

Course: FMH606-1 22V Master's Thesis, 2022

Title: AE-Sensors and Multimodal Sensor Data Fusion in Liquid Flow metering

Number of pages: 85 report + 41 appendices = 126

Keywords: Accelerometer, multiphase, flow rate, machine learning, neural network

Availability: Open

Student: Shailesh Kharche

Supervisor: Ru Yan (main supervisor)
Saba Mylvaganam (co-supervisor)

External partner: Kjetil Fjalestad, Equinor
Tonni Franke Johansen, SINTEF

Summary:

One of the biggest challenges in Oil and gas industries is finding convenient method for accurately measuring flow rate of multiphase materials flowing through a system. There are different approaches done to handle this situation and each ended up with different results. To continue research & development on this topic, two such experiments sites in this case rigs are present, one is in University of South-Eastern Norway and other one in Equinor.

This thesis objective is to estimate single phase flow velocity using clamp-on accelerometer sensors fitted on outer surface of pipes. Raw accelerometer data along with other sensor data like temperature and differential pressure was collected at both rigs. Since the main focus was on accelerometer data, complete thesis was done using only accelerometer data. The data was analyzed using FFT and PSD plots, filtered and pre-processed. Feature extraction was done.

The top three features were used to develop classification models to identify the type of flow material i.e., Gas, Oil or Water. The test accuracy of classification model is around 98 %. Then prediction model was developed for estimation of flow velocity. Top accelerometer features selected for prediction gave an RMSE of nearly 10.2.

The University of South-Eastern Norway takes no responsibility for the results and conclusions in this student report.

Preface

This thesis was completed as part of two-year master's study program – Industrial IT and Automation. It was worked on and written from January to May 2022. It is a result of 5 months of work, which included studying some additional concepts not limited to but including signal analysis and handling, which was completely new to the author in terms of experience.

Kjetil Fjalestad from EQUINOR provided raw experimental data consisting of sensor's data including accelerometer data. Another experimental data from USN rig was collected and given in April 2022.

Basic prior knowledge of vibrational analysis or basic signal concepts and basics of machine learning can be advantageous for reading this thesis.

I would like to thank Ru Yan for helping throughout the thesis. Also, I would like to thank Saba Mylvaganam for co-supervising the work done for this thesis. I would also like to thank Kjetil Fjalestad and Ashim Khadka for running experiments for data collection.

Microsoft Word is used for writing this report. Complete technical work is done in MATLAB.

Front page illustration is made by the author of this thesis.

Porsgrunn, 10th May 2022



Shailesh Kharche

Contents

Preface	3
List of Figures.....	6
List of Tables	9
Nomenclature	10
1 Introduction	11
1.1 Objectives	11
1.2 Workflow	11
1.3 Scope	12
1.4 Report Structure	12
2 Fluid Flow metering	13
2.1 Latest developments	13
2.2 Types of flow meters	13
2.3 Flow meters and their influence in multiphase flow	14
3 Single phase flow rate experiment	15
3.1 Equinor rig experiments	15
3.2 USN rig experiments.....	16
3.3 Accelerometer Sensor.....	17
4 Raw Data Analysis	18
4.1 Raw Data structure	18
4.2 Revamped Data Structure for ML.....	20
5 Accelerometer Data Analysis	22
5.1 Working of accelerometer sensor	22
5.2 Vibrations and flow rate	23
5.2.1 <i>Various Studies Based on Vibration & Flow velocity</i>	23
5.3 Spectral Analysis	25
5.3.1 <i>Raw Signal Plot</i>	25
5.3.2 <i>Fast Fourier Transform of vibration data</i>	25
5.3.3 <i>Power Spectral density of vibration data</i>	29
5.3.4 <i>Relative study of different flow types</i>	33
6 Pre-Processing of Accelerometer Data	34
6.1 Filtering of vibration signals.....	34
6.2 Designing of filter	34
6.3 Filtered signal output	35
6.3.1 <i>For Water flow experiments</i>	35
6.3.2 <i>For Gas flow experiments</i>	36
6.3.3 <i>For Oil flow experiments</i>	36
6.4 Splitting of filtered signal.....	36
6.5 Feature Engineering	37
6.5.1 <i>Accelerometer features</i>	37
6.6 Feature Dataset	40
6.7 Normalization of dataset	42
6.7.1 <i>Adding de-normalizing capability</i>	42
6.8 Final Dataset for ML models	43

6.8.1 *Tabular format of training and test data set*..... 43

7 Classification Model 46

7.1 Basics of Machine Learning 46

7.1.1 *Common Terminology* 47

7.2 Algorithms Explained 47

7.2.1 *Linear Discriminant Analysis*..... 47

7.2.2 *Naive Bayes* 48

7.2.3 *Support Vector Machine (SVM)*..... 49

7.2.4 *K-Nearest Neighbour (KNN)*..... 49

7.2.5 *Gaussian Processes (GP)* 50

7.2.6 *Ensemble Methods* 50

7.2.7 *Neural Network*..... 50

7.3 Flow type classification model 52

7.3.1 *KNN Model* 53

7.3.2 *SVM Model* 54

8 Flow rate Regression Model 55

8.1.1 *Accelerometer Channel 1 GP Model* 56

8.1.2 *Channel 2 GP Model* 58

8.1.3 *Channel 1,2,3 Ensemble Bagged*..... 60

8.1.4 *Channel 1 and 2 GP* 62

8.1.5 *Channel 1 and 2 Ensemble Bagged* 64

9 Results..... 66

9.1 MATLAB Live Editor 66

9.2 MATLAB Simulink Demonstration 69

9.3 Model Accuracy 70

9.4 USN Test Data 71

9.4.1 *Spectral Analysis of USN data*..... 71

9.4.2 *Power Spectral Density of accelerometer channel data* 75

9.5 Compatibility check of USN dataset with Equinor dataset..... 76

9.5.1 *Classification model test results* 77

9.5.2 *Regression model test results*..... 79

10 Discussion 81

10.1 Key Findings 81

10.2 Limitations..... 81

10.3 Sensor Fusion Possibility with ECT based approach 81

11 Conclusion..... 83

References..... 84

Appendices..... 86

List of Figures

Figure 1.1 Overview of workflow carried in this thesis	11
Figure 1.2 Block Diagram of report structure.....	12
Figure 3.1: Accelerometer sensors position in Equinor rig	15
Figure 3.2: Piping & Instrument Diagram of USN rig	16
Figure 3.3: USN rig site photo with focus on accelerometer sensor's location.....	17
Figure 3.4: Clamp-on HS-100 accelerometer sensor fitted on horizontal pipe in USN rig.....	17
Figure 4.1 Internal Structure of Raw Data files (.mat)	18
Figure 4.2 Screen Snip of rows showing values of Oil Choke experiments (OCxx)	20
Figure 5.1 Basic illustration of accelerometer sensor on pipe	22
Figure 5.2 Plot of Raw accelerometer channel 1 of first 25000 samples	25
Figure 5.3 FFT plot of Accelerometer channel 1 Water type experiments	26
Figure 5.4 FFT plot of Accelerometer channel 2 Water type experiments	26
Figure 5.5 FFT plot of Accelerometer channel 3 Water type experiments	26
Figure 5.6 FFT plot of Accelerometer channel 1 Gas type experiments	27
Figure 5.7 FFT plot of Accelerometer channel 2 Gas type experiments	27
Figure 5.8 FFT plot of Accelerometer channel 3 Gas type experiments	27
Figure 5.9 FFT plot of Accelerometer channel 1 Oil type experiments	28
Figure 5.10 FFT plot of Accelerometer channel 2 Oil type experiments	28
Figure 5.11 FFT plot of Accelerometer channel 3 Oil type experiments	28
Figure 5.12 PSD plot of Accelerometer channel 1 Water type experiments (Without Hanning Window)	30
Figure 5.13 PSD plot of Accelerometer channel 1 Water type experiments (With Hanning Window)	30
Figure 5.14 PSD plot of Accelerometer channel 1 Oil type experiments (Without Hanning Window)	31
Figure 5.15 PSD plot of Accelerometer channel 1 Oil type experiments (With Hanning Window)	31
Figure 5.16 PSD plot of Accelerometer channel 1 Gas type experiments (With Hanning Window)	32
Figure 5.17 PSD plot of Accelerometer channel 1 Gas type experiments (Without Hanning Window)	32

Figure 5.18 PSD plot of Accelerometer channel 1,2 and 3 for 40 m ³ /h flow rate (With Hanning Window).....	33
Figure 6.1 MATLAB filter design screen snip showing parameters	35
Figure 6.2 FFT plots of Water experiments (Unfiltered : Left) and (Filtered : Right)	35
Figure 6.3 FFT plots of Gas experiments (Unfiltered : Left) and (Filtered : Right).....	36
Figure 6.4 FFT plots of Oil experiments (Unfiltered : Left) and (Filtered : Right).....	36
Figure 6.5 One second split of accelerometer channel 1 signal of experiment G03	37
Figure 6.6 First 36 Rows out of 16,681 of feature dataset showing features values	41
Figure 6.7 Normalized Training dataset screen snip	44
Figure 6.8 Normalized Test dataset screen snip	45
Figure 7.1 Basic Machine Learning Diagram.....	46
Figure 7.2 Linear Discriminant Analysis illustration	48
Figure 7.3 Support Vector Machine plot [19].....	49
Figure 7.4 Illustration of KNN classification algorithm.....	49
Figure 7.5 Illustration of Gaussian Probability Function [21].....	50
Figure 7.6 Structure of Bagged Ensemble Algorithm	51
Figure 7.7 Simple Structure of Neural Network.....	51
Figure 7.8 Test Confusion Matrix of Fine KNN model.....	53
Figure 7.9 Test Confusion Matrix of Linear SVM model	54
Figure 8.1 Response plot of Accelerometer Channel 1 GP model	57
Figure 8.2 Predicted vs Actual Test plot of Accelerometer Channel 1 GP model	57
Figure 8.4 Response plot of Accelerometer Channel 2 GP model	59
Figure 8.3 Response plot of Accelerometer Channel 2 GP model	59
Figure 8.6 Predicted vs Actual Test plot of Accelerometer Channel 1,2,3 Ensemble bagged model.....	61
Figure 8.5 Response plot of Accelerometer Channel 1,2,3 Ensemble bagged model.....	61
Figure 8.8 Predicted vs Actual Test plot of Accelerometer Channel 1 and 2 GP model.....	63
Figure 8.7 Response plot of Accelerometer Channel 1 and 2 GP model	63
Figure 8.10 Predicted vs Actual Test plot of Accelerometer Channel 1 and 2 Ensemble Bagged model	65
Figure 8.9 Response plot of Accelerometer Channel 1 and 2 Ensemble Bagged model.....	65
Figure 9.1 Block Diagram showing testing scenario used to showcase the results.....	66
Figure 9.2 Screen Snips of MATLAB live editor showing testing of classification models...67	67
Figure 9.3 Screen Snips of MATLAB live editor showing testing of Prediction models	68

Figure 9.4 Screen Snips of MATLAB Simulink showing usage of classification model (NN)	69
Figure 9.5 Screen Snips of MATLAB Simulink showing usage of regression model (NN)	69
Figure 9.6 Work Flow Chart of test data handling to get accuracy	70
Figure 9.7 Work Flow Chart of test data handling to get accuracy	72
Figure 9.8 Work Flow Chart of test data handling to get accuracy	72
Figure 9.10 Work Flow Chart of test data handling to get accuracy	73
Figure 9.9 Work Flow Chart of test data handling to get accuracy	73
Figure 9.12 Work Flow Chart of test data handling to get accuracy	74
Figure 9.11 Work Flow Chart of test data handling to get accuracy	74
Figure 9.14 Work Flow Chart of test data handling to get accuracy	75
Figure 9.13 Work Flow Chart of test data handling to get accuracy	75
Figure 9.15 Work Flow Chart of test data handling to get accuracy	76
Figure 9.16 Test Confusion matrix of classification model with USN test data	78
Figure 9.17 Different classification model performances with USN test data	78
Figure 9.18 Response plot of GPR regression model	80
Figure 10.1 One possible sensor data fusion with electrical capacitance tomography	82

List of Tables

Table 2.1: Types of flow meters based on setup and working principle.	13
Table 3.1: Experiments performed at Equinor.....	15
Table 3.2: Experiments performed at USN.....	16
Table 4.1: Variables present in raw data from Equinor	19
Table 5.1: Summary of various studies based on vibrations and flow velocity	24
Table 6.1: Features used on accelerometer signals	38
Table 6.2: Manually separated training and test data	43
Table 7.1: Different classification model performance	52
Table 7.2: KNN model performance	53
Table 7.3: Linear SVM model performance	54
Table 8.1: Different Prediction Model Performance	55
Table 8.2: Accelerometer Channel 1 GP model performance	56
Table 8.3: Accelerometer Channel 2 GP model performance	58
Table 8.4: Accelerometer Channel 1,2,3 Ensemble model performance	60
Table 8.5: Accelerometer Channel 1 and 2 GP model performance	62
Table 8.6: Accelerometer Channel 1 GP model performance	64
Table 9.1: Flow Rate Prediction Model accuracy for each test experiment	71
Table 9.2: Flow Rate Prediction Model accuracy for each test experiment	76
Table 9.3: Linear Discriminant classification model performance.....	77
Table 9.3: GPR model performance	79
Table 10.1: GP regression model results showing true flow and predicted flow using 4 features of accelerometer channel 2.....	83

Nomenclature

Symbol	Explanation
AUC	Area Under Curve
CH	Channel (accelerometer)
dB	Decibel
ECT	Electrical Capacitance Tomography
FFT	Fast Fourier Transform
G	Gas
GP	Gaussian Process
GPR	Gaussian Process Regression
KNN	K-Nearest Neighbour
LDA	Linear Discriminant Analysis
ML	Machine Learning
MSE	Mean Squared Error
MMSE	Minimum Mean Square Error
NN	Neural Network
O	Oil
OT	Oil Test
PCA	Principal Component Analysis
PSD	Power Spectrum Density
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
RSSQ	Root Sum of Squares
USN	University of South-East Norway
W	Water

1 Introduction

For the last two decades, extensive research has been done for multiphase flow measurement in oil and gas production industry. Different approaches like non-invasive and invasive methods are tried to get better results of flow measurement. To continue further research, two such experimental setup is present, one in USN, Porsgrunn and one in Equinor, Herøya, Grenland. Recently the focus is on flow measurement using clam-on accelerometer sensors.

1.1 Objectives

Multiphase flow consists of three materials i.e., Oil, Gas and Water. The main objective of this thesis is to predict type of material flowing inside pipe and also to estimate flow velocity of that material using accelerometer data and machine learning models (see, Appendix A).

1.2 Workflow

Raw accelerometer data is collected at both the rigs. Data is imported in MATLAB. Since the accelerometer data is in the form of signal, signal analysis is done. Analysis like FFT is done to study frequencies in data and the change in frequency patterns when flow type changes and also when flow rate changes. Spectral analysis is also performed to study power spectrum of accelerometer signals and the variations in power due to change in flow type. Filtering of signals is performed. Signal is then split into few seconds duration. Feature extraction is done and this feature acts as an input to machine learning models. Classification model and Prediction model is developed.

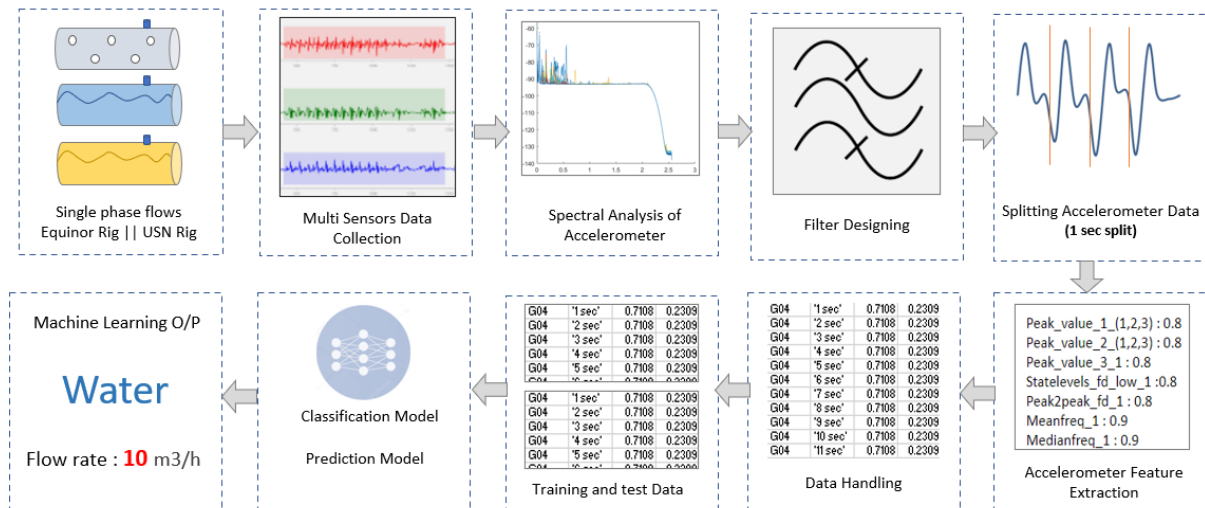


Figure 1.1 Overview of workflow carried in this thesis

1.3 Scope

The nature of accelerometer data is limited to experimental setup at mentioned locations. Also, the models developed are expected to work for single-phase flow metering. The minimum and maximum flow rate for estimation is limited to the flow rate at which the data is captured. The values are mentioned in respective chapters.

1.4 Report Structure

The coming chapters follows the workflow mentioned above and are organized as follows:

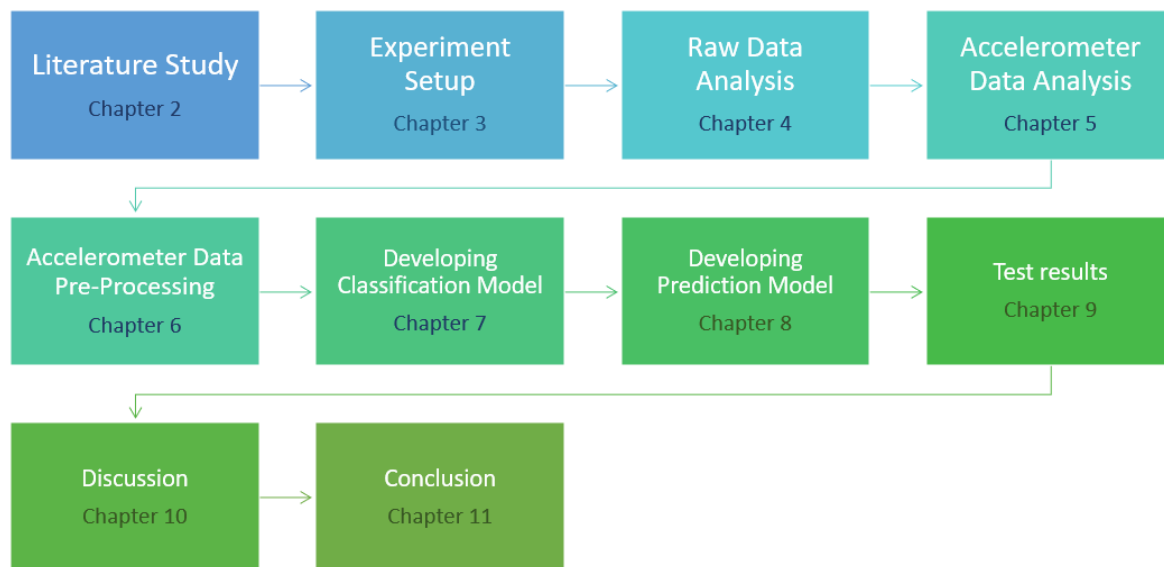


Figure 1.2 Block Diagram of report structure

Chapter 2 covers the literature study of latest developments in fluid flow metering and different approaches done to estimate flow velocity. Chapter 3 covers experimental setup scenario, types of experiments performed, and the raw data generated from these experiments. First raw data analysis is performed and is mentioned in chapter 4. Analyzing accelerometer data is then done in chapter 5. Pre-Processing of this accelerometer data is then done in chapter 6. Chapter 7 covers developing classification model for estimating flow type and chapter 8 covers developing prediction model for estimation of flow velocity. Chapter 9 includes testing of models developed in previous chapters. Since there is additional accelerometer data from USN rig, pre-processing of this data and testing of this data with ML models developed using Equinor data is covered in same chapter 9 as one separate section. Discussion based on outcome of work done in this thesis is covered in chapter 10. Finally, conclusion is covered in chapter 11.

2 Fluid Flow metering

In this chapter, brief survey of fluid flow metering is mentioned, particularly focusing on latest development in this field followed by different approaches to estimate flow velocity and different types of liquid flow meters.

2.1 Latest developments

Virtual Flow Metering is well-known term in latest developments related to fluid flow metering, especially done in multiphase flow scenarios. This kind of approach involves gathering not directly related sensor readings like pressure at different points in experiment, temperature of liquid and many more. In one such study, VFM was able to reconcile total oil and total water flow rates with average relative deviations of 0.87% and 17% respectively and maximum deviation of 2.3% for oil flow rates [1]. Another is thermal pulse time-of-flight based liquid flow meter. In this the heat pulse is imparted in flowing liquid and its detection in arrival downstream is used to predict flow velocity [2]. Ultrasound based flow velocity measurements is another non-invasive approach [3]. Electrical Capacitance Tomography which involves technique of reading several capacitance sensor's readings, which is a result of dielectric permittivity influence of liquid flowing through a pipe [4].

2.2 Types of flow meters

Below table shows different types of flow meters used till now to estimate flow velocity along with the principles they are based on.

Table 2.1: Types of flow meters based on setup and working principle.

Type	Setup	Description
Differential Pressure	Invasive	Based on the difference in pressure between upstream and downstream sides of a restriction in a confined fluid stream, which is related to square of fluid velocity
Differential Area	Invasive	A free moving float inside a glass tube to get the fluid velocity
Electromagnetic	Non - Invasive	Based on Faraday's law of magnetic induction which states that when a conductive material (in this case a conductive fluid) moves in a magnetic field, a voltage is generated between two electrodes at right angles to fluid velocity.
Ultrasonic	Non - Invasive	Acoustic waves are passed in between transmitter and receiver. Time difference to travel these waves varies in correspondence to fluid velocity.
Turbine	Invasive	Multi-bladed rotor mounted and suspended in the fluid stream to get flow velocity.

Vortex	Invasive	An obstruction placed inside a pipe creates vortices and this shedding frequency is directly proportional to fluid velocity.
Positive Displacement	Invasive	This meter repeatedly entraps the fluid into a known quantity and then passes it out. Rotor rotational velocity is directly proportional to flow rate, since the flow of fluid is causing the rotation.
Coriolis Mass	Invasive	Flow is passed through a tube which is continuously moving and flow rate causes change in frequency of this tube's movement. This movement is directly relating to mass flow rate.
Thermal Mass	Non - Invasive	Two temperature transducers are used out of which one monitors actual gas flow temperature. Flow velocity causes the change in temperature on one transducer and this difference is used to calculate flow velocity.

2.3 Flow meters and their influence in multiphase flow

The flow meters mentioned in table above are successfully used in other common applications where flow fluid is of one phase and the phenomenon is simple to model and understand like water, non-viscous and semi-viscous chemicals, only oil, different gas flow applications. But multiphase is complex phenomenon which is difficult to understand, predict and model [5]. Venturi meter based on differential pressure type is often used to determine velocity of multiphase flow. However, the equations for single phase can-not be directly applied to multiphase flows and thus are modified for use in multiphase flow measurement.

Multiphase flow metering usually comprises of combination of different techniques described above. For instance, a positive displacement meter will usually measure total volumetric multiphase flow rate (gas and liquid) [5].

Many meters are developed using electromagnetic measurement principles to apply cross-correlation techniques to calculate characteristic velocity of multiphase mixture [5].

Several Electrical Impedance techniques which are based on measuring the electrical permittivity and conductivity characteristics of materials of fluid flowing is used to determine the proportion of materials flowing which is further used to classify flow regimes in one of the studies [6].

Gamma Ray Meter is also used to find fluid density based on its multiphase components.

This thesis brings in non-invasive way of measuring flow type and flow velocity using vibrations caused by single-phase flow in the pipe.

3 Single phase flow rate experiment

The work done in this thesis is based on two large datasets. One dataset is from Equinor Rig and another dataset is from USN rig. This chapter presents the experimental setup with focus on location of accelerometer sensors. Also, the details of experiments along with structure of data obtained is mentioned.

3.1 Equinor rig experiments

The rig is a multiphase flow rig consisting of different flow meters of make Krohne and Enders. Also, Differential pressure transmitters of make Emco and Wika are present on the rig. Temperature and pressure sensors are fitted at certain locations. 4 accelerometer sensors are fitted on certain locations as shown in figure 3.1. Since the main focus is only on accelerometer sensor readings, in the figure only accelerometer sensors position is mentioned. Also, it is worth mentioning that accelerometer sensor 2 is defective at the time of performing these experiments.

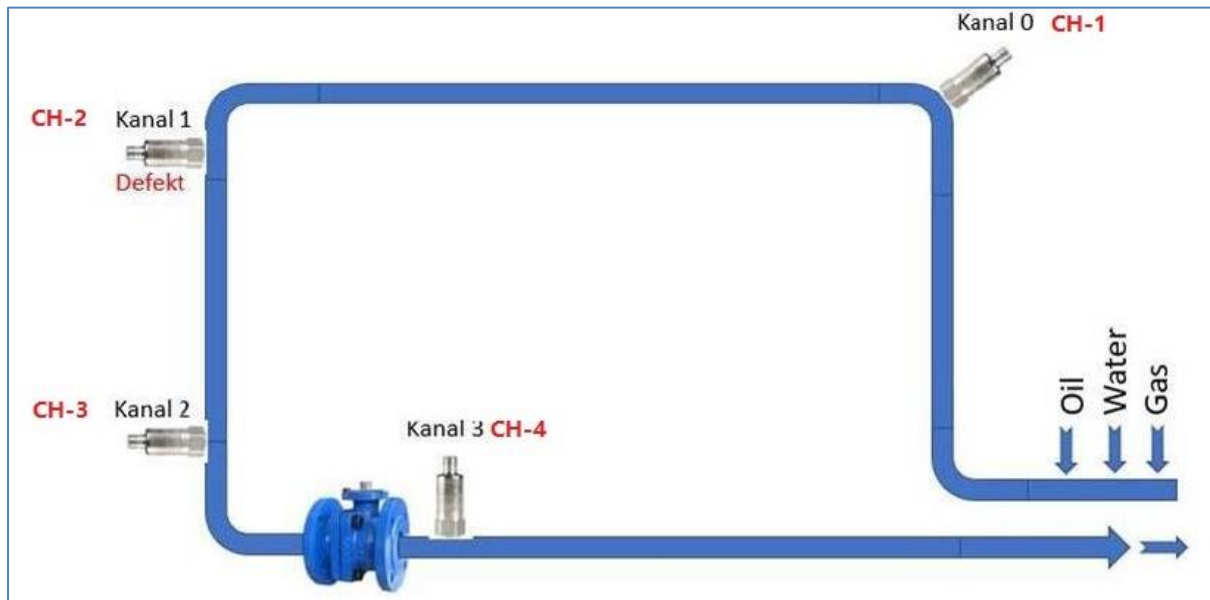


Figure 3.1: Accelerometer sensors position in Equinor rig (Simplified drawing, provided by Equinor)

Experiments performed are shown in table 3.1 below.

Table 3.1: Experiments performed at Equinor (“xx”: test sequence numbers)

	Experiment	Number of experiments	Data File Name	Flow Range (m ³ /h)
1	Water	7	Wxx	2 – 60
2	Oil	15	OTxx	2 – 40
3	Gas	10	Gxx	30 - 200

The pipe on which accelerometer channel 1 is fitted is vertical pipe with flow direction from bottom to top. Accelerometer channel 2 and 3 are also fitted on vertical pipe with flow direction from top to bottom. Accelerometer channel 4 is located after the choke valve. Channel 2 is defective in all these experiments hence no data is present from that channel. The details of experiments conducted is mentioned in appendix C. Gas and Water experiment's duration is around 10 minutes per experiment while Oil experiment's duration is around 15 minutes per experiment.

3.2 USN rig experiments

The rig is a multiphase flow rig consisting of various sensors like flow meters, pressure transmitters and accelerometers as shown in figure 3.2. The location of accelerometer sensors in this rig is as shown in figure below using naming convention of Loc.1 and Loc. 2 meaning location of accelerometer 1 and 2 respectively.

