VIBRATION BASED CONDITION MONITORING OF PUMPING SYSTEMS IN TEXTILE DYEHOUSES

by

Canberk Demircan

B.S., Electrical and Electronic Engineering, Boğaziçi University, 2015

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 Electronic Engineering
Boğaziçi University

2020

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. Mehmet Akar, for the patient guidance, encouragement and advice he has provided through this study. Also, I would like to thank all the members of staff at Eliar Elektronik San. A.Ş., especially to my colleagues from Smart Sensors and Embedded Systems Department.

This study was funded by Scientific and Technological Research Council of Turkey (TÜBİTAK) 1505 project 5180025, in collaboration of Bogazici University and Eliar Elektronik San. A.Ş and I would like to thank Ceren Coker for her effort and association during this project.

I would like to thank my wife and my family thousands time for everything. Finally, I thank my friends for supporting and motivating me during my thesis.

ABSTRACT

VIBRATION BASED CONDITION MONITORING OF PUMPING SYSTEMS IN TEXTILE DYEHOUSES

Condition monitoring of pumps is very important due to their critical role in industrial plants. This thesis focuses on vibration based fault detection and diagnosis of the flexible impeller pumps which are mostly used in pharmaceutical, food processing and textile finishing plants. The common problems of this type of positive displacement pumps are associated with the impellers and the cavitation is the main cause of impeller wear and damage. In the literature, there are many studies focused on the condition monitoring of positive displacement pumps. Nevertheless, none of these studies is particularly concentrated on the flexible impeller pumps. In this study, firstly, the data acquisition system has been devised for collecting vibration and motor current signals from the experimental setup. The separation ability of the features extracted from the vibration and motor current signals collected under different pump conditions has been investigated. The analysis regarding motor current signals has shown that they are not that distinctive to separate the conditions experienced by flexible impeller pumps. The classifiers exploiting features based on the time domain, frequency domain and time-frequency domain representations of vibration signals have been trained. The results regarding the performances of trained multi-class support vector machine and feedforward neural network classifiers have been presented as well. The findings of the thesis show that the feedforward neural networks exploiting wavelet variance features perform very well for classifying the flexible impeller pump conditions focused on. Eventually, a digital signal processing chain for fault detection and diagnosis of flexible impeller pumps has been proposed and realized in the embedded hardware.

ÖZET

TEKSTİL BOYAHANELERİNDEKİ POMPALAMA SİSTEMLERİNİN TİTREŞİM TABANLI DURUM İZLEMESİ

Pompalar, endüstriyel işletmelerde hayati öneme sahiptir ve bu nedenle durumlarının izlenmesi gerekmektedir. Bu tez, ilaç, yiyecek ve tekstil terbiye fabrikalarında sıklıkla kullanılan esnek kanatçıklı pompalarda meydana gelecek hataların titreşim işaretlerine dayalı tespit ve teşhisine odaklanmaktadır. Pozitif deplasmanlı bir pompa olan esnek kanatçıklı pompalarda en çok kanatçıklarla ilgili problemler gözlemlenmektedir. Kavitasyon ise kanatçık hasarına ve aşınmasına en çok sebep olan olgudur. Literatürde pozitif deplasmanlı pompalarla ilgili birçok çalışma olmasına rağmen, bu çalışmalardan hiçbirinde esnek kanatçıklı pompalara odaklanılmamıştır. Bu tezde, ilk olarak, deney düzeneğinden titreşim ve motor akımı işaretlerinin toplanabilmesi için gereken veri toplama sistemleri tasarlanmıştır. Sonrasında, pompanın farklı çalışma durumları altında toplanan verilerden titreşim ve motor akımı özellikleri çıkartılmıştır ve bu özelliklerin ayırt edicilikleri analiz edilmiştir. Motor akımı özelliklerinin pompanın farklı çalışma durumlarını ayırt etmekte başarılı olmadığı sonucuna varılmıştır. Titreşim işaretlerinin zaman, frekans ve zaman-frekans özelliklerini girdi olarak kullanan sınıflandırıcılar eğitilmiştir. Eğitilen karar destek makinesi ve yapay sinir ağı sınıflandırıcıları ile gerçekleştirilen performans testlerinin sonuçları sunulmuştur. Titreşim işaretlerinden elde edilen dalgacık varyansları kullanılarak eğitilen ileri beslemeli sinir ağları, esnek kanatçıklı pompanın çalışma durumlarını %99'un üstünde bir başarım oranıyla sınıflandırmıştır. Gerçekleştirilen çalışmaların son adımı olarak, esnek kanatçıklı pompalarda hata teşhis ve tespiti için önerilen sayısal işaret işleme zinciri, gömülü sistem üzerinde gerçeklenmiştir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES viii
LIST OF TABLES xi
LIST OF SYMBOLS xiv
LIST OF ACRONYMS/ABBREVIATIONS xv
1. INTRODUCTION 1
1.1. CM Techniques for Pumps 2
1.2. Overview of the Pumping Systems in Dyehouses 4
1.3. Motivation of the Thesis 5
1.4. Scope of the Thesis 6
2. SYSTEM DESCRIPTION 8
2.1. Accelerometer Selection 8
2.2. Experimental Setup 10
2.3. Acquisition of Vibration Signals 11
2.4. Acquisition and Analysis of Motor Current Signals 12
2.5. Summary of the Chapter 15
3. VIBRATION SIGNAL ANALYSIS METHODOLOGY 20
3.1. Vibration Signal Feature Extraction 20
3.1.1. Time Domain Signal Features 20
3.1.2. Frequency Domain Signal Features 25
3.1.3. Time-Frequency Domain Signal Features 26
3.2. SVM Classification of Vibration Signals 35
3.2.1. Overview of SVM Classifiers 35
3.2.2. Experiments with SVM Classifiers 42
3.3. Neural Network Classification of Vibration Signals 45
3.3.1. Overview of Feedforward Neural Network Classifiers 45
3.3.2. Evaluating the Error Gradients and Backpropagation 49

3.3.3. The Gradient Descent Algorithm and Its Variants Utilized . . . 52
3.3.4. Experiments with Feedforward Neural Network Classifiers . . . 55
3.4. Summary of the Chapter and Concluding Remarks 61
4. EMBEDDED SYSTEM REALIZATION OF THE PROPOSED METHOD . 62
5. CONCLUSION AND FUTURE WORK 67
REFERENCES 69

LIST OF FIGURES

Figure 2.1. Vibration axes on a pump. 8

Figure 2.2. Sensor placement on a pump. 10

Figure 2.3. The view of test setup. 11

Figure 2.4. Transimpedance amplifier circuit. 13

Figure 2.5. The magnitude response of “accumulate and average” stage vs normalized frequency in dB scale. 15

Figure 2.6. A window of the motor current measurements under normal operating condition. 16

Figure 2.7. A window of the motor current measurements for the condition with broken impeller. 16

Figure 2.8. A window of the motor current measurements under cavitation condition. 17

Figure 2.9. Oversampled ADC measurements of motor current for the condition with broken impeller. 17

Figure 2.10. The frequency spectrum of motor current signals, given in Figure 2.6. 18

Figure 2.11. The frequency spectrum of motor current signals, given in Figure 2.7. 18

Figure 2.12. The frequency spectrum of motor current signals, given in Figure 2.8. 19

Figure 3.1. Time domain vibration signal collected along the shaft axis. 21

Figure 3.2. The distribution of rms values of radial horizontal, x axis vibration measurements over all motor speeds. 22

Figure 3.3. The distribution of rms values of radial horizontal, x axis vibration measurements at 400 rpm. 23

Figure 3.4. The distribution of rms values of radial vertical, y axis vibration measurements at 400 rpm. 23

Figure 3.5. The distribution of variances of radial vertical, y axis vibration measurements at 400 rpm. 24

Figure 3.6. Autocorrelation sequence for the signal in Figure 3.1. 25

Figure 3.7. *db*16 decomposition low-pass filter. 30

Figure 3.8. *db*16 decomposition high-pass filter. 30

Figure 3.9. An example of third level detail coefficients of x, y and z axis vibration measurements under normal condition, respectively. 31

Figure 3.10. An example of third level detail coefficients of x, y and z axis vibration measurements under cavitation condition, respectively. . . 31

Figure 3.11. The distribution of wavelet variances estimated from fourth level detail coefficients of radial horizontal, x axis vibration measurements for each class. 32

Figure 3.12. The distribution of wavelet variances estimated from fourth level detail coefficients of radial horizontal, x axis vibration measurements at 300 rpm. 32

Figure 3.13. The distribution of wavelet variances estimated from fourth level detail coefficients of radial horizontal, x axis vibration measurements at 400 rpm. 33

Figure 3.14. The distribution of wavelet variances estimated from fourth level detail coefficients of radial horizontal, x axis vibration measurements at 500 rpm. 33

Figure 3.15. The distribution of wavelet variances estimated from third level detail coefficients of axial, z axis vibration measurements at 400 rpm. 34

Figure 3.16. A single neuron. 46

Figure 3.17. K parallel neurons. 47

Figure 3.18. The construction of a three layer, fully connected neural network. 48

Figure 3.19. Illustration of the forward and backpropagation along the network, in the direction of dashed and solid lines, respectively. 51

Figure 4.1. The flowchart of developed embedded system application. 64

LIST OF TABLES

Table 1.1. Improvements in plants applying condition based maintenance. . . 1

Table 2.1. Some specifications of the accelerometers examined. 9

Table 2.2. Cable connections for vibration data acquisition system. 12

Table 3.1. The coding scheme utilized for classifying four pump conditions. . 41

Table 3.2. The confusion matrix for the classification with SVM model trained using the first 9 features selected via MRMR. 43

Table 3.3. Test accuracies of the classifiers trained with different set of features. 43

Table 3.4. Test accuracies of the classifiers trained with wavelet variances estimated from vibration measurements on different axes. 44

Table 3.5. Test accuracies of the classifiers trained with combined wavelet variances estimated from vibration measurements on different axes. . . 44

Table 3.6. The confusion matrix for the classification with SVM model trained using wavelet variance features. 44

Table 3.7. The performance of different FFNN classifiers exploiting wavelet variances and trained with Levenberg-Marquardt algorithm and linear activation function at output layer. 56

Table 3.8. The performance of different FFNN classifiers exploiting wavelet variances and trained with Bayesian regularization backpropagation algorithm and the linear activation function at output layer. 57

Table 3.9. The performance of different FFNN classifiers exploiting wavelet variances and trained with Levenberg-Marquardt algorithm and softmax activation function at output layer. 57

Table 3.10. The performance of different FFNN classifiers exploiting wavelet variances and trained with Bayesian regularization backpropagation algorithm and softmax activation function at output layer. 58

Table 3.11. The training results for FFNN classifiers which exploit wavelet variances, have five hidden neurons, softmax activated output layer and are trained using CE loss function. 59

Table 3.12. The training results for FFNNs which are trained in the same way with the FFNNs in Table 3.11 excluding early stopping criterion. . 59

Table 3.13. The training results for FFNN classifiers exploiting wavelet variances and having five hidden neurons and softmax activated output layer. 60

Table 3.14. The confusion matrix belonging to test with the classifier whose hyperparameters shown in the last row of Table 3.13. 60

Table 3.15. The performances of networks exploiting time domain statistics and trained with Levenberg-Marquardt algorithm. 61

Table 4.1. Cable connections between EVAL-ADXL355 and STM32F746G-DISCO boards. 63

Table 4.2. The size of code and data in bytes for the application developed. . 65

Table 4.3. Test performance of the online CM system utilizing wavelet variance features. 66

Table 4.4. Test performance of the online CM system utilizing time domain statistics as features. 66

LIST OF SYMBOLS

b	Bias term
C_f	The value of feedback capacitor
f_r	Rotational frequency
f_{vp}	Vane pass frequency
I_{iMax}	Maximum output value of current sensor
g	Acceleration due to Earth's gravity, $\approx 9.81\ m/s^2$
h_a	Absolute pressure above the open tank
h_f	Friction loss in the piping system
h_{st}	Distance from the surface of the liquid to the impeller eye
h_{vpa}	Vapor pressure of the liquid at medium temperature
$L(\cdot)$	Lagrangian function
J_0	The level of wavelet transform
R_f	The value of feedback resistor
T_r	Rotational period
$\mathrm{w_{ij}}$	Weight from input i to output j
V_{ref}	The output voltage of external reference integrated circuit
V_{oMin}	The minimum digitized voltage
β	The inverse variance of Gaussian noise
η	The learning rate
λ	The regularization parameter
μ	The mean of measurement samples
ξ	Slack variables
σ	The standard of measurement samples
ϕ	Scaling function
Φ	Feature-space transformation
ψ	Wavelet function

LIST OF ACRONYMS/ABBREVIATIONS

3D	Three Dimensional
ADC	Analog to Digital Converter
ANN	Artifical Neural Network
CE	Cross Entropy
CM	Condition Monitoring
CMSIS	Cortex Microcontroller Software Interface Standard
CBM	Condition Based Monitoring
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DC	Direct Current
DSP	Digital Signal Processor
DWT	Discrete Wavelet Transform
ECOC	Error Correcting Output Code
IC	Integrated Circuit
IoT	Internet of Things
IIoT	Industrial Internet of Things
FFNN	Feedforward Neural Network
FFT	Fast Fourier Transform
FN	False Negative
FP	False Positive
LSTM	Long Short-Term Memory
MCSA	Motor Current Signature Analysis
MCU	Microcontroller Unit
MEMS	Microelectromechanical Systems
MODWT	Maximal Overlap Discrete Wavelet Transform
MRMR	Minimum Redundancy Maximum Relevance
MSE	Mean Squared Error
NN	Neural Network

NPSH	Net Positive Suction Head
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RMS	Root Mean Square
RPM	Revolutions Per Minute
SE	Shannon Entropy
SAR	Successive Approximation Register
SGD	Stochastic Gradient Descent
SPI	Serial Peripheral Interface
STFT	Short Time Fourier Transform
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
USB	Universal Serial Bus
WPT	Wavelet Packet Transform

1. INTRODUCTION

Maintenance strategies can be classified into three categories which are breakdown maintenance, preventive maintenance and predictive, i.e., condition based maintenance (CBM). Breakdown maintenance results in unexpected failures and productivity slowdown. Preventive maintenance is time based maintenance where maintenance is performed at scheduled intervals irrespective of the health status of the machinery with the aim of reducing downtime and increasing the quality of the final product. On the other hand, with the predictive maintenance strategy, maintenance action is only taken when there is a sign of abnormality in the condition of the equipment being monitored. Condition monitoring (CM) may be performed online or offline. Online CM requires permanently mounted transducers to the equipment being monitored. Collected data are conveyed to a central processing or logging unit via wired or wireless systems. Data sampling may be continuous or at decided intervals. Analysis of data is done simultaneously with the data collection. On the other hand, offline monitoring is performed in specified time ranges such as weekly or monthly using portable data acquisition devices. Data logged is analyzed and interpreted later, usually by experienced technicians or a specialized software program. Online CM systems have become popular in recent years although they require additional computation power, memory and extra sensors as well. These necessities initially may increase the cost of the overall system. However, this can be compensated gradually through the efficient utilization of equipment and increases in production efficiency. The results of a recent survey given in Table 1.1 show that predictive maintenance strategies offer several advantages to the companies.

Table 1.1. Improvements in plants applying condition based maintenance [1].

Uptime improvement	9%
Maintenance cost reduction	12%
Decrease in safety and quality risks	14%
Lifetime extension of equipment	20%

1.1. CM Techniques for Pumps

Cavitation, impeller damages, seal faults, rotor faults and bearing faults can be listed as the common failures encountered in pumps. Most rotating equipment gives certain signs, warnings or indications of their impending failure. Many CM techniques are available to monitor equipment conditions. Acoustic monitoring is one of these techniques. With this technique, the sound of equipment is recorded and analyzed. Another CM technique for rotating equipment, especially for the motors, is electrical signal analysis in which the current or voltage of the motor is measured and analyzed. On the other hand, vibration monitoring is probably the most popular CM technique for rotating equipment. Several other techniques such as oil, temperature analysis are also available in the literature.

Many time domain, time-frequency domain and frequency domain signal processing techniques have been developed and various feature extraction methods have been proposed in the literature over the years [2]. These studies are differentiated from each other in terms of the choice of sensors, the data acquisition methods, and feature extraction techniques.

The analysis of vibration and audio signals are widely applied to detect the cavitation in pumps [3]. Henríquez *et al.* studied fault diagnosis using both audio and vibration signals in a circulation pump [4]. Motor current signature analysis (MCSA) is also applied for diagnosing the impeller related faults and cavitation [5]. The methods using inlet and outlet pressure measurements for cavitation detection are also studied [6].

Descriptive statistics such as standard deviation, kurtosis, skewness, root mean square (rms) and variance are often used time domain vibration features. There are many other statistical parameters suggested in the literature that have been applied for fault diagnosis of rotating machinery such as peak factor, entropy, crest factor, range, minimum value, maximum value and sum [7].

The features extracted from the frequency domain and time frequency domain are generally more distinctive than time domain features due to the fact that characteristics of frequency components are more associated with the faults [8]. Fast Fourier Transform (FFT) is probably the most popular signal analysis technique and has been widely applied to identify the source of frequency components from vibration signals [9]. The power spectrum and the logarithm of the power spectrum can also be used to diagnose failure modes in machinery [10]. Statistical parameters such as variance, skewness, kurtosis [11] etc. extracted from the frequency domain may also form useful feature sets for fault diagnosis classifiers. Farokhzad *et al.* [12], extracted a set of statistical features from the frequency domain of vibration signals and those features were fed into an artificial neural network (ANN) for fault diagnosis in centrifugal pumps. The variants of the wavelet transform such as discrete wavelet transform (DWT) and discrete wavelet packet analysis have also been applied for extracting the features to diagnose rotating machinery faults [13].

Recently, applying approaches based on deep learning for the fault detection and diagnosis of rotating machines have become popular. The evolution of hardware and increasing data availability have contributed to this trend greatly and it seems that this trend will continue to grow. Rui *et al.* [14] reviewed the deep learning applications for condition based monitoring. Zhang *et al.* [15], compiled the state-of-the-art deep learning algorithms for the applications aiming to detect the bearing faults. Elsaid *et al.* [16] utilized the long short-term memory (LSTM) networks to predict the occurrence of excess vibration situations in aircraft engines. Fengtao *et al.* [17], have exploited time domain, time-frequency domain and frequency domain features of vibration signals to train LSTM networks in order to predict the remaining useful life time of rolling bearings. Xiaohan *et al.* [18] proposed a method utilizing the convolutional neural networks (CNNs) to extract vibration signal features and using a LSTM network trained exploiting automatically extracted features to detect the problems in bearings.

1.2. Overview of the Pumping Systems in Dyehouses

Pumping systems are utilized to transfer a wide range of fluids such as water based fluids, fuels, oils, acids and alkaline to the process tanks in industrial plants. In dye houses, those systems are utilized to transfer water and chemicals to the bath where the fabrics are dyed. These liquid transfer systems usually consist of more than one pump arranged in parallel, electric motors which drive the pumps and liquid storage tanks. Flexible impeller pumps are commonly utilised in food processing, pharmaceutical, oenological and textile finishing plants. They consist of mechanical housing and a rotating rubber impeller with vanes contacting with the inner wall of the housing. The rotation of flexible impeller creates a vacuum in which the pressure is low at the inlet of the pump and this permits the liquid to be pushed towards the pump by the atmospheric pressure. The impeller of pumps may wear out over time and this results in deterioration in the hydraulic performance, reduction in pumping pressure, flow rate and efficiency of the pump.

