, 2017.

15. Jing, L., M. Zhao, P. Li and X. Xu, "A Convolutional Neural Network Based Feature Learning and Fault Diagnosis Method for the Condition Monitoring of Gearbox", *Measurement*, Vol. 111, pp. 1–10, 2017.

16. Zhang, D., E. Stewart, M. Entezami, C. Roberts and D. Yu, "Intelligent Acoustic-Based Fault Diagnosis of Roller Bearings Using a Deep Graph Convolutional Network", *Measurement*, Vol. 156, p. 107585, 2020.

17. Lei, J., C. Liu and D. Jiang, "Fault Diagnosis of Wind Turbine Based on Long Short-Term Memory Networks", *Renewable Energy*, Vol. 133, pp. 422–432, 2019.

18. O'Brien, R. J., J. M. Fontana, N. Ponso and L. Molisani, "A Pattern Recognition System Based on Acoustic Signals for Fault Detection on Composite Materials", *European Journal of Mechanics-A/Solids*, Vol. 64, pp. 1–10, 2017.

19. Salamon, J. and J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification", *IEEE Signal Processing Letters*, Vol. 24, No. 3, pp. 279–283, 2017.

20. "BirdCLEF 2016 — ImageCLEF / LifeCLEF - Multimedia Retrieval in CLEF", `https://www.imageclef.org/lifeclef/2016/bird`, accessed in February 2021.

21. Sprengel, E., M. Jaggi, Y. Kilcher and T. Hofmann, *Audio Based Bird Species Identification Using Deep Learning Techniques*, Tech. rep., 2016.

22. Nanni, L., G. Maguolo and M. Paci, "Data Augmentation Approaches for Improving Animal Audio Classification", *Ecological Informatics*, Vol. 57, p. 101084, 2020.

23. Ali, S. M., K. Hui, L. Hee and M. S. Leong, "Automated Valve Fault Detection Based on Acoustic Emission Parameters and Support Vector Machine", *Alexandria Engineering Journal*, Vol. 57, No. 1, pp. 491–498, 2018.

24. Hasan, M. J., M. M. Islam and J.-M. Kim, "Acoustic Spectral Imaging and Transfer Learning for Reliable Bearing Fault Diagnosis under Variable Speed Conditions", *Measurement*, Vol. 138, pp. 620–631, 2019.

25. Henze, D., K. Gorishti, B. Bruegge and J. Simen, "AudioForesight: A process Model for Audio Predictive Maintenance in Industrial Environments", *18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 352–357, 2019.

26. Liu, X., D. Pei, G. Lodewijks, Z. Zhao and J. Mei, "Acoustic Signal Based Fault Detection on Belt Conveyor Idlers Using Machine Learning", *Advanced Powder Technology*, Vol. 31, No. 7, pp. 2689–2698, 2020.

27. Salamon, J., C. Jacoby and J. P. Bello, "A Dataset and Taxonomy for Urban Sound Research", *22nd ACM International Conference on Multimedia (ACM-MM'14)*,

pp. 1041–1044, Orlando, FL, USA, Nov. 2014.

28. "Bearing Data Center", `https://csegroups.case.edu/bearingdatacenter/pages/download-data-file`, accessed in January 2021.

29. Guo, X., L. Chen and C. Shen, "Hierarchical Adaptive Deep Convolution Neural Network and Its Application to Bearing Fault Diagnosis", *Measurement*, Vol. 93, pp. 490–502, 2016.

30. Jia, F., Y. Lei, J. Lin, X. Zhou and N. Lu, "Deep Neural Networks: A Promising Tool for Fault Characteristic Mining and Intelligent Diagnosis of Rotating Machinery with Massive Data", *Mechanical Systems and Signal Processing*, Vol. 72, pp. 303–315, 2016.

31. Mühlbauer, M., H. Würschinger, D. Polzer, S. Ju and N. Hanenkamp, "Automated Data Labeling and Anomaly Detection Using Airborne Sound Analysis", *Procedia CIRP*, Vol. 93, pp. 1247–1252, 2020.

32. Sun, M., H. Wang, P. Liu, S. Huang and P. Fan, "A Sparse Stacked Denoising Autoencoder With Optimized Transfer Learning Applied to the Fault Diagnosis of Rolling Bearings", *Measurement*, Vol. 146, pp. 305–314, 2019.

33. Shao, H., H. Jiang, H. Zhao and F. Wang, "A Novel Deep Autoencoder Feature Learning Method for Rotating Machinery Fault Diagnosis", *Mechanical Systems and Signal Processing*, Vol. 95, pp. 187–204, 2017.

34. Durbhaka, G. K. and B. Selvaraj, "Predictive Maintenance for Wind Turbine Diagnostics Using Vibration Signal Analysis Based on Collaborative Recommendation Approach", *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1839–1842, 2016.

35. Paolanti, M., L. Romeo, A. Felicetti, A. Mancini, E. Frontoni and J. Loncarski, "Machine Learning Approach for Predictive Maintenance in Industry 4.0", *14th*

IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), pp. 1–6, 2018.

36. Shao, H., H. Jiang, Y. Lin and X. Li, "A Novel Method for Intelligent Fault Diagnosis of Rolling Bearings Using Ensemble Deep Auto-Encoders", *Mechanical Systems and Signal Processing*, Vol. 102, pp. 278–297, 2018.