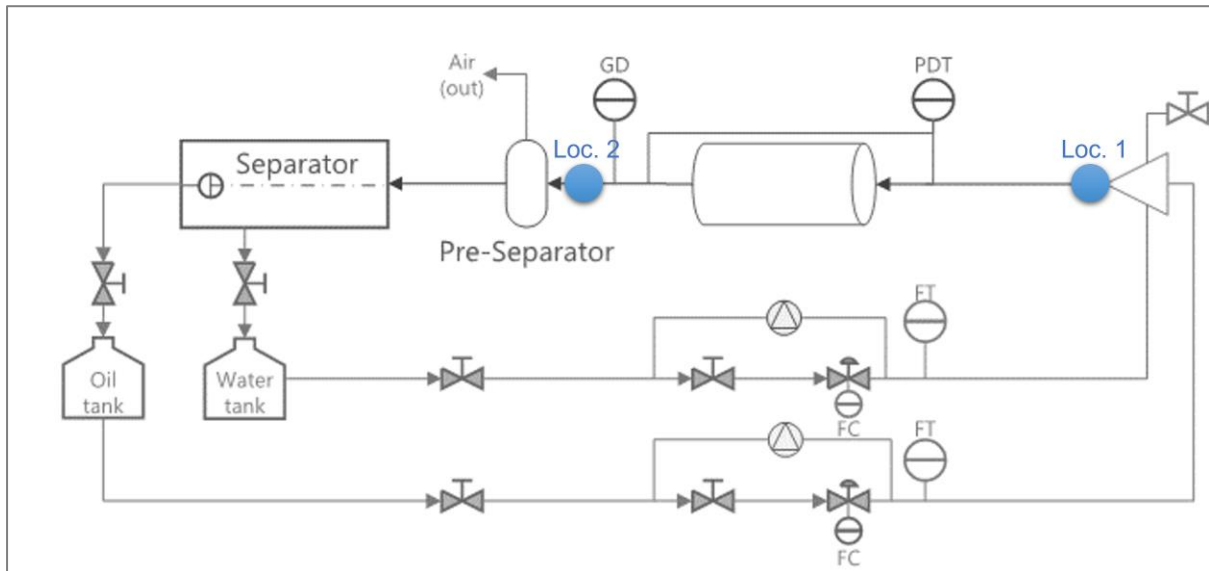


Figure 3.2: Piping & Instrument Diagram of USN rig

Unlike to Equinor rig, the location of accelerometer sensors in USN rig is on horizontal pipe. Single phase flow experiments are performed and only accelerometer sensors data is recorded along with reference flow rate. The experiments brief summary is as shown in table below.

Table 3.2: Experiments performed at USN (“x” and “xx”: flow rates)

	Experiment	Number of experiments	Data File Name (x: flow xx: channel)	Flow Range (kg/min)	Flow Range (m ³ /hr.)
1	Water	5	Water_x_acc_xx	2 - 50	0.12 - 3
2	Oil	5	Oil_x_acc_xx	2 - 50	0.12 - 3
3	Gas	7	Air_x_acc_xx	0.2 - 2	0.01 – 0.12

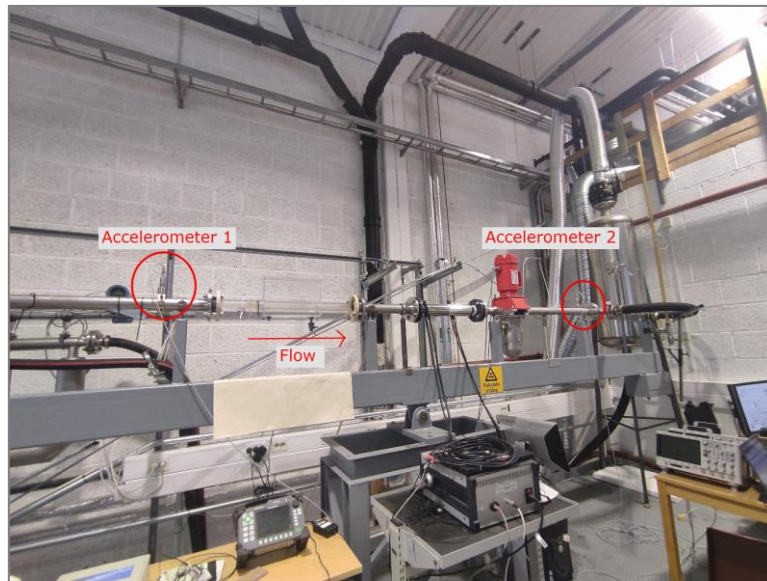


Figure 3.3: USN rig site photo with focus on accelerometer sensor's location

3.3 Accelerometer Sensor

An accelerometer sensor measures the acceleration forces acting on an object, which enables to monitor object's movement and position in space. There are two types of acceleration forces: static forces and dynamic forces. Static forces are forces that are constantly being applied to the object (such as friction or gravity). Dynamic forces are “moving” forces applied to the object at various rates (such as vibration, or the force exerted on a cue ball in a game of pool). In the experiments mentioned in this Thesis, the accelerometer of make Hansford Sensors having model number HS-100 is used having a frequency response with minimum sensitivity changes of $\pm 3\text{dB}$ in between 0.8 Hz to 15 kHz [7]. However, the mounted resonant frequency of this sensor is 30 kHz. As the name implies, it is the result of the natural resonance of the mechanical structure of the accelerometer itself.



Figure 3.4: Clamp-on HS-100 accelerometer sensor fitted on horizontal pipe in USN rig

4 Raw Data Analysis

In this chapter analysis of raw data from Equinor is performed. This covers data handling like getting all data in MATLAB, putting data in tabular format, finding missing values, find outliers with respect to single phase experiments.

4.1 Raw Data structure

The Data obtained is in the form of MATLAB data file and is named according to type of flow material and corresponding number of experiment, for example one such file is G02.mat which contains sensor readings of one gas experiment with flow rate of 200 m³/h. In total 32 such files are present from Equinor rig experiments.

For each experiment, 52 variables are collected. Variables in this context is the values of different sensors located at various positions and includes values of temperature, differential pressure, density, choke valve position, mass flow rate, volumetric flow rate and accelerometer sensor. Table 4.1 shows variables present in raw data along with their meaning and units.

Custom made MATLAB functions mentioned in appendix is used to extract data from each raw data experiment file and data is put in tabular format for further processing. After performing loop to find missing values, 4 values of Krohne flowmeter were found missing. 4 experiments named W12, OT30, OT28 and OT26 doesn't have Krohne flow rate. Internal structure of files is as shown in figure 4.1.

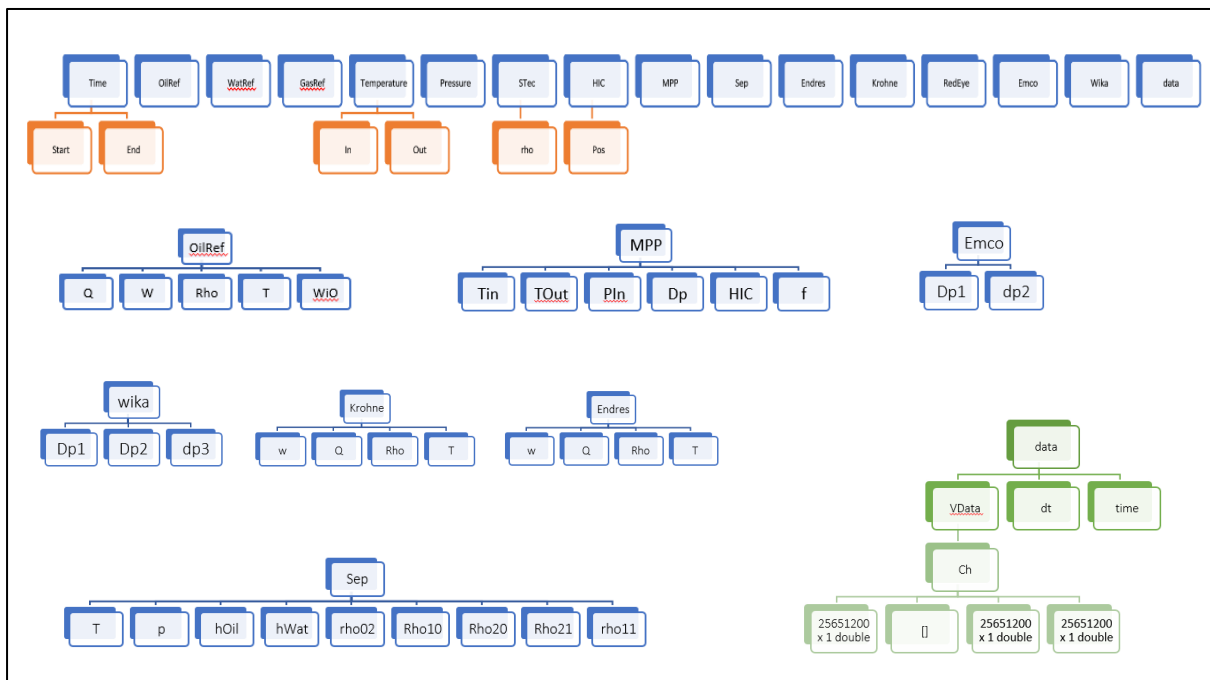


Figure 4.1 Internal Structure of Raw Data files (.mat)

Table 4.1: Variables present in raw data from Equinor

Sr No	Variable	Variable Sub Set	Meaning	Unit
1	oilRef	q	Volumetric Flow	m3/h
2		w	mass flow rate	t/h
3		rho	density	kg/m3
4		T	Temp	deg C
5		WiO	Water in Oil	%
6	watRef	q	Volumetric Flow	m3/h
7		w	mass flow rate	t/h
8		rho	density	kg/m3
9		T	Temp	deg C
10	gasRef	q	Volumetric Flow	m3/h
11		w	mass flow rate	t/h
12		rho	density	kg/m3
13		T	Temp	deg C
14	temp	in	In Temperature	deg C
15		out	Out Temperature	deg C
16	press	in	In pressure	bar
17		out	Out Pressure	bar
18	STec	rho	density (measured by gamma densitometer)	kg/m3
19	HIC	pos	choke valve position	%
20	MPP	TIn	In Temperature	deg C
21		TOut	Out Temperature	deg C
22		pIn	In pressure	bar
23		dp	differential pressure	psi
24		HIC	choke valve position	%
25		f	??	--
26	Sep	T	Temp	deg C
27		p	pressure	bar
28		hOil	Height of interaface level : Oil	--
29		hWat	Height of interaface level : Water	--
30		rho02	density	kg/m3
31		rho10	density	kg/m3
32		rho11	density	kg/m3
33		rho20	density	kg/m3
34		rho21	density	kg/m3
35	Endres	w	mass flow rate	t/h
36		q	Volumetric Flow	m3/h
37		rho	density	kg/m3
38		T	Temp	deg C
39	Krohne	w	mass flow rate	t/h
40		q	Volumetric Flow	m3/h
41		rho	density	kg/m3
42		T	Temp	deg C
43	RedEye	WC	Water Cut (Ratio of water compared to Total Volume)	%
44	Emco	dp1	Differential pressure # 1	psi
45		dp2	Differential pressure # 2	psi
46	Wika	dp1	Differential pressure # 1	psi
47		dp2	Differential pressure # 2	psi
48		dp3	Differential pressure # 3	psi
49	Data	ch:1	Accelerometer data from channel 1	g
50		ch:2	Accelerometer data from channel 2 (defective)	g
51		ch:3	Accelerometer data from channel 3	g
52		ch:4	Accelerometer data from channel 4	g

At this stage main work is to get data from different .mat files, combine them and put them in tabular format. MATLAB code mentioned in Appendix E is used for the same. Also, it is observed that Oil Choke experiments were creating outliers in many sensor readings which in turn were expanding the distribution of sensor readings range in histogram and box plot. The values in Oil Choke experiments (OCxx) can be seen in figure 4.2 below.

name	MPP_Tin	MPP_TOut	MPP_pln	MPP_dp	MPP_f	Emco_dp1	Emco_dp2	Wika_dp1	Wika_dp2	Wika_dp3
G02	69.03883106	59.64922919	38.19124616	-0.066083208	-2.99245062	256.9980342	112.4737137	558.5980624	97.96292706	105.9640592
G03	72.69905393	60.83960028	37.8759364	-0.065349572	-3.102617782	211.162822	96.40397644	448.8005239	77.36415504	80.68733877
G04	72.82123022	61.40107213	37.49725314	-0.064659091	-2.908291692	174.8174644	87.79443931	367.6444314	63.0376615	63.19997781
G05	72.66622869	61.30639852	37.07390732	-0.063968611	-2.708493479	136.089255	79.20513861	282.6767566	50.1882517	46.87320302
G06	72.35848572	60.88243353	36.67122474	-0.06327813	-2.753310442	103.8589531	73.71037422	207.9319598	39.84846899	33.22516451
G07	71.39145335	60.22076537	36.34207494	-0.06250134	-2.753310442	78.94132243	74.73431008	147.7317867	36.964293	27.32735762
G08	70.22840084	59.29780252	35.97635293	-0.061638239	-2.896064538	52.88614511	71.607141	92.94169618	34.89655759	22.16612815
G09	68.7972259	57.68958089	35.75403615	-0.060451475	-2.581936628	36.2379805	63.76998328	50.55108491	22.52701123	7.849848544
G10	67.80194341	56.75234267	35.67349065	-0.059502064	-2.475767522	22.31999322	60.23238382	18.27605278	16.15673182	0.220921362
G11	66.26107321	57.2614636	35.60392863	-0.058682119	-2.813265007	17.24107265	59.88178986	5.700006256	14.59918565	-1.786671405
OC01	66.84768037	95.09517534	59.05091352	22.75979797	956.4756611	17.97088913	43.32408755	8.831129101	6.900555648	-0.84449652
OC02	63.77661083	81.91829524	59.28583477	25.00445078	1009.094606	32.90163749	44.58861541	50.41352042	14.52368043	2.543718072
OC03	73.88542868	83.82438169	50.96988598	20.19186228	905.6000825	28.74559978	39.0338019	50.73307614	15.93452274	-0.612171918
OC04	71.81255711	78.93644435	40.73473104	14.36469438	782.2138122	24.82553737	41.10219021	46.22681232	16.30564941	0.624883716
OC07	71.41507372	81.31171198	59.49858024	26.28461436	1101.762968	107.6955621	60.20834667	221.8254801	38.20087003	30.19794842
OC08	72.64333401	79.46991794	59.73570386	26.15925398	1163.279787	220.6170735	81.97865832	492.3793875	73.03937812	68.28964859
OC09	75.71676569	80.10660632	60.20235803	24.94657033	1203.513219	373.9894651	120.3748425	854.0523083	123.2072439	139.5056748
OC10	74.96973028	78.67573288	60.59249014	24.87085949	1266.719256	560.2520374	164.1627469	1314.302469	185.9669423	223.6161675
OC11	79.83794456	80.62917907	51.60427222	14.00989673	1008.723583	563.8340055	164.4461568	1314.760773	181.279195	218.4229783
OC12	79.48861304	80.07160184	41.57663231	10.93251721	946.7587457	569.5304008	158.3954144	1311.238803	177.5595267	221.9443116
OC13	80.02754319	80.38295641	41.72836495	9.27619613	956.8069552	817.6706781	212.554042	1880.71472	246.6521322	317.8115662
OC14	78.9754777	80.90271459	50.14212917	18.75127667	1214.802204	813.8638098	221.6286052	1876.885142	257.1624081	327.8195388
OC16	77.3598984	80.94301982	59.45191358	22.26498372	1142.84193	363.3520373	112.0309674	844.8934004	123.3076738	159.4664327
OC17	76.47555523	79.93269212	51.42721027	20.50116304	1121.885892	361.6505292	108.1675563	836.9827613	120.6236507	154.8336308
OC18	76.16688695	80.34591234	45.75358245	21.08774875	1183.194429	365.7416846	109.1170505	850.0817723	120.8192739	154.9519534
OT08	70.92204728	62.6253155	40.1716812	-0.075506533	-2.690433867	395.24352	115.6583936	867.2452033	128.2649021	157.2868578
OT09	71.26870755	66.36258675	37.87857573	-0.078991638	-2.829260427	229.522936	79.93515142	491.1100237	81.4555391	89.66128529
OT10	70.38339265	62.55285146	35.02794472	-0.075860026	-2.79977249	36.18285986	49.87011518	51.19552937	20.50281164	11.34000676
OT12	70.36780787	63.24509353	35.18461668	-0.076250041	-2.753310442	48.3203164	51.509829	77.25336128	23.91641597	19.46376572
OT14	70.35222309	63.93733561	35.34128863	-0.076640055	-2.753310442	60.55320033	53.22756105	106.1689084	27.75694626	25.93998466
OT16	70.33663831	64.62957769	35.49796059	-0.077030069	-2.753310442	75.31955852	54.74035643	140.1426973	32.6828008	30.20096431
OT2	66.22427213	67.1133534	34.81046592	-0.073597944	-2.835595127	14.70386581	42.92977571	-5.780247115	5.777913176	-2.880055337
OT20	70.23472869	65.97571673	35.81147394	-0.077810098	-2.787331205	110.2115335	60.35883312	221.2865403	44.98945414	43.24227832
OT22	70.06294825	65.99929113	36.0157419	-0.078200112	-2.825363046	131.4638307	63.56859892	268.9479927	52.08779861	54.56280173
OT24	69.8909813	65.80023882	36.29848258	-0.078590126	-2.971187237	153.8639887	66.91877438	320.1571438	59.46139966	62.03876782

Figure 4.2 Screen Snip of rows showing values of Oil Choke experiments (OCxx)

Since the focus of this work is to find relation between accelerometer data and single - phase flow rates, Oil Choke experiments were removed from the dataset and only single - phase experiments were considered for further analysis.

4.2 Revamped Data Structure for ML

At this point the data is split into two parts as follows :

1. All Variables except accelerometer data (Variable named 'Data' from figure 4.1 & table 4.2)

This data is sensor variables for each experiment and contains 51 variables for each experiment.
2. Accelerometer data

This data is accelerometer channel 1, 3 & 4 values are each experiment and contains 78 variables i.e., 26 features of each channel for each experiment.

5 Accelerometer Data Analysis

Since the main area of focus of this thesis is estimating flow velocity in single phase flows using accelerometer sensor network, further thesis continues with only 3 variables from total of 52 variables. The 3 variables are accelerometer channel 1, channel 3 and channel 4.

This chapter covers the working principle of accelerometer sensor. Relationship of accelerometer signals with flow velocity is studied. Spectral Analysis is performed to study effect of flow velocity and flow type on accelerometer signals.

5.1 Working of accelerometer sensor

Accelerometers are full-contact transducers typically mounted directly on high-frequency elements. They rely on the use of piezoelectric effect which occurs when a voltage is generated across certain types of crystals as they are stressed. The vibration of test structure on which these accelerometers are fitted, is transmitted to a seismic mass inside the accelerometer that generates a proportional force on the piezoelectric crystal. This external stress on the crystal then generates a high-impedance electrical charge proportional to the applied force and thus proportional to vibration.

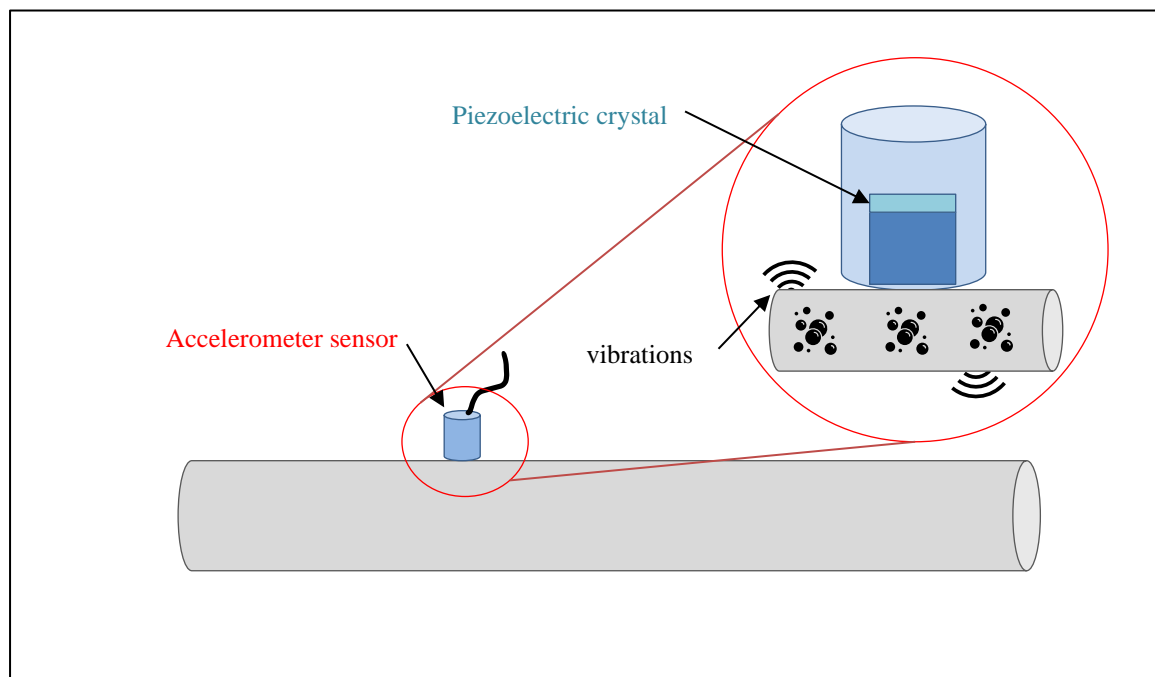


Figure 5.1 Basic illustration of accelerometer sensor on pipe

Piezoelectric or charge mode accelerometers require an external amplifier or inline charge converter to amplify the generated charge, lower the output impedance for compatibility with measurement devices, and minimize susceptibility to external noise sources and crosstalk. Other accelerometers have a charge-sensitive amplifier built inside them. This amplifier accepts a constant current source and varies its impedance with respect to a varying charge on the piezoelectric crystal. The benefits of an accelerometer include linearity over a wide frequency range and a large dynamic range.

5.2 Vibrations and flow rate

Accelerometer sensor measures vibrations caused by material flowing through pipes. Theoretically it is proved that the flow rates in pipes are linearly related to the transverse vibrations induced in pipes [8]. Also, relationship between fluid flow rates in pipes and vibrations due to it is mentioned in Blake [9].

In the literature, the experimental correlation between the fluid flow rate through a pipe (Q) and the acceleration affecting the pipe wall in the radial direction has been described with a series of linear relations (\propto), expressed by (5.1)

$$Q = AU \propto u' \propto \tau_w \propto \frac{\partial^2 \tau_w}{\partial t^2} \quad (5.1)$$

Where,

A = cross sectional area of pipe

U = average flow velocity

u' = flow velocity fluctuations along axial

τ_w = shear stress in the pipe

Direct mathematical relation between vibration and flow rate in third order root function of water flow rate is shown by Equation (5.2) [10].

$$f(t) = \alpha^3 \sqrt{v(t)} + \beta \sqrt{v(t)} + \gamma v(t) + \delta \quad (5.2)$$

Where, $f(t)$ = flow rate, $v(t)$ = measured vibration and α , β and γ are function parameters that must be adjusted according to study case.

Since the nature of study which is dealt in this thesis is the basis for complex process of multi-phase flows, it's difficult to make mathematical model relationship between vibration and flow rate. Hence considering that there is relation between vibration patterns induced on pipe walls due to flow velocity, further spectral analysis is done to obtain the vibrations patterns due to Oil, Water and Gas flow type. And this vibrations patterns forms as basis for feature extraction. But there are many things to cover before getting there.

5.2.1 Various Studies Based on Vibration & Flow velocity

As part of the literature study, previous studies based on vibration analysis and its relation to flow velocity are studied and summarized in table 5.1 below. This acts as a strong support for this thesis in relation to type of approach and features selection.

S. No	Reference	Study Name	Description	Vibration Features Used	Outcome
1	11	Towards flow measurement with passive accelerometers	Finding suitable flow measurement and characterization with passive accelerometers to estimate flow quantity.	Mean frequency, Lag between signals	KNN algorithm had a classification accuracy of 83 %. Low accuracy compared to traditional flow meter.
2	12	Fluid Flow Rate Estimation using Acceleration Sensors	Water experiments used to improve measurement of fluid flow through measurement of vibrations	First harmonic amplitude	amplitude of the first harmonic increases as the flow rate grows
3	13	Flow Measurement by Piezoelectric Accelerometers: Application in the Oil Industry	The technique used consists of measuring the vibrations induced by the passage of flow through the pipeline, known as the flow induced vibration (FIV), so that the flow rate is estimated from the standard deviation of the measurement of this vibration	Standard deviation between two accelerometers	FIV method based on standard deviation is not yet acceptable in context of fiscal measurement as it showed uncertainty between 2.5 % and 5 %
4	14	Prediction of Flow Velocity from the Flexural Vibration of a Fluid-Conveying Pipe Using the Transfer Function Method	The components of wavenumbers changes at low frequencies and converge at high frequencies and these are then used in transfer function to predict flow velocities	High and low frequency components	At lower frequencies, the prediction based on transfer function decreased. However, on high frequencies prediction rate was good.

Table 5.1: Summary of various studies based on vibrations and flow velocity

5.3 Spectral Analysis

The raw data obtained from accelerometer sensor is of form continuous time series data which gives gravity (g) against time (t). The data collected for this thesis has a sampling frequency of 51.2 kHz i.e., 51200 samples are collected every second and that too the experiment's length is around 10-15 minutes.

5.3.1 Raw Signal Plot

Directly plotting accelerometer channel 1 data gives output figure like shown below. The figure shown below is of experiment G02 and channel 1. Hence corresponding signal processing is done on raw data and is covered in the following sections.

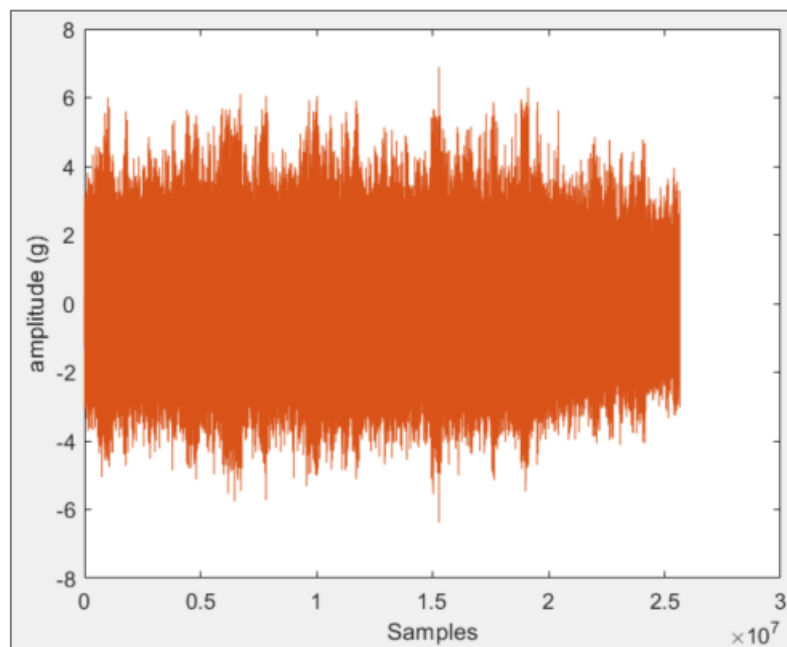


Figure 5.2 Plot of Raw accelerometer channel 1 of first 25000 samples

5.3.2 Fast Fourier Transform of vibration data

Direct plots of accelerometer signals in time domain are not informational. In order to extract relevant information from them, an algorithm named FFT is used. This algorithm converts original domain i.e., time domain data of signals to a representation in frequency domain. The accelerometer data which is in the form of waveform is actually a sum of serious of different frequencies, amplitudes and phases. To deconstruct this waveform into individual components, Fourier analysis is used. FFT plots in this case enables to study the presence of certain frequencies in accelerometer data and identify different frequencies with different amplitudes in Gas, Water and Oil type flow and also helps to study change in frequencies and amplitudes when flow rate is changed. Plots are plotted according to flow type i.e., all the experiments with only Water flow but with different flow rate is shown in figure 5.3 to 5.5. Likewise, Gas and Oil FFT plots are shown in figure 5.6 to 5.8 and figure 5.9 to 5.11 respectively.