The most common failure mode for flexible impeller pumps is damaged, worn impellers. The primary cause of impeller wear is cavitation. Operating the pump when there is no liquid and transferring incompatible chemicals with impeller material also result in impeller faults. Cavitation occurs when the inlet pressure of the pump falls below the saturation vapor pressure of the liquid [19]. Consequently, small vapor-filled cavities are formed inside the pump. These cavities may collapse and generate an intense shock wave leading to loud noise and a high level of vibration which may damage the components of the pump.

The key consideration parameter against cavitation is the net positive suction head (NPSH). The required NPSH is the minimum pressure needed at the inlet of the pump to prevent cavitation and it is generally provided by the pump manufacturer. The available NPSH for a flooded suction, $NPSA_a$ is calculated as follows:

$$NPSH_a = h_a - h_{vpa} + h_{st} - h_f \tag{1.1}$$

where h_a denotes the value of absolute pressure above the open tank, h_{vpa} is vapor pressure of the liquid at medium temperature, h_{st} is distance from the surface of the liquid to the impeller eye and h_f represents the friction loss in the piping system. The pumping systems must be designed in a way that available absolute pressure at the pump inlet is always greater than the required NPSH [20]. However, designing the system may be complicated when taking several factors such as the range of pump speed, temperature, impeller material and lubrication properties of the liquid being pumped into consideration. The other causes of impeller damage are debris, incompatible chemicals, and running the pump when there is no liquid.

The maintenance reports obtained from the technical service and dyehouses shows that the problems related to sealing elements are encountered as well. This problem mostly emerges due to the eccentricity and looseness of the pump body, set screws. The looseness and mechanical imbalance of the pump body may wear the other parts such as shaft and bearing and reduces the service life of the transfer system.

1.3. Motivation of the Thesis

Most of the modern textile machines are highly productive. However, when there is any fault in the liquid transfer system, the manufacturing is interrupted and the productivity decreases. To prevent this, the fault must be detected and corrected before emerging. In turn, utilizing an online CM system which tracks the pump condition, detects and diagnoses the faults and creates alerts and notifications accordingly increases manufacturing productivity and extends the service life of pumping system. But, many textile finishing plants suffer the lack of CM of the components of liquid transfer systems. This need constitutes the source of motivation for this study.

There are various studies available in the literature as mentioned in Section 1.1. Although the different types of positive displacement pumps have been focused on these studies, none of them is specifically focused on the flexible impeller pumps. In this thesis, classifying operating conditions experienced by flexible impeller pumps with the utilization of vibration and motor current sensors has been targeted. This study

focuses on four different pump conditions which are cavitation, broken or worn impeller, mechanical looseness and normal operating condition. Due to the lack of research on the flexible impeller pumps, the novelty of this study is to develop CM algorithms for this type of pump.

There is a wide range of online and offline CM solutions currently available in the market. However, with the advancements in the Industrial Internet of Things (IIoT), the plants are becoming smart and data driven. Therefore, the data must be ready at any time to make inferences and take actions accordingly. These needs and requirements place online CM solutions ahead of offline CM solutions. On the other hand, the number of IoT devices grows exponentially and the data collection and analysis infrastructures need to be changed since the network bandwidth is limited. At this point, it could be said that "edge computing" would complement "cloud computing" in the near future of IIoT applications. This study has been conducted as a part of project named "Developing Integrated Sensor Fusion Algorithms for Fault Detection and Diagnosis of Mechatronics Systems in Textile Dyehouses" and supported by TUBITAK under the "1505 University and Industry Cooperation Call". The objective of this project is to provide a basis for the data acquisition systems and signal processing algorithms for an online CM solution running on the edge.

1.4. Scope of the Thesis

The following chapters in this report are committed to the details of the experimental setup, motor current and vibration signal analyzes conducted for the classification of flexible impeller pump's condition. Different feature extraction methods and classifiers are visited as well. Also, the implementation of the developed signal processing chain on top of the embedded hardware is elaborated.

Chapter 2 is reserved for the preliminary study to collect data reliably from the experimental setup. Moreover, the accelerometer selection process and vibration data acquisition system are detailed. Motor current signal acquisition system and the results obtained through MCSA are presented as well.

Chapter 3 deals with the analysis and classification of vibration measurements collected from the experimental setup. Feature extraction techniques exploited in this study and the feedforward neural network as well as support vector machine multi-class classifiers are visited. Also, the results related to training and test of multi-class classifiers are presented.

In Chapter 4, the implementation of the proposed signal processing chain on top of the embedded system is detailed. The test results of the developed online CM system are provided as well.

In the concluding Chapter 5, the findings of the study and future work are discussed.

2. SYSTEM DESCRIPTION

In this chapter, vibration sensor selection process and the elements and construction of experimental setup are detailed. Also, the data acquisition systems devised to collect vibration and motor current signals are elaborated. Lastly, the results of motor current signal analysis are discussed.

2.1. Accelerometer Selection

The vibration can be simply defined as the backward and forward motion of a mass from its fixed position. Figure 2.1 demonstrates the sides of pump and directions in which the vibration measurements can be collected. Accelerometers are commonly utilized to track the mechanical vibration levels of an equipment. Displacement probes and velocity sensors may be utilized as well. The accelerometer specifications such as measurement range, operating principle, frequency response, power consumption and resolution of the accelerometers differ from one accelerometer to another [21]. The piezoelectric and microelectromechanical systems (MEMS) capacitive type accelerometers are commonly used in condition based monitoring systems.

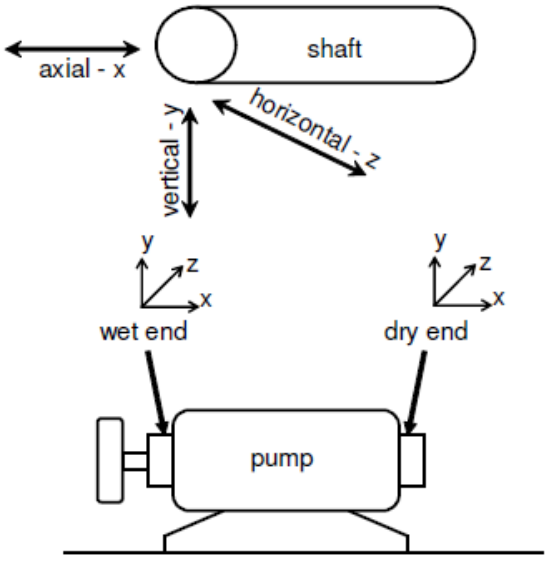

Figure 2.1. Vibration axes on a pump. Reprinted from [22].

The key selection criteria for condition based monitoring accelerometers are having wide bandwidth, minimal measurement drift over the time and low noise. However, the piezoelectric accelerometers perform poorly at low frequencies, but certain mechanical faults such as looseness and misalignment can be diagnosed by analysing vibration freuency spectrum in low speed applications [23]. On the other hand, MEMS based capacitive accelerometers offer better direct current (DC) response and they may have integrated high and low pass filters which simplifies the signal conditioning front end. Upon completing the research on accelerometer technologies, several MEMS based accelerometer models have been investigated. Table 2.1 shows the characteristics of those accelerometers. In this table, the measurement resolution is only specified for the accelerometers with digital interface.

Table 2.1. Some specifications of the accelerometers examined.

Accelerometer model	**Noise density** ($\mu g/\sqrt{Hz}$)	**Maximum range (g)**	**Bandwidth (kHz)**	**Resolution (bits)**
ADXL1001	30	± 100	11	-
ADXL1002	25	± 50	11	-
ADXL354	20	± 8	1.5	-
ADXL355	20	± 8.192	1	20
ADXL356	80	± 10	1.5	-
ADXL357	80	± 40.96	1	20
LSM6DSM	130	± 16	3.3	16

The location of sensors are very important to get reliable vibration measurements for further signal analysis. The sensor must be located at a point where the direct path to the vibration sources is available through the equipment. With rotating machines, this path is almost always passed through the machine's bearings since dynamic forces generated by a rotating equipment are transferred to the structure over the bearings. Figure 2.2 shows an example pump casing and sensor locations on it. When the most direct path to the vibration sources is considered for Figure 2.2, it could be said that the most suitable sensor location for axial measurements is on the edge of flange, not

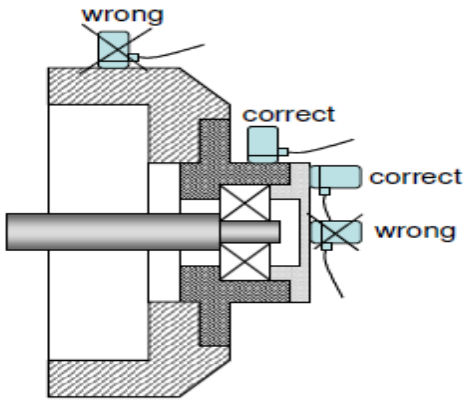

Figure 2.2. Sensor placement on a pump. Reprinted from [22].

in the middle of it. Considering this discussion, the accelerometer has been located on the pump cover. A plastic body has been produced using 3D printer to fasten the accelerometer board. After fastening the accelerometer to 3D printed body, it has been fixed to the pump cover using adhesive material.

2.2. Experimental Setup

The test setup is consisted of a water tank, pump, a three phase induction motor, an inverter, two manual valves and water pipes as shown in Figure 2.3. A three axis accelerometer and electrical current sensor are utilized to collect vibration and current signals for different conditions of the flexible impeller pump. In order to decide the target speeds of the induction motor, the databases of dyehouses have been examined and accordingly, the measurements have been recorded at different motor speeds between 250 and 800 revolutions per minute (rpm).

The signals have been acquired under four several cases which are pump with healthy impellers and worn impellers, cavitation and mechanical looseness. The worn impellers have been obtained from the dyehouses. Cavitation condition is generated by throttling the valve in the inlet of the pump. Lastly, the mechanical looseness condition has been produced by loosing the screw that is used to fasten the pump to the motor body.

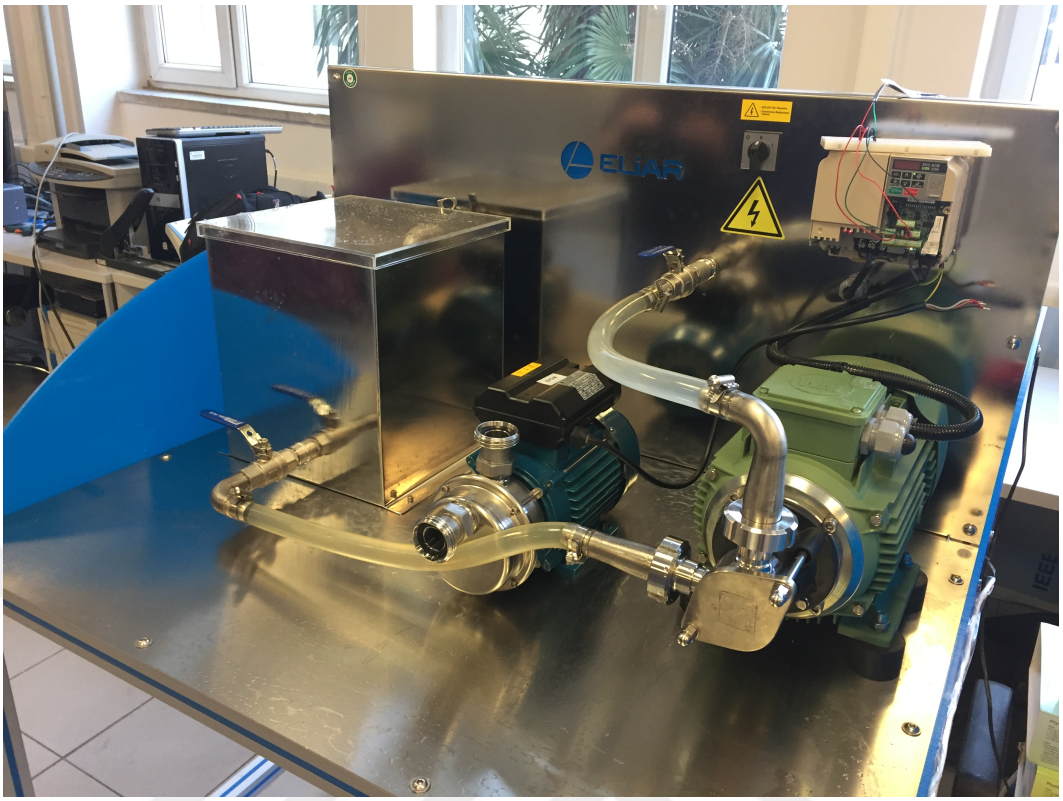

Figure 2.3. The view of test setup.

2.3. Acquisition of Vibration Signals

The vibration signals are measured using EVAL-ADXL355 evaluation board containing 3-axis, MEMS type accelerometer. The measurements are read from the ADXL355 through Serial Peripheral Interface (SPI) protocol by a main microcontroller unit (MCU). Then, the MCU transmits collected data through Universal Serial Bus (USB) to the computer. A Python script listening to serial port writes the incoming measurement and timestamp data to a text file. The embedded software needed for acquiring vibration measurements and transferring them to the computer has been developed on top of the STM32F4 discovery board. Table 2.2 demonstrates the connections between the EVAL-ADXL355 and STM32F4 discovery boards. The cutoff frequency of the low pass filter integrated to the accelerometer has been set to 1 kHz and -3 dB point of the integrated high pass filter has been set to 2.5 Hz. Moreover, the measurement range and the output data rate of the accelerometer was set to $\pm$8.192g and 4 kHz, respectively. Vibration signals have been collected in radial horizontal, radial vertical and axial planes, corresponding to x, y and z axes measurements, re-

Table 2.2. Cable connections for vibration data acquisition system.

STM32F4 discovery board connector and pin	**EVAL-ADXL355 connector and pin**	**Pin function**
P2-3V3	P1-V_{DDIO}	Digital side voltage supply
P2-3V3	P1-VDD	Voltage supply
P2-GND	P1-GND	Ground
P1-A6	P2-MISO	Master in slave out
P1-A7	P2-MOSI	Master out slave in
P1-A5	P2-SCLK	Clock pin
P1-A3	P2-CS	Chip select pin
P2-D5	P1-DRDY	Data ready interrupt pin

spectively at different motor speeds between 250 and 800 rpm when the motor rotates at its target speed.

2.4. Acquisition and Analysis of Motor Current Signals

A clamp type, Hall effect current sensor that is capable of measuring the motor currents up to 30 A has been used as the main sensing component. It has 0-10 mA current output type. In order to measure the output of current sensor, an analog front end integrated circuit (IC) which consists of an transimpedance amplifier and an analog to digital converter (ADC) driver has been utilized. Also, fast switching schottky diodes has been used to protect ADC inputs against overvoltage situations.

The transimpedance operational amplifier circuit configuration which is demonstrated in Figure 2.4, works as a current to voltage converter. The feedback resistor, R_f determines the current to voltage gain. Then, R_f should be selected such that the maximum reachable output voltage corresponds to the maximum current output of the sensor, I_{iMax}, permits the utilization of the full range of ADC.

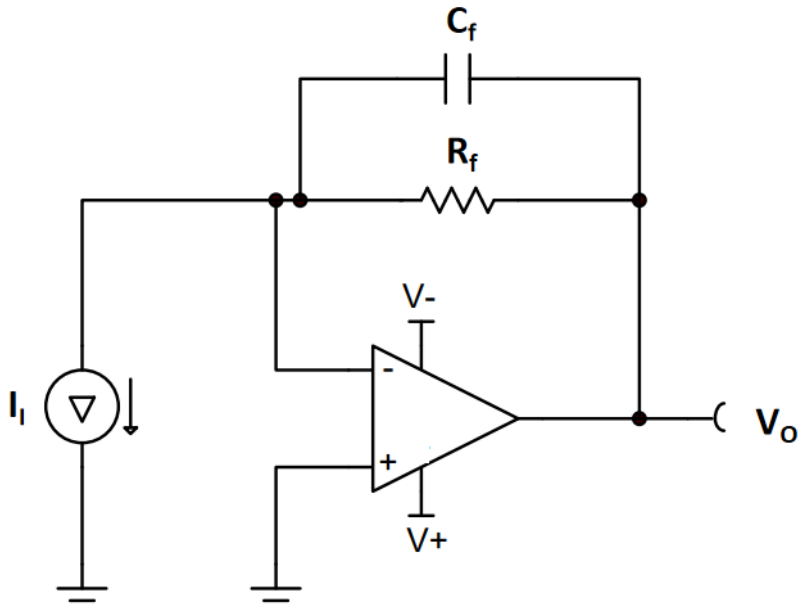

Figure 2.4. Transimpedance amplifier circuit.

The maximum current drawn by the motor is around 15 A. In this case, the current sensor outputs I_{iMax} which is 5 mA. On the other hand, the analog part of the MCU is supplied with an external reference IC which has 3 V output, V_{ref}. Also, the minimum voltage, V_{oMin}, which can be digitized by ADC peripheral is 0 V. The feedback resistor, R_f is chosen as 600 Ω by using the following equation:

$$R_f = \frac{V_{ref} - V_{oMin}}{I_{iMax}}. \tag{2.1}$$

Then, the feedback capacitor, C_f must be chosen in a way that the following inequality is satisfied,

$$C_f \leq \frac{1}{2 \times \pi \times R_f \times f_p}. \tag{2.2}$$

where f_p denotes the required bandwidth. Accordingly, the feedback capacitor, C_f, value is selected as 18 nF. The frequency spectrum between 0 and 5 kHz captures enough information for motor current signature analysis [24]. It is generally preferred to sample the input at a frequency of 5 or 10 times needed bandwidth in practical applications. In this application, the microcontroller unit (MCU) which has a successive approximation register (SAR) ADC peripheral have been programmed to sample the

output of the ADC driver with a frequency of 29 kHz. Figure 2.6, Figure 2.7 and Figure 2.8 show the example motor current measurements collected, respectively for normal operating condition, broken impeller and cavitation cases. The calibration procedure for analog front end is not performed. But, the output voltage of voltage reference IC, V_{ref} and the value of feedback resistor used in the transimpedance amplifier circuit have been utilized in order to convert the raw ADC measurements, M_{ADC}, to motor current values:

$$I = M_{ADC}/(2^{14} - 1)/R_f \times V_{ref} \times 3000 \tag{2.3}$$

where the nominal current transformation ratio of current sensor is 3000. Also, to filter high pass noise, the measurements are collected by utilizing hardware oversampling unit of the MCU. This unit has been programmed to simply accumulate N samples and output the division of the sum by N without overloading the central processing unit (CPU). Since the sampling cycle of ADC unit is modified accordingly, the oversampler unit again outputs measurements at a frequency of 29 kHz. The result is simply the average of summed samples as follows:

$$Result = \frac{1}{N} \times \sum_{0}^{N-1} Conversion(t_n). \tag{2.4}$$

The frequency response "accumulate and average" stage is equivalent to a first order cascaded integrator-comb Hogenauer filter. The magnitude response of this stage is demonstrated in Figure 2.5. The phase response of this stage is linear which means that all frequency components of the input signal are delayed in time by the same constant amount.