37. Zhang, W., C. Li, G. Peng, Y. Chen and Z. Zhang, "A Deep Convolutional Neural Network With New Training Methods for Bearing Fault Diagnosis Under Noisy Environment and Different Working Load", *Mechanical Systems and Signal Processing*, Vol. 100, pp. 439 – 453, 2018.

38. Sudarsanam, S., "Fuzzy Logic Based Predictive Maintenance of Complex Equipments", *Second International Conferene on Business Analytics and Intelligence*, IISc Bangalore, 2014.

39. Baban, M., C. F. Baban and B. Moisi, "A Fuzzy Logic-Based Approach for Predictive Maintenance of Grinding Wheels of Automated Grinding Lines", *2018 23rd International Conference on Methods Models in Automation Robotics (MMAR)*, pp. 483–486, 2018.

40. Guo, L., Y. Lei, S. Xing, T. Yan and N. Li, "Deep Convolutional Transfer Learning Network: A New Method for Intelligent Fault Diagnosis of Machines with Unlabeled Data", *IEEE Transactions on Industrial Electronics*, Vol. 66, No. 9, pp. 7316–7325, 2018.

41. Zhang, R., H. Tao, L. Wu and Y. Guan, "Transfer Learning with Neural Networks for Bearing Fault Diagnosis in Changing Working Conditions", *IEEE Access*, Vol. 5, pp. 14347–14357, 2017.

42. Han, T., C. Liu, W. Yang and D. Jiang, "Learning Transferable Features in Deep Convolutional Neural Networks for Diagnosing Unseen Machine Conditions", *ISA*

Transactions, Vol. 93, pp. 341–353, 2019.

43. Carr, J. J. and J. M. Brown, *Introduction to Biomedical Equipment Technology*, Prentice Hall, 2001.

44. Mohammed, A. A., W. A. Moussa and E. Lou, "High Sensitivity MEMS Strain Sensor: Design and Simulation", *Sensors*, Vol. 8, No. 4, pp. 2642–2661, 2008.

45. "ReSpeaker 4-Mic Linear Array Kit for Raspberry Pi - Seeed Wiki", https://wiki.seeedstudio.com/ReSpeaker_4-Mic_Linear_Array_Kit_for_Raspberry_Pi/, accessed in May 2020.

46. "NASA, Raspberry Pi and a Mini Rover - Raspberry Pi", https://www.raspberrypi.org/blog/nasa-raspberry-pi-and-a-mini-rover/, accessed in May 2020.

47. "Buy a Raspberry Pi 3 Model B+ – Raspberry Pi", https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/, accessed in May 2020.

48. "Raspberry Pi Downloads - Software for the Raspberry Pi", https://www.raspberrypi.org/downloads/, accessed in May 2020.

49. Çoker, C., C. Demircan and M. Akar, *Proceedings of 21st National Committee of Automatic Control, Muğla, Turkey, September 2019*, p. 41–46.

50. Phinyomark, A., S. Thongpanja, H. Hu, P. Phukpattaranont and C. Limsakul, "The Usefulness of Mean and Median Frequencies in Electromyography Analysis", *Computational Intelligence in Electromyography Analysis-A Perspective on Current Applications and Future Challenges*, pp. 195–220, 2012.

51. Cabral, F. S., H. Fukai and S. Tamura, "Feature Extraction Methods Proposed for Speech Recognition are Effective on Road Condition Monitoring Using Smartphone

Inertial Sensors", *Sensors*, Vol. 19, No. 16, p. 3481, 2019.

52. Picone, J. W., "Signal Modeling Techniques in Speech Recognition", *Proceedings of the IEEE*, Vol. 81, No. 9, pp. 1215–1247, 1993.

53. Harris, F. J., "Chapter 3 - Multirate FIR Filters for Interpolating and Desampling", D. F. Elliott (Editor), *Handbook of Digital Signal Processing*, pp. 173–287, Academic Press, San Diego, 1987.

54. Giannakopoulos, T. and A. Pikrakis, *Introduction to Audio Analysis: A MATLAB® Approach*, Elsevier Science, 2014.

55. Demircan, C., *Vibration Based Condition Monitoring of Pumping Systems in Textile Dyehouses*, Master's Thesis, Boğaziçi University, Istanbul, Turkey, 2020.

56. Beale, H. D., H. B. Demuth and M. Hagan, "Neural Network Design", *PWS, Boston*, 1996.

57. Haykin, S. S. *et al.*, "Neural Networks and Learning Machines", , 2009.

58. Goodfellow, I., Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.

59. Buduma, N. and N. Locascio, *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*, O'Reilly Media, Inc., 2017.

60. Kingma, D. P. and J. Ba, "Adam: A Method for Stochastic Optimization", *arXiv preprint arXiv:1412.6980*, 2014.

61. Feng, L., S. Liu and J. Yao, "Music Genre Classification with Paralleling Recurrent Convolutional Neural Network", *arXiv preprint arXiv:1712.08370*, 2017.

62. Sharan, R. V. and T. J. Moir, "Acoustic Event Recognition Using Cochleagram Image and CNN", *Applied Acoustics*, Vol. 148, pp. 62–66, 2019.