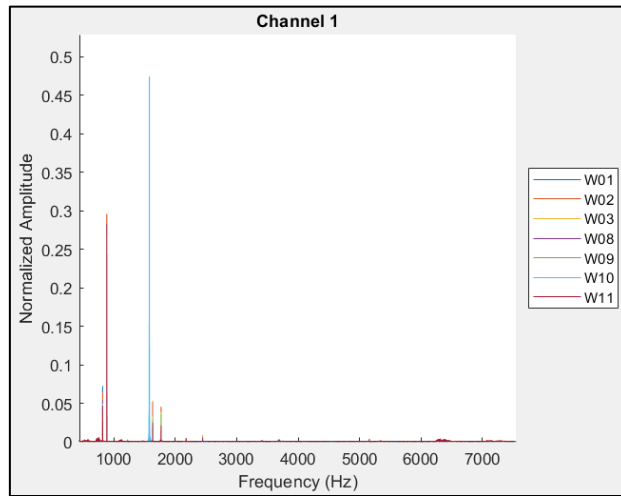


Figure 5.3 FFT plot of Accelerometer channel 1 Water type experiments

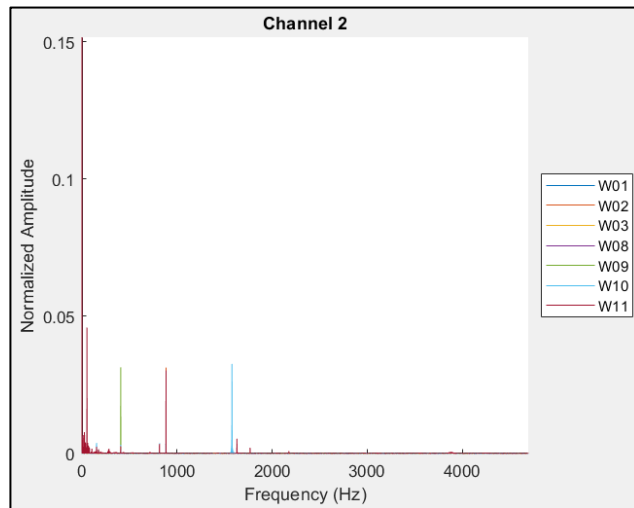


Figure 5.4 FFT plot of Accelerometer channel 2 Water type experiments

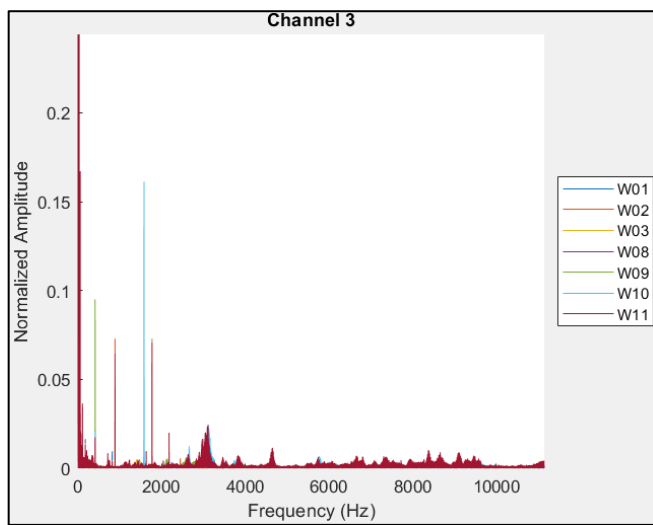


Figure 5.5 FFT plot of Accelerometer channel 3 Water type experiments

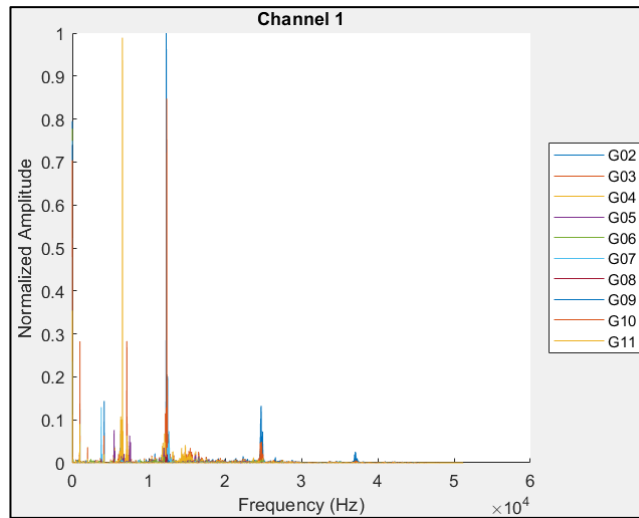


Figure 5.6 FFT plot of Accelerometer channel 1 Gas type experiments

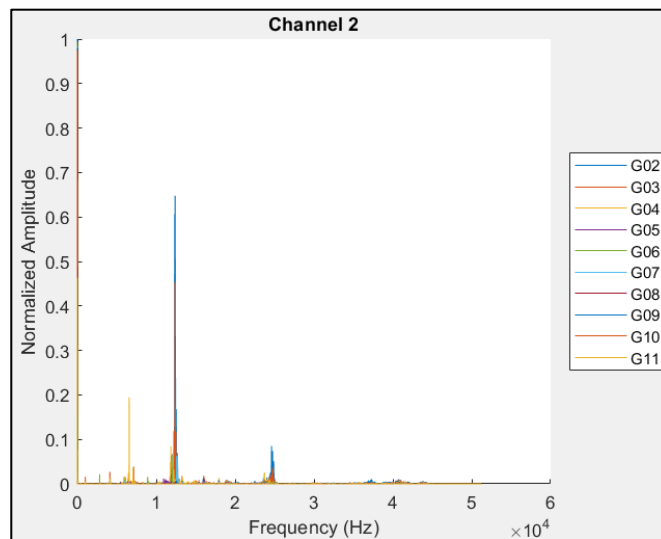


Figure 5.7 FFT plot of Accelerometer channel 2 Gas type experiments

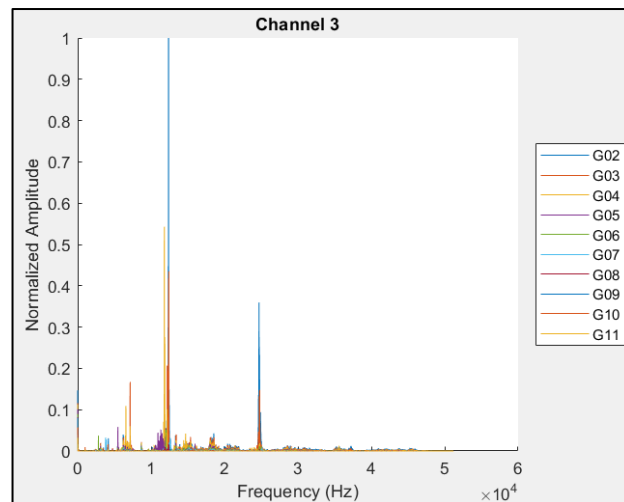


Figure 5.8 FFT plot of Accelerometer channel 3 Gas type experiments

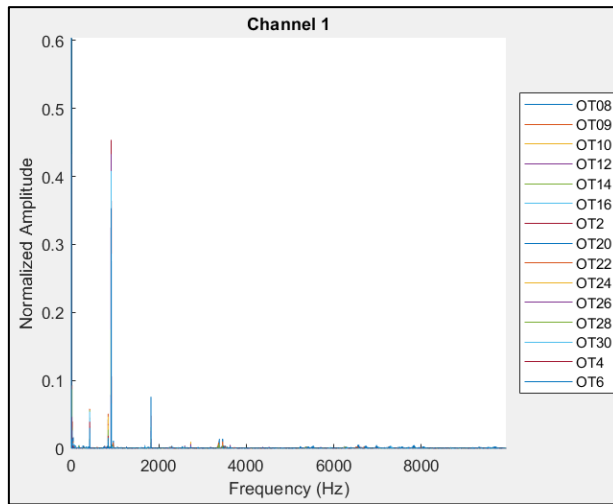


Figure 5.9 FFT plot of Accelerometer channel 1 Oil type experiments

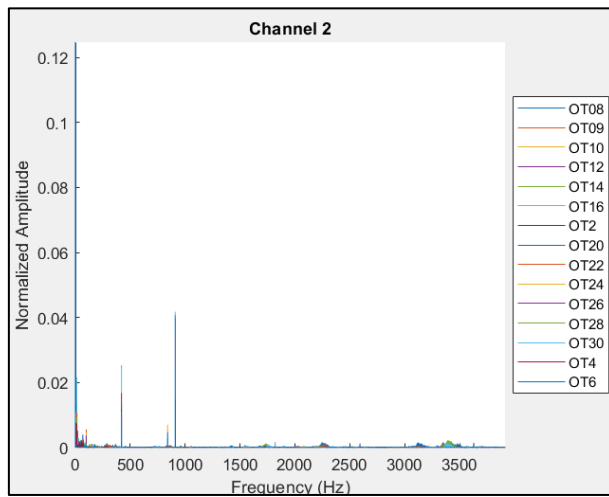


Figure 5.10 FFT plot of Accelerometer channel 2 Oil type experiments

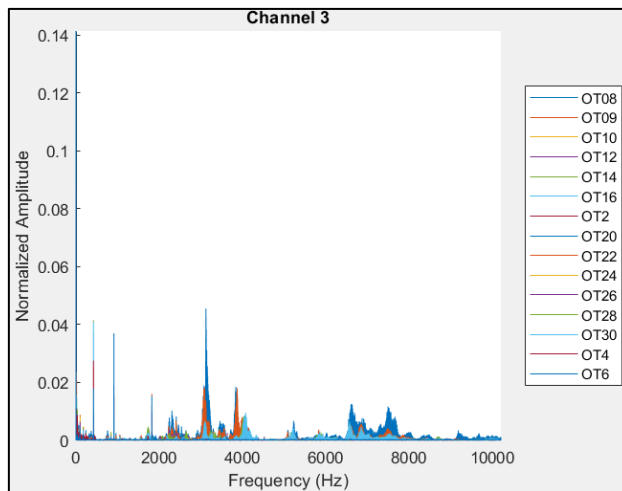


Figure 5.11 FFT plot of Accelerometer channel 3 Oil type experiments

Following observations can be made on basis of FFT plots.

- For Water experiments (“W”) dominant frequencies lies within range 0 to 2 kHz and below amplitude 0.5 (Refer Figure 5.3 to Figure 5.5).
- For Oil experiments (“OT”) dominant frequencies lies within range 0 to 5 kHz and below amplitude 0.5 (Refer Figure 5.9 to Figure 5.11).
- For Gas experiments (“G”) dominant frequencies lies within range 0 to 15 kHz and up to amplitude 1.0 (Refer Figure 5.6 to Figure 5.8).
- Accelerometer channel 2 is showing less amplitudes for each experiment as compared to other 2 channels.
- Accelerometer channel 3 is showing large noise levels especially in higher frequencies in liquid experiments like water and oil, most probably due to presence of Oil Choke Valve just before the channel 3.

5.3.3 Power Spectral density of vibration data

Analysis of vibration data is incomplete and mostly inaccurate without doing Power Spectral density (PSD) analysis since the nature of vibration in real world is random. The main reason why PSD is preferred over FFT is that these PSD plots are normalized to frequency bin width, preventing the duration of the data set from changing the amplitude of the result. This removes dependency over duration of an experiment and enables the developed system to give real time accurate analysis of accelerometer data. PSD plots are frequency (x-axis) vs dB/frequency. They show the power of frequency present in spectrum. **Pwelch()** MATLAB method is used to get PSD plots. Along with this, windowing parameters are also passed so as to smooth the signal by eliminating spectral leakages. The process of windowing a signal involves multiplying the time record by a smoothing window of finite length whose amplitude varies smoothly and gradually towards zero at the edges. The length, or time interval, of a smoothing window is defined in terms of number of samples. Multiplication in the time domain is equivalent to convolution in the frequency domain. Therefore, the spectrum of the windowed signal is a convolution of the spectrum of the original signal with the spectrum of the smoothing window. Windowing changes the shape of the signal in the time domain, as well as affecting the spectrum that you see.

Hanning Window :

Equation 5.3 [15]

$$w(\tau) = \begin{cases} 0.5(1 + \cos(\pi\tau/T)) & \text{for } |\tau| < T \\ 0 & \text{elsewhere} \end{cases} \quad (5.3)$$

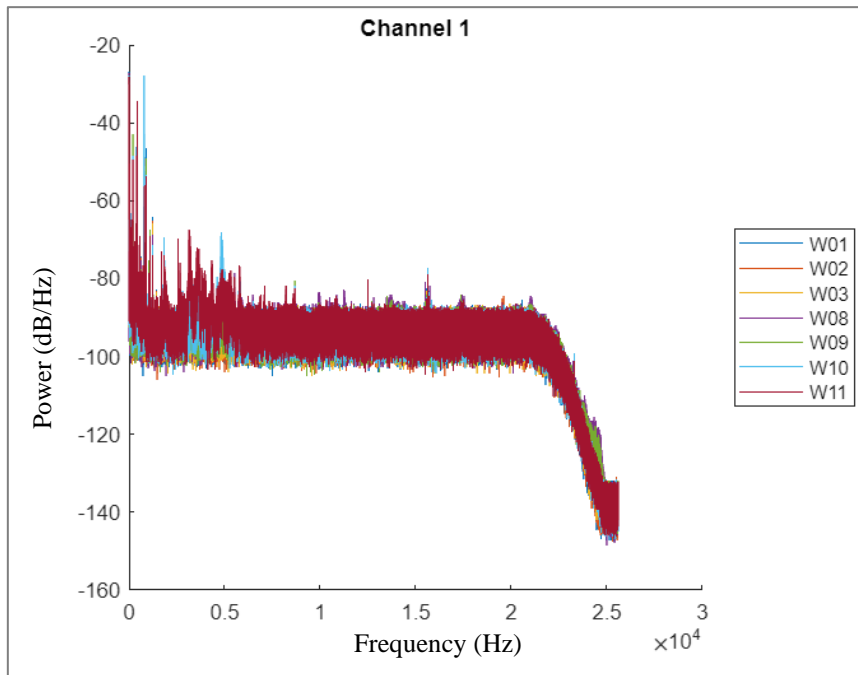


Figure 5.12 PSD plot of Accelerometer channel 1 Water type experiments (Without Hanning Window)

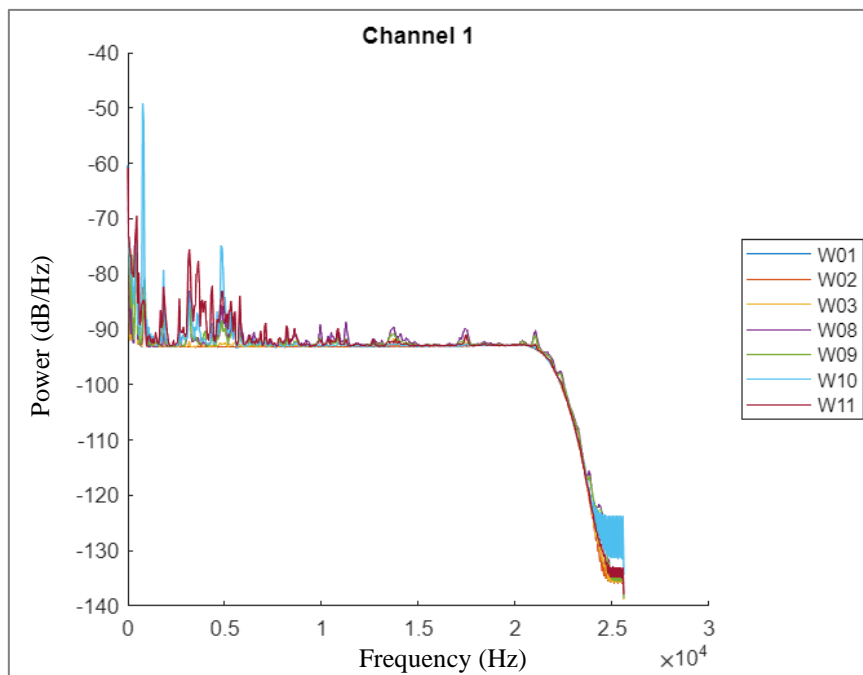


Figure 5.13 PSD plot of Accelerometer channel 1 Water type experiments (With Hanning Window)

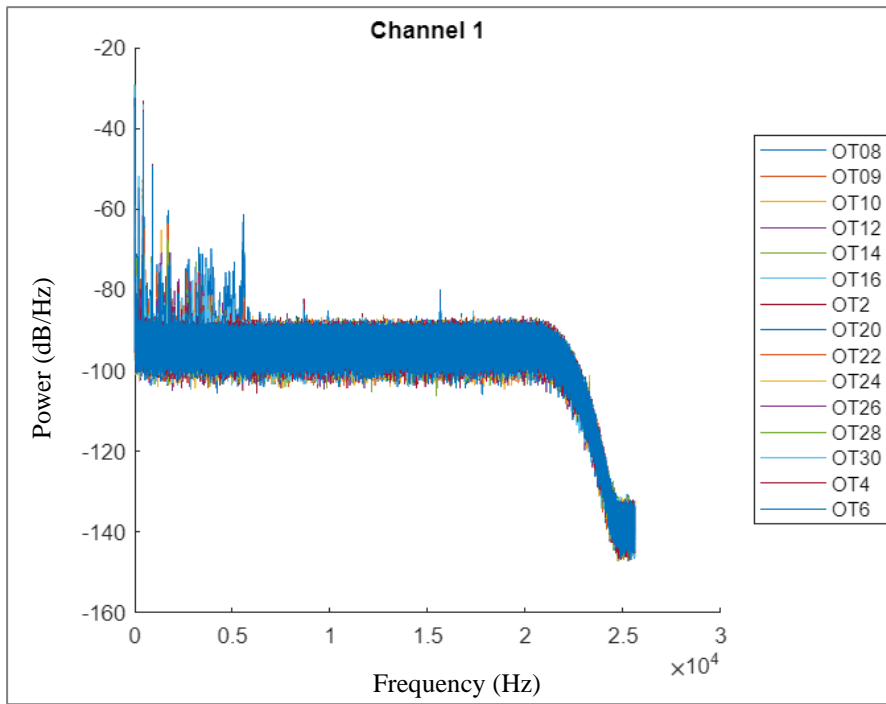


Figure 5.14 PSD plot of Accelerometer channel 1 Oil type experiments (Without Hanning Window)

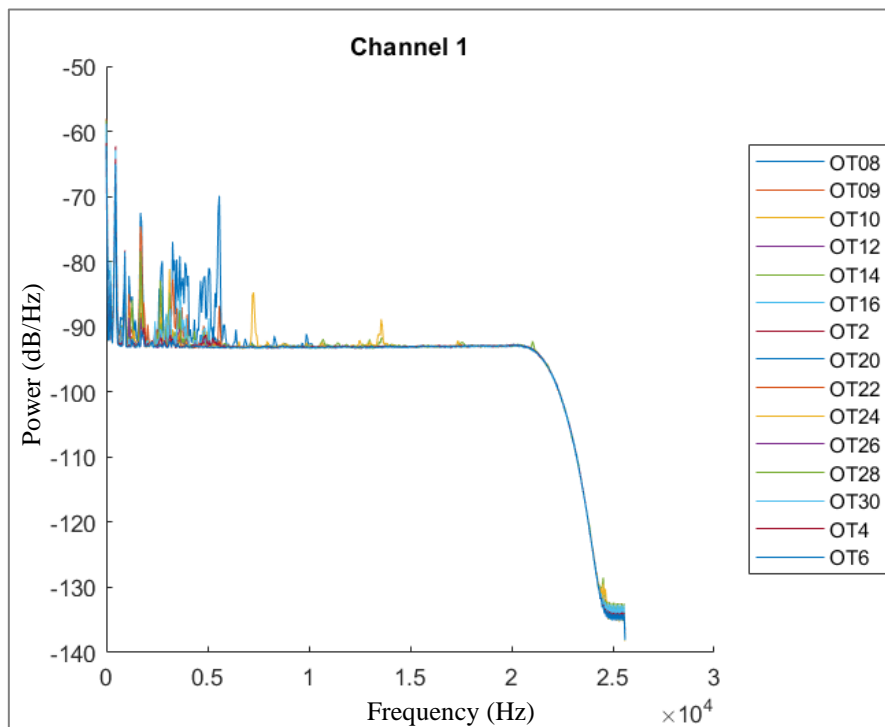


Figure 5.15 PSD plot of Accelerometer channel 1 Oil type experiments (With Hanning Window)

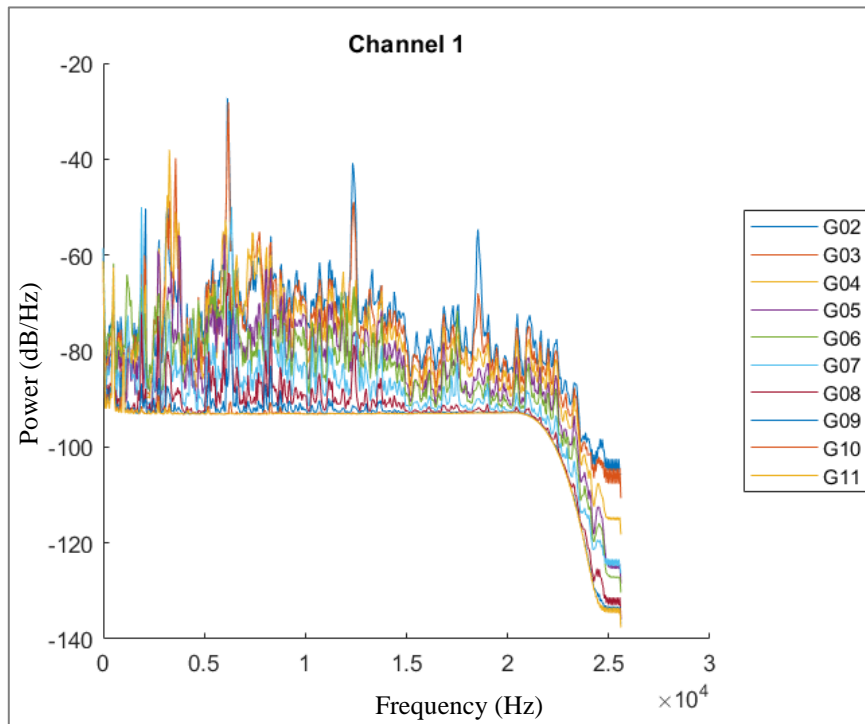


Figure 5.16 PSD plot of Accelerometer channel 1 Gas type experiments (With Hanning Window)

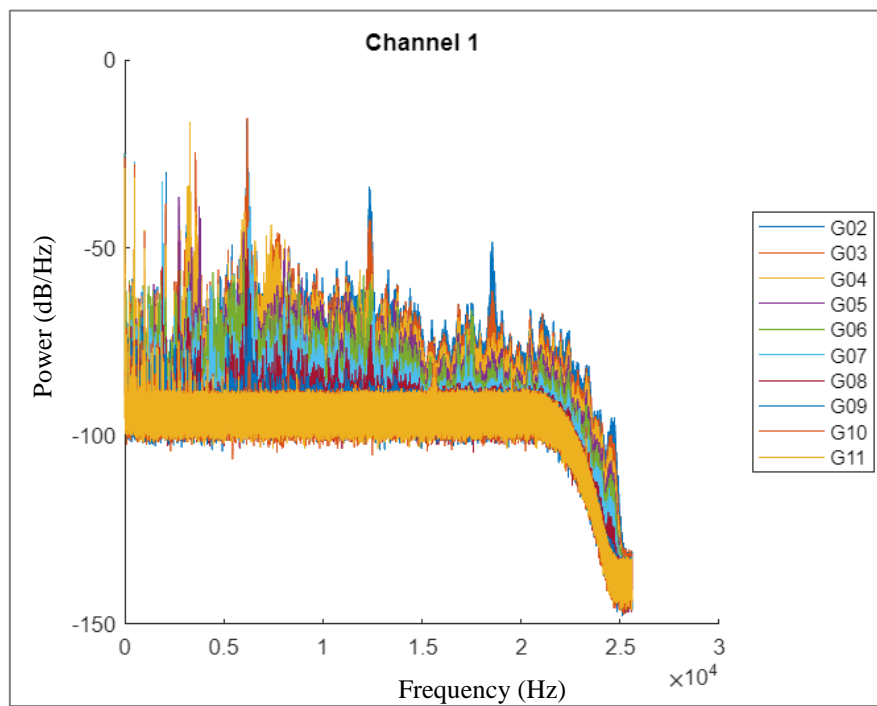


Figure 5.17 PSD plot of Accelerometer channel 1 Gas type experiments (Without Hanning Window)

5.3.4 Relative study of different flow types

In this section Power spectrum density of same flow rate i.e., $40 \text{ m}^3/\text{h}$ is analyzed as shown in Figure 5. Different vibration profile is observed for different flow type. This forms as a basis for classification model.

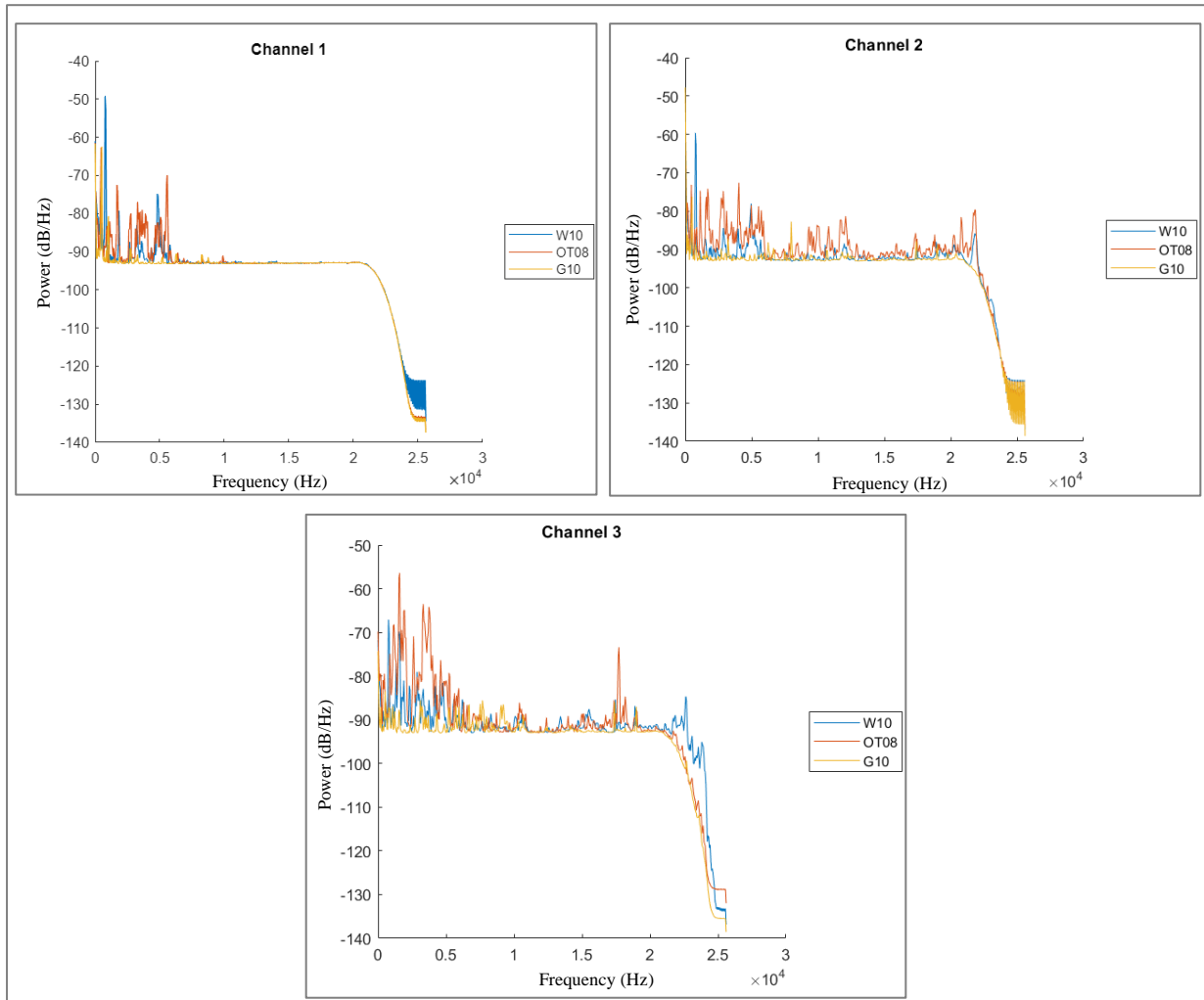


Figure 5.18 PSD plot of Accelerometer channel 1,2 and 3 for $40 \text{ m}^3/\text{h}$ flow rate (With Hanning Window)

6 Pre-Processing of Accelerometer Data

This chapter covers filtering of accelerometer data using observations obtained from the previous chapter. Then splitting of signal is done since the experiment is conducted for 10 – 15 minutes and for real time usage of machine learning models, it becomes necessary to train the models with data from few seconds time span.