After the acquisition of motor current measurements under different operating conditions, time domain statistics and the frequency spectrum of current signals have been examined. Some examples of the frequency spectra of motor current measurements are illustrated in Figure 2.10, Figure 2.11 and Figure 2.12. Oversampled motor current signals have been analyzed as well. The pump speed may be easily obtained by

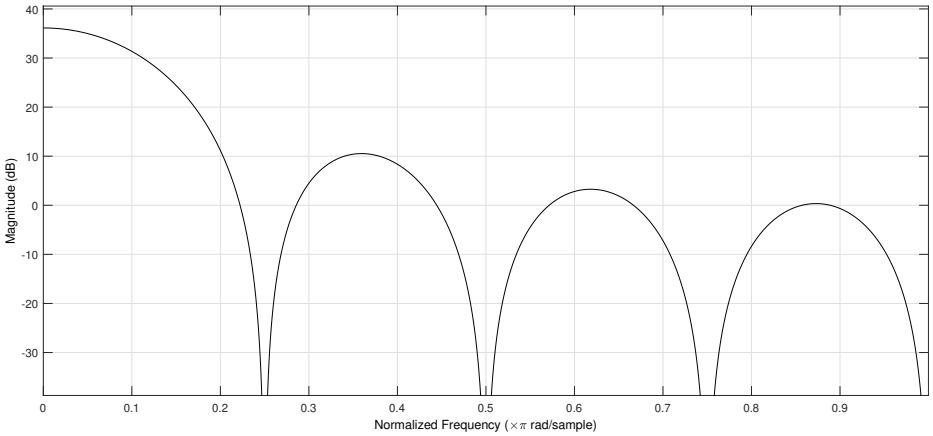

Figure 2.5. The magnitude response of "accumulate and average" stage vs normalized frequency in dB scale.

determining the duration between the two consecutive peak points or zero crossings. But, no other significant differences were found between the motor current measurements. Consequently, it has been decided to focus on vibration measurements of the pump instead of current measurements of the motor.

2.5. Summary of the Chapter

The implementation of CM mainly consists of four steps. The first step is data acquisition referring to collection and storage of condition data from the asset being monitored. In the second step of CM, the data are prepared for further analysis. This step generally includes data cleaning and preprocessing. In Chapter 2, the studies conducted in order to complete the requirements for the first two steps of CM have been elaborated. Also, the results of analyzes regarding motor current signals have been presented. It has been reached that the motor current signals do not provide distinctive features to classify the conditions experienced by the flexible impeller pump. The reason for this result may be that the inverter eliminates the fluctuations sourced by the load faults. Subsequently, it has been decided to analyze the vibration signals thoroughly.

The third step of CM is feature extraction and selection. Lastly, the fourth step of CM includes assessing the data and giving diagnostic or prognostic recommendations. Chapter 3 addresses the analyzes conducted to complete the last two steps of CM.

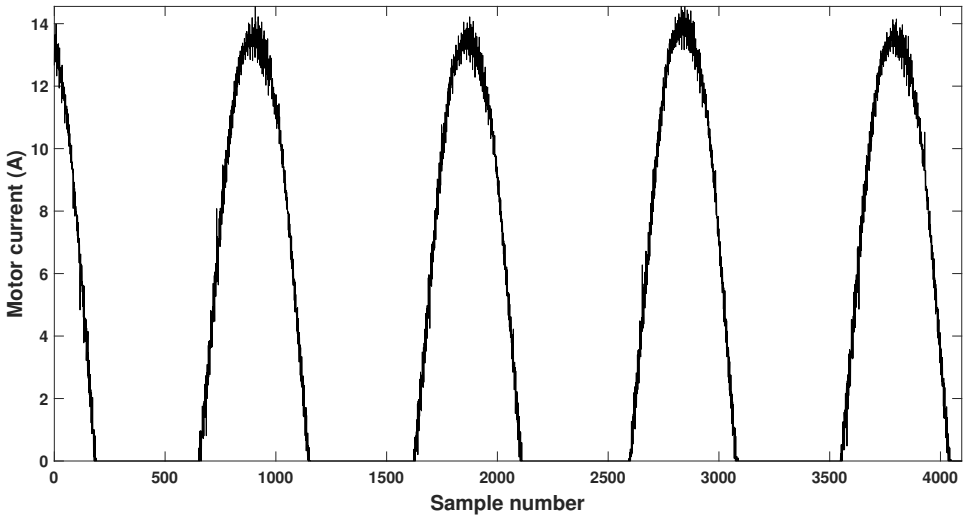

Figure 2.6. A window of the motor current measurements under normal operating condition.

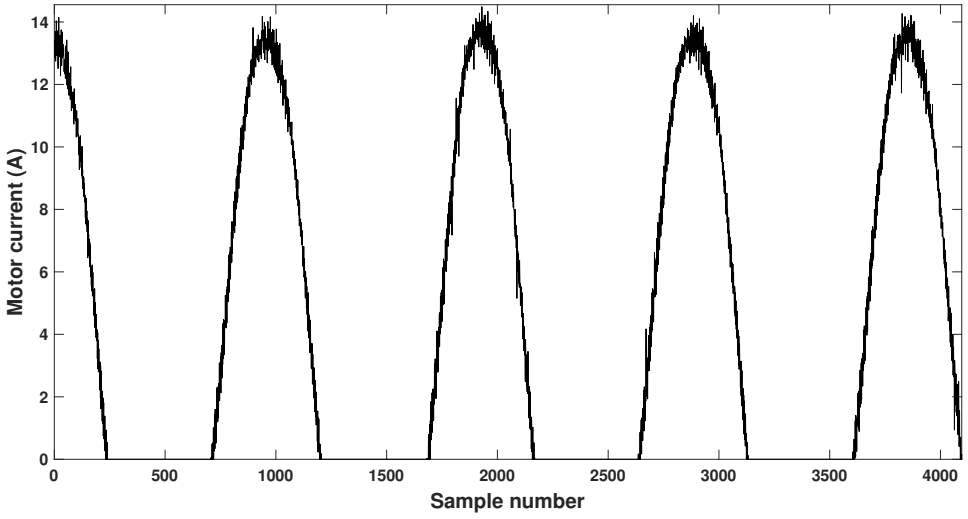

Figure 2.7. A window of the motor current measurements for the condition with broken impeller.

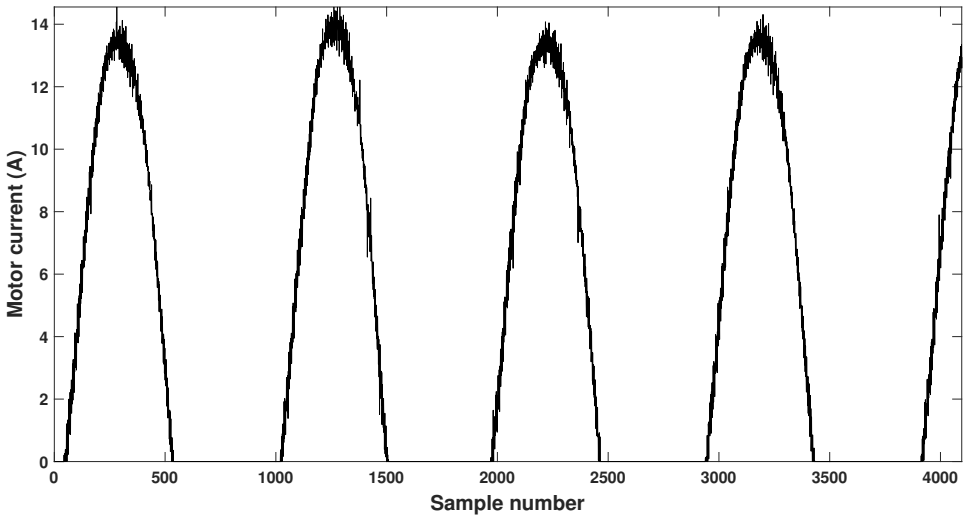

Figure 2.8. A window of the motor current measurements under cavitation condition.

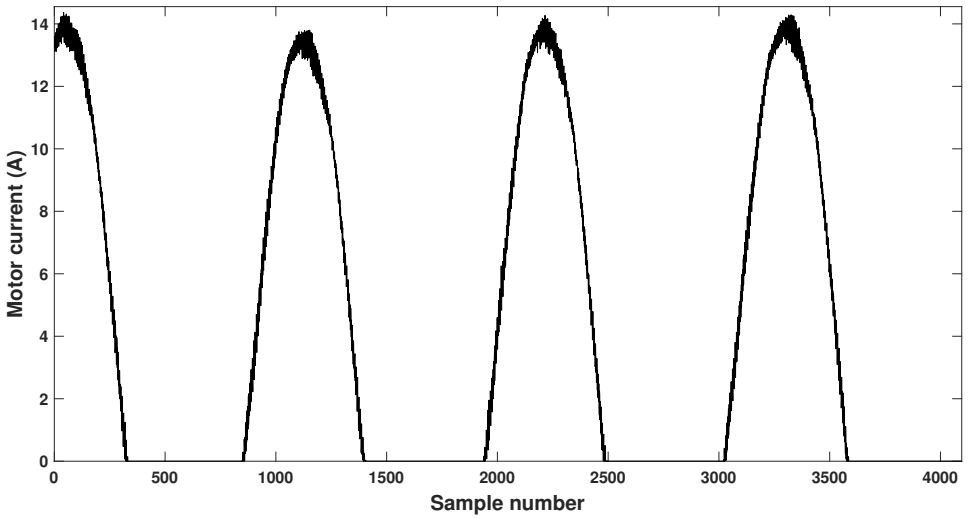

Figure 2.9. Oversampled ADC measurements of motor current for the condition with broken impeller.

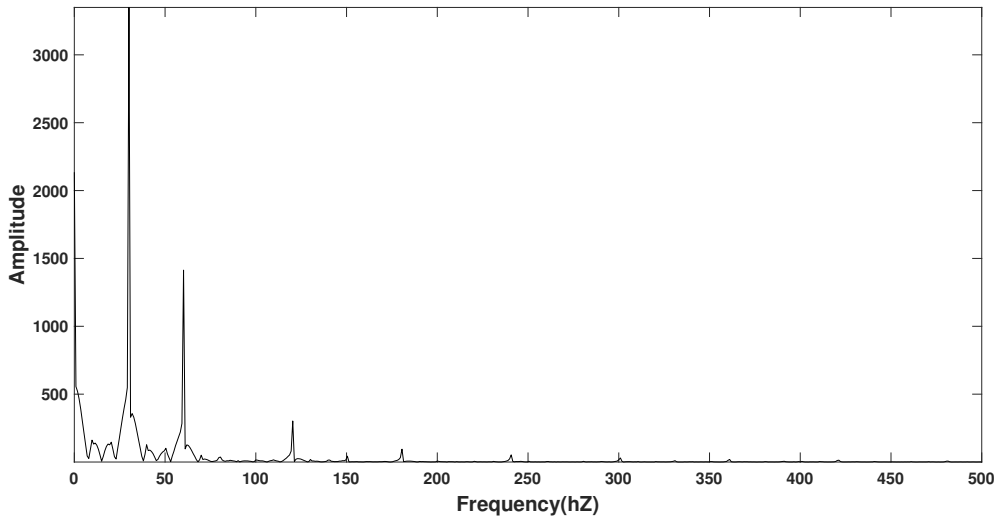

Figure 2.10. The frequency spectrum of motor current signals, given in Figure 2.6.

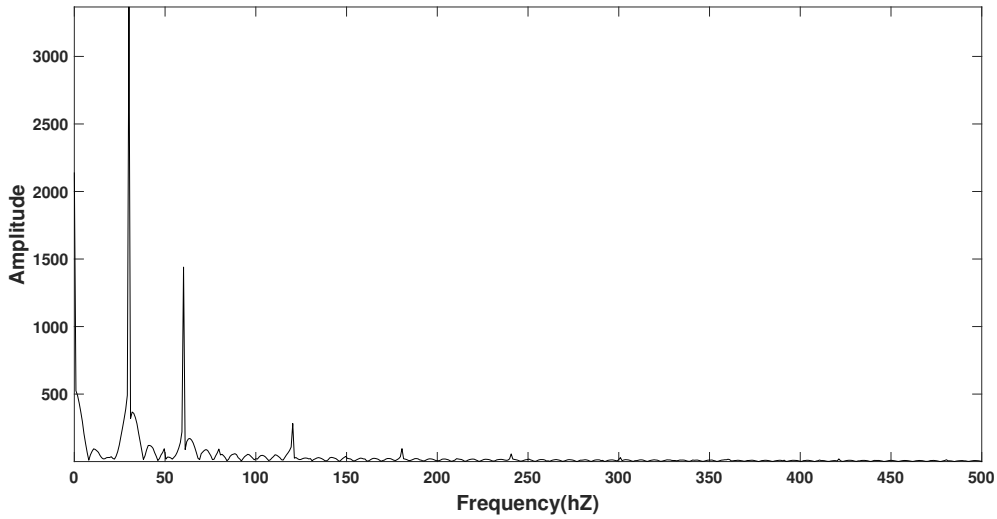

Figure 2.11. The frequency spectrum of motor current signals, given in Figure 2.7.

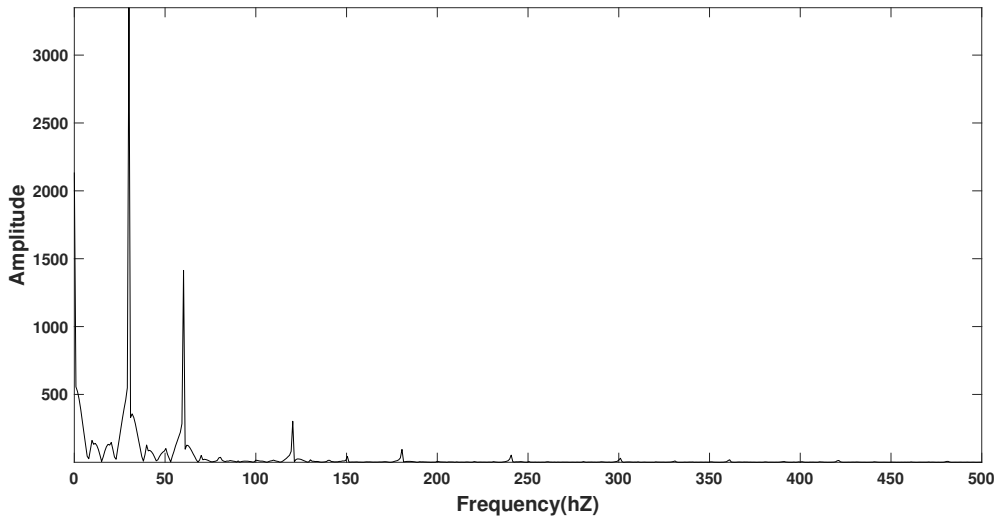

Figure 2.12. The frequency spectrum of motor current signals, given in Figure 2.8.

3. VIBRATION SIGNAL ANALYSIS METHODOLOGY

In Chapter 2, the first two steps of CM, namely data acquisition and preprocessing, have been completed and the data sets needed for the analysis of vibration signals have been constructed. The next steps of CM include extracting features, selecting features and taking decisions regarding the condition of equipment accordingly. This chapter investigates feature extraction and selection procedures applied to vibration signals. The mathematical backgrounds and performances regarding the feedforward neural network (FFNN) and support vector machine (SVM) classifiers are presented as well.

3.1. Vibration Signal Feature Extraction

The dimension of raw measurements collected from accelerometer is too large and the classifier cannot be trained effectively with redundant raw data. Thus, data should be reduced and transformed into its set of features containing the relevant information of corresponding class. Time-domain, frequency-domain and time-frequency domain analysis are common methods for feature extraction. After obtaining the signal features, a feature selection may be applied to determine distinctive subsets of initial signal features. In this study, vibration signal features have been extracted from the measurements collected along the x, y and z directions. Each measurement is consisted of 2000 sample points, i.e. the duration of the measurements to be analysed is 500 ms. Before extracting frequency domain and time-frequency domain signal features, mean subtraction is applied to measurements. In the following subsections, utilized feature extraction methods are elaborated.

3.1.1. Time Domain Signal Features

As mentioned in Section 1.1, time domain statistics such as rms, variance, skewness and kurtosis are conventionally used as the vibration signal features for fault

diagnosis of rotating machinery. These statistics are calculated as follows:

$$rms = \sqrt{\frac{1}{N}\sum_{i=1}^{N} X_i^2} \tag{3.1}$$

$$variance = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \mu)^2 \tag{3.2}$$

$$skewness = \frac{1}{N-1}\sum_{i=1}^{N}\frac{(X_i - \mu)^3}{\sigma^3} \tag{3.3}$$

$$kurtosis = \frac{1}{N-1}\sum_{i=1}^{N}\frac{(X_i - \mu)^4}{\sigma^4} \tag{3.4}$$

where X_i is the ith sample of measurement window whose length is denoted by N. Also, μ and σ represent the mean and standart deviation of measurement samples, respectively. The signal statistics given above have been calculated for each measurement window to determine whether they are distinctive or not for impeller related problems, mechanical looseness and cavitation. Figure 3.1 depicts an example of vibration measurements collected along the shaft axis in time domain.

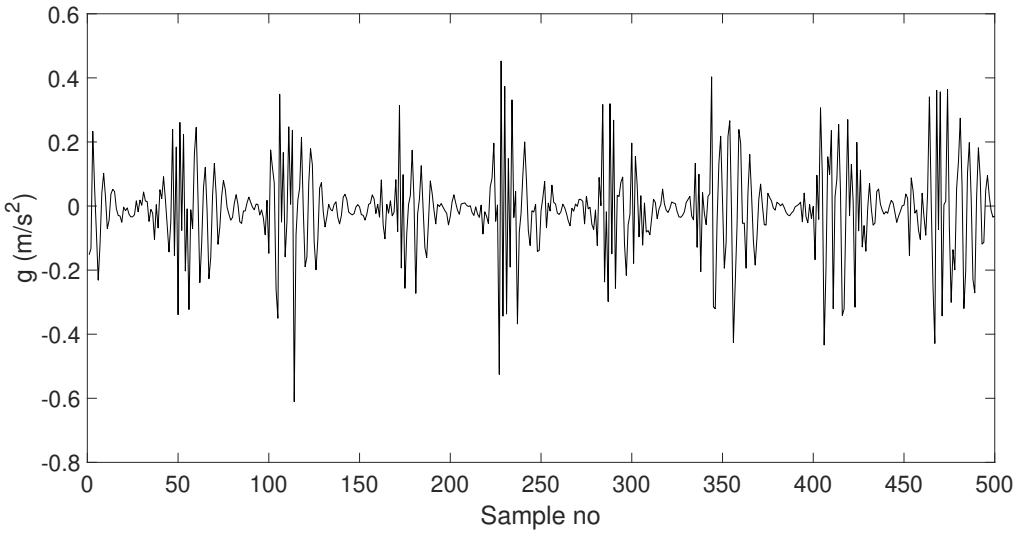

Figure 3.1. Time domain vibration signal collected along the shaft axis.

After calculating the time domain features aforementioned, box plots which show the distribution of the features for different operating conditions and motor speeds have been drawn. To illustrate, Figure 3.2 demonstrates the rms value distribution of vibration signals at radial horizontal axis for all measurements whereas Figure 3.3 shows the rms value distribution only for the measurements recorded when the motor speed is 400 rpm. As it can be noticed that classifying the pump condition by only focusing on the rms values of measurements seems unfeasible without knowing the motor speed. On the other hand, if the motor speed is known it may be possible to classify the pump condition by focusing the rms values. Figure 3.4 and Figure 3.5 show the distribution of rms and variance values of vibration signals at radial vertical axis, respectively, for the measurements recorded when the motor speed is 400 rpm. As can be seen in the figures, these specific time domain features perform well for separating pump conditions at 400 rpm motor speed.