6.1 Filtering of vibration signals

FFT plots covered in the previous chapter revealed dominating frequencies in Water, Oil and Gas flow experiments. Also, PSD plots revealed the intensity of these frequencies over the span of the complete experiment. The main vibration frequencies are located at the lower frequency range. This frequency range forms the basis for selecting design parameters of the filters. Hence from plots study and frequency response of accelerometer, a range of 10 Hz to 15 kHz is selected for designing a band-pass filter. The range for only water and oil experiments vibration data can be selected much less in order to get better resolution but since dominating frequencies in Gas experiments appear in the high frequency range, in order to cover all three flow types, a range of 10 Hz to 15 kHz is selected. Low frequency cut-off removed the frequency harmonics likely to originate from the experiment setup and high frequency cut-off removed the added noise since the sensitivity of the sensor changes above 15 kHz, which is likely to give unwanted noise above this frequency.

Fourth order band pass Butterworth filter of range 10 Hz to 15 kHz is selected to use to filter accelerometer sensor data. Butterworth filter is selected due to its maximally flat frequency response in the passband. This flat top characteristic is known to give very accurate amplitudes. Also, Butterworth filter is ripple free. In this thesis, a lower order filter is selected i.e., 4th order because high order filters tend to give sharper cutoff at both the edges and this can lead to loss of important data especially for Gas experiments whose dominant frequencies lie very close to 15 kHz.

6.2 Designing of filter

MATLAB filter designer app is used to design a filter of specifications shown in figure 6.1 below.

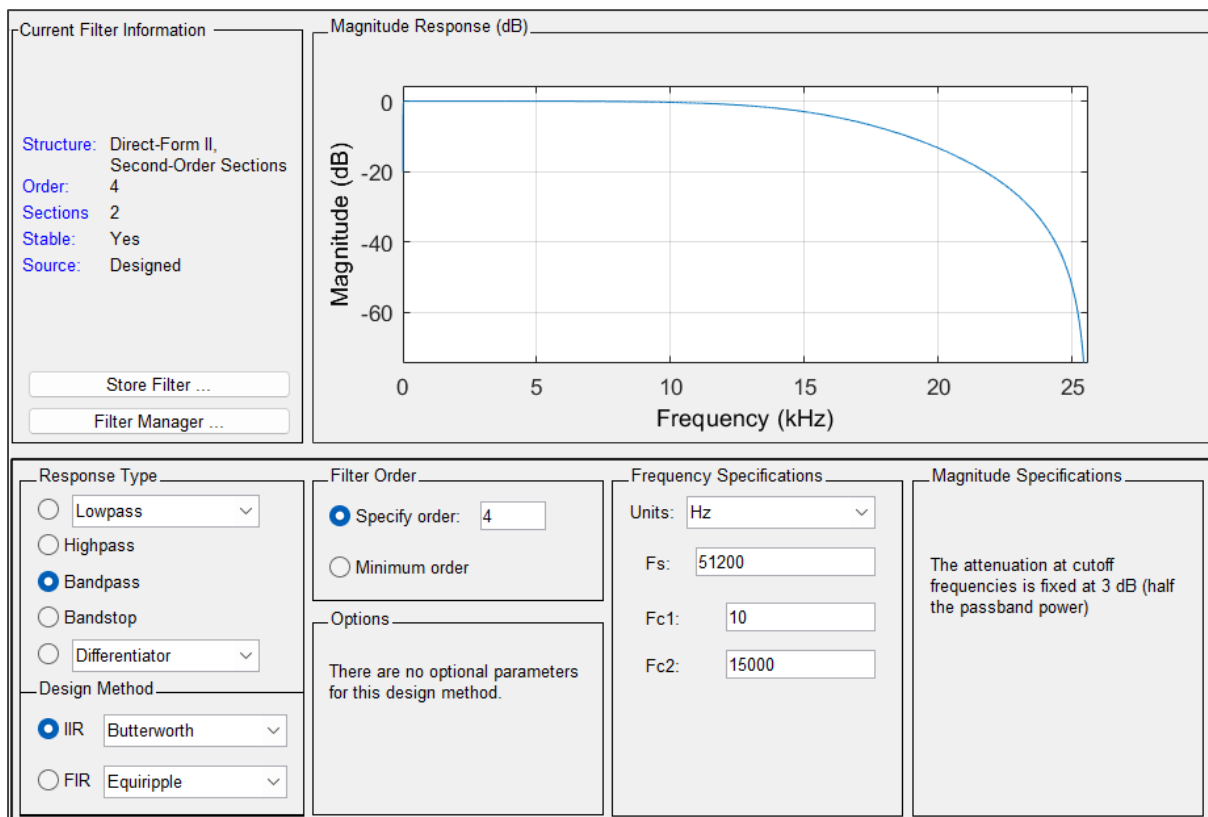


Figure 6.1 MATLAB filter design screen snip showing parameters

6.3 Filtered signal output

6.3.1 For Water flow experiments

It is observed that since filtering removed the effect of higher frequencies, dominant frequencies in lower range got visible, as its amplitude is increased and one such effect can be seen for W09 experiment (green line) visible in filtered output at 500 Hz.

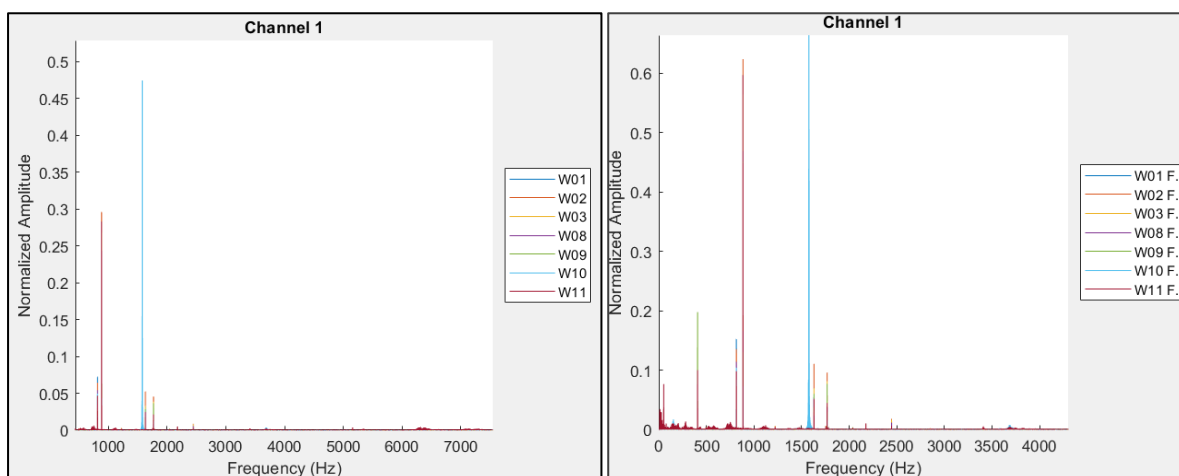


Figure 6.2 FFT plots of Water experiments (Unfiltered : Left) and (Filtered : Right)

6.3.2 For Gas flow experiments

For Gas flows, vibration profile is spread over the range so all the dominant frequencies are already visible with and without filter as shown in plots below. But what is observed is increase in amplitudes of dominant frequencies which can help in differentiating flow type and flow rates better due to increased visibility. This in turns make ML models more accurate.

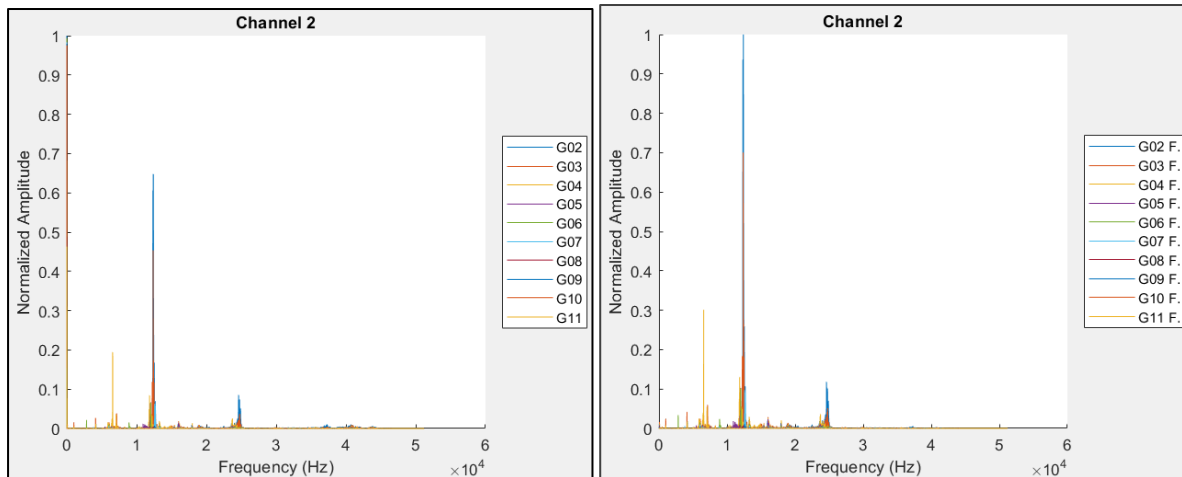


Figure 6.3 FFT plots of Gas experiments (Unfiltered : Left) and (Filtered : Right)

6.3.3 For Oil flow experiments

Filtering in this case revealed the dominant frequencies since their amplitudes are increased and also peaks for each experiment are now more clearly visible. This peak will act as one of the features for ML models. Looking at y-axis i.e., amplitude range, the peaks of each experiment are more clearly distinguishable, forming a basis for training ML models.

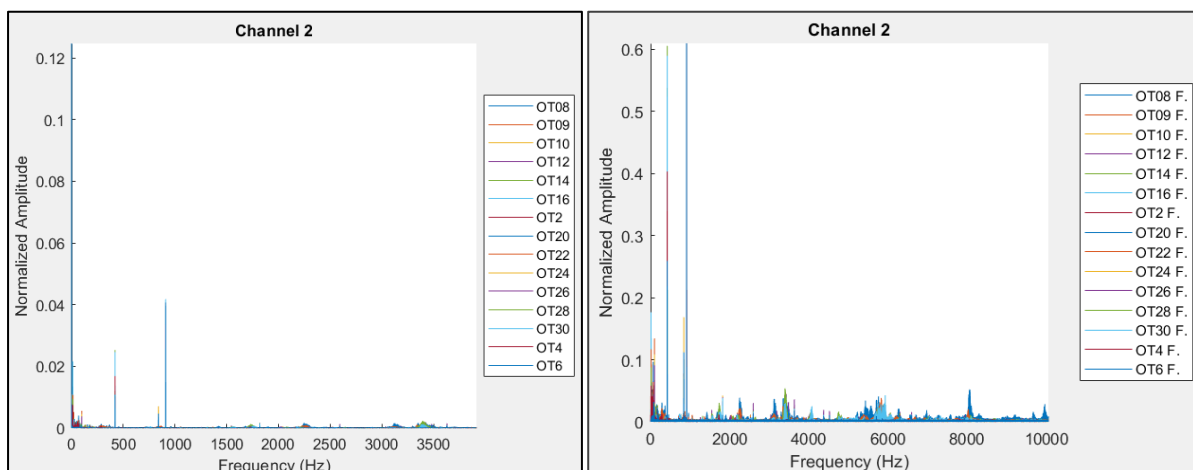


Figure 6.4 FFT plots of Oil experiments (Unfiltered : Left) and (Filtered : Right)

6.4 Splitting of filtered signal

At this point, filtered accelerometer signal from all 3 channels is available. But the signal for each experiment is over a timespan of around 10-15 minutes. In order to develop ML models which can classify and predict in real-time, it is necessary to split each signal in duration of

few seconds. Using the data of this split signal which is of duration of certain seconds is then used to train ML models. Usually, real-time systems give output immediately when an input is given to them but since this thesis is still on research level, to be on safe side, duration of 1 second is used for splitting signal. Based on the sampling frequency of 51.2 kHz, 1 second duration contains 51200 samples, which contains enough information of signal. To avoid loss of data due to split of signals, signals are being split with 50 % overlapping technique. To explain, consider plot shown in figure 6.5 below, showing filtered signal of first 200 samples of accelerometer channel 1 of an experiment G03. The split of 1 sec based on x – axis coordinates is just for demonstration in the figure.

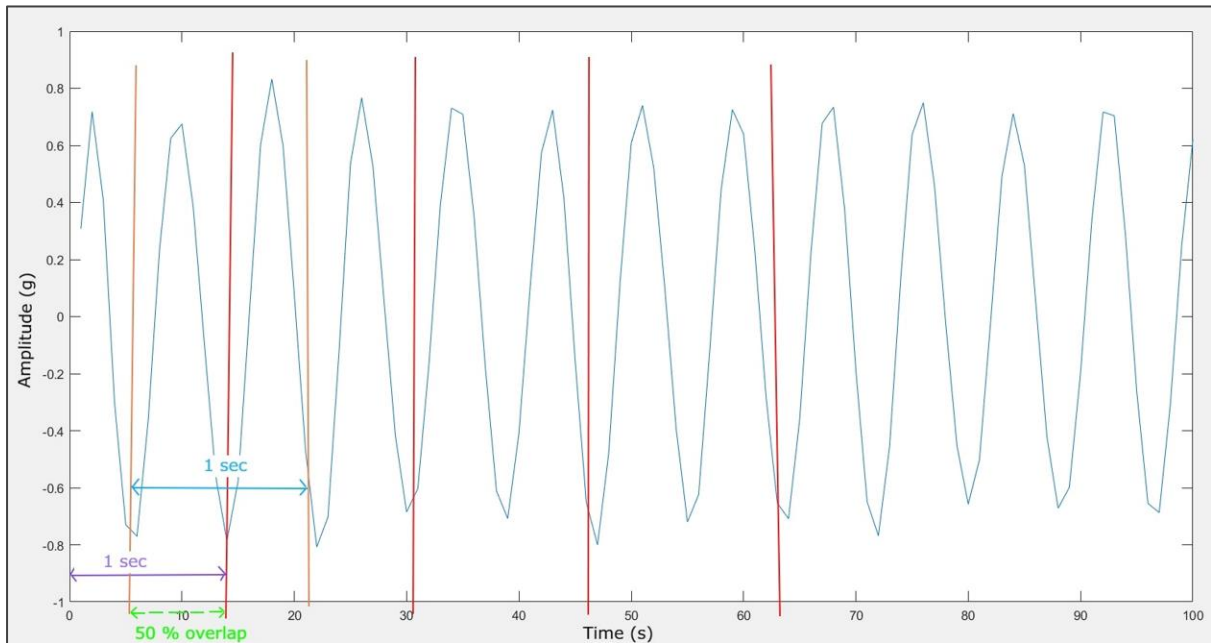


Figure 6.5 One second split of accelerometer channel 1 signal of experiment G03

6.5 Feature Engineering

The data at this point is filtered accelerometer signal of duration 1 second. Even though it is filtered, it is still a raw signal. This raw signal cannot be applied directly to machine learning models. Feature engineering is the process of transforming raw data into features that better represent the characteristics of raw data to machine learning models, resulting in improved model accuracy on unseen data. Better features mean increased flexibility and more open ML models. Wrong models will still give good results since they can pick up on good structure in data. But flexibility of good features will allow to use fewer complex models that run faster, easier to understand and easier to maintain. Selecting good features to develop less complex machine learning model is desirable in almost all ML related developments.

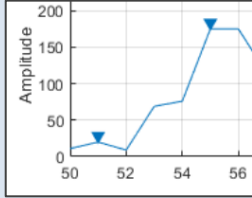
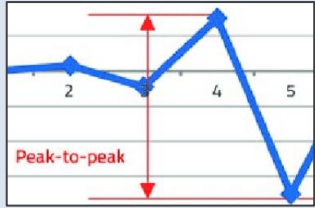
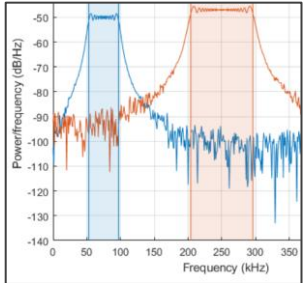
6.5.1 Accelerometer features

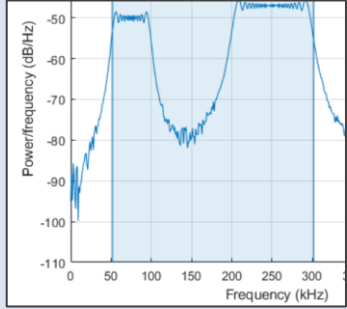
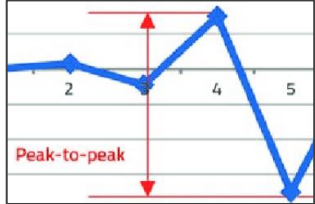
Feature functions which can be applied on 1 second vibration signal can be divided in to 3 categories.

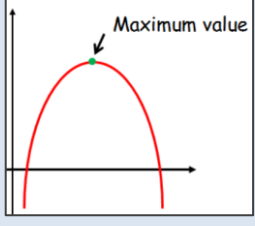

1. Basic Statistical features
2. Time-domain features
3. Frequency domain features

Following table gives brief review of the top features used on the signal.

Table 6.1: Features used on accelerometer signals

Sr No	Category	Feature Name	Definition	Formula / Derived
1	Frequency domain	Peak value 1, 2 and 3	Use of top 3 peaks from FFT of accelerometer signal	
2		State levels	High and Low level of signal	Using histogram : Lower : $i_{low} \leq i \leq \frac{1}{2}(i_{high} - i_{low})$ Higher : $i_{low} + \frac{1}{2}(i_{high} - i_{low}) \leq i \leq i_{high}$
3		Peak to peak	Difference between maximum positive and maximum negative amplitude	
4	Time Domain	Zero cross rate	Rate at which signal changes from positive to zero to negative or vice versa.	$zcr = \frac{1}{T-1} \sum_{t=1}^{T-1} 1_{\mathbb{R}_{<0}}(s_t s_{t-1})$
5		Spurious free dynamic range	Dynamic range between the fundamental tone and largest spur	SFDR = Amplitude of fundamental (dB) – amplitude of largest spur (dB)
6		Power Band-width	Difference between upper frequency and lower frequency where the response of both is 3 dB down	

7	Basic Statistic al	Occupied bandwidth	Bandwidth of the frequency band that contains a specified percentage of total power of signal	
8		Band power	Average power in accelerometer signal	$P_{[\omega_1, \omega_2]} = \frac{1}{2\pi} \int_{\omega_1}^{\omega_2} [S(\omega) + S(-\omega)] d\omega$
9		Peak to RMS	Ratio of largest value in signal to root-mean-square value of that signal	$\frac{\ X\ _{\infty}}{\sqrt{\frac{1}{N} \sum_{n=1}^N X_n ^2}}$
10		RSSQ	Root Sum of Squares level of signal	$x_{RSS} = \sqrt{\sum_{n=1}^N x_n ^2}$
11		RMS	Square root of average of squared value of signal	$x_{RMS} = \sqrt{\frac{1}{\tau} \int_0^{\tau} x^2(t) dt}$
12		Peak to peak	Difference between maximum and minimum values	
13		Median frequency	Represents the midpoint of power distribution of signal	$\text{Median} = 1 + \left[\frac{\frac{n}{2} - c}{f} \right] \times h$
14		Mean frequency	Mean frequency of power spectrum	$f_{\text{mean}} = \frac{\sum_{i=0}^n I_i \cdot f_i}{\sum_{i=0}^n I_i}$
15		State levels	High and Low level of signal	Using histogram : Lower : $i_{\text{low}} \leq i \leq \frac{1}{2}(i_{\text{high}} - i_{\text{low}})$ Higher : $i_{\text{low}} + \frac{1}{2}(i_{\text{high}} - i_{\text{low}}) \leq i \leq i_{\text{high}}$
16		Standard Deviation	Measure of how far the signal fluctuates from mean	$S = \sqrt{\frac{1}{N-1} \sum_{i=1}^N x_i - \mu ^2}$

17		Max	Largest value in signal vector	
18		Range	difference between the maximum and minimum values in signal vector	
19		Interquartile range	Spread of the values in signal calculated on basis of lower and higher quartile	Lower quartile = median of smallest values Higher quartile = median of largest values
20		mean	Mean of time series of signal	$\mu_x = \frac{1}{N}(x(1) + x(2) + \dots + x(N))$

The details of symbols mentioned in equations [23] in table 6.2 is as following:

l_{low} = lowest-indexed histogram

l_{high} = highest-indexed histogram

S = signal of length T

$1_{\mathbb{R}<0}$ = indicator function

$S(\omega)$ = power spectral density

$[\omega_1, \omega_2]$ = band limits

X = signal vector (1 sec signal in time-series form)

τ = signal length

n = number of frequencies

c = cumulative frequency preceding to the median class frequency

h = width of the class interval

μ = weighted mean of x

6.6 Feature Dataset

All the features engineering for the features mentioned in table 6.1 is performed on 1 second split signal of all three accelerometers channel. MATLAB inbuilt functions are used for the same. The output of each feature function is then stored in newly created column in existing dataset. Column name in the dataset is kept similar to function name used and corresponding channel. Dataset generated by feature extraction on all three accelerometers is shown in figure 6.6.

6.7 Normalization of dataset

Normalizing is done in categories i.e.; it is done based on the type of variable under consideration. For example, accelerometers features are not normalized with other variables like temperature and pressure. Same features like meanfreq_1 and meanfreq_2 is normalized together to not lose their spatial relationship. Let's consider an example using values to explain further:

For example, imagine these values for meanfreq_1 and meanfreq_2. Note that first three elements are the same.

```
meanfreq_1_example = [1 2 3 3 5 6 2 2];
```

```
meanfreq_2_example = [1 2 3 9 11 12 14];
```

If they are normalized separately, output is the different values for first three elements although they have the same unit and magnitude:

```
normalize(meanfreq_1_example)
```

```
ans = 1×8
```

```
-1.1832 -0.5916 0 0 1.1832 1.7748 -0.5916 -0.5916
```

```
normalize(meanfreq_2_example)
```

```
ans = 1×7
```

```
-1.2087 -1.0207 -0.8327 0.2955 0.6715 0.8595 1.2356
```

To solve this, signal features are combined in one vector, normalize that vector, and then split it back into 4 features. Continuing with the example:

```
meanfreq_all = [meanfreq_1_example, meanfreq_2_example]
```

```
meanfreq_all = 1×15
```

```
1 2 3 3 5 6 2 2 1 2 3 9 11 12 14
```

```
meanfreq_all_normalized = normalize(meanfreq_all);
```

```
meanfreq_1_normalized = meanfreq_all_normalized(1:8)
```

```
meanfreq_1_normalized = 1×8
```

```
-0.9384 -0.7076 -0.4769 -0.4769 -0.0154 0.2154 -0.7076 -0.7076
```

```
meanfreq_2_normalized = meanfreq_all_normalized(9:end)
```

```
meanfreq_2_normalized = 1×7
```

```
-0.9384 -0.7076 -0.4769 0.9076 1.3691 1.5999 2.0614
```

Now, same normalized values for the first three elements is obtained.

6.7.1 Adding de-normalizing capability

While performing normalization on the dataset, the corresponding mu and sigma value of each variable is stored in separate variable named 'normalization'. This variable can be later used to de-normalize the dataset for further analysis. Also, this data from normalization can be used while trying to use completely new data for this thesis.

6.8 Final Dataset for ML models

Since one of the machine learning models is for classifying flow type based on Water, Oil and Gas, extra column named 'category' is added for each row and corresponding alphabet is added in that row for each experiment i.e., G for Gas, OT for Oil and W for Water. Each experiment is conducted for 10 to 15 minutes and in total 32 experiments were performed by Equinor. Initially there is only one row per each experiment but splitting accelerometer signal in duration of 1 second for each experiments caused 1000 rows for each experiment. So final dataset is of table : 16,680 Rows and 114 columns. For machine learning purpose, two datasets are required i.e., training dataset and test data. While there is in-built mechanism in MATLAB to randomly separate training and test data, to make the models more robust and to justify it better, manually training data and test data are separated. So, 6 experiments, 2 of each flow type are kept totally separated from training of Machine learning models in next section and test results are entirely from experiments not at all included in training data set. Following Table 6.2 illustrates manually separated training and test data.

Table 6.2: Manually separated training and test data

Training Data						Test Data	
Experiment Name	Flow Rate (m3/h)	Experiment Name	Flow Rate (m3/h)	Experiment Name	Flow Rate (m3/h)	Experiment Name	Flow Rate (m3/h)
Water Type		OT8	8.0	Gas Type		G04	160.0
W01	2.0	OT28	28.0	G11	30.0	G06	120.0
W02	5.0	OT26	26.0	G02	200.0	OT09	30.0
W08	20.0	OT24	24.0	G05	140.0	OT22	22.0
W10	40.0	OT6	6.0	G03	180.0	W03	10.0
W11	50.0	OT20	20.0	G07	100.0	W09	30.0
Oil Type		OT18	18.0	G08	80.0		
OT4	4.0	OT16	16.0	G09	60.0		
OT2	2.0	OT14	14.0	G10	40.0		
OT08	40.0	OT12	12.0				
OT10	20.0	OT30	30.0				

6.8.1 Tabular format of training and test data set

Separating the dataset caused following sizes:

- Training dataset : 14860 Rows x 114 Columns
- Test dataset : 2000 Rows x 114 Columns

7 Classification Model

For the purpose of estimating flow velocity in these experiments, it is needed to also identify what is the type of material that is flowing. Since this is part of multiphase flow meters in oil & gas applications, the type of material flowing can be anything from Oil, Gas, Water or a combination of any two or three. Since the main focus here is single phase flow analysis, the estimation will be of only Gas, Oil or Water. This chapter covers development of classification model in order to predict flow type i.e., Oil, Water or Gas based on accelerometer channel input.

7.1 Basics of Machine Learning

Machine learning can be briefly defined as a system of computer algorithms that are initially programmed using historical inputs and corresponding outputs. So, these algorithms can predict new output values when similar type of inputs are given to them. Like humans, ML applications learn from experiences without new for direct programming. Machine Learning is complex, which is why it has been divided into two primary areas, supervised learning and unsupervised learning. Each one has a specific purpose and action, yielding results and utilizing various forms of data.

In this thesis, since the data is known, supervised learning approach is used for classification and prediction models.

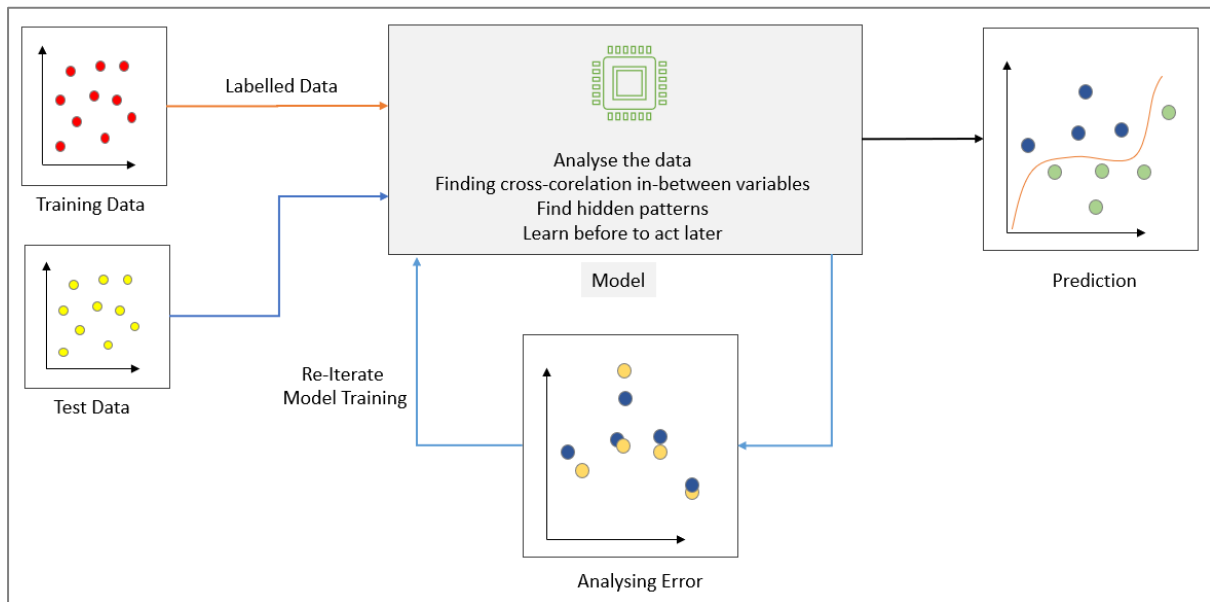


Figure 7.1 Basic Machine Learning Diagram

7.1.1 Common Terminology

This section covers common terms used in machine learning application.

Regression :

A method that attempts to determine the strength and character of the relationships between one dependent variable and series of other variables. Mostly commonly used regression techniques are Linear regression and logistic regression.

Mean Squared Error (MSE) :

Average of squared differences between predicted and actual output. This is usually used to showcase the performance of ML model developed and compare different types of models.

Confusion Matrix :

A table which defines the performance of a classification algorithm. It visualizes and summarizes the performance of a classification algorithm. Basically, it shows how correctly the inputs in test data is classified in desired category. Higher the percentage, higher is the accuracy of that model.

True Positive Rates (TPR) : Unlike the false alarm situation encountered in our day to day lives, true positive is an outcome where the model correctly predicts the positive class. They are the actual positives which are correctly identified.

Receiver Operating Characteristic (ROC) curve :

It is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The method was originally developed for operators of military radar receivers starting in 1941, which led to its name [16].

Area Under the Curve (AUC):

It is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. Higher the value of AUC i.e., as close as possible to 1 or 1, the better the performance of model to distinguish between positive and negative classes.

7.2 Algorithms Explained

There are many algorithms being used in machine learning applications and many new are being developed. But some basic algorithms which can serve as basis for this study are used here and only that algorithms are explained in this section.

7.2.1 Linear Discriminant Analysis

It is a classification method that projects high-dimensional data onto a line and performs classification in this one-dimensional space. The projection maximizes the distance between the means of the two classes while minimizing the variance within each class. Each variable in the data is shaped in the form of a bell curve when plotted i.e., Gaussian. The values of each variable vary around the mean by the same amount on the average i.e., each attribute has the same variance. Figure 7.2 shows basic illustration [17] of LDA approach.

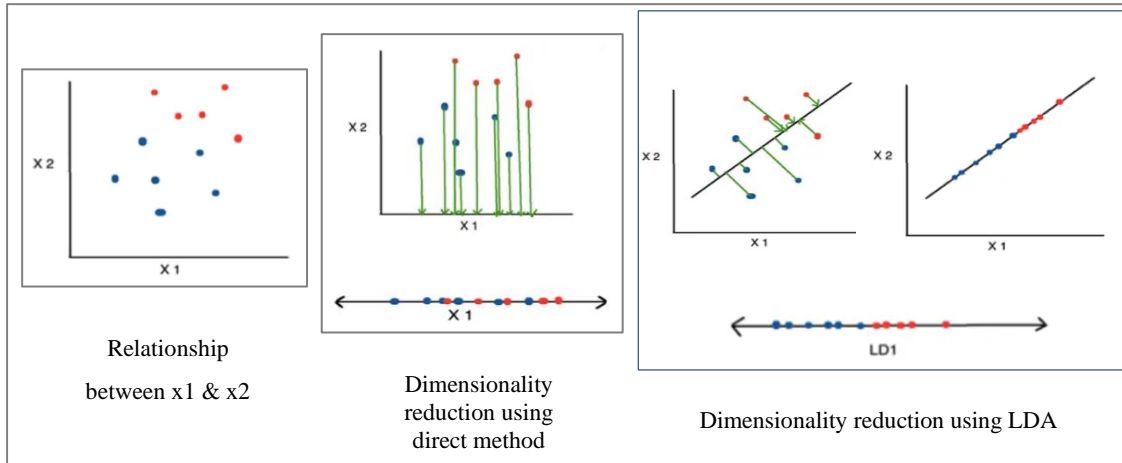


Figure 7.2 Linear Discriminant Analysis illustration

7.2.2 Naive Bayes

It is a classification method based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem states the following relationship [18], given class variable y and dependent feature vector x_1 through x_n :

$$P(y | x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n | y)}{P(x_1, \dots, x_n)} \quad (7.1)$$

for all , this relationship is simplified to

$$P(y | x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i | y)}{P(x_1, \dots, x_n)} \quad (7.2)$$

Since $P(x_1, \dots, x_n)$ is constant given the input, we can use the following classification rule:

$$\begin{aligned}
 P(y | x_1, \dots, x_n) &\propto P(y) \prod_{i=1}^n P(x_i | y) \\
 \Downarrow \\
 \hat{y} &= \arg \max_y P(y) \prod_{i=1}^n P(x_i | y)
 \end{aligned} \quad (7.3)$$

7.2.3 Support Vector Machine (SVM)

It is one of the most robust classification models developed at AT&T Bell Laboratories by Vladimir Vapnik [19]. It creates a hyperplane which acts as a border between positive and negative class and the data is classified based on the position in relation to this border.

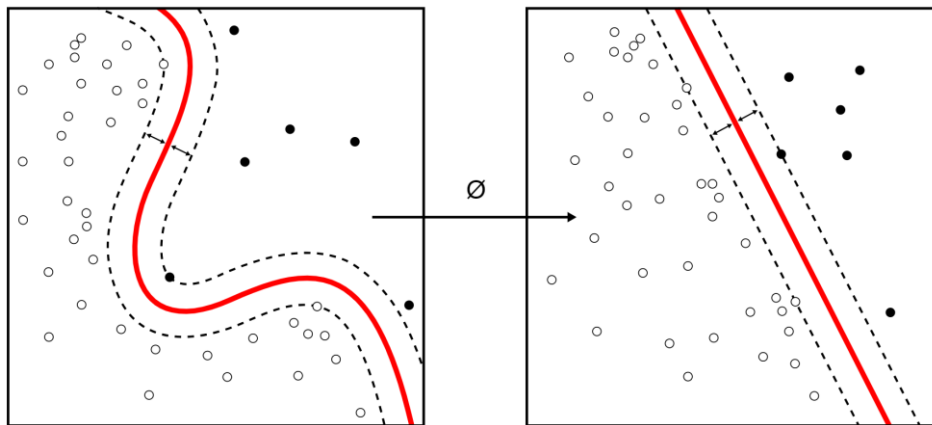


Figure 7.3 Support Vector Machine plot [19]

7.2.4 K-Nearest Neighbour (KNN)

It is a classification model where object is classified by the plurality vote of its neighbours with the most being assigned to the class most common among its k nearest neighbors. Consider a data point in n dimensional space which is defined by n features. This algorithm calculates the distance between one point to another and then assign the label of unobserved data based on the labels of nearest observed data points.

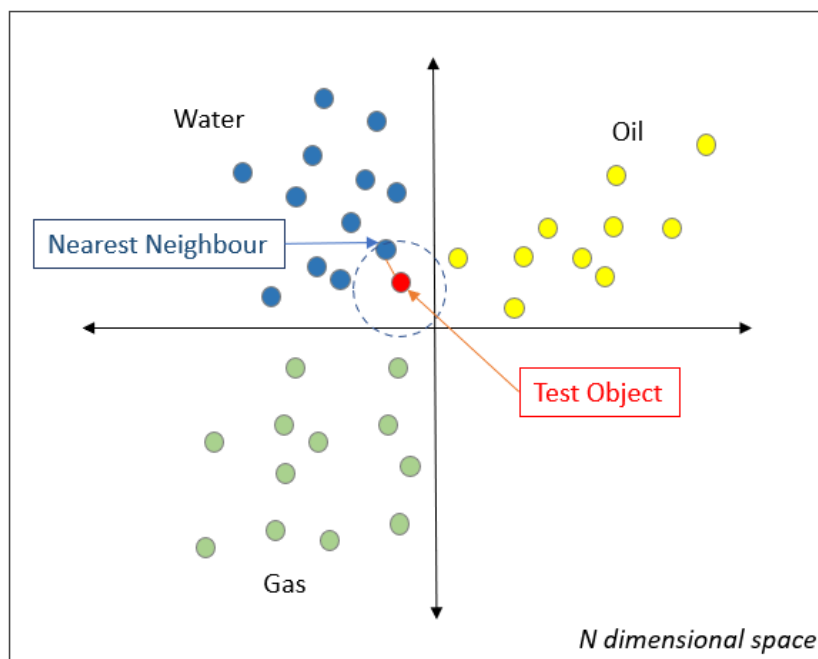


Figure 7.4 Illustration of KNN classification algorithm

7.2.5 Gaussian Processes (GP)

These are the generalization of gaussian probability distribution. Whereas a probability distribution describes random variables which are scalars or vectors (for multivariate distributions), a stochastic process governs the properties of functions. Leaving mathematical sophistication aside, one can loosely think of a function as a very long vector, each entry in the vector specifying the function value $f(x)$ at a particular input x [20].

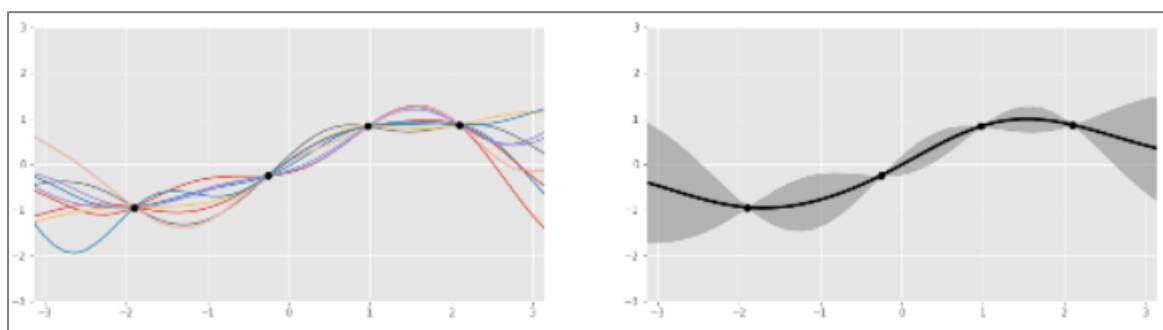


Figure 7.5 Illustration of Gaussian Probability Function [21]

On the left in figure 7.5, each line is a sample from the distribution of functions and each feature as an input to model is reflected in the wide range of possible functions and diverse function shapes on display. Sampling from Gaussian process is like getting outputs of unknown function at various points as shown in right side in figure 7.5.

7.2.6 Ensemble Methods

It is a machine learning technique that combines several base models in order to produce one optimal predictive model. A Decision Tree determines the predictive value based on series of questions and conditions. Rather than just relying on one Decision Tree and hoping to make the right decision at each split, Ensemble Methods takes a sample of Decision Trees into account, calculate which features to use or questions to ask at each split, and make a final predictor based on the aggregated results of the sampled Decision Trees. The three main classes of ensemble learning methods are bagging, stacking, and boosting [22].

- Bagging : Fitting many decision trees on different samples of the same dataset and averaging the predictions.
- Stacking : Fitting many different models' types on the same data and using another model to learn how to best combine the predictions.
- Boosting : Adding ensemble members sequentially that correct the predictions made by prior models and outputs a weighted average of the predictions.

Popular Bagging ensemble algorithms are Random Forest, Bagged Decision Trees and Extra Trees. Since bagging algorithm is used in this thesis, its structure is shown in figure 7.6.

7.2.7 Neural Network

These systems are inspired by biological neural networks that constitute animal brains. So, it is a collection of connected nodes called artificial neurons. The signal at the connection is a real number and the output of each neuron is computed by some-nonlinear function of sum of its input. Each node has an associated weight and threshold and changes based on learning due to past inputs. The layers of functions between the input and the output are what make up the

neural network. In practice, the neural network is slightly more complicated than the figure 7.7 shown below.

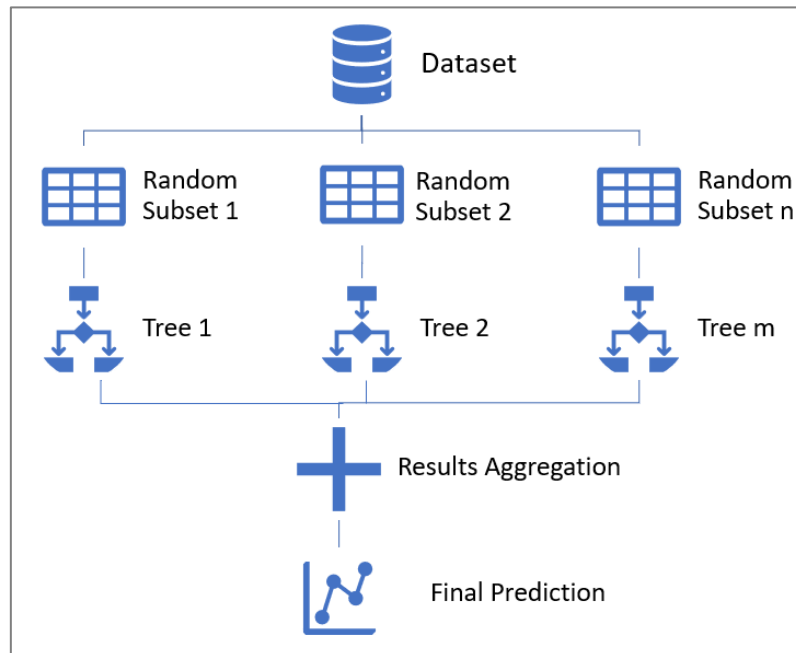


Figure 7.6 Structure of Bagged Ensemble Algorithm

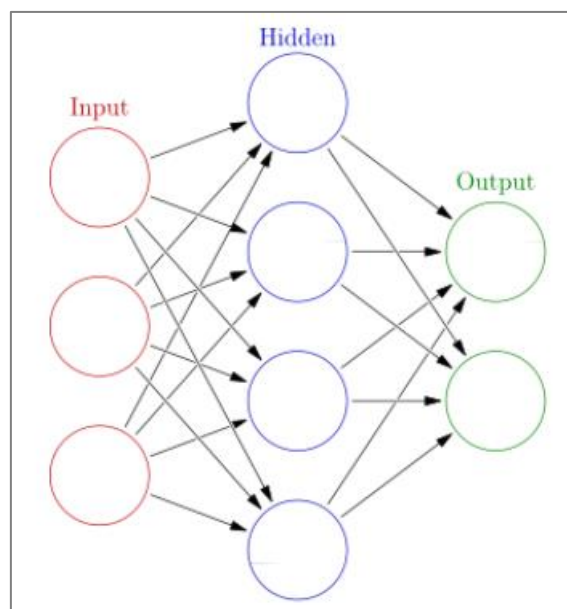


Figure 7.7 Simple Structure of Neural Network

7.3 Flow type classification model

Classification is a process of categorizing a given set of data into classes. Over 100 inputs for single row are present in dataset. But to maintain the focus of this thesis on accelerometer signals, only accelerometer channel inputs are used for training and testing these models. Selecting all 25 features of channel 1 gave testing accuracy of 99%. So, to stretch models a bit more and to limit the input data to just top features, only 3 features of just 1 channel is used further in this thesis. MATLAB classification learner app is used for training and testing using various classification algorithms and the model accuracy with total cost is mentioned in table 7.1 below.

Table 7.1: Different classification model performance

	Inputs Accelerometer channel 1 (Numbers) : Feature name	Algorithm	Test Accuracy (%)	Confusion Matrix True Positive Rates (%)		
				Gas	Oil	Water
1	(25) All features	Linear Discriminant	91.5	85.5	96.5	99.8
2		Naive Bayes	84.9	95.1	99.0	63.3
3		SVM	98.4	97.2	99.5	100
4		KNN	99.2	98.6	100	99.9
5		Neural Network	97.2	95.1	99.5	100
6	(3) Median frequency Mean frequency Zero cross-rate	Linear Discriminant	98.9	98.3	100	99.5
7		Naive Bayes	98.9	99.7	99.2	97.6
8		SVM	97.0	95.5	98.5	99
9		KNN	93.6	88.6	100	99.7
10		Neural Network	93.6	89	98	99.8
11	(2) Median frequency Peak value 1	SVM	78.0	92	7.8	78
12		KNN	98.2	97.6	98.8	99.2
13		Neural Network	79.7	96.7	1	77.8
14	(1) Median frequency	SVM	87.8	99.9	0	96.8
15		KNN	97.2	95.8	99.2	98.8
16		Neural Network	84.6	96.8	0	92.6
17	(2) Median State levels (low & high)	SVM	67.4	99.7	100	2.6
18		KNN	98.2	96.9	99.5	99.9
19		Neural Network	95.2	96.8	73.0	100

Looking at the results of different models along with different features, the best model for classification of flow type is found to be :

- KNN model with 3 inputs i.e., Median frequency, state levels low and state levels high. Here one feature is time domain feature i.e., median frequency and another one is frequency domain feature i.e., state levels.

7.3.1 KNN Model

Model Hyperparameters :

- Preset : Fine KNN
- Number of neighbors : 1
- Distance metric : Euclidean
- Distance weight : Equal
- Standardize data : false

PCA : Disabled

Features : Median frequency, state levels low and state levels high

Table 7.2: KNN model performance

Training Results		Test Results	
Accuracy (Validation)	99.8%	Accuracy	98.2 %
Total Cost (Validation)	31	Total Cost	66
Prediction Speed	~62000 obs/sec		
Training Time	7.3088 sec		

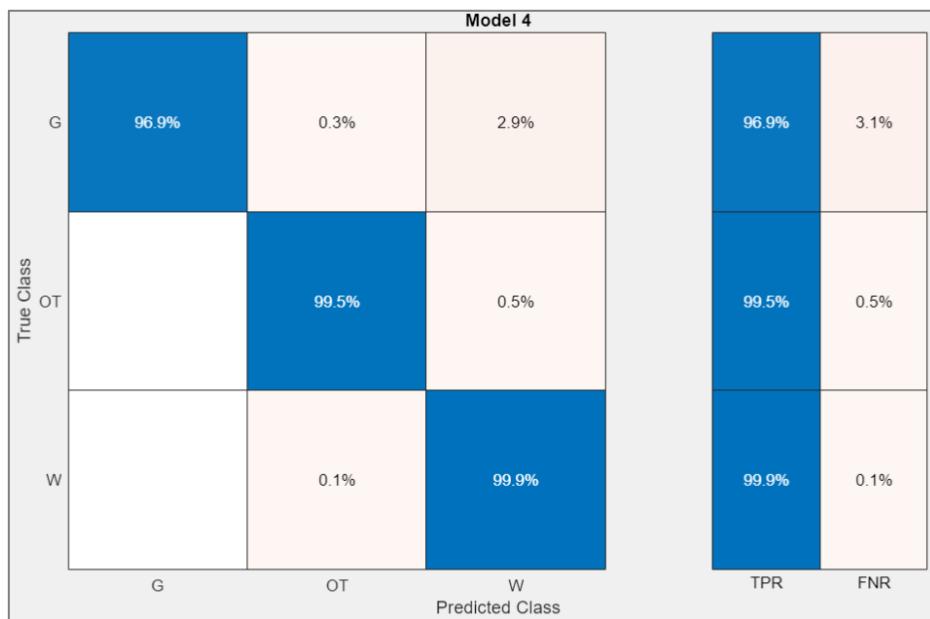


Figure 7.8 Test Confusion Matrix of Fine KNN model

7.3.2 SVM Model

Model Hyperparameters :

- Preset : Linear SVM
- Kernel function : Linear
- Kernel Scale : Automatic
- Box Constraint level : 1
- Multiclass method : One-vs-One
- Standardize data : false

PCA : Disabled

Features : Median frequency, Mean frequency and Zero cross-rate

Table 7.3: Linear SVM model performance

Training Results		Test Results	
Accuracy (Validation)	91.3%	Accuracy	97 %
Total Cost (Validation)	1141	Total Cost	108
Prediction Speed	~220000 obs/sec		
Training Time	47.969 sec		



Figure 7.9 Test Confusion Matrix of Linear SVM model

8 Flow rate Regression Model

Machine learning algorithms are described as learning a target function (f) that best maps input variables (x) to an output variable (y): $y = f(x)$. This is a general learning task to make predictions in the future (y) given new input variables (x). In this scope, input variables are features of accelerometer channels and output i.e., to be predicted variable is flow rate. Following table gives overview of performance of different models with different inputs.

Table 8.1: Different Prediction Model Performance

Sr No	Input (Accelerometer)		Algorithm	RMSE (Test)	R-Squared (Test)
	Channel	Features			
1	1	26	SVM	18.995	0.90
2		7	GP Regression	15.713	0.94
3		26	Neural Network	12.877	0.96
4		5	GP Regression	13.846	0.95
5		7	Ensemble Bagged	13.35	0.95
6	2	26	SVM	32.389	0.72
7		26	GP Regression	9.113	0.98
8		26	Neural Network	11.162	0.97
9		4	GP Regression	10.207	0.97
10		4	Neural Network	13.454	0.95
11	3	26	SVM	17.917	0.92
12		26	GP Regression	17.426	0.92
13		26	Neural Network	17.323	0.92
14	1, 2, 3	76	Linear Regression	11.701	0.96
15		76	SVM	12.016	0.96
16		8	Neural Network	16.41	0.93
17		8	Ensemble Bagged	12.739	0.96
18	1,2	51	GP Regression	8.756	0.98
19		51	SVM	9.2741	0.98
20		13	Ensemble Bagged	12.717	0.96
21		7	Ensemble Bagged	14.263	0.95

Following Section covers the details of model with lowest RMSE and based on accelerometer channels.

8.1.1 Accelerometer Channel 1 GP Model

Model Hyperparameters :

- Preset : Exponential GPR
- Basis function : Constant
- Kernel function : Exponential
- Use isotopic kernel : true
- Kernel Scale : Automatic
- Signal Standard Deviation : Automatic
- Sigma : Automatic
- Standardize data : false
- Optimize numeric parameters : true

PCA : Disabled

Features :

- Median Frequency
- Category
- Peak to RMS
- Peak value 1
- Inter quartile range

Table 8.2: Accelerometer Channel 1 GP model performance

Training Results		Test Results	
RMSE (Validation)	5.01	RMSE	13.84
MSE (Validation)	25.19	R-Squared	0.95
Prediction Speed	~10,000 obs/sec	MSE (Test)	191.7
Training Time	255.9 sec		

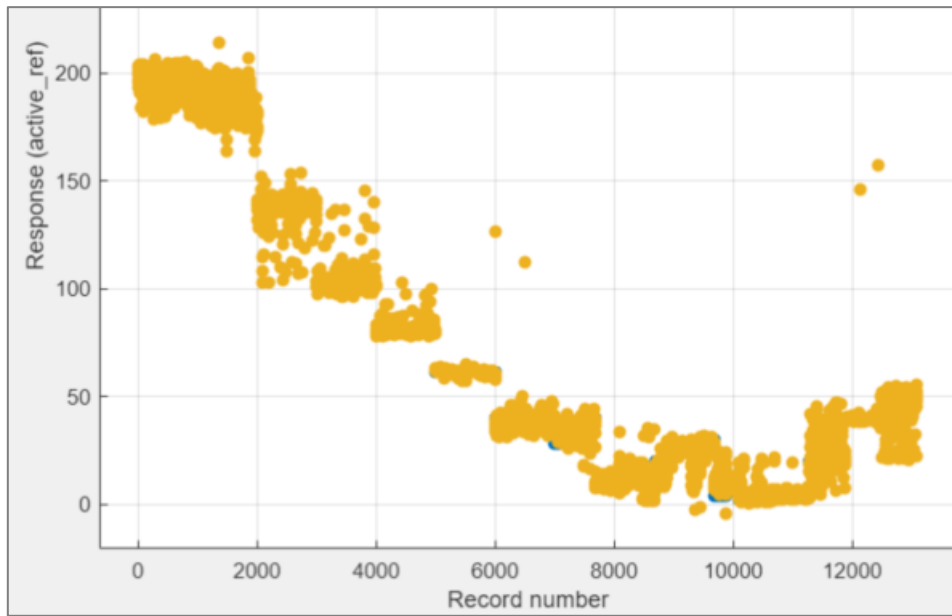


Figure 8.1 Response plot of Accelerometer Channel 1 GP model

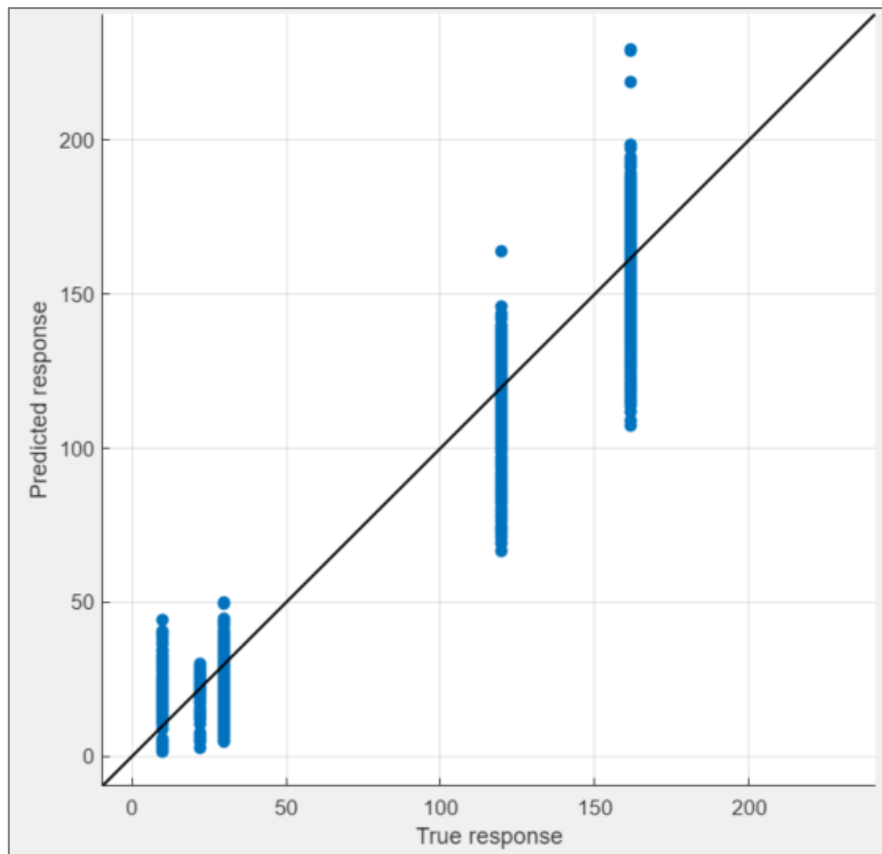


Figure 8.2 Predicted vs Actual Test plot of Accelerometer Channel 1 GP model

8.1.2 Channel 2 GP Model

Model Hyperparameters :

- Preset : Rational Quadratic GPR
- Basis function : Constant
- Kernel function : Rational Quadratic
- Use isotopic kernel : true
- Kernel Scale : Automatic
- Signal Standard Deviation : Automatic
- Sigma : Automatic
- Standardize data : false
- Optimize numeric parameters : false

PCA : Disabled

Features :

- Category
- Peak value 1
- Median Frequency
- Inter quartile range

Table 8.3: Accelerometer Channel 2 GP model performance

Training Results		Test Results	
RMSE (Validation)	7.83	RMSE	10.20
MSE (Validation)	61.43	R-Squared	0.97
Prediction Speed	~6100 obs/sec	MSE (Test)	104.18
Training Time	81.5 sec		