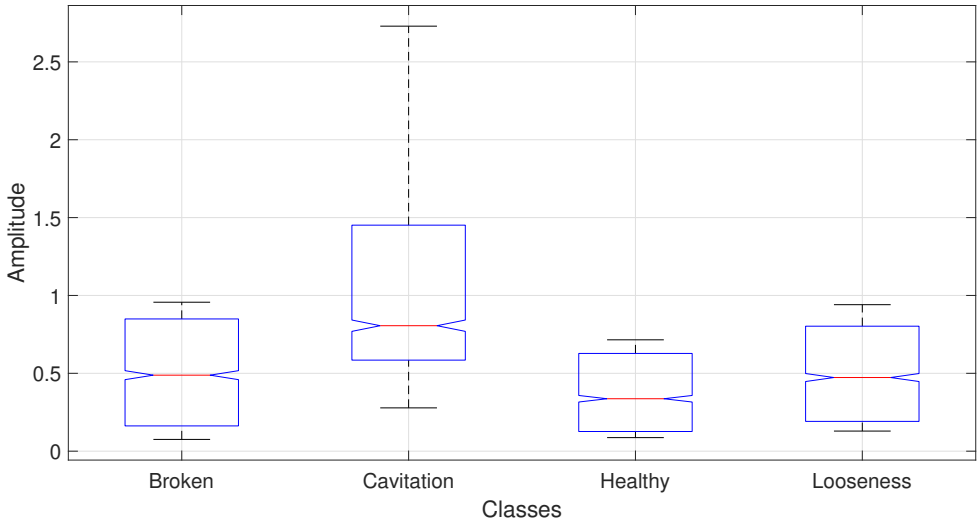

Figure 3.2. The distribution of rms values of radial horizontal, x axis vibration measurements over all motor speeds.

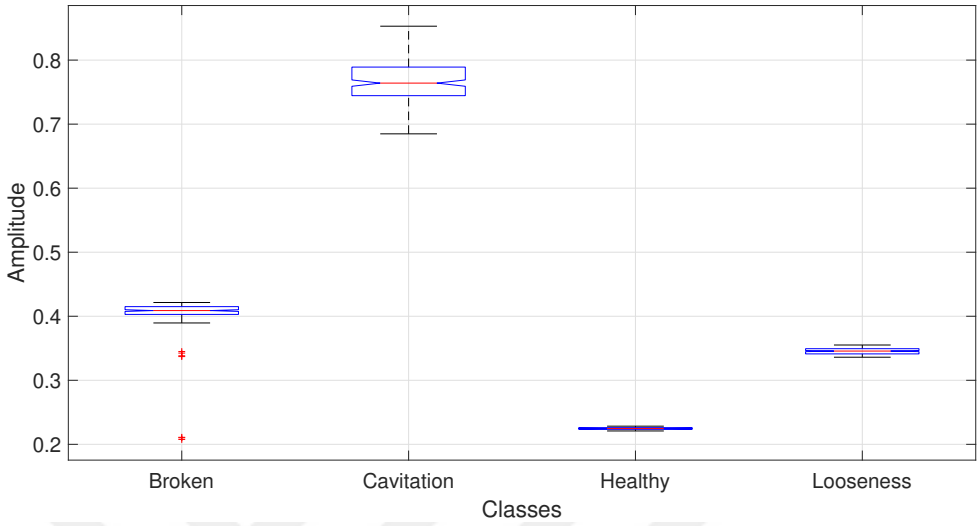

Figure 3.3. The distribution of rms values of radial horizontal, x axis vibration measurements at 400 rpm.

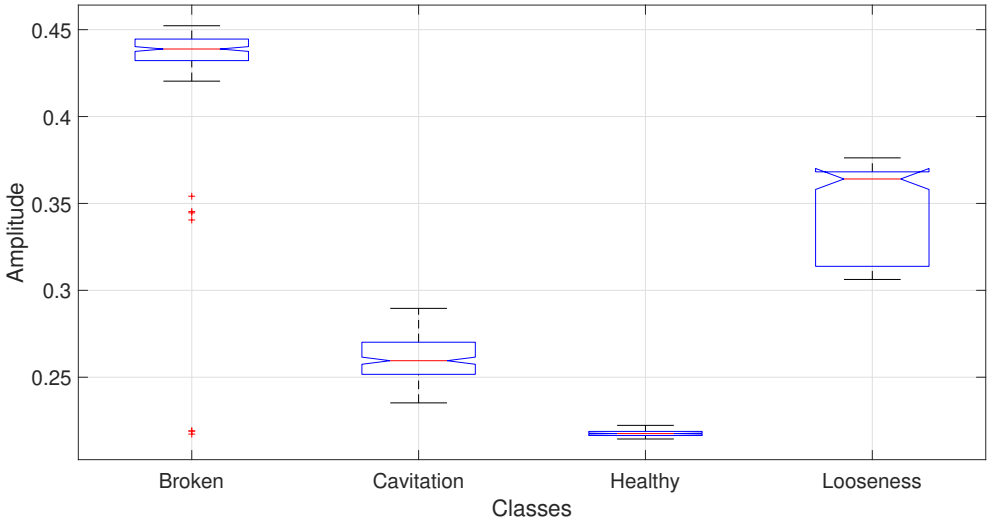

Figure 3.4. The distribution of rms values of radial vertical, y axis vibration measurements at 400 rpm.

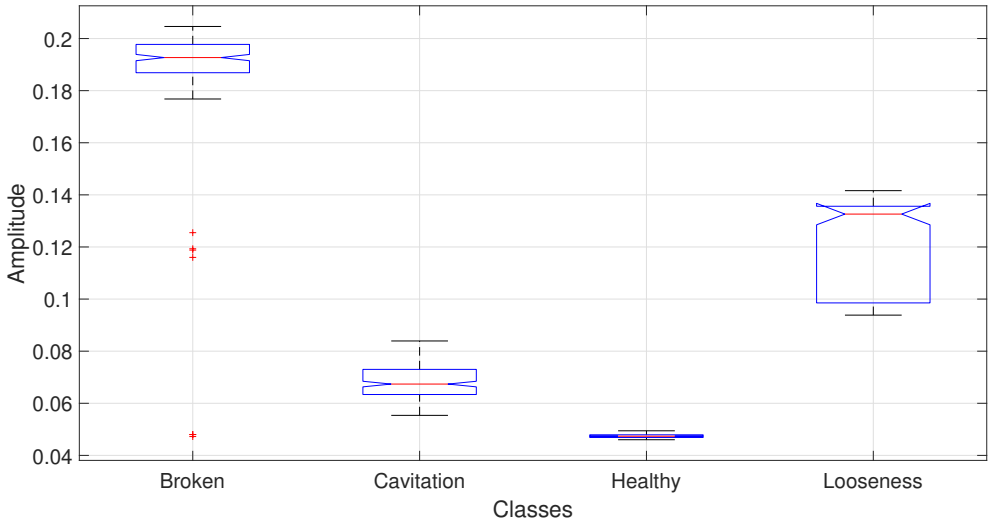

Figure 3.5. The distribution of variances of radial vertical, y axis vibration measurements at 400 rpm.

3.1.2. Frequency Domain Signal Features

Determining the rotational frequency of the pump is very important for extracting distinctive frequency domain features of vibration signals. Since the vanes of flexible impeller pump are always in contact with the housing, synchronous vibrations are generated. The periodic structure of acquired vibration signals can be seen in Figure 3.1. The autocorrelation function of a periodic function is also periodic and they have the same period. In this case, it is possible to determine the rotational frequency using autocorrelation analysis. The minimum and maximum pump speed during the experiments are 250 rpm and 800 rpm, respectively. Then, the peak location of autocorrelation function between 75 ms and 240 ms lags dashed in Figure 3.6, gives T_r, the rotational period. The rotational frequency, f_r and the vane pass frequency, f_{vp} are equal to $1/T_r$ and f_r times the number of vanes, respectively. At this point, there is a limitation which must be paid attention to determine the rotational frequency correctly. The requirement is that the measurement window must cover at least two cycles of the impeller. Moreover, autocorrelation analysis must be applied to vibration measurements along the shaft axis in order to obtain accurate results. Once the rotational frequency is found, the local maxima, the estimates of the mean normalized frequency and power around the frequencies, f_r and f_{vp} are calculated as vibration signal features.

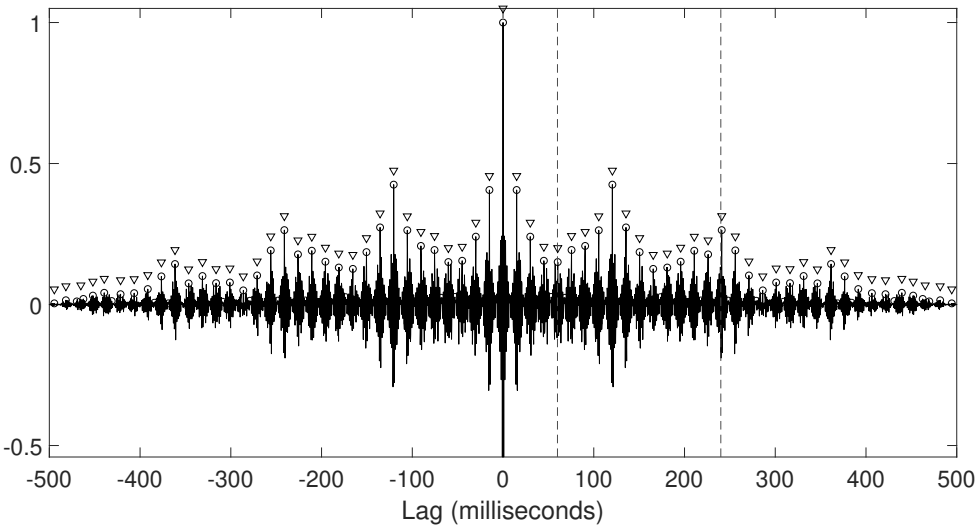

Figure 3.6. Autocorrelation sequence for the signal in Figure 3.1.

In order to obtain reliable results under all operating conditions, the duration of measurement window must be extended. This increases the need of computational power and extends the duration of analysis as well. Thus, the current sensors or magnetometers must be preferred for determining the rotational frequency of the motor and pump reliably.

3.1.3. Time-Frequency Domain Signal Features

Time-frequency domain analysis techniques such as discrete wavelet transform (DWT) and short time Fourier transform (STFT) use basis functions localized in both frequency and time domain in order to decompose the signals. They are commonly used for vibration signal analysis and also well suited for analyzing non-stationary vibration signals [25, 26]. In this study, the maximal overlap discrete wavelet transform (MODWT) and wavelet packet transform (WPT) methods are adopted. In the following section, it is investigated how the variances of MODWT applied series and Shannon entropies of wavelet packets are distinctive for pump conditions which are focused on this study.

The MODWT, which is very similar to the DWT, simply applies a linear filtering operation to a discrete time series. As a result, the data are transformed into the coefficients related to variations at different scales and translations. The function, $f(x)$ evaluated at N-many points can be represented as a linear combination of the wavelet function, $\psi(x)$ and scaling function, $\phi(x)$:

$$f(x) = \sum_{k=0}^{N-1} c_k 2^{-J_0/2} \phi(2^{-J_0}x - k) + \sum_{j=1}^{J_0} f_j(x) \tag{3.5}$$

where

$$f_j(x) = \sum_{k=0}^{N-1} d_{j,k} 2^{-j/2} \psi(2^{-j}x - k). \tag{3.6}$$

MODWT of level J_0 requires $O(N \log_2 N)$ multiplications whereas DWT of level J_0 can be obtained using $O(N)$ multiplications for a sample size N which is an integer multiple of 2^{J_0}. Although DWT is advantageous over MODWT in terms of computational price, the main disadvantage of DWT is that variances estimated using DWT change according to the initial point of input time series, while variances estimated utilizing MODWT do not. In this study, the MODWT is adopted for wavelet analysis since it has a number of attractive features over DWT [27]. Using MODWT creates a computational burden compared with DWT but its computational price is the same as commonly used FFT and pretty acceptable.

MODWT has basis vectors located at a time and a scale for each decomposition level $j = 1, ..., J_0$. For a J_0 level transform, it yields the vectors $\mathbf{W_1}, \mathbf{W_2}, ..., \mathbf{W_{J_0}}$ containing the wavelet coefficients related with changes in signal, $\mathbf{X}$ over different scales,

$$W_{j,t} = \sum_{l=0}^{L_j-1} h_{j,l} \mathbf{X}_{t-l \bmod N} \tag{3.7}$$

and $\mathbf{V_{J_0}}$ vector containing scaling coefficients related with variations at scale 2^{J_0} and higher,

$$V_{j,t} = \sum_{l=0}^{L_j-1} g_{j,l} \mathbf{X}_{t-l \bmod N} \tag{3.8}$$

where $t = 0, ..., N-1$, $\{h_{j,l} : l = 0, ..., (L_j - 1)\}$ and $\{g_{j,l} : l = 0, ..., (L_j - 1)\}$ are the jth level MODWT wavelet and scaling filters, respectively. Also, the width of the base filter is denoted by L and L_j is equal to $(2^j - 1)(L - 1) + 1$.

Most of the mother wavelet selection methods proposed in the literature relies on the similarity between the signals to be analyzed and mother wavelet [28]. A series of analysis has been conducted with $db6, db8, db12$ and $db16$. It has been deduced that the wavelet variance features extracted using $db16$ performed slightly better than others and $db16$ wavelet family has been adopted for wavelet analysis.

Figure 3.7 and Figure 3.8 demonstrate the wavelet and scaling functions of $db16$ wavelet family, respectively. MODWT is applied to each measurement window containing one thousand sample points. Then, unbiased multiscale variance estimates of all detail coefficients and final level scale coefficients are calculated as

$$\hat{v}_j^2 = \frac{1}{N_j} \sum_{t=0}^{N_j-1} W_{j,t}^2 \tag{3.9}$$

where N_j, is the effective width of jth level filter. Figure 3.9 and Figure 3.10 show the examples series of detail coefficients at third level for normal and cavitation conditions, respectively.

The maximal overlap discrete wavelet packet transform (MODWPT) is an extension of MODWT. The MODWPT distributes the total signal energy among the wavelet packets at each level and it is an energy preserving transform. Wavelet packet tree at four levels using $db16$ wavelets is obtained. Then, Shannon entropy (SE) which is a measure of uncertainty related to random variables in information theory, is calculated based on energy. The energy in the kth node at jth level can be calculated as

$$E_{j,k} = \sum_{t=0}^{N-1} \|W_{j,k,t}\|^2 . \tag{3.10}$$

The probability of the tth coefficient in the kth node is calculated as

$$P_{j,k,t} = \frac{\|W_{j,k,t}\|^2}{E_{j,k}} \tag{3.11}$$

which is used to obtain SE in the kth mode at jth level as

$$SE_{j,k} = -\sum_{t=0}^{N-1} P_{j,k,t} log(P_{j,k,t}). \tag{3.12}$$

Figure 3.11 shows the distribution of wavelet variance values calculated from the 4th level detail coefficients of the vibration measurements on radial horizontal axis

over each class. Figure 3.12, Figure 3.13 and Figure 3.14 depict the box plots of same feature computed for the measurements recorded when the motor speed is 300 rpm, 400 rpm and 500 rpm, respectively. As can be seen from the figures, similar to the findings reached in time domain feature analyzes, it seems that individual wavelet variance features are not able to classify pump condition without the information of the motor speed. It can be concluded that the wavelet variances computed from the 3rd level detail coefficients of the axial vibration measurements performs well when the motor speed is known. The box plot shown in Figure 3.15 illustrates how this wavelet variance feature serves to separate the classes.

To summarize, 30 features have been extracted from the vibration measurements collected along each direction using the methods detailed in this section. 16 Shannon entropy features from the 4th level MODWPT nodes and 6 wavelet variance estimates are calculated using the vibration measurements on each axis. Moreover, engineered frequency domain features which are mean frequencies and band powers around corresponding vane pass frequencies for the acceleration measurements on each axis are computed. Variance, kurtosis, skewness and rms statistics as well have been calculated from the time domain signals. Eventually, 90 features in total have been extracted from each record. In Section 3.1, the feature extraction techniques exploited during the study have been detailed. The next step is to assign those features to one of four classes. To achieve this, feedforward neural network and support vector machine multi-class classifiers have been trained. In the subsequent sections, the theoretical backgrounds of those classifiers and conducted experiments are elaborated.

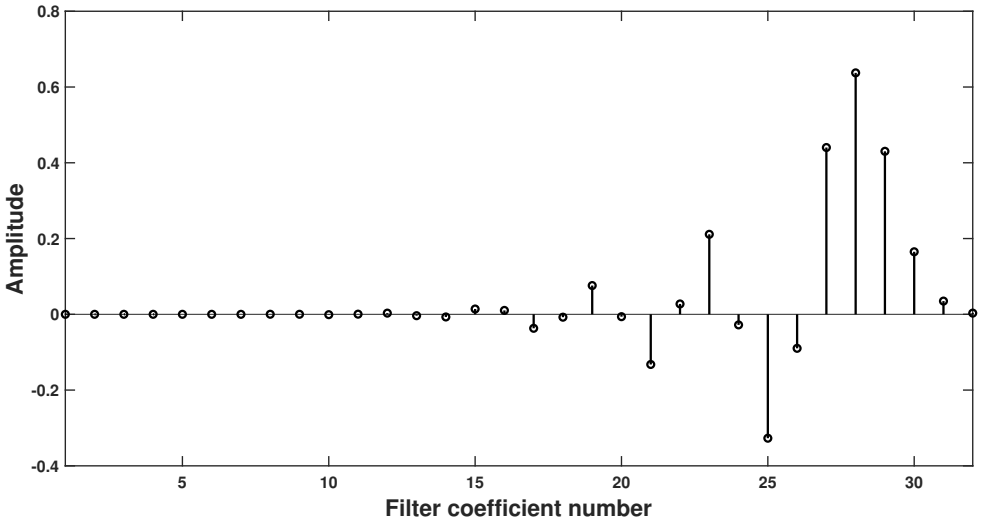

Figure 3.7. *db*16 decomposition low-pass filter.

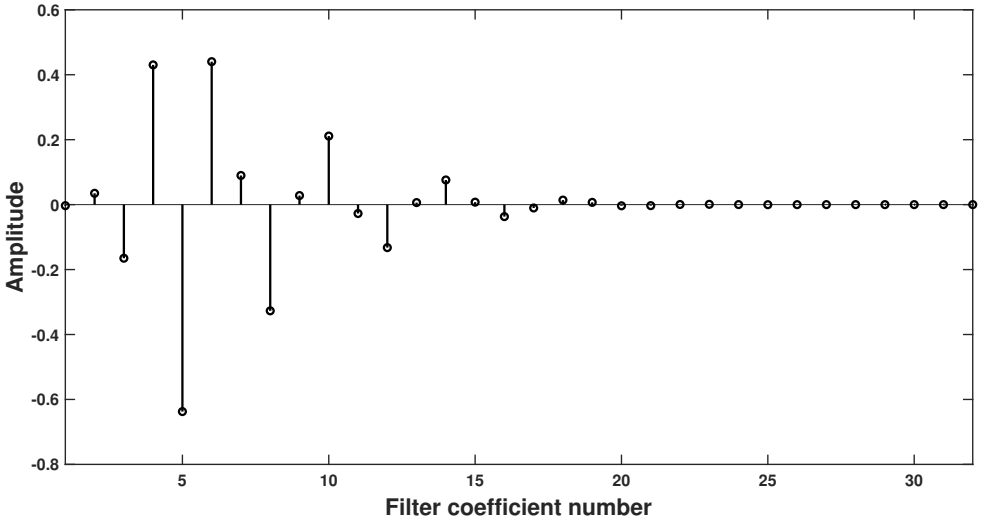

Figure 3.8. *db*16 decomposition high-pass filter.