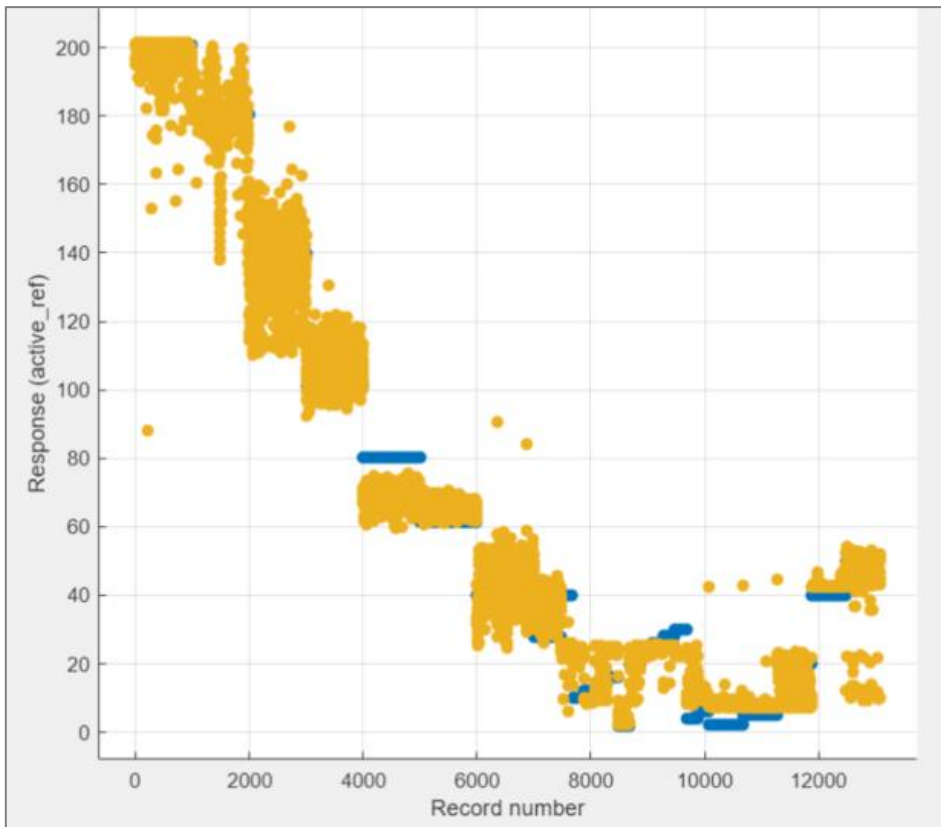


Figure 8.3 Response plot of Accelerometer Channel 2 GP model

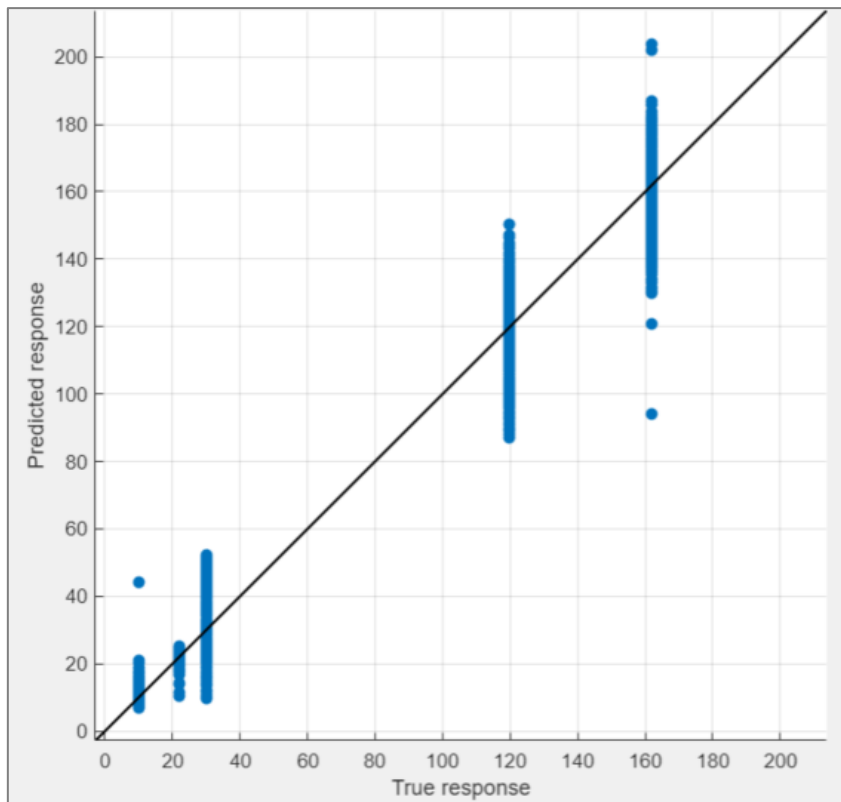


Figure 8.4 Response plot of Accelerometer Channel 2 GP model

8.1.3 Channel 1,2,3 Ensemble Bagged

Model Hyperparameters :

- Preset : Bagged Trees
- Minimum leaf size : 8
- Number of learners : 30
- PCA : Disabled

Features :

- Peak to RMS (all 3 channels)
- Category
- Peak value 1 (all 3 channels), Median frequency

Table 8.4: Accelerometer Channel 1,2,3 Ensemble model performance

Training Results		Test Results	
RMSE (Validation)	2.44	RMSE	12.73
MSE (Validation)	5.98	R-Squared	0.96
Prediction Speed	~78,000 obs/sec	MSE (Test)	169.8
Training Time	4.18 sec		

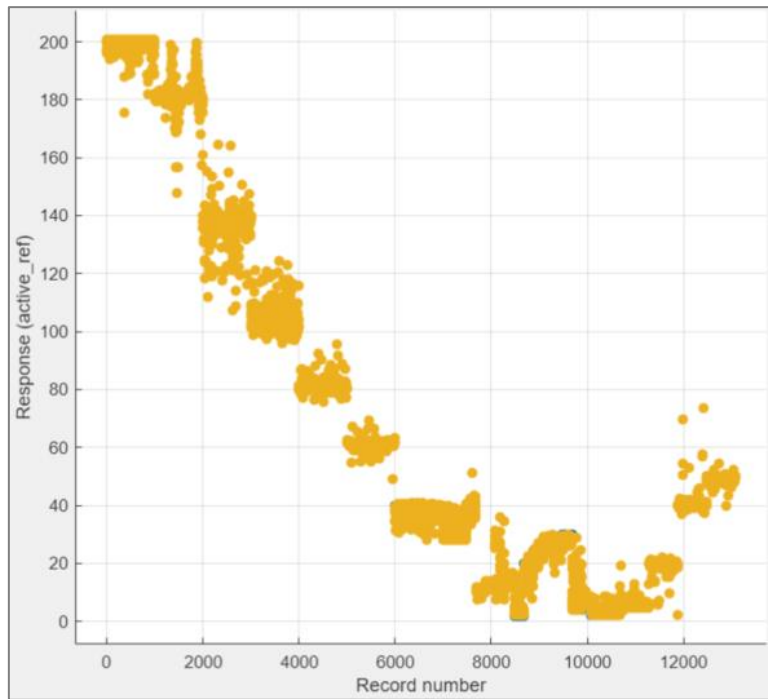


Figure 8.5 Response plot of Accelerometer Channel 1,2,3 Ensemble bagged model

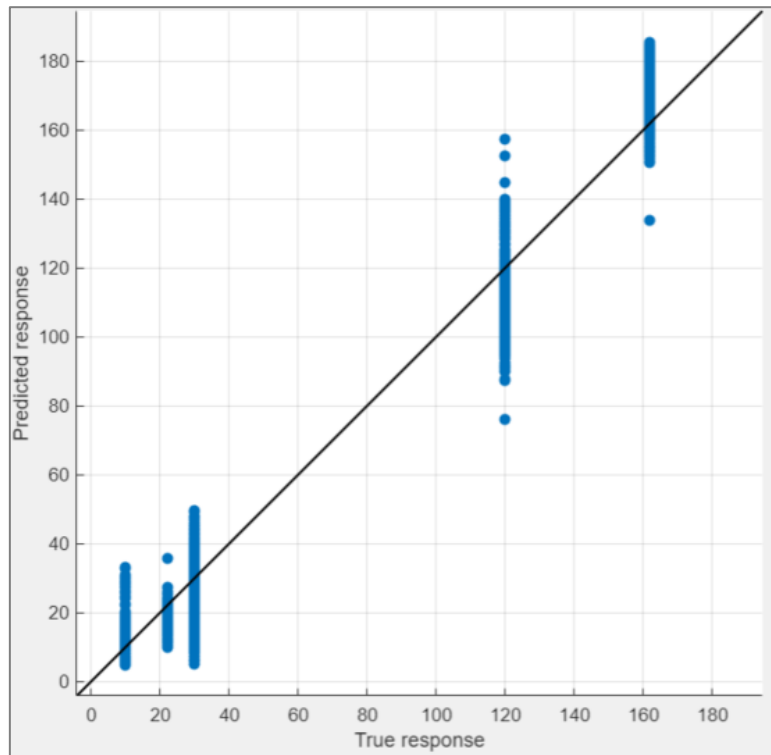


Figure 8.6 Predicted vs Actual Test plot of Accelerometer Channel 1,2,3 Ensemble bagged model

8.1.4 Channel 1 and 2 GP

Model Hyperparameters :

Preset : Exponential GPR

Kernel function : Exponential

Use isotopic kernel : true

Kernel Scale : Automatic

Signal Standard Deviation : Automatic

Sigma : Automatic

Standardize data : true

PCA : Disabled

Features : All 51 features

Table 8.5: Accelerometer Channel 1 and 2 GP model performance

Training Results		Test Results	
RMSE (Validation)	2.4216	RMSE	8.7561
MSE (Validation)	5.864	R-Squared	0.98
Prediction Speed	~3800 obs/sec	MSE (Test)	76.67
Training Time	1433.5 sec		

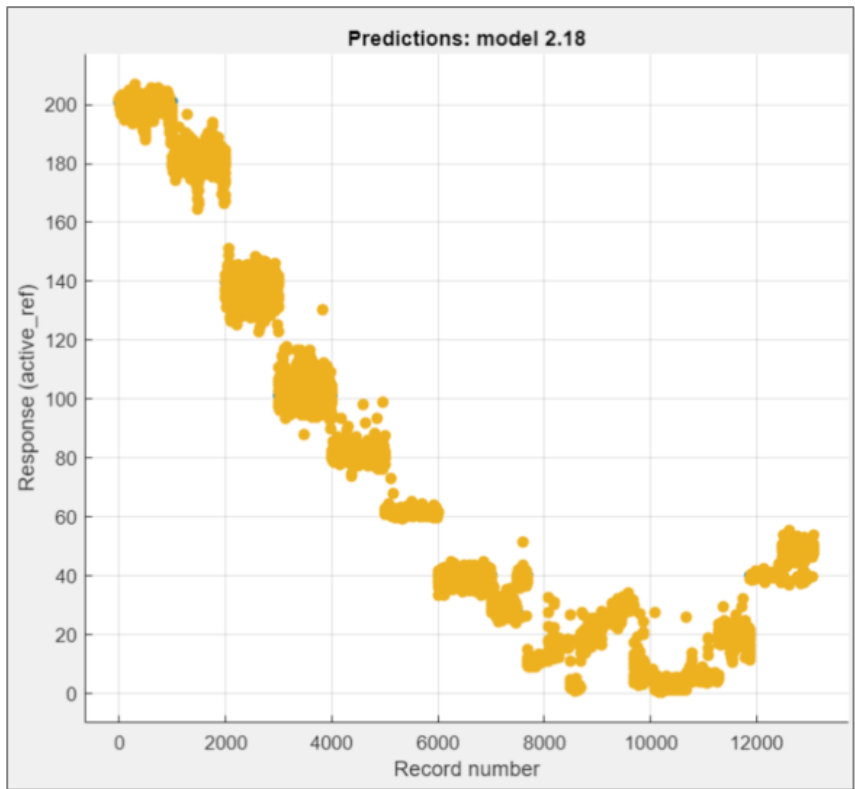


Figure 8.7 Response plot of Accelerometer Channel 1 and 2 GP model

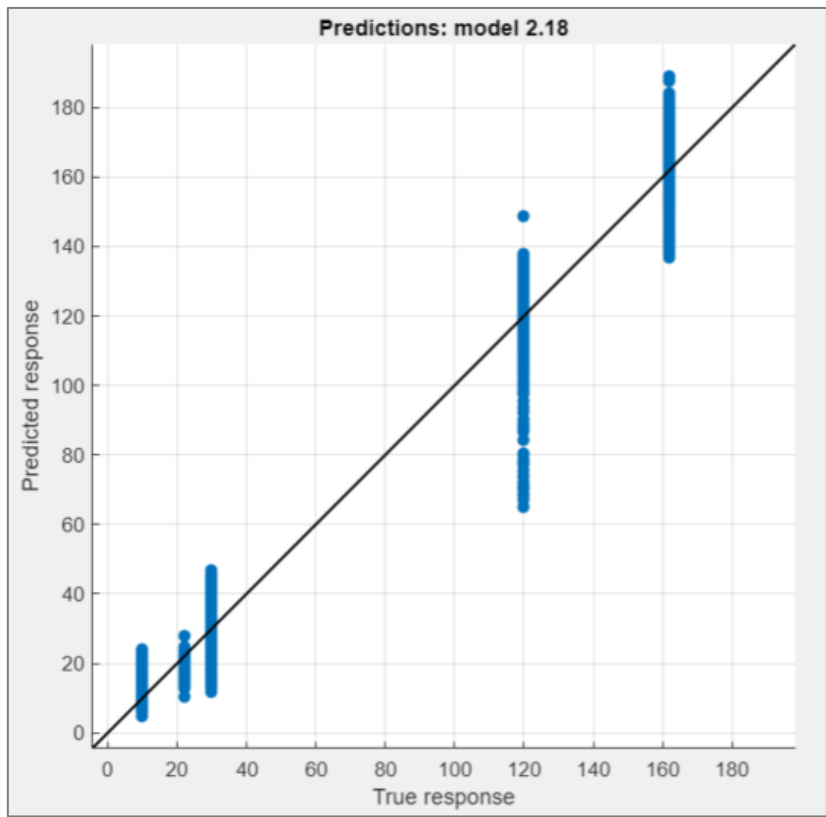


Figure 8.8 Predicted vs Actual Test plot of Accelerometer Channel 1 and 2 GP model

8.1.5 Channel 1 and 2 Ensemble Bagged

Model Hyperparameters :

- Preset : Bagged Trees
- Minimum leaf size : 8
- Number of learners : 30

PCA : Disabled

Features :

- Median Frequency (both channels)
- Mean (both channels)
- Category
- Peak value 1 (both channels)

Table 8.6: Accelerometer Channel 1 GP model performance

Training Results		Test Results	
RMSE (Validation)	2.4914	RMSE	14.263
MSE (Validation)	6.206	R-Squared	0.95
Prediction Speed	~77000 obs/sec	MSE (Test)	203.44
Training Time	4.721 sec		

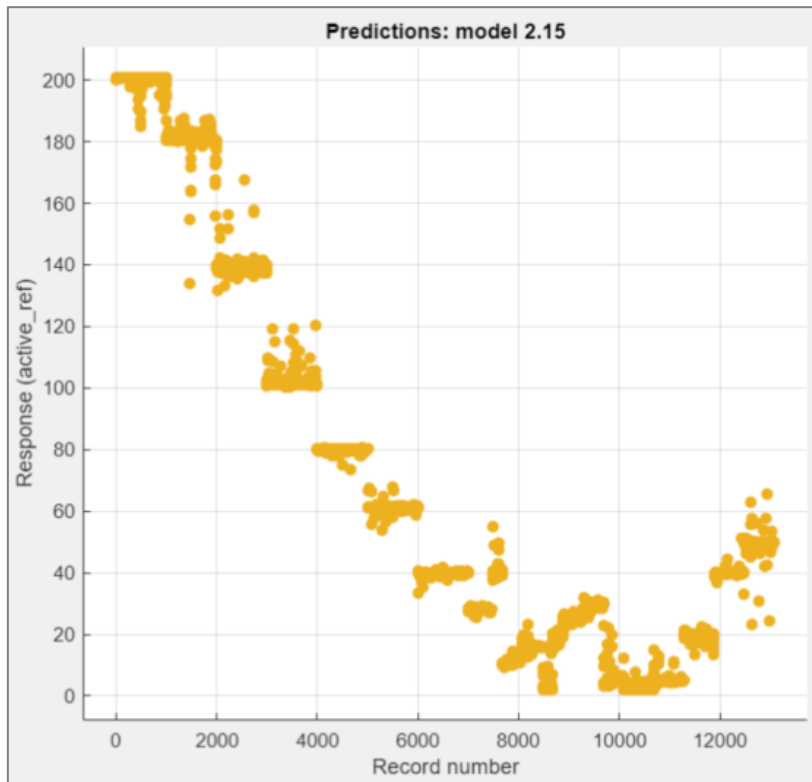


Figure 8.9 Response plot of Accelerometer Channel 1 and 2 Ensemble Bagged model

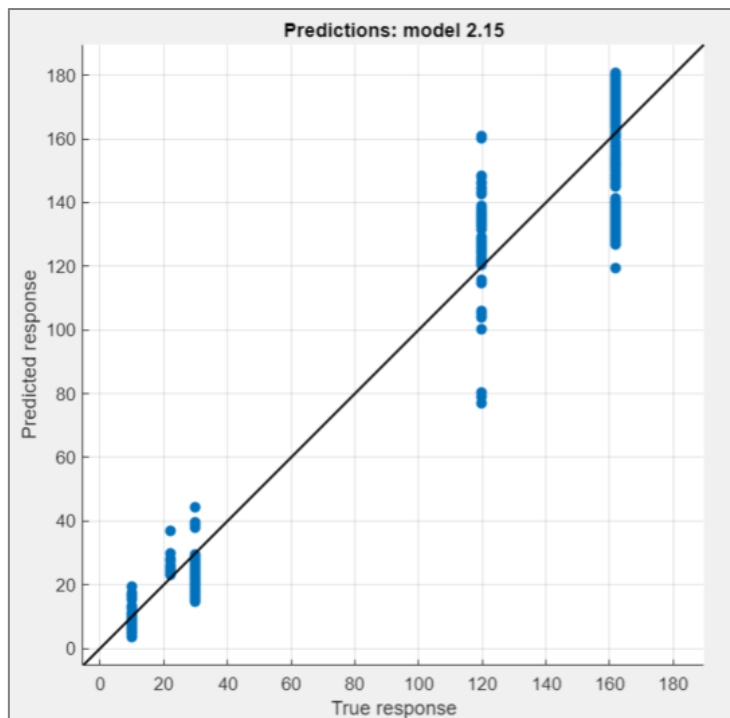


Figure 8.10 Predicted vs Actual Test plot of Accelerometer Channel 1 and 2 Ensemble Bagged model

9 Results

This chapter details the combined performance of classification and prediction models developed in previous section. The results are showcased based on the testing of 6 experiments which were kept isolated from training dataset. So, the models discussed next doesn't have any prior information of these 6 experiments. Following block diagram showcases the workflow for testing.

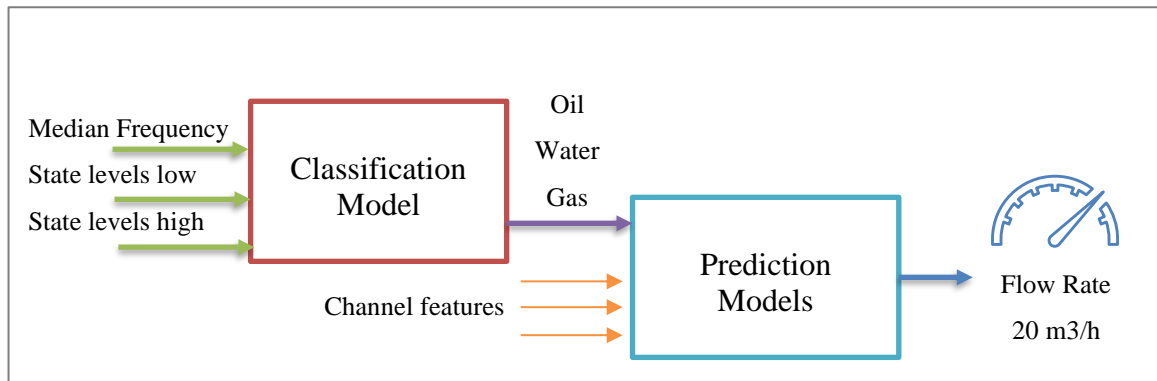


Figure 9.1 Block Diagram showing testing scenario used to showcase the results

As shown in figure 9.1, first the input is given to classification model. Here the input is accelerometer channel 1 features named median frequency, state levels low and high. The output of this classification model is type of flow material i.e., Gas, Oil or Water. This categorical output type acts as one of the inputs to prediction model. Prediction model also has other inputs and they depend upon the model and accelerometer channel, as mentioned in previous chapter. Although many models with different combination were tested. Only the robust models from Table 7.1 and Table 8.1 are selected for this section.

Here, following mentioned models are used :

Classification Model :

1. KNN with accelerometer channel 1 features (3)
2. SVM with accelerometer channel 1 features (3)
3. Neural Network with accelerometer channel 1 features (3)

Prediction Model :

1. GPR with Accelerometer channel 1 (5)
2. GPR with Accelerometer channel 2 (4)
3. GPR with Accelerometer channel 1 and 2 (51)
4. Ensemble bagged with Accelerometer channel 1 and 2 (8)
5. Ensemble bagged with Accelerometer channel 1, 2 and 3 (8)

9.1 MATLAB Live Editor

This section covers the testing performed on a dataset of 3600 rows of 6 experiments. Here screen snippets of live editor are shown.

Random Row selected from test data

```
test_data = 1x114 table
```

	category	name	time	temp_in	temp_out	press_in	...
1	OT (Oil)	"OT22"	76 sec	0.4331	0.5636	-0.4022	

Original Flow Type

```
test_data.category
```

```
ans = categorical  
      OT
```

```
acc_data = 1x3 table
```

	statelevels_fd_low_1	statelevels_fd_high_1	medianfreq_1
1	-0.3486	-0.3118	-1.6358

Predicted Flow Type

```
SVM
```

```
category_prediction_1 = svm_classify.predictFcn(acc_data)
```

```
category_prediction_1 = categorical  
      OT
```

```
KNN
```

```
category_prediction_2 = knn_classify.predictFcn(acc_data)
```

```
category_prediction_2 = categorical  
      OT
```

```
Neural Network
```

```
category_prediction_3 = nn_classify.predictFcn(acc_data)
```

```
category_prediction_3 = categorical  
      OT
```

Figure 9.2 Screen Snips of MATLAB live editor showing testing of classification models

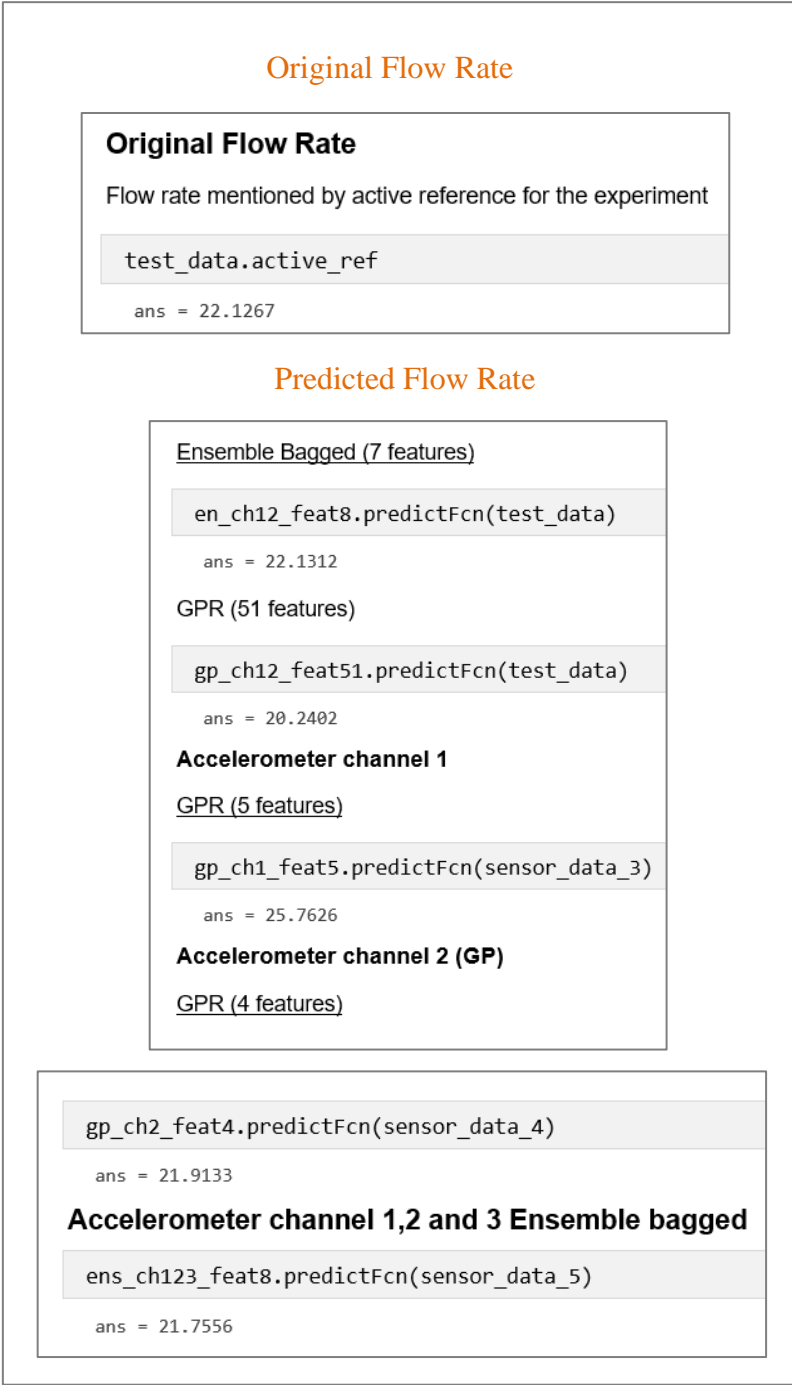


Figure 9.3 Screen Snips of MATLAB live editor showing testing of Prediction models

9.2 MATLAB Simulink Demonstration

To represent the real time performance of work done in this thesis, Simulink model is developed using classification and regression models developed before. Screen snip of usage of classification model is shown in figure 9.4 and usage of regression model is shown in figure 9.5.

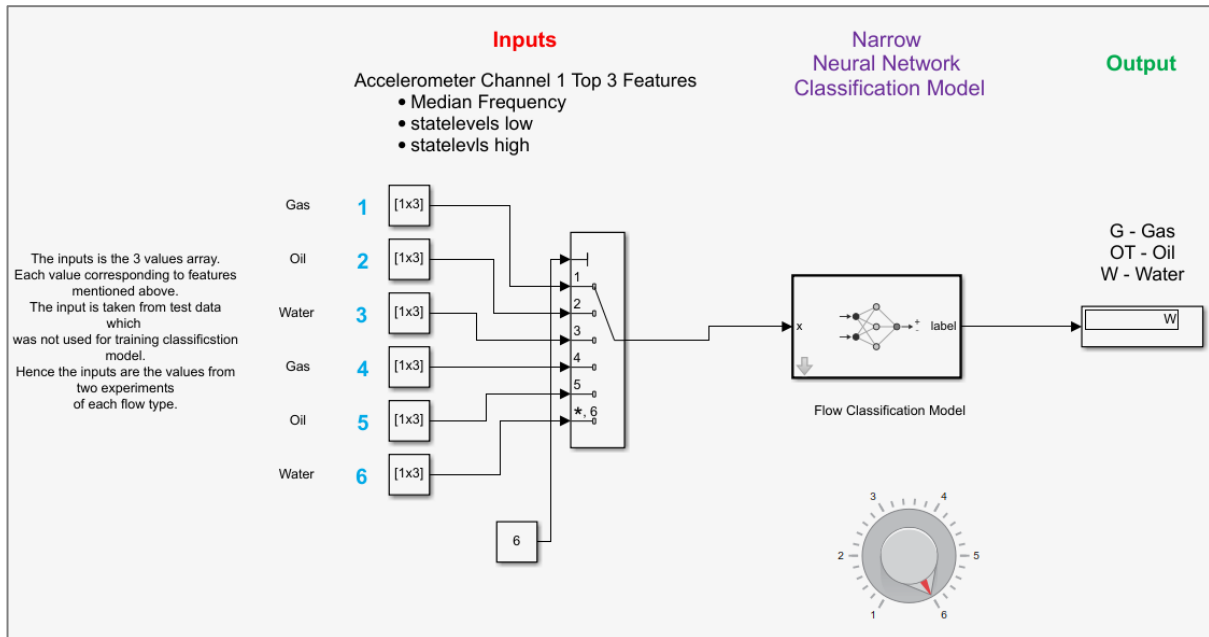


Figure 9.4 Screen Snips of MATLAB Simulink showing usage of classification model (NN)

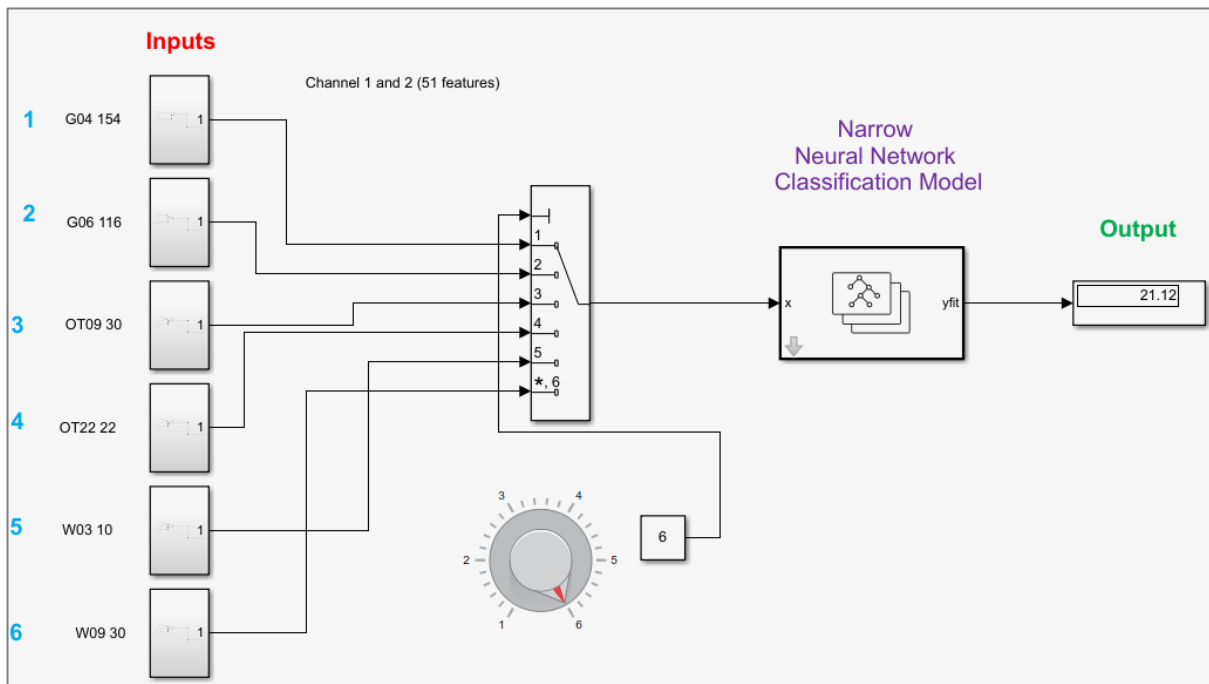


Figure 9.5 Screen Snips of MATLAB Simulink showing usage of regression model (NN)

9.3 Model Accuracy

Classification Models accuracy is directly plotted by MATLAB in form of confusion matrix. It is mentioned in Table 7.1. For KNN model with 3 features of Median frequency, state levels low and high, accuracy is as follows :

- Gas : 96.9 %
- Oil : 99.5 %
- Water : 99.9 %

Overall Model accuracy is 98.2 %.