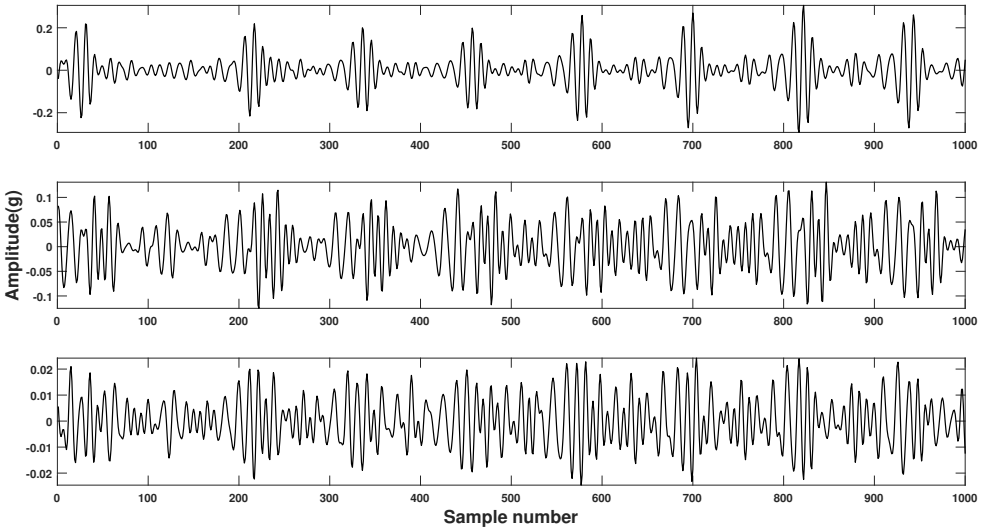

Figure 3.9. An example of third level detail coefficients of x, y and z axis vibration measurements under normal condition, respectively.

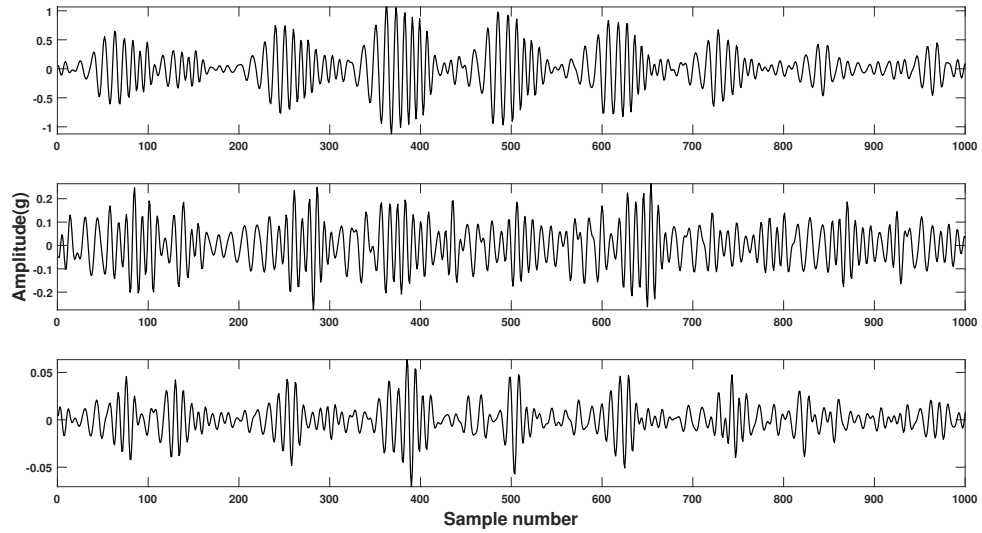

Figure 3.10. An example of third level detail coefficients of x, y and z axis vibration measurements under cavitation condition, respectively.

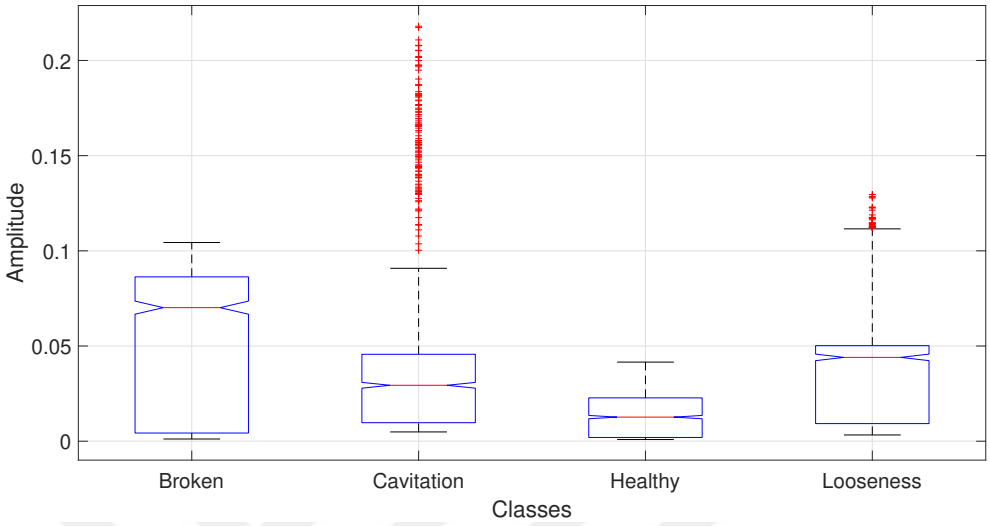

Figure 3.11. The distribution of wavelet variances estimated from fourth level detail coefficients of radial horizontal, x axis vibration measurements for each class.

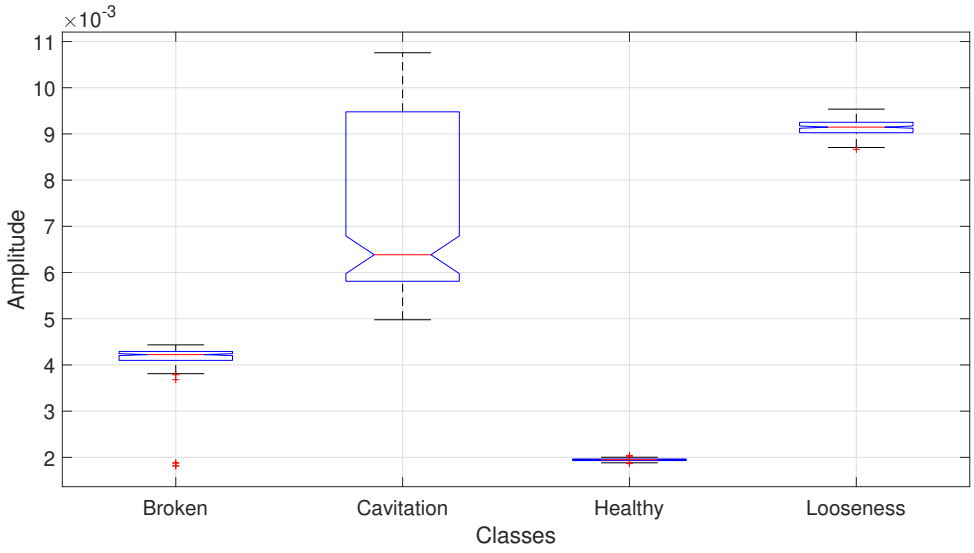

Figure 3.12. The distribution of wavelet variances estimated from fourth level detail coefficients of radial horizontal, x axis vibration measurements at 300 rpm.

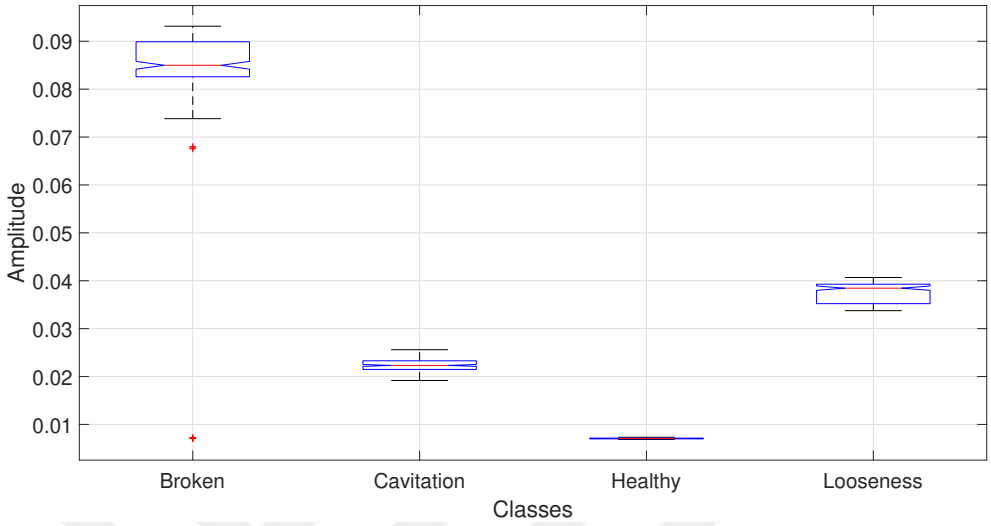

Figure 3.13. The distribution of wavelet variances estimated from fourth level detail coefficients of radial horizontal, x axis vibration measurements at 400 rpm.

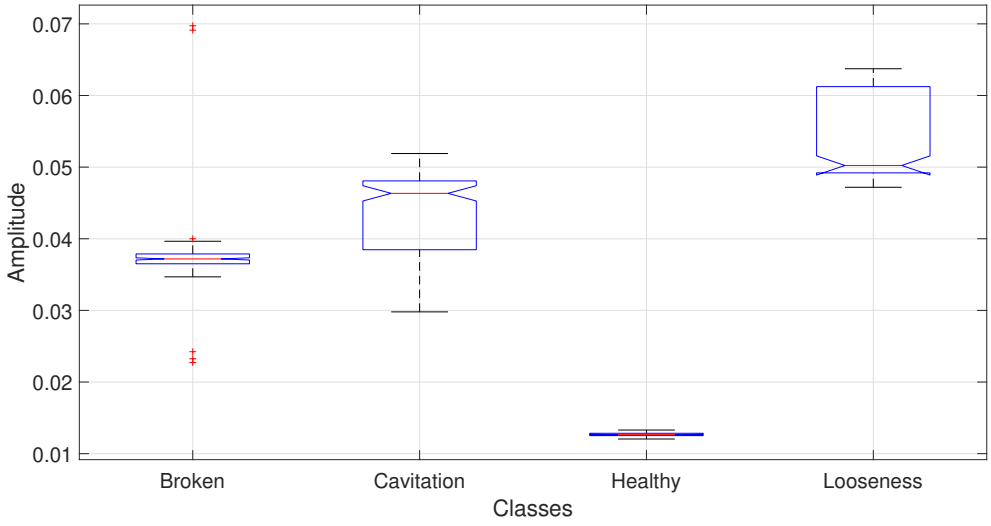

Figure 3.14. The distribution of wavelet variances estimated from fourth level detail coefficients of radial horizontal, x axis vibration measurements at 500 rpm.

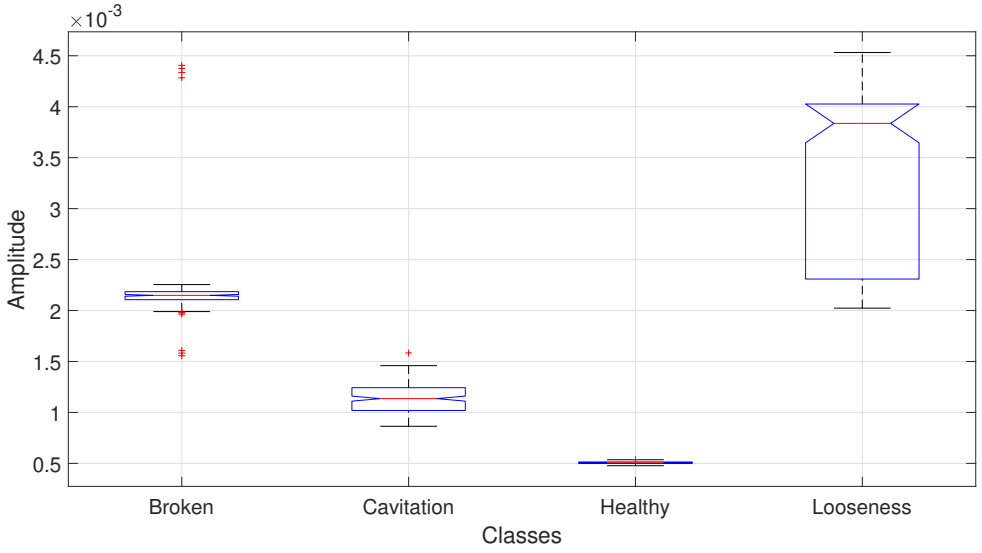

Figure 3.15. The distribution of wavelet variances estimated from third level detail coefficients of axial, z axis vibration measurements at 400 rpm.

3.2. SVM Classification of Vibration Signals

This section addresses the mathematical background of SVM classifiers and analyzes carried out with SVM classifiers. The approach adopted to train multi-class SVM classifiers is also detailed. Finally, the results regarding the performance tests of trained SVM classifiers are presented.

3.2.1. Overview of SVM Classifiers

Two-class classification problem using linear models is visited to introduce support vector machines in the form of,

$$y(\mathbf{x}) = \mathbf{w}^T \Phi(\mathbf{x}) + b \tag{3.13}$$

where $\Phi(\mathbf{x})$ and b represent an specific feature-space transformation and bias, respectively. The training data set is consisted of N-many input vectors $x_1, ..., x_N$ and target values $t_1, ..., t_N$ where $t_n \in \{-1, 1\}$ for each input vector. Accordingly, test points $\mathbf{x}$ are classified by checking the sign of $y(\mathbf{x})$. For now, let us assume that it is possible to partition the training data set using a line in the feature space. Then, at least one set of the parameters $\mathbf{w}$ and b can be found such that a function in the form of (3.13) meets $y(\mathbf{x}_n) > 0$ for the points with $t_n = +1$ and $y(\mathbf{x}_n) < 0$ the for points with t_n = -1 as well, hence $t_n y(\mathbf{x}_n) > 0$ for all points in training data set. The feedforward neural network classifiers that will be introduced in Section 3.3.1 ensures to obtain a solution in a limited number of steps. However, the order in which the training data is fed to network and as well the initial values selected for $\mathbf{w}$ and b alters the solution. If there exists many solutions which are capable of classifying the training data points exactly, therefore, the solution providing the smallest generalization error is searched. Before discussing how support vector machines approach this problem, "margin" must be introduced. The margin may be described as the shortest distance between the decision boundary and any of the data points. In support vector machines, the decision boundary is selected so that the margin is maximized.

The perpendicular distance between a point $\mathbf{x}$ and a hyperplane determined by $y(\mathbf{x}) = 0$ where $y(\mathbf{x})$ has the form of (3.13) is given by $|y(\mathbf{x})| \,/\, \|\mathbf{w}\|$. Since the solutions classifying all samples correctly are searched, $t_n y(\mathbf{x}_n) > 0$ must hold for all n. Thereby, the distance between a point $\mathbf{x}_n$ and the decision surface is determined by

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n(\mathbf{w}^T \Phi(x_n) + b)))}{\|\mathbf{w}\|}. \tag{3.14}$$

As discussed above, the perpendicular distance to the closest point $\mathbf{x}_n$ among training data points determines the margin. In order to maximize this distance, the parameters $\mathbf{w}$ and b must be optimized. Hence, solving the following optimization problem gives the maximum margin solution,

$$\arg\max_{w,b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[t_n(\mathbf{w}^T \Phi(\boldsymbol{x}_n) + b) \right] \right\}. \tag{3.15}$$

This problem can be converted into an equivalent problem whose solution is easier to obtain. Scaling $\mathbf{w}$ to $\varphi\mathbf{w}$ and b to φb does not change the distance between the decision surface and any data point x_n, given by (3.14). In this case, φ is chosen to set

$$t_n(\mathbf{w}^T \Phi(\boldsymbol{x}_n) + b) = 1 \tag{3.16}$$

for the closest data point to the decision surface. Then, the following constraints are fulfilled by all data points:

$$t_n(\mathbf{w}^T \Phi(\boldsymbol{x}_n) + b) \geq 1 \qquad n=1,\ldots,N. \tag{3.17}$$

For the data points satisfying the equality, their constraints are called as “active”, whereas for other data points that do not meet the equality, corresponding constraints are “inactive”. Since there will always be a closest point, at least one active constraint is available. When the margin is maximized, at least two active constraints will exist. The optimization problem specified in (3.15) basically necessitates to maximize $\|\mathbf{w}\|^{-1}$,

equivalently, to minimize $\|\mathbf{w}\|^2$. Hence, it is transformed to

$$\arg\min_{w,b} \frac{1}{2}\|\mathbf{w}\|^2 \tag{3.18}$$

which is subjected to the constraints specified in (3.17). The Lagrangian function is obtained by assigning the Lagrange multipliers $a_n \geq 0$, with one multiplier a_n for each constraint given in (3.17) as

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left\{ t_n(\mathbf{w}^T \Phi(\boldsymbol{x}_n) + b) - 1 \right\} \tag{3.19}$$

where $\mathbf{a} = (a_1, ..., a_N)^T$. Calculating the derivative of $L(\mathbf{w}, b, \mathbf{a})$ and setting it to zero, ends up with the following conditions:

$$\begin{aligned} \mathbf{w} &= \sum_{n=1}^{N} a_n t_n \Phi(\mathbf{x}_n) \\ 0 &= \sum_{n=1}^{N} a_n t_n \end{aligned} \tag{3.20}$$

Utilizing these conditions to put away $\mathbf{w}$ and b from (3.19), the *dual representation* of the maximum margin problem can be written as:

$$\tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{3.21}$$

which is subjected to the constraints:

$$\begin{aligned} &C \geq a_n \geq 0, \quad n = 1, ..., N. \\ &\sum_{n=1}^{N} a_n t_n = 0 \end{aligned} \tag{3.22}$$

where C denotes the "box constraint" which is a parameter utilized to limit the Lagrange multiplier values. At this point, the kernel function that is specified by $k(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^T \Phi(\mathbf{x}')$ comes into the play. The dual problem permits us to recom-

pose the model making use of kernels. In this way, it becomes possible to apply the maximum margin classifier effectively to feature spaces that has higher dimensionality than the number of samples. Additionally, the kernel formulation ensures that the Lagrangian function specified in (3.21) is bounded below. By the way, the dual problem is also a quadratic programming problem that has $O(N^3)$ computational complexity.

To classify the test data points utilizing the trained models, the sign of $y(\mathbf{x})$ is evaluated. After introducing the Lagrangian multipliers $\{a_n\}$ and the kernel function, the expression given in (3.13) can be rewritten utilizing the condition specified in (3.20) as

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b. \tag{3.23}$$

The following properties must hold so that the constrained optimization problem fulfills the Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{aligned} a_n &\geq 0 \\ t_n y(\mathbf{x}_n) - 1 &\geq 0 \\ a_n \left\{ t_n y(\mathbf{x}_n) - 1 \right\} &= 0. \end{aligned} \tag{3.24}$$

To satisfy the last property given in (3.24), either $a_n = 0$ or $t_n y(\mathbf{x}_n) = 1$ must hold for each data point. Any data point whose Lagrangian multiplier, $a_n = 0$ will not change the sum given in (3.26) and thus, does not influence the predictions which is made for test data points. The other data points satisfying $t_n y(\mathbf{x}_n) = 1$ are called as *support vectors*. Those points reside on the top of maximum margin hyperplanes in features space and they only play role in making predictions. This property of support vector machines makes them much suitable for practical applications.

After solving the quadratic optimization problem to obtain a value for a, b value can be determined having noted that $t_n y(\mathbf{x}_n) = 1$ holds for any support vector, by

definition. Utilizing (3.26) gives that,

$$t_n \left(\sum_{m \in S} a_n t_n k(\mathbf{x}_n, \mathbf{x}_m) + b \right) = 1 \tag{3.25}$$

where the set of indices belonging to the support vectors is denoted by S. Solving this equation by utilizing an randomly selected support vector x_n gives b. However, multiplying both sides by t_n, taking advantage of $t_n^2 = 1$, brings a numerically more stable solution. Taking average of the equations written for each support vector and solving it for b results in

$$b = \frac{1}{N_S} \sum_{n \in S} \left(t_n - \sum_{m \in S} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right) \tag{3.26}$$

where the total number of support vectors is denoted by N_S. Also, a quadratic regularization parameter may be introduced to the maximum margin classifier error function that is intended to be minimized, so that,

$$\sum_{n=1}^{N} E_\infty \left(y(\mathbf{x}_n) t_n - 1 \right) + \lambda \left\| \mathbf{w} \right\|^2 \tag{3.27}$$

where

$$E_\infty(z) = \begin{cases} 0, & \text{if } z \geq 0 \\ \infty, & \text{otherwise} \end{cases} \tag{3.28}$$

where λ represents the regularization parameter which must be chosen such that $\lambda > 0$. The function specified in (3.28) guarantees that the constraints (3.17) are fulfilled.