However, accuracy of prediction or regression model is not directly mentioned in MATLAB regression. The model performance is given out in form of Root Mean Square Error (RMSE). It is mentioned in table 8.1.

To mention the testing results for the work done in this thesis in terms of flow rate prediction, following workflow scenario is performed to show the model performance in form of accuracy (%). Microsoft Excel is used to perform this action.

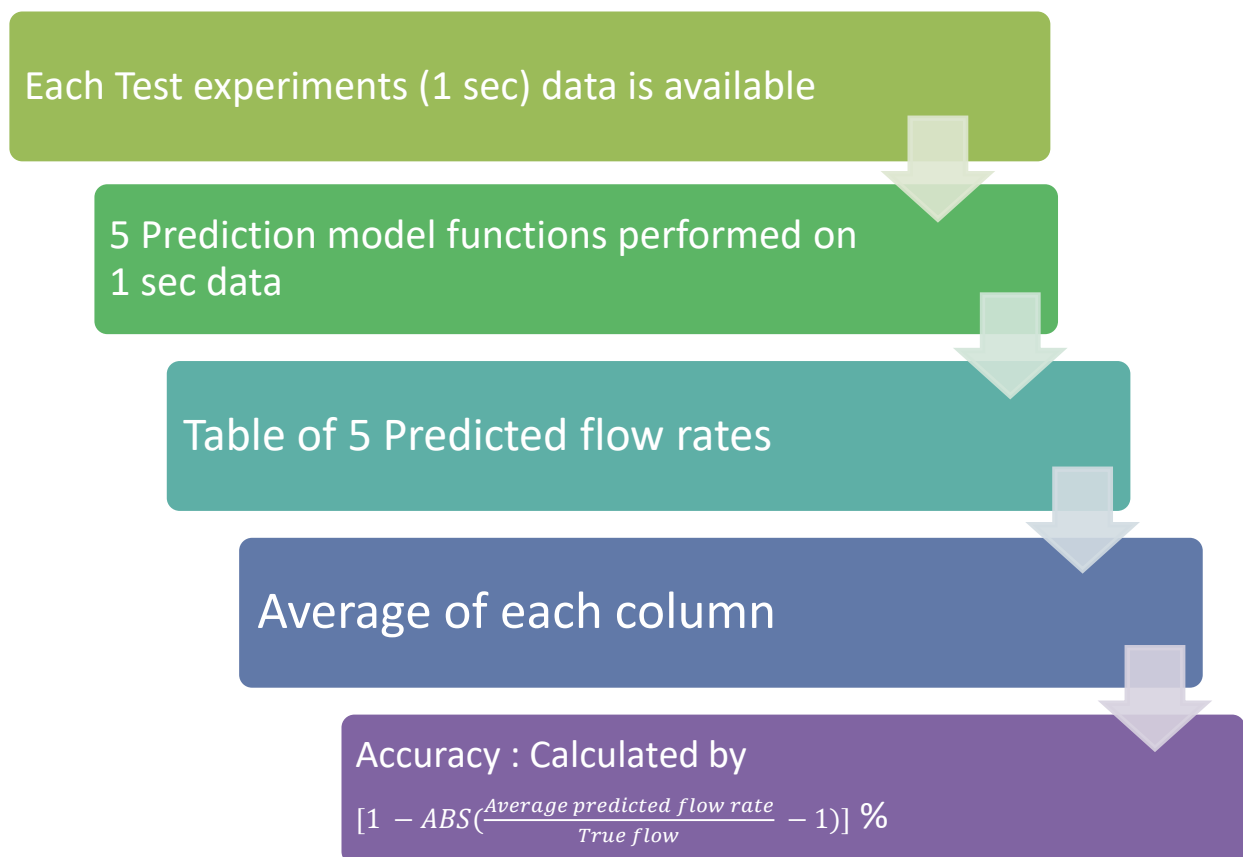


Figure 9.6 Work Flow Chart of test data handling to get accuracy

Performing following action gives results mentioned in table 9.1 below. Models are as follows :

1. GPR with Accelerometer channel 1 (5)
2. GPR with Accelerometer channel 2 (4)
3. GPR with Accelerometer channel 1 and 2 (51)
4. Ensemble bagged with Accelerometer channel 1 and 2 (8)
5. Ensemble bagged with Accelerometer channel 1, 2 and 3 (8)

Table 9.1: Flow Rate Prediction Model accuracy for each test experiment

Experiment	True Flow (m3/hr.)	Predicted Flow Rate Average (m3/hr.) Accuracy (%)									
		Model 1		Model 2		Model 3		Model 4		Model 5	
G04	161.9	164.0	99	157.2	97	166.5	97	157.5	97	170.5	95
G06	119.8	116.2	97	121.0	99	122.4	98	136.0	86	124.3	96
OT09	24.3	20.57	85	22.78	94	26.3	92	19.0	78	13.3	55
OT22	22.1	23.4	94	22.1	100	18.7	91	20.2	91	18.7	85
W03	10.0	15.4	45	9.0	90	12.8	72	11.5	85	15.6	43
W09	30.0	23.4	78	41.1	63	29.2	98	34.3	85	32.7	91

9.4 USN Test Data

This section covers analysis of experimental data obtained from USN rig. Also, compatibility check of USN rig data with Equinor rig data is performed and then testing results of USN data with Equinor data trained model is mentioned.

9.4.1 Spectral Analysis of USN data

After studying the raw FFT plots of both accelerometer channels, 4th order Butterworth band pass filter is used. But the range used here is 10 Hz to 10 kHz. The range is selected based on following two points. Low frequency cut-off removed the frequency harmonics likely to originate from experiment setup and high frequency cut-off removed the added noise since the sensitivity of sensor changes above 10 kHz, which is likely to give unwanted noise above this frequency. Also, frequencies like 12 and 13 kHz are known noise from surrounding and is observed in all FFT plots. So high cut-off of 10 kHz is selected.

9.4.1.1 Air flow experiments plots

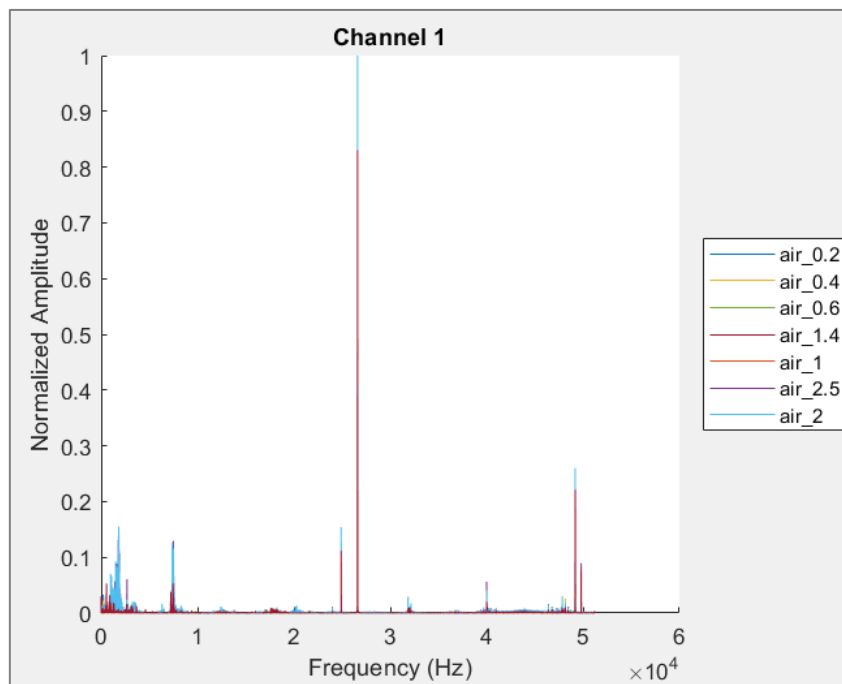


Figure 9.7 FFT of air experiments at USN rig (Unfiltered)

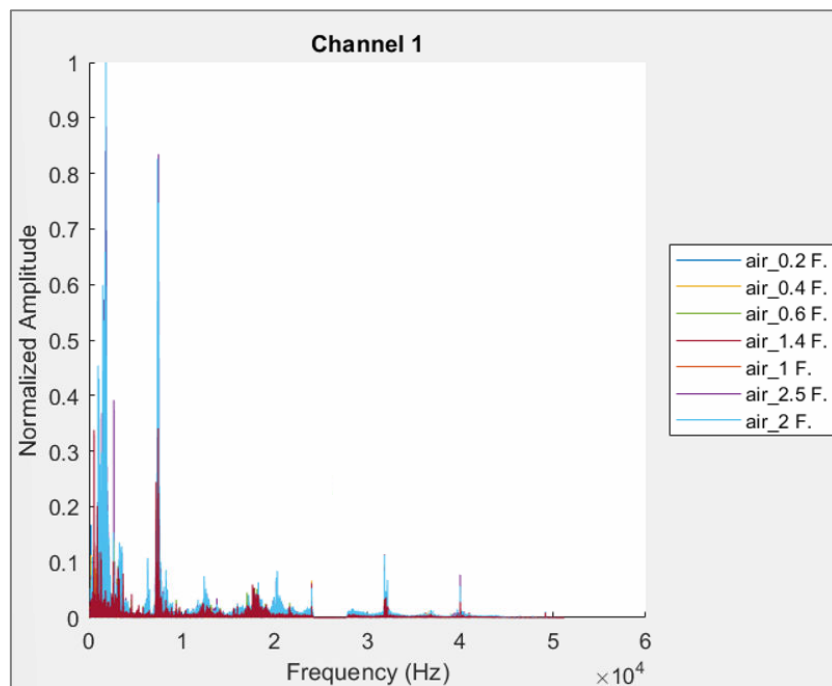


Figure 9.8 FFT of air experiments at USN rig (Filtered)

9.4.1.2 Oil flow experiments plots

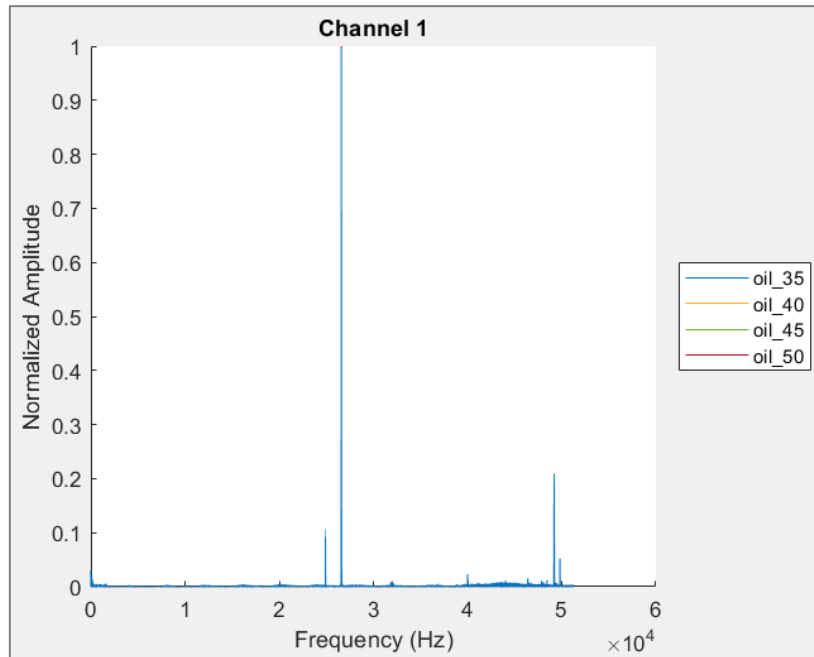


Figure 9.9 FFT of oil experiments at USN rig (Unfiltered)

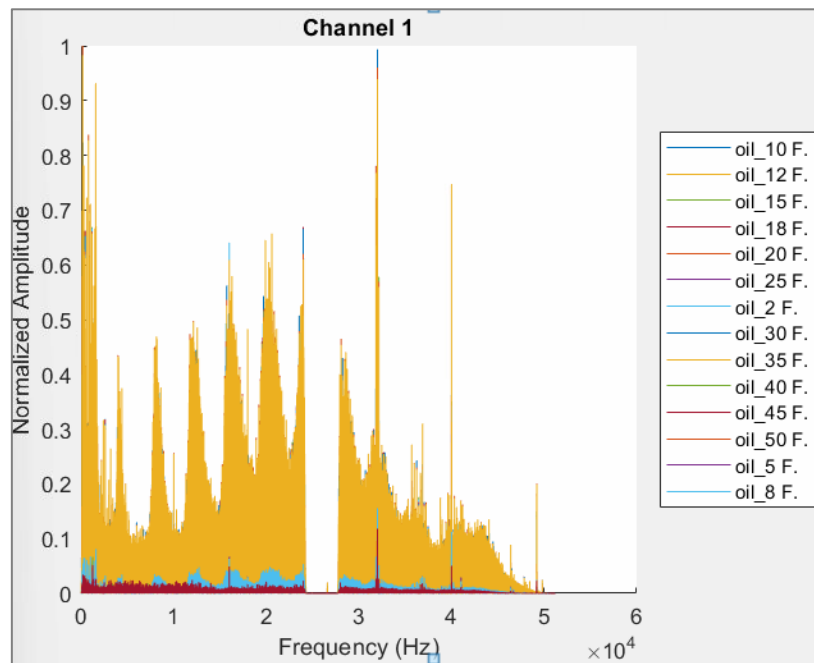


Figure 9.10 FFT of oil experiments at USN rig (Filtered)

9.4.1.3 Water flow experiments plots

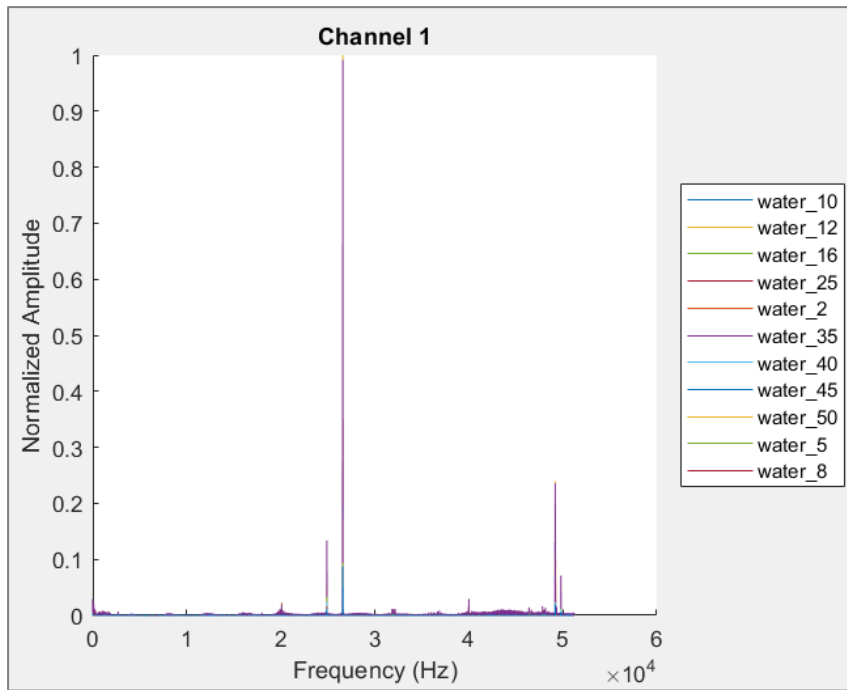


Figure 9. FFT of water experiments at USN rig (Unfiltered)

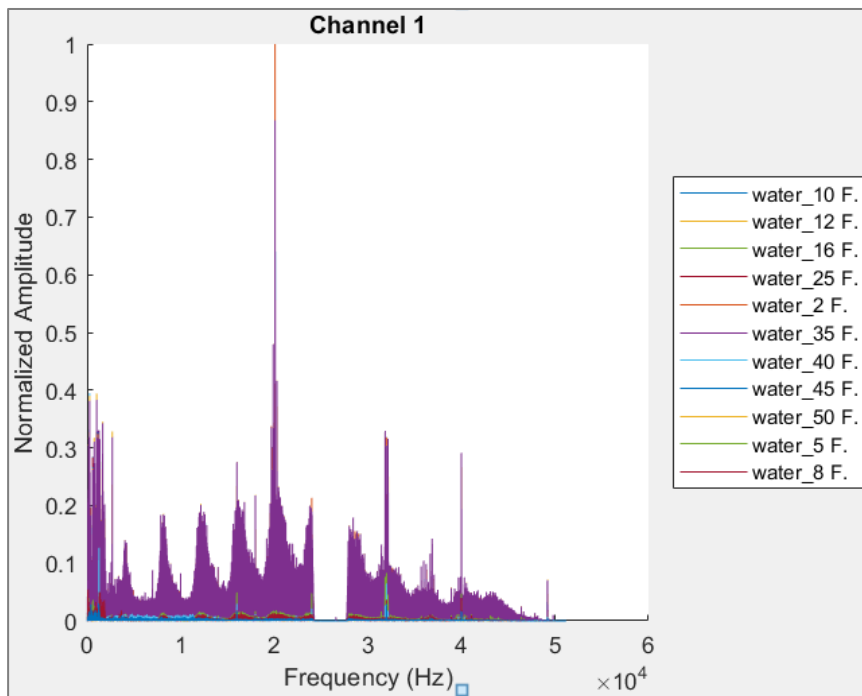


Figure 9.12 FFT of water experiments at USN rig (Filtered)

9.4.2 Power Spectral Density of accelerometer channel data

To study the intensity of frequencies, present in vibration data, power spectrum density plots of each flow type i.e., air water and oil are plotted as shown in figure 9.13, 9.14 and 9.15 respectively.

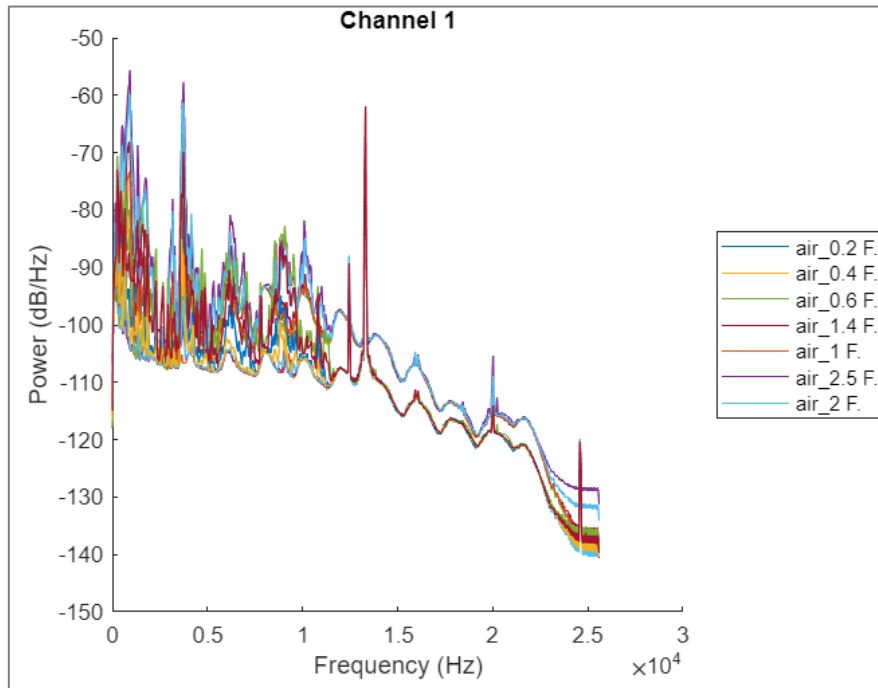


Figure 9.13 PSD plot of air experiments at USN rig (Using Hanning Window)

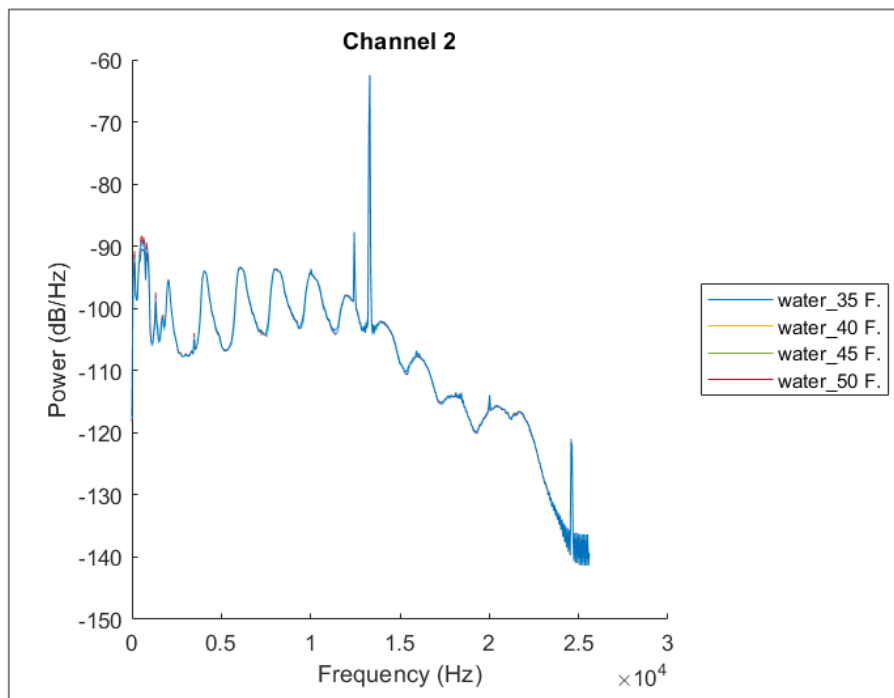


Figure 9.14 PSD plot of water experiments at USN rig (Using Hanning Window)

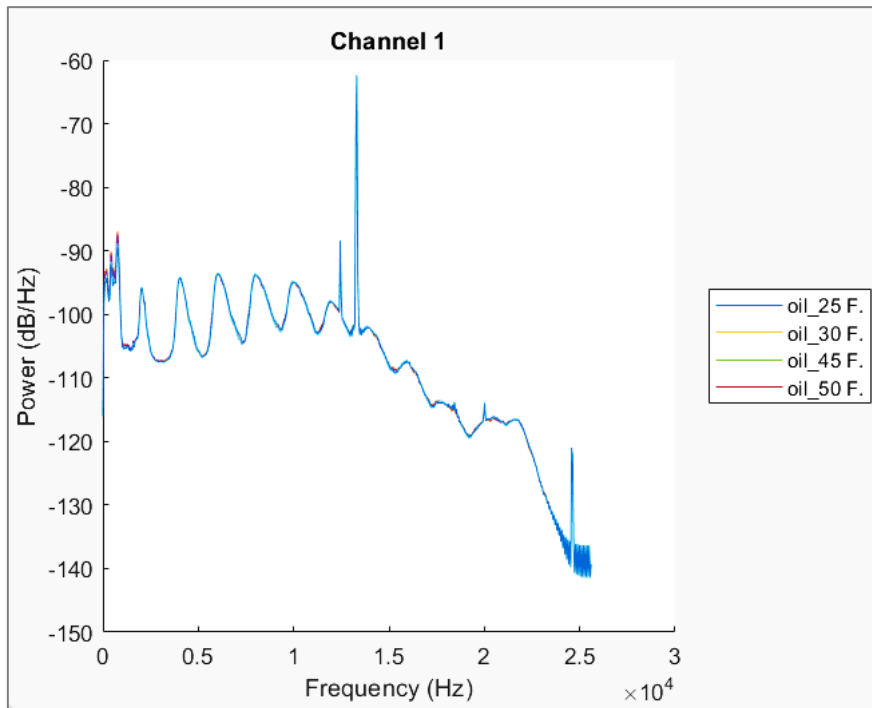


Figure 9.15 PSD plot of oil experiments at USN rig (Using Hanning Window)

Following Observations can be made based on plots :

- High amplitudes of noise frequencies i.e., 12 to 13 kHz are observed in unfiltered plots in each flow material type. This causes the amplitudes of desired frequencies to appear very small in plots.
- Amplitudes of frequencies in USN data set appears to be very less as compared to amplitudes of frequencies in Equinor dataset. This is most likely due to experiments conducted at very low flow rate as compared to Equinor flow rate experiments.
- Also, power of vibration frequencies is not that high as can be in PSD plots. Also, PSD plot of water and oil shows same behavior. This is interesting thing as it affects classification model developed in later section.

9.5 Compatibility check of USN dataset with Equinor dataset

Experiments at both the rigs are conducted at different flow rates as shown in table 9.2 below.

Table 9.2: Flow Rate Prediction Model accuracy for each test experiment

Experiment Flow Type	Equinor flow range (m3/hr.)	USN flow range (m3/hr.)
Water	2 – 60	0.12 - 3
Oil	2 – 40	0.12 - 3
Gas	30 - 200	0.01 – 0.12

The table 9.2 implies following things :

- Model trained using Equinor dataset is not directly compatible with USN dataset due to mismatch of flow range since the data in Equinor dataset is normalized before training and normalizing USN dataset with same parameters causes error in values.
- Equinor trained model for Water, Oil and Gas experiments has no values of low flow rates i.e., below 2 m³/hr. as desired by USN dataset.
- Hence Gas experiments from USN dataset will be completely eliminated for testing since it will only cause incorrect results.
- Also, a mini dataset from Equinor is formed including only values of Water and Oil with low flow rates to again train classification and regression models to test with USN dataset of water and oil only.