Up to this point, it has been assumed that it is possible to separate the training data points in the feature space $\Phi(\mathbf{x})$. However, absolute separation of the training data points might result in generalization error in practical applications. Therefore, the support vector machine must be modified so that the misclassification of some training data points is permitted with a penalty linearly increasing along with the

distance between the point and the corresponding decision boundary. To do this, "slack variables", ξ_n are defined such that $\xi_n = 0$ for the data points within the correct side of decision boundary and $\xi_n = |t_n - y(x_n)|$ for other points where $n = 1, ..., N$. Then, the constraints specified in (3.17) are replaced with

$$t_n y(\mathbf{x}_n) \geq 1 - \xi_n \qquad n=1,...,N. \tag{3.29}$$

Also, the function to be minimized takes the form of

$$C \sum_{n=1}^{N} \xi_n + \frac{1}{2} \|\mathbf{w}\|^2 \tag{3.30}$$

where the box constraint, C, is the parameter controlling the trade-off between slack variable penalty and the margin.

The support vector machines are basically defined as two-class classifiers. To adapt them to multi-class classification tasks, multiple two-class SVM learners are combined. Various methods have been proposed to tackle this task. One frequently exploited approach is building K-many SVM models for K-class classification task. With this approach, during the training of kth model, $k = 1, ..., K$, the data from class k are accepted as the positive instances whereas the data from other $K - 1$ classes are accepted as the negative instances. Adapting this approach inherently poses an imbalance in training sets. Also, the decisions of individual classifiers may result in inconsistent results and an input may be assigned to multiple classes concurrently. Another commonly used approach is constructing $K(K - 1)/2$ different SVMs and classifying test points according to the votes of those classifiers. This approach may lead to uncertainties in the classification results as well. In this study, the approach relying on error-correcting output codes (ECOCs) proposed by Allwein *et al.* [29] has been adopted for multi-class classification. This approach generalizes the voting scheme of the one-versus-one approach. SVM classification based on ECOCs needs a coding scheme for mapping the inputs to the classes. Since the classification of four different pump conditions is targeted in this study, the coding scheme given in Table 3.1 has been

Table 3.1. The coding scheme utilized for classifying four pump conditions.

SVM learner / **Class**	1	2	3	4	5	6
Broken impeller	1	1	1	0	0	0
Cavitation	-1	0	0	1	1	0
Healthy	0	-1	0	-1	0	1
Looseness	0	0	-1	0	-1	-1

utilized. To build the learners of multi-class classifier, the ECOC algorithm applies the following steps:

(i) Learner 1 is trained using the measurements recorded for class 1 and class 2, and treats class 1 as the positive instance and class 2 as the negative instance. The other SVM learners are trained in a similar way.

(ii) Let M be the coding matrix given in Table 3.1 with elements m_{kl}, and s_l be the predicted classification score for the positive class of learner l. Then, a new observation is assigned to the class $\widehat{k}$ that produces the minimum average of the losses calculated for the L learners. This is called as loss-weighted decoding [30].

$$\widehat{k} = \arg\min_{k} \frac{\sum_{l=1}^{L} |m_{kl}|\, g(m_{kl}, s_l)}{\sum_{l=1}^{L} |m_{kl}|} \tag{3.31}$$

3.2.2. Experiments with SVM Classifiers

In this section, the training of SVM classifiers and test results are addressed. During the analyzes, 5600 measurements in total, i.e. 1400 measurements from each condition class, have been utilized. The training of SVM classifiers has been conducted on MATLAB development environment. 70% of the data have been used for training and the remaining data have been utilized for testing the trained classifier.

Average test accuracy is used for evaluating the performances of different classifiers in this study. This metric is calculated as

$$\frac{1}{C}\sum_{i=1}^{C}\frac{TP_i + TN_i}{TP_i + FN_i + FP_i + TN_i} \tag{3.32}$$

where C denotes the number of classes, TP is true positive rate which is the percentage of correct classification, FP is false positive rate which is the percentage of incorrect classification, FN is false negative rate which is the number of misses and TN is true negative rate which is the number of correct rejections.

SVM models with different hyperparameters and kernels were trained for classifying vibration signals. The kernel providing best classification accuracy were determined experimentally. The models using the second order polynomial kernel instead of linear, radial basis function and gaussian kernels performed better.

Several SVM classifiers have been trained utilizing different set of features detailed in Section 3.1. 99.90% test accuracy is reached by combining 9 features selected by minimum redundancy maximum relevance (MRMR) algorithm [31]. Related confusion matrix can be seen in Table 3.2. Test accuracies of classifiers trained utilizing different feature sets can be seen in Table 3.3. The interpretation of this table gives that the classification based on wavelet variance features reaches 99.17% test accuracy and 98.50% data reduction rate may be achieved owing to wavelet variance feature set. Also, another comment which can be made for this table is that engineered frequency

domain features are also more distinctive than shannon entropies of wavelet packet tree nodes and time domain features for vibration signals.

Table 3.4 shows that 95.24% test accuracy is achieved by utilizing only wavelet variance features extracted from vibration measurements on radial horizontal axis. It is apparent from Table 3.5 that combining wavelet variances calculated using vibration signals acquired on X and Z axes improves the accuracy of classification. The corresponding confusion matrix is given in Table 3.6. Hence, 99% data reduction rate is achieved owing to combined wavelet variance feature set.

Table 3.2. The confusion matrix for the classification with SVM model trained using the first 9 features selected via MRMR.

Actual / **Predicted**	**Healthy**	**Broken impeller**	**Cavitation**	**Looseness**
Healthy	420	0	0	0
Broken impeller	0	420	0	0
Cavitation	0	2	418	0
Looseness	0	0	0	420

Table 3.3. Test accuracies of the classifiers trained with different set of features.

Features	**Time domain**	**Shannon entropies**	**Frequency domain**	**Wavelet variances**
Test accuracy	75.95	81.90	93.80	99.17

Table 3.4. Test accuracies of the classifiers trained with wavelet variances estimated from vibration measurements on different axes.

Axis	**X**	**Y**	**Z**
Test accuracy	95.24	93.81	78.57

Table 3.5. Test accuracies of the classifiers trained with combined wavelet variances estimated from vibration measurements on different axes.

Axes	**X and Y**	**X and Z**	**Y and Z**
Test accuracy	94.76	99.40	94.52

Table 3.6. The confusion matrix for the classification with SVM model trained using wavelet variance features.

Actual / **Predicted**	**Healthy**	**Broken impeller**	**Cavitation**	**Looseness**
Healthy	420	0	0	0
Broken impeller	0	420	0	0
Cavitation	4	0	414	2
Looseness	0	0	4	416

3.3. Neural Network Classification of Vibration Signals

This section addresses the mathematical background of feedforward neural network classifiers. Network training procedure is detailed and the training algorithms exploited for this study are introduced. Finally, the results regarding the performance tests of trained classifiers are provided.

3.3.1. Overview of Feedforward Neural Network Classifiers

Artificial neural network (ANN) models have been constructed by taking the inspiration from the human brain. The brain is consisted of a large number of processing units, that is to say, neurons, operating in parallel. The neuron is the fundamental processing element of a neural network. It takes inputs that are sourced from the environment or may be the outputs of other perceptrons as depicted in Figure 3.16. The output of the summation unit of the neuron, y, may be written as a dot product

$$\mathrm{y} = \mathbf{W}\mathbf{x} + \mathrm{b} \tag{3.33}$$

where $\mathbf{W} = [\mathrm{w}_{1,1}, \mathrm{w}_{1,2}, ..., \mathrm{w}_{1,\mathrm{d}}]$ and $\mathbf{x} = [\mathrm{x}_1, ..., \mathrm{x}_\mathrm{d}]^T$ are vectors including weights and inputs, respectively. Also, b denotes the weight of the bias term which is +1. The overall output of a neuron, z may be written as

$$\mathrm{z} = f(y) = f\left(\sum_{j=1}^{d} \mathrm{w}_{1,\mathrm{j}}\mathrm{x}_\mathrm{j} + \mathrm{b}\right) \tag{3.34}$$

where $\mathrm{x}_\mathrm{j} \in \Re$, $j = 1, ..., d$ are the inputs, $\mathrm{w}_{1,\mathrm{j}} \in \Re$ are the connection weights and b is the bias weight. The function, f, is generally called as "transfer" or "activation" function. If this function is a simple thresholding function, the neuron actually defines a hyperplane dividing the input space into two half spaces. In this case, only two linearly separable classes can be distinguished by controlling the sign of the output, z,

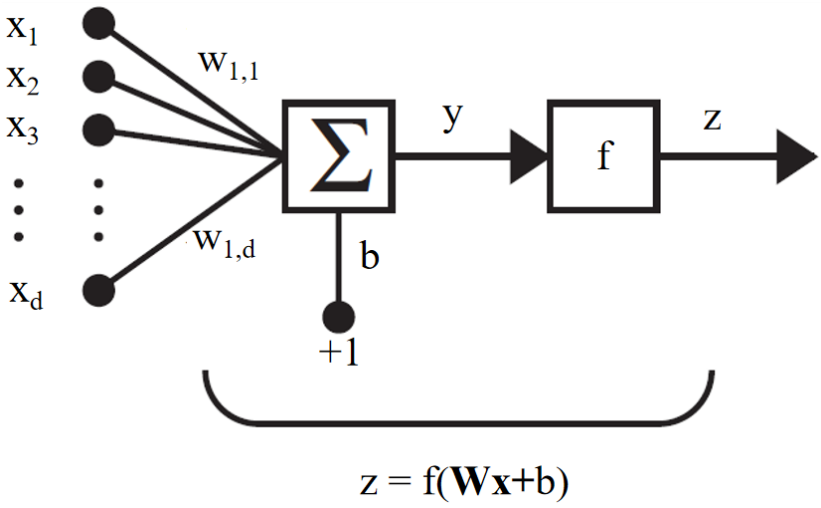

Figure 3.16. A single neuron.

as follows:

$$\begin{aligned} &\text{choose } C_1 \quad \text{if z } > \ 0 \\ &\text{choose } C_2 \quad \text{otherwise} \end{aligned} \tag{3.35}$$

where C_1 and C_2 represent the two classes.

Generally, one neuron, even with several inputs, may not be sufficient and many of them, operating in parallel, is needed. Figure 3.17 shows a "layer" of K neurons. Each neuron has a weight vector $[\mathrm{w_{i,1}}, \mathrm{w_{i,2}}, ..., \mathrm{w_{i,d}}]$ and bias vector, b_i, where $i = 1, ..., K$ denotes the corresponding neuron among the K parallel neurons. As a result, the input vector elements enter the neural network over the weight matrix,

$$\begin{bmatrix} \mathrm{w_{1,1}} & \mathrm{w_{1,2}} & \cdots & \mathrm{w_{1,d}} \\ \mathrm{w_{2,1}} & \mathrm{w_{2,2}} & \cdots & \mathrm{w_{2,d}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{w_{K,1}} & \mathrm{w_{K,2}} & \cdots & \mathrm{w_{K,d}}. \end{bmatrix} \tag{3.36}$$

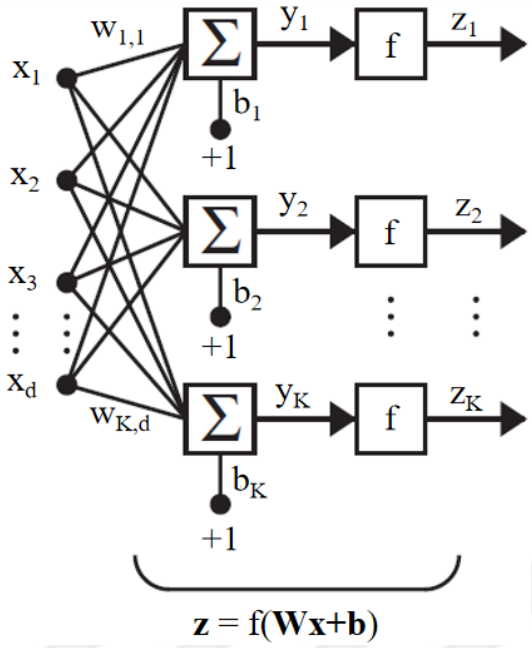

Figure 3.17. K parallel neurons.

More layers can be added after the first layer as well. Figure 3.18 depicts a three layer neural network. In this scheme, each layer has its weight matrix constructed as in (3.36). The weight matrices are denoted by $\mathbf{W^{(1)}}$, $\mathbf{W^{(2)}}$ and $\mathbf{W^{(3)}}$ for the first layer, second layer and third layer, respectively. Also, $\mathbf{b^{(1)}}$, $\mathbf{b^{(2)}}$ and $\mathbf{b^{(3)}}$ denote the corresponding bias vectors. There are s^1, s^2 and s^3 neurons in the first, second and third layer of the network given in Figure 3.18 and the network is fed with d inputs. By adding hidden layers to the network, more complicated functions of the inputs may be implemented.

In multi-layer neural networks, differentiable nonlinear transfer functions are generally exploited. The differentiability of transfer functions plays a central role in network training. For example, sigmoid is the continuous and differentiable version of thresholding and commonly used:

$$Sigmoid(x) = \frac{1}{1+e^{-x}}. \tag{3.37}$$

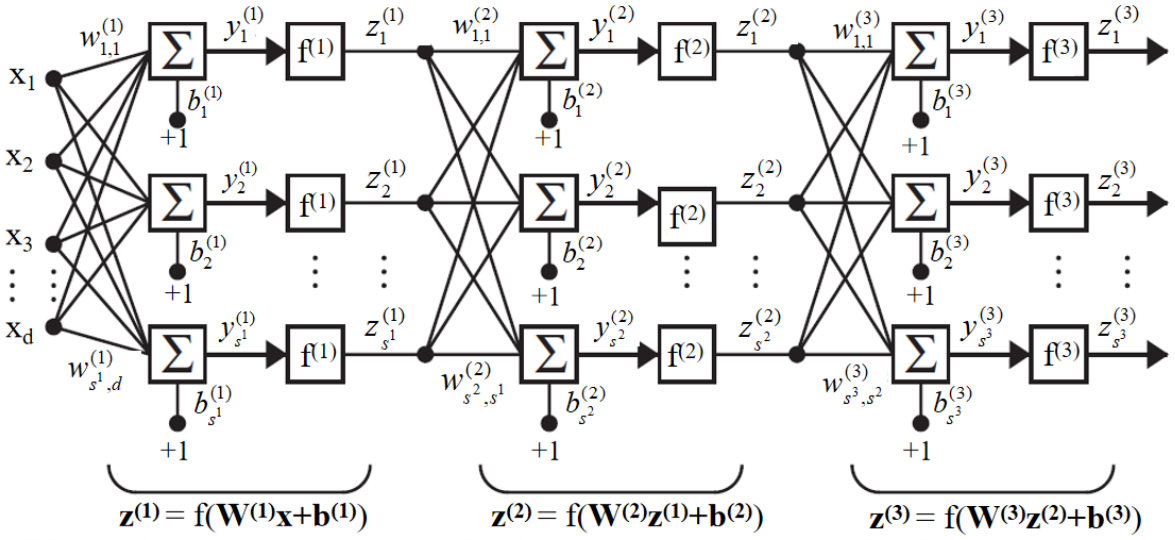

Figure 3.18. The construction of a three layer, fully connected neural network.

As noted, different layers of the network may consist of different numbers of neurons and may utilize different activation functions as well. Making use of nonlinear activation functions in hidden neurons provides the output nonlinearity. If the hidden neurons' outputs were linear, the hidden layer would be useless since a linear combination of linear combinations produces another linear combination. Then, it can be stated that the first hidden layer implements a nonlinear transformation from the d-dimensional input space to s^1-dimensional space spanned. Similarly, the second hidden layer performs a nonlinear transformation from the s^1-dimensional space to s^2-dimensional space and so on.

In brief, the input vector, $\mathbf{x}$ is fed to the input layer and the weighted sum of inputs including bias terms is calculated. Then, each hidden neuron applies the nonlinear activation function, f, to its weighted sum. The hidden neuron outputs, z_i are calculated where $i = 1, ..., s^n$ and s^n denotes the number of neurons in the corresponding hidden layer, "activation" propagates in the forward direction and reaches to the "output layer". The output neurons take the activation outputs from the last hidden layer as inputs and calculates the weighted sum of them, y_k where $k = 1, ..., K$ denotes the corresponding class among the K different classes. Sigmoid, rectified linear unit (ReLU) and softmax are commonly utilized as the output activation function. For

the neural networks that need to classify inputs into multiple categories, the softmax function, is widely applied to determine the posterior probability for each class as follows:

$$Softmax(\mathrm{y_k}) = \frac{\exp \mathrm{y_k}}{\sum_i \exp \mathrm{y_i}}. \tag{3.38}$$

The neural networks can be thought as a general class of parametric nonlinear functions from an input vector, $\mathbf{x}$ to an output vector, $\mathbf{y}$. The overall network function of a two layer network takes the form of

$$\mathrm{y_k}(\mathbf{x}, \mathbf{w}) = \gamma \left(\sum_{i=0}^{s^1} \mathrm{w}_{\mathrm{k,i}}^{(2)} f \left(\sum_{j=0}^{d} \mathrm{w}_{\mathrm{i,j}}^{(1)} \mathrm{x_j} \right) \right) \tag{3.39}$$

where γ denotes the activation function at the output layer. The evaluation process of (3.39) is called as "forward propagation" of information throughout the network.

3.3.2. Evaluating the Error Gradients and Backpropagation

Given a training set consisting of N independent, identically distributed observations, $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ and corresponding target vectors $\mathbf{t} = \{\mathbf{t}_1, ..., \mathbf{t}_N\}$, the corresponding likelihood function,

$$p\left(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta\right) = \prod_{n=1}^{N} p\left(\mathbf{t}_n \mid \mathbf{x}_n, \mathbf{w}, \beta\right) \tag{3.40}$$

where β is the inverse variance of Gaussian noise in observations. In the literature of neural networks, minimizing the error function is considered instead of maximizing the likelihood function given in (3.40). The maximization of the this function is same with the minimization of the error function given by,

$$E(\mathbf{w}) = \sum_{n=1}^{N} \left\{ \mathrm{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n \right\}^2. \tag{3.41}$$

In the case of multi-class classification problem where each observation is assigned to single class among K mutually exclusive classes indicated by 1-of-K coding scheme consisting of the binary target variables $t_k \in \{0, 1\}$, the error function may be rewritten as

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} \mathrm{t_{kn}} \ln \mathrm{y_k}(\mathbf{x}_n, \mathbf{w}). \tag{3.42}$$

The main objective of network training is finding a weight vector, $\mathbf{w}$ such that the error function, $E(\mathbf{w})$ given in (3.41) is minimized. Taking a small step from $\mathbf{w}$ to $\mathbf{w} + \delta\mathbf{w}$ in the weight space changes the error function by $\delta E \simeq \delta\mathbf{w}^T \nabla E(\mathbf{w})$, where the vector, $\nabla E(\mathbf{w})$ indicates the direction in which the error function increases most. $E(\mathbf{w})$ gets minimized at the points where the gradient of the error function becomes zero, so that $\nabla E(\mathbf{w}) = 0$. Those points where the gradient vanishes are said to be stationary points. There will be many points where the gradient of the error is numerically very small or zero. For a successful neural network application, reaching the global minimum may not be required, but several local minima must be certainly compared.

Various network training algorithms have been developed and proposed over the years. All of those algorithms require the evaluation of error gradients. The procedure followed to evaluate the derivatives of the error function with respect to the network parameters is known as "error backpropagation" which is illustrated in Figure 3.19. To constitute the base of further discussion, it must be remembered that the output of each unit in the network gives a weighted sum of its inputs. Thereby, the unit outputs may be rewritten in the form

$$\mathrm{a_j} = \sum_i \mathrm{w_{ji} z_i} \tag{3.43}$$

where $\mathrm{z_i}$ denotes a unit activation or an input that has the connection with unit j, and $\mathrm{w_{ji}}$ represents the weight corresponding to that connection. A nonlinear activation

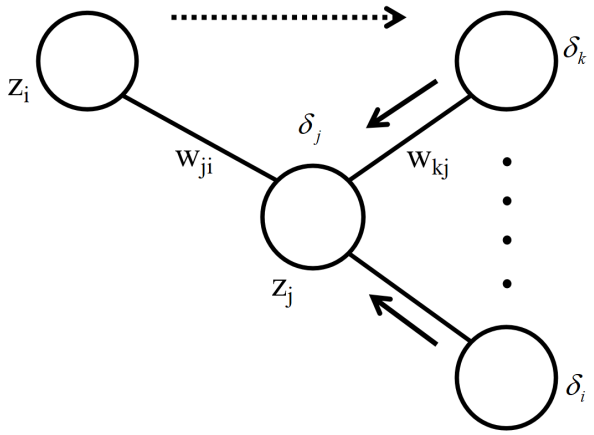

Figure 3.19. Illustration of the forward and backpropagation along the network, in the direction of dashed and solid lines, respectively.

function, f is applied to the sum in (3.43) to obtain the activation z_j of unit j, so that,

$$z_j = f(a_j). \tag{3.44}$$

Most of the training algorithms involve an iterative procedure to minimize the error function given in (3.41). It may also be rewritten as the sum of errors for each training data, thereby,

$$E(\mathbf{w}) = \sum_{n=1}^{N} E_n(\mathbf{w}). \tag{3.45}$$

The chain rule may be applied to evaluate the derivative of E_n with respect to a weight w_{ji} to obtain

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j}\frac{\partial a_j}{\partial w_{ji}}. \tag{3.46}$$

Introducing the notation for the errors, $\delta_i = \dfrac{\partial E_n}{\partial a_j}$ and using $\dfrac{\partial E_n}{\partial w_{ji}} = z_i$ from (3.43), we may write

$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i. \tag{3.47}$$

Hence, after calculating the value of δ_j for each network unit, (3.47) is applied in order to evaluate the derivatives. The error for the output units is $\delta_k = \mathrm{y_k} - \mathrm{t_k}$ and to calculate the errors for hidden units, the chain rule may be applied again as follows:

$$\delta_j \equiv \frac{\partial E_n}{\partial \mathrm{a_j}} = \sum_k \frac{\partial E_n}{\partial \mathrm{a_k}} \frac{\partial \mathrm{a_k}}{\partial \mathrm{a_j}}. \tag{3.48}$$

Then, by making use of (3.43) and (3.44), the errors for hidden units can be computed with the following formula

$$\delta_j = f'(\mathrm{a_j}) \sum_k \mathrm{w_{kj}} \delta_k. \tag{3.49}$$

The "error backpropagation" procedure can be summed up as follows:

(i) An input vector from the training set, $\mathbf{x_n}$ is fed into the network and propagate in forward direction throughout the network using (3.39).

(ii) Error terms, $\delta_k = \mathrm{y_k} - \mathrm{t_k}$ are evaluated for each output unit.

(iii) Error at output units is backpropagated to obtain the errors in hidden units making use of (3.49).

(iv) Lastly, the required derivatives are found using (3.47).

(v) For batch methods, the first four steps are repeated for each sample in the training set. Then, averaging the individual errors gives the derivative of the total error:

$$\frac{\partial E}{\partial \mathrm{w_{ji}}} = \sum_n \frac{\partial E_n}{\partial \mathrm{w_{ji}}}. \tag{3.50}$$

3.3.3. The Gradient Descent Algorithm and Its Variants Utilized

The derivatives obtained with the backpropagation procedure are utilized to adjust the weights of the network. There are several types of optimization algorithms to be exploited for the second stage. The simplest approach among those optimization schemes is the "gradient descent". With this approach, the weights are updated in the

direction of $-\nabla E(\mathbf{w})$ since a local minimum of the error function is searched, so that,

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \eta \nabla E(\mathbf{w}^{\tau}) \tag{3.51}$$

where η is called as the learning rate parameter which adjusts the step size. After updating the weights, the gradient is evaluated again to determine the direction of next weight update and this procedure is repeated until it converges. Each epoch necessitates the evaluation of ∇E through the entire training set. Computing the gradient over a "batch" of training set as in (3.52) instead of the gradient for a single sample may perform significantly better

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \frac{\eta}{n} \sum_{i=1}^{n} \nabla E_i(\mathbf{w}^{\tau}) \tag{3.52}$$

where n denotes the batch size, that is, the number of samples in the training set. This scheme also provides a smoother convergence since the gradient is averaged over more training samples at each step. Many extensions and variants of the basic stochastic gradient descent algorithm have been proposed and exploited over the years. The first variant utilized in this study is the stochastic gradient descent with momentum [32] in which the next update is calculated as a linear combination of the gradient and previous update in the following way:

$$\begin{aligned} \Delta\mathbf{w}^{\tau} &= \alpha \Delta \mathbf{w}^{\tau-\mathbf{1}} - \eta \nabla E(\mathbf{w}^{\tau}) \\ \mathbf{w}^{\tau+1} &= \mathbf{w}^{\tau} + \Delta \mathbf{w}^{\tau} \\ \mathbf{w}^{\tau+1} &= \mathbf{w}^{\tau} - \eta \nabla E(\mathbf{w}^{\tau}) + \alpha \Delta \mathbf{w}^{\tau-\mathbf{1}} \end{aligned} \tag{3.53}$$

where α denotes the momentum parameter which takes a value between 0 and 1. The training algorithm is accelerated in the relevant direction and the oscillations are prevented thanks to this parameter.

Another training algorithm exploited in this study is the Levenberg-Marquardt algorithm [33]. It fits perfectly for the optimization tasks in which the minimization

of the nonlinear functions including the sums of squares is aimed. Therefore, the Levenberg-Marquardt algorithm is typically used along with mean squared error (MSE) loss function. Computing the Jacobian matrix including the partial derivatives of errors at each unit with respect to the biases and weights constitutes the key step for this algorithm. The Jacobian matrix, $\mathbf{J}$, for multilayer network training can be written in the form of

$$\begin{bmatrix} \frac{\partial e_{1,1}}{\partial \mathrm{w}_{1,1}^{(1)}} & \frac{\partial e_{1,1}}{\partial \mathrm{w}_{1,2}^{(1)}} & \cdots & \frac{\partial e_{1,1}}{\partial \mathrm{w}_{\mathrm{s}^1,\mathrm{d}}^{(1)}} & \frac{\partial e_{1,1}}{\partial \mathrm{b}_1^{(1)}} & \cdots \\ \frac{\partial e_{2,1}}{\partial \mathrm{w}_{1,1}^{(1)}} & \frac{\partial e_{2,1}}{\partial \mathrm{w}_{1,2}^{(1)}} & \cdots & \frac{\partial e_{2,1}}{\partial \mathrm{w}_{\mathrm{s}^1,\mathrm{d}}^{(1)}} & \frac{\partial e_{2,1}}{\partial \mathrm{b}_1^{(1)}} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \cdots \\ \frac{\partial e_{s^M,1}}{\partial \mathrm{w}_{1,1}^{(1)}} & \frac{\partial e_{s^M,1}}{\partial \mathrm{w}_{1,2}^{(1)}} & \cdots & \frac{\partial e_{s^M,1}}{\partial \mathrm{w}_{\mathrm{s}^1,\mathrm{d}}^{(1)}} & \frac{\partial e_{s^M,1}}{\partial \mathrm{b}_1^{(1)}} & \cdots \\ \frac{\partial e_{1,2}}{\partial \mathrm{w}_{1,1}^{(1)}} & \frac{\partial e_{1,2}}{\partial \mathrm{w}_{1,2}^{(1)}} & \cdots & \frac{\partial e_{1,2}}{\partial \mathrm{w}_{\mathrm{s}^1,\mathrm{d}}^{(1)}} & \frac{\partial e_{1,2}}{\partial \mathrm{b}_1^{(1)}} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{3.54}$$

where e terms denote the errors at each hidden unit. In order to calculate J, a modified version of the backpropagation algorithm is generally exploited. After that, the next step is to compute Hessian matrix, $\mathbf{H}$, which can be approximated as:

$$\mathbf{H} = \mathbf{J}^{\mathbf{T}}\mathbf{J} \tag{3.55}$$

when the error function has the form of a sum of squares, that is typical in case of training neural networks. Then, the gradient of error function can be computed using

$$\nabla E(\mathbf{w}) = \mathbf{J}^{\mathbf{T}}\mathbf{e} \tag{3.56}$$

where $\mathbf{e}$ represents the vector consisting of network errors. Then, the weight update rule of the Levenberg-Marquardt optimization algorithm may be written as follows:

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \left[\mathbf{J}^{\mathbf{T}}\mathbf{J} + \mu\mathbf{I}\right]^{-1}\mathbf{J}^{\mathbf{T}}\mathbf{e} \tag{3.57}$$

where μ is called as the Levenberg's damping factor and adjusted at each epoch in such a way that if the error decreases rapidly, then μ is divided by a constant, whereas the error decreases slowly, it is multiplied by the same constant.

3.3.4. Experiments with Feedforward Neural Network Classifiers

In this section, the training process of feedforward neural network classifiers and test results are addressed. During the analyzes, the data set exploited to train SVM classifiers has been used. Accordingly, 5600 measurements in total, i.e. 1400 measurements from each condition class, have been analyzed. Various feedforward neural network classifiers with different parameters and configurations have been trained. These trainings have been conducted on MATLAB development environment and Python development environment as well. The Levenberg-Marquardt and Bayesian regularization backpropagation training algorithms have been exploited on MATLAB environment whereas the training on the Python environment has been conducted using SGD algorithm. Similar to the experiments with SVM classifiers, average test accuracy metric specified in (3.32) has been adopted to evaluate the performance of the classifiers.

By taking the results given in previous sections into consideration, it has been decided to use wavelet variances firstly as features. The input size of the networks equals the number of features utilized, which is eighteen for the case of utilizing wavelet variances. On the other hand, the number of the outputs is four since four different conditions of the pump have been investigated. During the training, various early stopping strategies have been adopted in order to prevent over-fitting. Validation data sets have been used to evaluate the networks while tuning the hyperparameters. The training of classifiers has been terminated after six consecutive epochs passes without improving validation error.

Hyperbolic tangent was utilized as activation function in hidden layers, whereas softmax and linear activation functions were utilized in output layer. Mean squared error and cross entropy (CE) loss functions were exploited to calculate the losses during the study. The data set were divided into training, validation and test set at the rates

Table 3.7. The performance of different FFNN classifiers exploiting wavelet variances and trained with Levenberg-Marquardt algorithm and linear activation function at output layer.

Number of hidden neurons	**Number of epochs**	**Validation MSE**	**Test MSE**	**Test accuracy**
12	190	2.5291×10^{-4}	3.5628×10^{-4}	99.9%
10	363	3.9612×10^{-5}	5.9231×10^{-4}	99.9%
8	67	0.0139	0.0129	99.4%
5	36	0.0054	0.0054	99.8%
3	24	0.0067	0.0074	98.8%
2	40	0.0645	0.0655	74.4%

of $70\%, 15\%, 15\%$, respectively. The networks having only single layer and relatively fewer parameters have been constructed and tested to prevent the network memorizing the training set.

Various single layer feedforward neural networks were trained using different training algorithms such as the Levenberg-Marquardt and Bayesian regularization backpropagation [34] on MATLAB development environment. The former detailed in Section 3.3.3 is one of the fastest backpropagation algorithm and preferred as a first-choice supervised algorithm, although it requires more memory than the most of other algorithms. Whereas, the latter one offers good generalization qualities by minimizing a linear combination of weights and squared errors. Trained neural networks whose performances are given in Table 3.7 and 3.8 uses hyperpolic tangent and linear activation functions in their hidden layer and output layer, respectively. The training of networks was terminated if the validation performance has improved more than six epochs since the last time it declined. The results given in Table 3.9 and 3.10 show that the shallow networks with a couple of hidden neurons are able classify the pump conditions accurately.

Table 3.8. The performance of different FFNN classifiers exploiting wavelet variances and trained with Bayesian regularization backpropagation algorithm and the linear activation function at output layer.

Number of hidden neurons	**Number of epochs**	**Validation MSE**	**Test MSE**	**Test accuracy**
12	92	0.0015	0.0021	99.9%
10	90	0.0011	0.0015	100.0%
8	71	0.0188	0.0196	100.0%
5	240	0.0042	0.0038	99.9%
3	42	0.0102	0.0068	99.0%
2	49	0.0640	0.0643	74.5%

Table 3.9. The performance of different FFNN classifiers exploiting wavelet variances and trained with Levenberg-Marquardt algorithm and softmax activation function at output layer.

Number of hidden neurons	**Number of epochs**	**Validation MSE**	**Test MSE**	**Test accuracy**
12	2207	0.0053	0.0057	100.0%
10	3412	0.0016	0.0017	100.0%
8	480	0.0026	0.0014	100.0%
5	42	0.0018	4.5687×10^{-4}	99.9%
3	33	0.0034	0.0023	99.6%
2	61	5.9529×10^{-5}	7.1291×10^{-5}	100%

A series of network training sessions has been conducted on Python development environment as well, using the Stochastic Gradient Descent (SGD) optimizer. In order to determine the changes that will be applied to the weights and biases with this algorithm, the learning rate, η is multiplied by the negative of the gradient as in (3.51).

Table 3.10. The performance of different FFNN classifiers exploiting wavelet variances and trained with Bayesian regularization backpropagation algorithm and softmax activation function at output layer.

Number of hidden neurons	**Number of epochs**	**Validation MSE**	**Test MSE**	**Test accuracy**
12	3582	7.2071×10^{-4}	5.6212×10^{-4}	99.9%
10	2754	0.0013	0.0015	99.9%
8	79	0.0011	4.3057×10^{-4}	100.0%
5	41	0.0018	5.8995×10^{-4}	99.9%
3	32	0.0012	3.4564×10^{-5}	100.0%
2	42	5.9537×10^{-4}	4.9063×10^{-4}	99.9%

Larger learning rates result in taking bigger steps over the error function. If the learning rate is selected too small, the training algorithm may take a long time to converge whereas the learning rate is chosen too large, the algorithm may become unstable. Also, SGD with momentum was used during the training. The results obtained in this section show that the shallow networks having up to five hidden neurons are able classify the pump conditions accurately and exploiting the softmax at output activation function offers better performance. Accordingly, single layer feedforward neural networks whose performances are given in Table 3.11, Table 3.12 and Table 3.13 have been trained by utilizing different optimizer parameters. Table 3.11 and Table 3.13 demonstrate the results obtained for the training sessions which were terminated after six epochs passed consecutively without an improvement in validation error. CE loss function was utilized when training the networks in Table 3.11 while MSE loss function was used to train the networks in Table 3.13. On the other hand, the training sessions whose results are shown in Table 3.12 were terminated after ten epochs passed consecutively without improving the validation CE error. Basically, mean squared error loss function is generally preferred for regression tasks while cross entropy loss function is commonly used for the tasks in which the final output is a probability distribution. Therefore, it is expected that training the neural networks which have softmax activated outputs by

Table 3.11. The training results for FFNN classifiers which exploit wavelet variances, have five hidden neurons, softmax activated output layer and are trained using CE loss function.

Learning rate	**Momentum parameter**	**Number of epochs**	**Validation CE error**	**Test CE error**	**Test accuracy**
0.001	-	22904	0.0562	0.0589	100.0%
0.001	0.5	23085	0.0696	0.0713	100.0%
0.001	0.9	21134	0.0726	0.0751	99.88%
0.01	-	21255	0.0697	0.0704	100.0%
0.01	0.5	20796	0.0747	0.0784	99.88%
0.01	0.9	19915	0.0850	0.0881	99.88%

Table 3.12. The training results for FFNNs which are trained in the same way with the FFNNs in Table 3.11 excluding early stopping criterion.

Learning rate	**Momentum parameter**	**Number of epochs**	**Validation CE error**	**Test CE error**	**Test accuracy**
0.001	-	36149	0.0054	0.0077	99.88%
0.001	0.5	26510	0.0416	0.0436	99.88%
0.001	0.9	34862	0.0085	0.0090	100.0%
0.01	-	23238	0.0549	0.0559	100.0%
0.01	0.5	37622	0.0042	0.0126	99.88%
0.01	0.9	35316	0.0060	0.0145	99.88%

using cross entropy loss functions offers better results. However, most of the networks in Table 3.12 and Table 3.13 performs similarly in terms of validation, test error and accuracy. Also, training the networks which have similar performances by using CE loss function instead of MSE loss function takes longer time.

Table 3.13. The training results for FFNN classifiers exploiting wavelet variances and having five hidden neurons and softmax activated output layer.

Learning rate	**Momentum parameter**	**Number of epochs**	**Validation MSE**	**Test MSE**	**Test accuracy**
0.001	-	22356	0.0059	0.0060	100.0%
0.001	0.5	22436	0.0055	0.0059	100.0%
0.001	0.9	21175	0.0078	0.0082	100.0%
0.01	-	20224	0.0085	0.0088	100.0%
0.01	0.5	19023	0.0098	0.0105	99.88%
0.01	0.9	22229	0.0062	0.0065	99.88%

Table 3.14. The confusion matrix belonging to test with the classifier whose hyperparameters shown in the last row of Table 3.13.

Actual / **Predicted**	**Healthy**	**Broken impeller**	**Cavitation**	**Looseness**
Healthy	210	0	0	0
Broken impeller	0	210	0	0
Cavitation	0	1	209	0
Looseness	0	0	0	210

Finally, networks exploiting time domain statistics such as skewness, kurtosis, variance and rms have been trained using the Levenberg-Marquardt algorithm. Accordingly, the input size of networks is twelve since these features have been extracted from the vibration measurements on each axis. Table 3.15 demonstrates the performances of networks having different activation functions in their hidden layers.

Table 3.15. The performances of networks exploiting time domain statistics and trained with Levenberg-Marquardt algorithm.

Hidden neurons	**Number of epochs**	**Validation MSE**	**Test MSE**	**Test accuracy**	**Output activation**
5	49	6.29×10^{-4}	0.0017	99.6%	Linear
8	26	3.74×10^{-4}	0.0011	99.9%	Linear
10	80	2.82×10^{-9}	5.95×10^{-4}	100.0%	Linear
5	25	$2.42\times\ 10^{-9}$	$4.22\times\ 10^{-7}$	100.0%	Softmax
8	169	3.12×10^{-8}	2.41×10^{-4}	100.0%	Softmax
10	27	1.93×10^{-8}	1.16×10^{-8}	100.0%	Softmax

3.4. Summary of the Chapter and Concluding Remarks

In this chapter, firstly, the methods applied to extract vibration signal features have been detailed and the manual selection of best performing features have been completed. Then, several classifiers utilizing those features have been trained. The results of the analyzes conducted show that both SVM and FFNN classifiers utilizing wavelet variance features as inputs perform well for classifying the conditions experienced by the flexible impeller pump. However, it could be concluded that FFNN classifiers perform slightly better than SVM classifiers. Moreover, the performance of trained FFNN classifiers exploiting time domain features is acceptable.

The next task is to implement the proposed digital signal processing chain in the embedded system. To accomplish this task, two different software applications have been developed. The first one utilizes the FFNN as classifier and wavelet variances as features. On the other hand, FFNN classifier and time domain features which are kurtosis, skewness, rms and variance are adopted for the second application. The next chapter presents the details of those embedded system implementations.

4. EMBEDDED SYSTEM REALIZATION OF THE PROPOSED METHOD

This chapter summarizes the works that have been completed to realize the proposed digital signal processing chain on top of the embedded system. It is possible to realize SVM classifiers and feedforward neural network classifiers on the MCUs based on specific ARM cores using the Cortex Microcontroller Software Interface Standard (CMSIS) libraries. The results discussed in the previous sections show that feedforward neural network classifier trained exploiting wavelet variances or time domain features performs quite well for classifying different pump conditions. Then, they have been adopted and realized in the embedded system for online pump CM. STM32F746G-DISCO board including a digital signal processor (DSP) with floating point unit and ARM Cortex M7 core has been preferred for this task. Table 4.1 demonstrates the connections between the EVAL-ADXL355 and STM32F746G-DISCO boards. The clocks have been set in such a way that the MCU core runs at a frequency of 200 MHz and the communication speed for serial peripheral interface (SPI) is 1.5 MBits/s.

A demonstration application whose states can be viewed in Figure 4.1 has been implemented in the STM32F746G-DISCO board. The demo application consists of two main tasks. The first one manages the user interface operations and the second one does the digital signal processing operations. The application is initially in the idle mode and waits for measurement trigger. Once it takes the trigger for reading acceleration measurements, it reads the current acceleration values at x, y and z axes from the accelerometer during 500 ms. Since the output data rate of the accelerometer is set to 4 kHz, 2000 acceleration measurements digitized with 20-bit resolution for each axis are read and placed to the random access memory (RAM).

Table 4.1. Cable connections between EVAL-ADXL355 and STM32F746G-DISCO boards.

STM32F746G-DISCO board connector and pin	**EVAL-ADXL355 connector and pin**	**Pin function**
CN6-IOREF	P1-V_{DDIO}	Digital side voltage supply
CN6-3V3	P1-VDD	Voltage supply
CN6-GND	P1-GND	Ground
CN7-D12	P2-MISO	Master in slave out
CN7-D11	P2-MOSI	Master out slave in
CN7-D13	P2-SCLK	Clock pin
CN7-D9	P2-CS	Chip select pin
CN7-D10	P1-DRDY	Data ready interrupt pin

The raw acceleration measurements are converted to g values utilizing the following equation:

$$\text{Acceleration in } g = (\text{Raw measurement}) * (8.192) \ / \ (2^{19}\text{-}1). \tag{4.1}$$

After obtaining acceleration values, the MODWT is applied to measurements and wavelet variances of transformed series are calculated in one of the applications. Time domain statistics, skewness, rms, kurtosis and variance, are computed and utilized as features in the other application developed for online CM. Then, the extracted features are fed into the feedforward neural network to determine the current pump condition.

The networks deployed to the MCU consist of a single hidden layer and eight neurons. They exploit sigmoid and softmax activation functions in their hidden layers and output layers, respectively. To deploy the trained networks to the MCU, they have been exported in $h5$ format and the required software libraries and converted network model have been generated using a software package called X-CUBE-AI which is a product of ST Microelectronics. Moreover, the software libraries used to apply

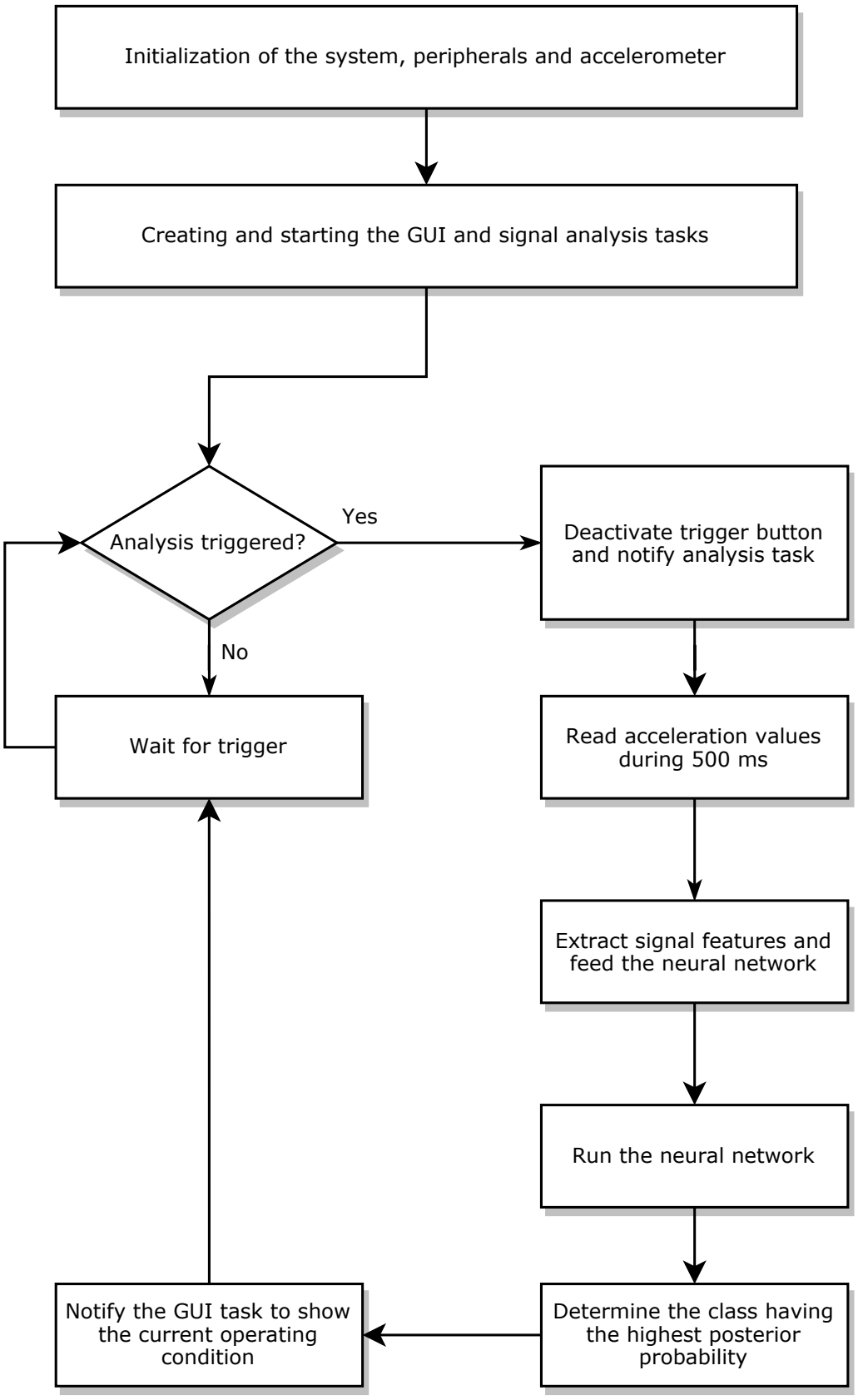

Figure 4.1. The flowchart of developed embedded system application.

Table 4.2. The size of code and data in bytes for the application developed.

Code size	**Read-only data**	**Read-write data**	**Zero initialized data**
101340	455772	435900	288636

MODWT to real time measurements have been generated using MATLAB Coder. TOUCHGFX framework has been utilized for the design and development of the user interface. The application handling the required operations has been built on top of FreeRTOS real time operating system. The memory used by the application is shown in Table 4.2. Source codes developed for the embedded application have been pushed into a private repository as well.

The final outputs of the digital signal processing chain are the posterior probabilities calculated as in (3.38) for each class. In order to test the performance of the embedded implementation, several laboratory experiments have been conducted. During the tests of online CM system, the motor has been operated at different speeds between 250 rpm and 800 rpm. The pump condition analysis has been triggered after the pump reached to its target speed.

Table 4.3 and Table 4.4 show the test performances of the analysis system using wavelet variances and time domain statistics, respectively. The online CM system utilizing wavelet variances have predicted the pump condition incorrectly only once. In that case, a measurement collected under cavitation condition have been associated with impeller fault. On the other hand, the real time CM system exploiting time domain features have predicted the pump condition correctly with an accuracy of 97.67%. Six measurements associated with mechanical looseness and one measurement associated with cavitation condition have been classified incorrectly. At this point, it turns out that there is a difference between the offline and online test performances of feed forward neural networks exploiting time domain statistics. The reason may be that during the experiments, it is needed to take actions such as loosing and tightening the

screws, disassembling and reassembling the pump. Those actions change the vibration profile generated by the pumping system which directly affects the time domain statistics. The online CM systems whose performances are presented take their decisions according to the analysis of single measurement window. Instead, applying majority voting would boost the performance of the online CM systems, although the duration needed to take decision gets longer.

Table 4.3. Test performance of the online CM system utilizing wavelet variance features.

Actual / **Predicted**	**Healthy**	**Broken impeller**	**Cavitation**	**Looseness**
Healthy	75	0	0	0
Broken impeller	0	75	1	0
Cavitation	0	0	74	0
Looseness	0	0	0	75

Table 4.4. Test performance of the online CM system utilizing time domain statistics as features.

Actual / **Predicted**	**Healthy**	**Broken impeller**	**Cavitation**	**Looseness**
Healthy	75	0	0	2
Broken impeller	0	75	0	0
Cavitation	0	0	74	4
Looseness	0	0	1	69

5. CONCLUSION AND FUTURE WORK

The feedforward neural network and support vector machine classifiers trained by utilizing vibration signal features are able to detect the pump state with high accuracy. It is presented that more than 99.0% test accuracy can be achieved with the NN model utilizing only wavelet variances extracted from the vibration signals. Also, the classifier utilizing time domain statistics performs well. The results of this study show that the vibration sensors perform quite well for sensing the flexible impeller pump conditions. On the other hand, the motor current signals collected under different pump conditions have not become distinctive since the motor driver tolerates the changes in the load mostly. Yet, the motor current measurements obtained through the Hall type clamp type current sensor enable the estimation of motor speed and it makes possible to extract frequency domain features which may be distinctive for certain faulty conditions.

Developed digital signal processing chain has been implemented on top of a memory constrained embedded hardware. In the future, sensor nodes capable of detecting operating conditions of a flexible impeller pump could be designed. Yet, another MCU consuming less power than the one utilized during the development can be used in the product to be deployed to the field since the current trends in IIoT applications lean toward low power devices. Up to now, the classifier has been trained offline and then, transferred to the MCU using debug adapters. When the wireless communication and over the air update capabilities are integrated into the nodes, it is possible to update the classifier running on the nodes. Also, the device may send only the requested features to an external device which results in reducing the amount of data which must be transferred and saving the network bandwidth.

During the late tests of real time CM system, swollen impellers obtained from the dyehouses have been used as well. Impeller swelling mostly occurs due to the interaction between impeller material and incompatible chemicals to be transferred. Without defining this condition to the fault detection and diagnosis system, some tests have been conducted. The operation with swollen impellers has not been classified as normal condition by the online CM system. This shows that the trained neural network model is able to detect the abnormal operating conditions as well as normal operating conditions.

A massive effort for feature engineering have been put during this study. However, in a real scenario, there will be many types of liquids having different characteristics such as viscosity and density to be transferred. Accordingly, the vibration generated by the flexible impeller pump and liquid interaction might be different. In this case, previously determined features may not perform well in terms of the separation ability of different pump conditions. If such challenges are taken into consideration, the methods with automated feature extraction capabilities could be adopted. Recurrent neural network (RNN) based auto-encoders works very well for calculating features from raw vibration data. These features can be exploited as features to train a supervised classifier. Moreover, long short-term memory networks which have RNN architecture could be directly utilized to classify the raw vibration measurements. Lastly, a decision system which classifies the pump condition only as “normal” and “abnormal” might be more practically applicable.

REFERENCES

1. PwC and Mainnovation, "Beyond the Hype: Predictive Maintenance 4.0 Delivers Results", September 2018, `https://www.pwc.be/en/documents/20180926-pdm40-beyond-the-hype-report.pdf`, accessed in July 2020.

2. Henríquez, P., J. Alonso, M. Ferrer and C. Travieso, "Review of Automatic Fault Diagnosis Systems Using Audio and Vibration Signals", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, pp. 642-652, May 2014.

3. Albraik, A, F. Althobiani, F. Gu and A. Ball, "Diagnosis of Centrifugal Pump Faults Using Vibration Methods", *Journal of Physics: Conference Series*, vol. 364, IOP Publishing, May 2012.

4. Henríquez, P., J. Alonso, M. Ferrer, C. Travieso and G. Gómez, "Fault Diagnosis Using Audio and Vibration Signals in a Circulating Pump", *Journal of Physics: Conference Series*, vol. 364, IOP Publishing, May 2012.

5. Mohanty, A., P. Pradhan, N. Mahalik and S. Dastidar, "Fault Detection in a Centrifugal Pump Using Vibration and Motor Current Signature Analysis", *International Journal of Automation and Control*, vol. 6, no. 3/4, pp. 261-276, 2012.

6. Jensen, J. and K. Dayton, "Detecting Cavitation in Centrifugal Pumps", *Orbit*, pp. 26-30, 2000.

7. Sakthivel N., V. Sugumaran and S. Babudevasenapati, "Vibration Based Fault Diagnosis of Monoblock Centrifugal Pump Using Decision Tree", *Expert Systems with Applications*, vol. 37, no. 6, pp. 4040-4049, 2010.

8. Scheffer, C. and P. Girdhar, "Practical Machinery Vibration Analysis and Predictive Maintenance", pp. 128-131, Elsevier Press, 2004.

9. Birajdar, R. S., R. S. Patil and K. Khanzode, "Vibration and Noise in Centrifugal Pumps - Sources and Diagnosis Methods", *3rd International Conference on Integrity, Reliability and Failure*, Porto/Portugal, 20-24 July 2009.

10. Tandon, N. and A. Choudhury, "A Review of Vibration and Acoustic Measurement Methods for the Detection of Defects in Rolling Element Bearings", *Tribology International*, vol. 32, no. 8, pp. 469-480, 1999.

11. Pan, Y., J. Chen and L. Guo, "Robust Bearing Performance Degradation Assessment Method Based on Improved Wavelet Packet-Support Vector Data Description", *Mechanical Systems and Signal Processing*, vol. 23, no. 3, pp. 669-681, 2009.

12. Farokhzad, S., H. Ahmadi, A. Jaefari, M. Abad and M. Kohan, "Artificial Neural Network Based Classification of Faults in Centrifugal Water Pump", *Journal of Vibroengineering*, vol. 14, pp. 1734-1744, December 2012.

13. Saravanan, N. and K. Ramachandran, "Incipient Gear Box Fault Diagnosis Using Discrete Wavelet Transform (DWT) for Feature Extraction and Classification Using Artificial Neural Network (ANN)", *Expert Systems with Applications*, vol. 37, no. 6, pp. 4168-4181, 2010.

14. Rui, Z., Y. Ruqiang, C. Zhenghua, M. Kezhi, W. Peng and X.G. Robert, "Deep Learning and Its Applications to Machine Health Monitoring", *Mechanical Systems and Signal Processing*, vol. 115, pp. 213-237, 2019.

15. Zhang, S., S. Zhang, B. Wang and T.G. Habetler, "Deep Learning Algorithms for Bearing Fault Diagnostics—A Comprehensive Review", *IEEE Access*, vol. 8, pp. 29857-29881, February 2020.

16. Elsaid, A., B. Wild, J. Higgins and T. Desell, "Using LSTM Recurrent Neural Networks to Predict Excess Vibration Events in Aircraft Engines", *IEEE 12th International Conference on e-Science*, pp. 260-269, October 2016.

17. Fengtao, W., L. Xiaofei, D. Gang, Y. Xiaoguang, L. Hongkun and H. Qingkai, "Remaining Life Prediction Method for Rolling Bearing Based on the Long Short-Term Memory Network", *Neural Processing Letters*, vol. 50, no. 3, pp. 2437-2454, December 2019.

18. Xiaohan, C., Z. Beike and G. Dong, "Bearing Fault Diagnosis Base on Multi-scale CNN and LSTM Model", *Journal of Intelligent Manufacturing*, June 2020.

19. Brennen, C., "Cavitation and Bubble Dynamics", *Expert Systems with Applications*, p. 21, Oxford University Press, 1995.

20. Girdhar, P., O. Moniz and S. Mackay, "Practical Centrifugal Pumps", pp. 70-72, Elsevier Press, 2005.

21. Eliesi S., "Performance Analysis of Commercial Accelerometers: A Parameter Review", *Sensors and Transducers*, vol. 193, no. 10, pp. 179-190, 2015.

22. Scheffer, C., "Pump Condition Monitoring Through Vibration Analysis", *Pumps: Maintenance, Design, and Reliability Conference 2008, IDC Technologies*, pp. 4-5, 2008.

23. Murphy, C., "Choosing the Most Suitable Accelerometer for Your Application-Part 2", *Analog Dialogue*, vol. 51, no. 11, November 2017, `https://www.analog.com/en/analog-dialogue/articles/choosing-the-most-suitable-accelerometer-for-your-application-part-2.html`, accessed in July 2020.

24. Bonaldi, E., L. Oliveira, J. Borges da Silva, G. Lambert-Torres and L. Silva, "Predictive Maintenance by Electrical Signature Analysis to Induction Motors", *Induction Motors-Modelling and Control*, pp. 487-520, IntechOpen, 2012.

25. Koo, I. S. and W. W. Kim, "The Development of Reactor Coolant Pump Vibration Monitoring and a Diagnostic System in the Nuclear Power Plant", *ISA Transactions*, vol. 39, no. 3, pp. 309-316, Elsevier Press, 2000.

26. Rafiee, J., M. Rafiee and P. Tse, "Application of Mother Wavelet Functions for Automatic Gear and Bearing Fault Diagnosis", *Expert Systems with Applications*, vol. 37, no. 6, pp. 4568-4579, Elsevier Press, 2010.

27. Percival, D. B. and A. T. Walden, "Wavelet Methods for Time Series Analysis", pp. 159-160, Cambridge University Press, 2000.

28. Ngui, W. K., M. S. Leong, L. M. Hee and A. M. Abdelrhman, "Wavelet Analysis: Mother Wavelet Selection Methods", *Applied Mechanics and Materials*, vol. 393, pp. 953-958, Trans Tech Publications, 2013.

29. Allwein, E. L., R. E. Schapire and Y. Singer, "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers", *Journal of Machine Learning Research*, vol. 1, pp. 113–141, September 2001.

30. Escalera, S., O. Pujol and P. Radeva, "On the Decoding Process in Ternary Error-Correcting Output Codes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 120-134, 2010.

31. Ding, C. and H. Peng, "Minimum Redundancy Feature Selection from Microarray Gene Expression Data", *Proceedings of the 2003 IEEE Computational Systems Bioinformatics Conference*, pp. 523-528, 2003.

32. Rumelhart, D. E., G.E. Hinton and R. J. Williams, "Learning Representations by Back-Propagating Errors", *Neurocomputing: Foundations of Research*, pp. 696–699, MIT Press, 1988.

33. Marquardt, D.W., "An Algorithm for Least-Squares Estimation of Nonlinear Parameters", *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431-441, 1963.

34. MacKay, D. J., "Bayesian Interpolation", *Neural Computation*, vol. 4, pp. 415-447, 1991.