9.5.1 Classification model test results

Training dataset :

- Equinor dataset (Oil and water experiments with flow range : 2 to 5 m³/hr.)
- 1600 Rows and 54 Columns

Test dataset :

- USN dataset (Oil and water experiments with flow range : 1 to 3 m³/hr.)
- 20,253 Rows and 54 Columns

Table 9.3: Linear Discriminant classification model performance with USN test data

Training Results		Test Results	
Accuracy (Validation)	100 %	Accuracy	57.8 %
Total Cost (Validation)	0	Total Cost	11617
Prediction Speed	~44000 obs/sec		
Training Time	1.7 sec		

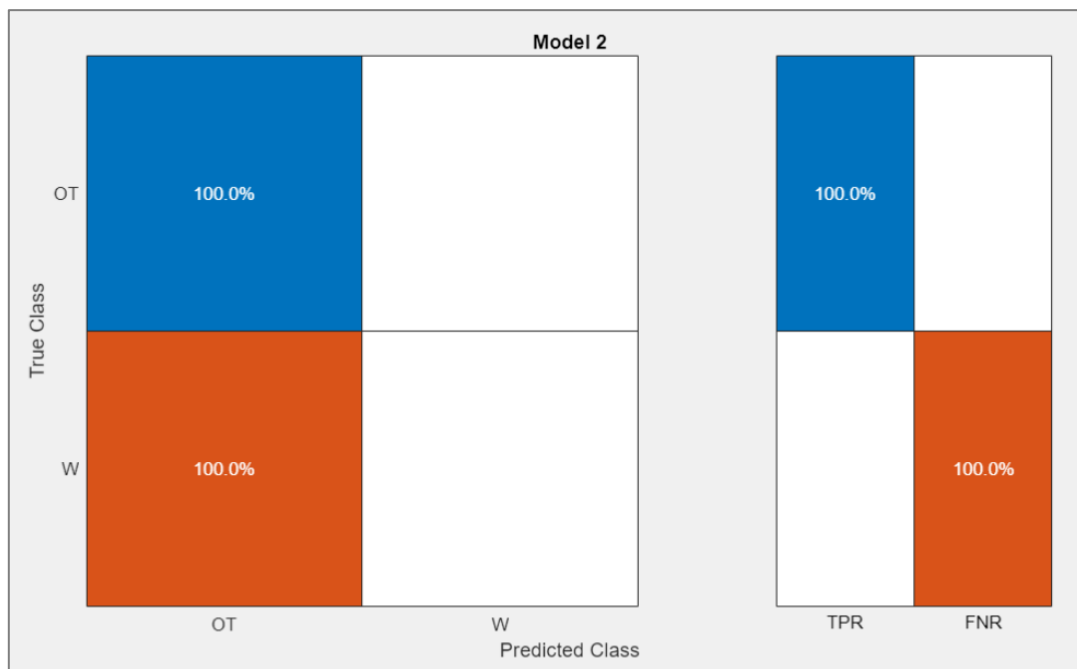


Figure 9.16 Test Confusion matrix of classification model with USN test data

☆ 1 Tree	Accuracy (Test): 57.6%
Last change: Fine Tree	25/25 features
☆ 2 Linear Discriminant	Accuracy (Test): 57.8%
Last change: Linear Discriminant	25/25 features
☆ 3 Naive Bayes	Accuracy (Test): 42.2%
Last change: Gaussian Naive Bayes	25/25 features
☆ 4 SVM	Accuracy (Test): 57.8%
Last change: Linear SVM	25/25 features
☆ 5 KNN	Accuracy (Test): 42.2%
Last change: Fine KNN	25/25 features
☆ 6 Ensemble	Accuracy (Test): 57.6%
Last change: Bagged Trees	25/25 features
☆ 7 Neural Network	Accuracy (Test): 42.2%
Last change: Hyperparameter option(s)	25/25 features

Figure 9.17 Different classification model performances with USN test data

9.5.2 Regression model test results

Model Hyperparameters :

- Preset : Rational Quadratic GPR
- Basis function : Constant
- Kernel function : Rational Quadratic
- Use isotopic kernel : true
- Kernel Scale : Automatic
- Signal Standard Deviation : Automatic
- Sigma : Automatic
- Standardize data : true
- Optimize numeric parameters : true

PCA : Disabled

Features :

- All features of Accelerometer Channel 1 & 2 & category

Table 9.3: GPR model performance with USN test data

Training Results		Test Results	
RMSE (Validation)	0.32	RMSE	1.01
MSE (Validation)	0.10	R-Squared	-5.0
Prediction Speed	~29000 obs/sec	MSE (Test)	1.02
Training Time	126.8 sec		



Figure 9.18 Response plot of GPR regression model with USN test data

Following Observations can be made based on classification model and regression model performance.

- All of the Water rows are also wrongly classified as Oil. This was expected based on compatibility check section. The vibration profiles are almost same or both the flow material type of experiments as seen in FFT and PSD plots.
- Regarding flow rate estimation the range of flow was not that diverse to predict i.e., test data range is 1 to 3 m³/hr. and training data range is 2 to 5 m³/hr.. So, even low flow rate is predicted as higher flow rate and this make sense since trained model has no information for low flow rate.
- But in case of rows where flow rate matches, it can be seen that at that point prediction of flow rate is better. But it has limitations due to a smaller number of experiments in this range.

10 Discussion

This chapter covers the interpretation of results mentioned in previous chapter, the implications of the results found in this thesis in terms of the field of oil and gas sector of flow metering, the limitations of the results and recommendation from the author point of view.

10.1 Key Findings

Different type of approaches is used to find accurate flow measurements in oil and gas, multiphase process environment. This study brings into attention the vibration data type of non-invasive approach which gives promising results in terms of finding type of flow material and estimating its flow velocity. High correlations are observed between some accelerometer features and flow material type and also with flow velocity.

10.2 Limitations

Based on the total workflow performed in this thesis and analyzing the model performances of classification and regression models, it can be said that better correlations between accelerometer features can be achieved and accuracy of prediction models can be further increased with following recommendations :

- Accelerometer data at no flow state.
- Experiments at linear flow rate difference. For example, one experiment at 10 m³/hr. and another at 11 m³/hr. This can help in analyzing the change in vibration profile at 1 m³/hr. change.
- The results mentioned here are limited to flow range mentioned in tables in chapter 3.
- Equinor Dataset and USN Dataset : Inter-compatibility of these datasets can be confirmed with more data like experiments performed in both the rigs are carried with same flow rate.

10.3 Sensor Fusion Possibility with ECT based approach

Different possibilities open when one system working on one principle is combined with another system which is working on different principle. One such data fusion possibility explained here in Figure 10.1 is combination of accelerometer features along with Electrical Capacitance Tomography system working on electrical permittivity and conductivity characteristics of material flowing.

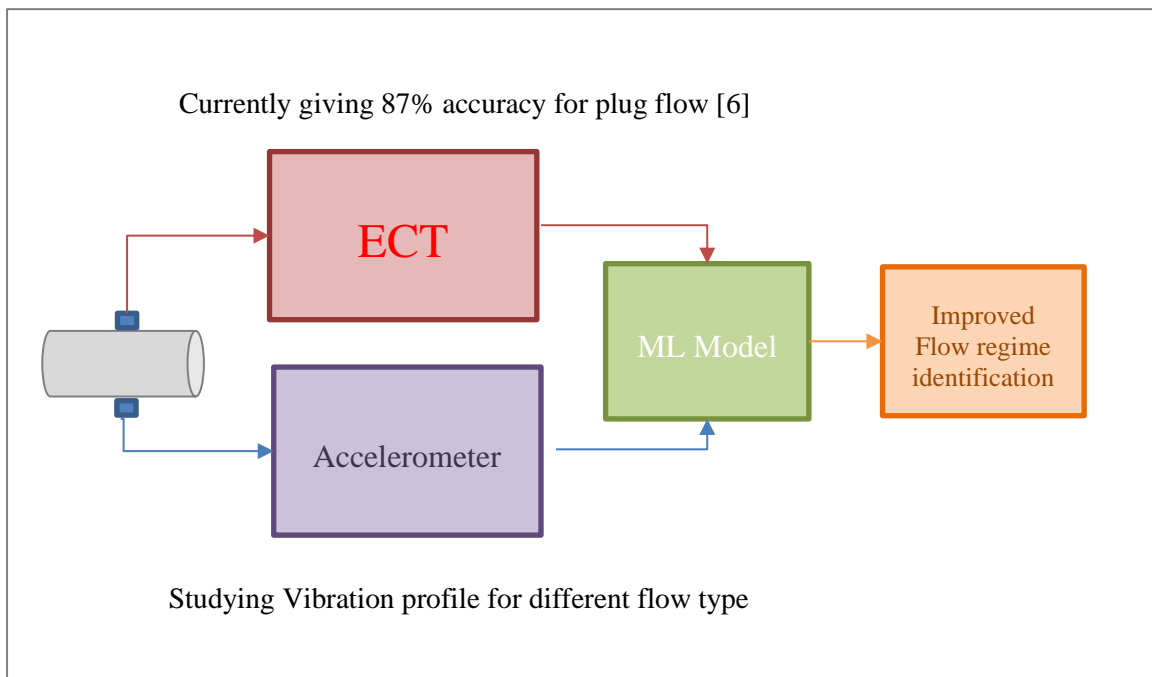


Figure 10.1 One possible sensor data fusion with electrical capacitance tomography

11 Conclusion

The work done in this thesis brings in the approach of vibration data in estimating flow material and estimating flow velocities in oil and gas section of multi-phase flow metering. Many accelerometer features are tested and this can be used as basis for further selecting suitable features which gave promising results in this thesis.

Machine learning models trained and tested showed the ability to classify the flow material type based on vibrations and also estimate flow velocities based on vibration profile. Models used were simple models without any kind of optimization. Further models can be developed to get even better results. Also, deep learning methods can be tested using different accelerometer features mentioned in this thesis.

Fine KNN classification model with accelerometer channel 1 features like median frequency, state levels low and high as an input gave accuracy of 98.2 %.

Rational Quadratic GPR model with Test RMSE of nearly 10.2 with accelerometer channel 2 features like Category, Peak value 1, Median Frequency and Inter quartile range gave an accuracy as mentioned in table below.

Table 11.1: GP regression model results showing true flow and predicted flow using 4 features of accelerometer channel 2

Experiment	True Flow (m ³ /hr.)	Predicted Flow (m ³ /hr.)	Accuracy (%)
G04	161.9	157.2	97
G06	119.8	121.0	99
OT09	24.3	22.78	94
OT22	22.1	22.1	100
W03	10.0	9.0	90
W09	30.0	41.1	63

References

- [1] Gabriel M.P. Andrade, Diego Q.F. de Menezes, Rafael M. Soares, Tiago S.M. Lemos, Alex F. Teixeira, Leonardo D. Ribeiro, Bruno F. Vieira, José Carlos Pinto. (2022). Virtual flow metering of production flow rates of individual wells in oil and gas platforms through data reconciliation. *Journal of Petroleum Science and Engineering*, Volume 208, Part E, 109772, ISSN 0920-4105.
<https://doi.org/10.1016/j.petrol.2021.109772>
- [2] Theodore E. Miller and Hamish. *Small Analytical Chemistry* 1982 54 (6), 907-910
<https://doi.org/10.1021/ac00243a016>
- [3] J. Hitomi, Y. Murai, H. J. Park and Y. Tasaka, *Ultrasound Flow-Monitoring and Flow-Metering of Air–Oil–Water Three-Layer Pipe Flows*, in *IEEE Access*, vol. 5, pp. 15021-15029, 2017, doi: 10.1109/ACCESS.2017.2724300.
<https://www.mdpi.com/1424-8220/20/1/306>
- [4] Mosorov V, Zych M, Hanus R, Sankowski D, Saoud A, Improvement of Flow Velocity Measurement Algorithms Based on Correlation Function and Twin Plane Electrical Capacitance Tomography. *Sensors* 2020, 20, 306.
<https://doi.org/10.3390/s20010306>
- [5] Eivind Dahl, Christian Michelsen Research AS, (2005), *Handbook of Multiphase Flow Metering*. Retrieved from
https://nfogm.no/wp-content/uploads/2014/02/MPFM_Handbook_Revision2_2005_ISBN-82-91341-89-3.pdf
- [6] Aleksander Tokle Poverud. (2019). Flow-Analytics using Multiphase Flow Rig with Multimodal Sensor Suite – with focus on Void Fraction, Water-Cut and Flow Regimes (Master Thesis). USN, Porsgrunn.
- [7] Hansford Sensors, 2022. Retrieved from
<https://www.hansfordsensors.com/wp-content/uploads/datasheets/TS015U.pdf>
- [8] R. P. Evans, J. D. Blotter, A. G. Stephens, *Flow rate measurements using flow-induced pipe vibration*, *Trans. ASME*, vol 126, pp. 280-285, March 2004.
- [9] W. K. Blake, *Mechanics of flow-induced sound and vibration*, Ac. Press. Inc., *Harcort Brace Jovanovich Publishers*, Orlando, FL, 1986, pp. 1-43, Chap.1.
- [10] M. M. Campagna, G. Dinardo, L. Fabbiano, and G. Vacca, *Fluid flow measurements by means of vibration monitoring*, *Meas. Sci. Technol.*, vol. 26, no. 11, p. 115306, 2015, DOI: 10.1088/0957-0233/26/11/115306.
- [11] Olle Penttinen, Marcus Ulveström, Kristina Karlsson, Veronika Andersson, Håkan Andersson, Johan Pettersson, Oliver Bükér. (2021). *Towards flow measurement with passive accelerometers*, *Flow Measurement and Instrumentation*, <https://doi.org/10.1016/j.flowmeasinst.2021.101992>.

- [12] Fabbiano, Laura & Vacca, Gaetano & Dinardo, Giuseppe. (2013). Fluid Flow Rate Estimation using Acceleration Sensors. *Proceedings of the International Conference on Sensing Technology*, ICST. 10.1109/ICSensT.2013.6727646. https://www.researchgate.net/publication/281652853_Flow_Measurement_by_Piezoelectric_Accelerometers_Application_in_the_Oil_Industry
- [13] De Oliveira, Elcio & Medeiros, K. & Barbosa, C. (2015). Flow Measurement by Piezoelectric Accelerometers: Application in the Oil Industry. *Petroleum Science and Technology*. 33. 1402-1409. 10.1080/10916466.2015.1044613.
- [14] Yang, Wonseok. 2021. *Prediction of Flow Velocity from the Flexural Vibration of a Fluid-Conveying Pipe Using the Transfer Function Method*, Applied Sciences 11, no. 13: 5779.
<https://doi.org/10.3390/app11135779>
- [15] Stankovic, L., Dakovic, M., & Thayaparan, T. (2013). Time-frequency signal analysis with applications. Artech House.
- [16] Wikipedia contributors. (2022, April 8). Receiver operating characteristic. In Wikipedia, The Free Encyclopedia. Retrieved 09:31, May 24, 2022, Retrieved from https://en.wikipedia.org/w/index.php?title=Receiver_operating_characteristic&oldid=1081635328
- [17] Priyanka Sarkar, (2022). What is LDA: Linear Discriminant Analysis for Machine Learning. Retrieved from <https://www.knowledgehut.com/blog/data-science/linear-discriminant-analysis-for-machine-learning>
- [18] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011. Retrieved from https://scikit-learn.org/stable/modules/naive_bayes.html
- [19] Wikipedia contributors. (2022, March 25). Support-vector machine. In Wikipedia, The Free Encyclopedia. Retrieved 09:49, May 24, 2022, Retrieved from https://en.wikipedia.org/w/index.php?title=Support-vector_machine&oldid=1079167701
- [20] C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006, ISBN 026218253X, Retrieved from <http://gaussianprocess.org/gpml/chapters/RW1.pdf>
- [21] Oscar Knagg, (2019), *An intuitive guide to Gaussian processes*, Retrieved from <https://towardsdatascience.com/an-intuitive-guide-to-gaussian-processes-ec2f0b45c71d>
- [22] Zhi-Hua Zhou. (2012). *Ensemble Methods: Foundations and Algorithms* (1st Edition), Chapman & Hall/CRC Machine Learning & Pattern Recognition.
- [23] MATLAB. (2022). Version (9.12.0.1884302) (R2022a). Natick, Massachusetts: The MathWorks Inc.

Appendices

Appendix A Task Description

Appendix B Gantt Chart

Appendix C Experiment Details

Appendix D Tools Used in Thesis : Specifications

Appendix E Importing Raw Data to MATLAB

Appendix F Accelerometer Data Plots : MATLAB code

Appendix G Accelerometer Data Processing

Appendix H Manual Separation of Training Data and test data

Appendix I Normalization of data

Appendix J Designed Filter

Appendix K USN Data Processing

Appendix A

Task Description

Final Version of Task description that outlines the work done in this thesis.

FMH606 Master's Thesis

Title: AE-Sensors and Multimodal Sensor Data Fusion in Liquid Flowmetering

USN supervisor: Ru Yan; Saba Mylvaganam

External partner: Kjetil Fjalestad/EQUINOR/, Tonni Franke Johansen/ SINTEF

Task background:

Multiphase flow rig in USN built and modified many times with funding from the industries and Research Council of Norway, has been used in various CFD studies, testing different multiphase and single flowmetering principles and phenomena.

EQUINOR in Herøya, Grenland has a multiphase flow rig for similar purposes, has performed various measurements, and is planning to perform more measurements. EQUINOR, with extensive experience in different single and multiphase flowmetering, has recently focused on single phase flowmetering using clamp-on AE-sensors.

Along with conventional measurements such as temperature, pressure, flow, and absorption of gamma rays, tomographic measurements using electrical resistance and capacitance tomographic equipment have also been used in the experimental campaigns in USN.

This project aims to build upon these results and has focused on single phase flowmetering based on the fusion of data from different sensor modalities to estimate fluid flow velocity. The aim is to estimate the flow velocity using a sensor network consisting of clamp-on AE-sensors. For validation and fusion of data for enhancing performance of the AE-sensor network, data from other sensor modalities will be fused.

Task description:

The tentative list of tasks for this thesis work is as follows:

- (1) Brief survey of fluid flowmetering with focus on the latest developments
- (2) Description of the different types of liquid flowmeters
- (3) HW/SW modifications in existing measurement systems incorporating latest developments in data acquisition and storage
- (4) Analysing data from liquid flow experiments done in USN and EQUINOR using the sensor suite available and collecting data from all the sensors, including the ECT and ERT modules.
- (5) Estimating liquid flow velocity with data from a single sensor and multimodal sensors
- (6) Developing new models (conventional as well as AI/ML) or extending already existing models in estimating using AE-sensor network for the estimation of liquid flow velocity

(7) Submitting a Master Thesis following the guidelines of USN with necessary programs and including a well-documented and complete set of all experimental data from the measurements

Student category: IIA. Reserved for the master student: Shailesh Kharche

The task is suitable for online students (not present at the campus): No.

Practical arrangements

Necessary experimental data will be provided by USN and possibly EQUINOR. This work is closely coupled to an ongoing project SAM ([SAM Self Adapting Model-based system for Process Autonomy - SINTEF](#)).

Supervision:

Generally, the student is entitled to 15-20 hours of supervision. This includes necessary time for the supervisor to prepare for supervision meetings (reading material to be discussed, etc).

Signatures:

Supervisor (date and signature):  18.02.2022

Student (write clearly in all capitalized letters):

SHAILESH KHARCHE

Student (date and signature):



11.02.2022

Appendix B

Gantt Chart

This appendix contains the screen snip of the Gantt chart used for doing this thesis. Even though the Gantt chart is finished at the end, for illustration purpose, Gantt chart somewhere in the middle is shown here. Gantt chart was made in SharePoint and put as one of the tabs in Microsoft teams in order to make it more interactive and easily visible for supervisors.

General Thesis RoadMap - Thesis Workspace and... presentation_thesis_v1									
Title	TaskStart	TaskDue	GenIt	Duration	Status	Link to Documents	Supervisor's comm...	Task Finish Comm...	
Existing Scenario Preparation	1/13/2022	1/21/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	8	Finished			Prof Ru explained in meeting the current status of experiments going on to people from SINTEF & Equinor	
Final Topic Deadline	1/17/2022	2/1/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	15	Finished	Final Topic Folder			
[Meet] Topic Finalizing	1/20/2022	1/21/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	1	Finished			Finalized to conduct study for single phase and then two phase	
Literature Study	2/2/2022	2/14/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	12	Finished	Literature Study Fol...			
[Meet] Discussion regarding pre-requis...	2/3/2022	2/4/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	1	Finished				
[Meet] Experiment Data (Old and New)	2/17/2022	2/18/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	1	Finished				
Data related to single phase flow	2/21/2022	3/7/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	14	Finished	Data Analysis Folder			
Software Adjustment for new data	3/1/2022	3/14/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	13	Finished				
Analyzing Data Collected	3/14/2022	3/21/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	7	Finished				
[Meet] Discussion regarding Data Han...	3/23/2022	3/25/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	2	Finished				
Data Handling for ML model	3/25/2022	4/11/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	17	Finished				
Testing various ML algorithms	4/11/2022	4/22/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	11	In-Progress				
[Meet] ML Model final algorithm	4/20/2022	4/22/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	2	Not-Started				
Model Validation & Testing	4/22/2022	4/29/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	7	Not-Started				
Thesis Submission	5/1/2022	5/18/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	17	Not-Started				
Expo	5/17/2022	5/24/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	7	Not-Started				
Oral Presentation & Exam	6/1/2022	6/24/2022	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec	23	Not-Started				

Appendix C

Experiment Details

Test ID	Start		Stop		Planned flow rates		Measured mass flow rate		Measured single phase rates								Pressure		Temperature		Choke		
	Date and time	Date and time	Date and time	Date and time	Gas rate	Oil rate	Water rate	Ref	Endress	Krohne	Gas	Oil	Oil	Oil	Gas	Water	Water	Water	Upstream	Downstream	Upstream	Downstream	Position
W01	10/02/2020 17:25	10/02/2020 17:35	0	0.0	0	0.0	2.0	2.2	2.1	2.0	0.00	26.9	0.00	810.0	2.0	1077.2	35.92	35.82	68.72	68.72	100.0	100.0	
W02	10/02/2020 17:05	10/02/2020 17:15	0	0.0	0	0.0	5.0	5.4	5.3	5.2	0.00	26.9	0.00	809.2	5.0	1077.0	35.93	35.81	68.43	68.84	100.0	100.0	
W03	10/02/2020 16:45	10/02/2020 16:55	0	0.0	0	0.0	10.0	10.8	10.7	10.6	0.00	26.9	0.00	808.6	10.0	1076.8	36.01	35.79	68.53	68.94	100.0	100.0	
W08	10/02/2020 16:25	10/02/2020 16:35	0	0.0	0	0.0	20.0	21.6	21.5	21.5	0.00	26.8	0.00	809.1	20.0	1076.6	36.31	35.95	68.47	68.90	100.0	100.0	
W09	10/02/2020 16:05	10/02/2020 16:15	0	0.0	0	0.0	30.0	32.3	32.3	32.4	0.00	26.7	0.00	809.5	30.0	1076.5	36.81	36.14	69.31	69.75	100.0	100.0	
W10	10/02/2020 15:50	10/02/2020 16:00	0	0.0	0	0.0	40.0	43.1	43.1	43.3	0.00	26.4	-0.01	809.9	40.0	1076.3	37.48	36.31	69.88	70.42	100.0	100.0	
W11	10/02/2020 15:25	10/02/2020 15:35	0	0.0	0	0.0	50.0	53.8	53.8	54.1	0.00	26.1	-0.01	810.4	50.0	1076.1	38.54	36.83	70.99	71.37	100.0	100.0	
W12	10/02/2020 15:00	10/02/2020 15:10	0	0.0	0	0.0	60.0	64.6	64.6	65.0	0.00	23.8	0.00	811.0	60.0	1075.9	39.52	37.16	70.88	71.34	100.0	100.0	
OT30	07/02/2020 08:40	07/02/2020 08:55	0	30.0	0	0.0	24.4	24.9	24.9	24.9	0.00	23.4	29.99	812.8	0.0	1084.5	36.59	35.96	66.59	66.86	100.0	100.0	
OT38	07/02/2020 09:00	07/02/2020 09:10	0	28.0	0	0.0	22.9	22.5	22.5	22.5	0.00	23.3	28.17	812.0	0.0	1083.9	36.41	35.83	67.60	67.87	100.0	100.0	
OT26	07/02/2020 09:15	07/02/2020 09:30	0	26.0	0	0.0	21.2	21.2	21.2	21.2	0.00	23.2	26.10	811.2	0.0	1083.1	36.12	35.61	68.17	68.41	100.0	100.0	
OT24	07/02/2020 09:30	07/02/2020 09:45	0	24.0	0	0.0	19.5	19.6	19.6	19.6	0.00	23.2	24.12	810.5	0.0	1082.5	35.86	35.39	68.44	68.73	100.0	100.0	
OT22	07/02/2020 09:45	07/02/2020 10:00	0	22.0	0	0.0	17.9	18.0	18.0	17.8	0.00	23.1	22.13	809.8	0.0	1081.8	35.61	35.19	68.64	68.86	100.0	100.0	
OT20	07/02/2020 10:00	07/02/2020 10:15	0	20.0	0	0.0	16.3	16.3	16.2	16.2	0.00	23.0	20.10	809.3	0.0	1081.1	35.35	34.99	68.72	68.96	100.0	100.0	
OT18	07/02/2020 10:15	07/02/2020 10:30	0	18.0	0	0.0	14.7	14.7	14.6	14.6	0.00	23.0	18.12	809.2	0.0	1080.2	35.14	34.82	68.75	69.06	100.0	100.0	
OT16	07/02/2020 10:30	07/02/2020 10:45	0	16.0	0	0.0	13.0	13.1	13.0	13.0	0.00	23.0	16.12	809.1	0.0	1079.3	34.96	34.71	68.76	69.08	100.0	100.0	
OT14	07/02/2020 10:45	07/02/2020 11:00	0	14.0	0	0.0	11.4	11.4	11.4	11.3	0.00	22.9	14.12	809.0	0.0	1078.9	34.78	34.60	68.73	69.06	100.0	100.0	
OT12	07/02/2020 11:00	07/02/2020 11:15	0	12.0	0	0.0	9.8	9.8	9.7	9.7	0.00	22.9	12.13	808.9	0.0	1078.7	34.63	34.49	68.83	69.04	100.0	100.0	
OT8	07/02/2020 11:36	07/02/2020 12:02	0	8.0	0	0.0	6.5	6.5	6.3	6.3	0.00	22.8	8.08	808.6	0.0	1078.0	34.42	34.30	69.14	69.03	100.0	100.0	
OT6	07/02/2020 12:05	07/02/2020 12:20	0	6.0	0	0.0	4.9	4.8	4.6	4.6	0.00	22.8	6.00	808.4	0.0	1077.7	34.34	34.25	69.29	68.98	100.0	100.0	
OT4	07/02/2020 12:25	07/02/2020 12:40	0	4.0	0	0.0	3.2	3.2	3.0	3.0	0.00	22.8	4.00	808.2	0.0	1077.4	34.31	34.26	69.01	68.40	100.0	100.0	
OT2	07/02/2020 12:42	07/02/2020 12:57	0	2.0	0	0.0	1.6	1.5	1.4	1.4	0.00	22.8	2.00	808.1	0.0	1077.1	34.35	34.35	68.99	67.68	100.0	100.0	
OT08	07/02/2020 13:10	07/02/2020 13:25	0	40.0	0	0.0	32.4	32.5	32.6	32.6	0.00	22.6	40.00	810.3	0.0	1076.8	39.59	38.64	68.92	69.32	100.0	100.0	
OT09	07/02/2020 13:30	07/02/2020 13:45	0	30.0	0	0.0	24.3	24.3	24.3	24.3	0.00	22.7	30.00	809.6	0.0	1076.7	37.37	36.79	69.72	70.02	100.0	100.0	
OT10	07/02/2020 13:50	07/02/2020 14:05	0	20.0	0	0.0	16.2	16.2	16.1	16.1	0.00	22.7	20.00	808.8	0.0	1076.7	35.74	35.52	69.80	70.07	100.0	100.0	
G02	10/02/2020 09:07	10/02/2020 09:17	200.0	0.0	0.0	0.0	4.9	4.9	4.7	4.7	200.31	24.5	0.00	808.8	-0.1	1091.1	37.69	37.17	65.06	63.83	100.0	100.0	
G03	10/02/2020 09:24	10/02/2020 09:34	180.0	0.0	0.0	0.0	4.4	4.4	4.2	4.2	180.37	24.5	0.00	808.5	0.0	1089.9	37.35	36.92	68.65	67.50	100.0	100.0	
G04	10/02/2020 09:40	10/02/2020 09:50	160.0	0.0	0.0	0.0	4.0	3.9	3.7	3.7	161.91	24.6	0.00	808.2	0.0	1089.1	37.00	36.71	69.12	68.28	100.0	100.0	
G05	10/02/2020 09:56	10/02/2020 10:06	140.0	0.0	0.0	0.0	3.5	3.4	3.2	3.2	139.91	24.7	0.00	808.0	0.0	1088.4	36.63	36.33	69.27	68.32	100.0	100.0	
G06	10/02/2020 10:12	10/02/2020 10:22	120.0	0.0	0.0	0.0	3.0	2.9	2.7	2.7	119.90	24.8	0.00	807.7	0.0	1087.7	36.27	36.08	69.22	68.19	100.0	100.0	
G07	10/02/2020 10:30	10/02/2020 10:40	100.0	0.0	0.0	0.0	2.5	2.5	2.3	2.3	101.15	24.9	0.00	807.4	0.0	1087.9	35.97	35.80	69.11	68.02	100.0	100.0	
G08	10/02/2020 10:50	10/02/2020 11:00	80.0	0.0	0.0	0.0	2.0	1.9	1.7	1.7	80.29	24.9	0.00	807.0	0.0	1088.1	35.62	35.52	68.08	66.89	100.0	100.0	
G09	10/02/2020 11:15	10/02/2020 11:30	60.0	0.0	0.0	0.0	1.5	1.5	1.3	1.3	61.48	24.9	0.00	807.1	0.0	1088.4	35.33	35.30	67.47	66.05	100.0	100.0	
G10	10/02/2020 11:37	10/02/2020 11:52	40.0	0.0	0.0	0.0	1.0	1.0	0.8	0.8	40.81	24.9	0.00	807.5	0.0	1088.7	35.15	35.18	66.76	64.72	100.0	100.0	
G11	10/02/2020 11:58	10/02/2020 12:09	30.0	0.0	0.0	0.0	0.7	0.7	0.4	0.4	28.89	24.8	0.00	808.5	6.4	1077.1	35.03	35.08	66.67	64.00	100.0	100.0	

Appendix D

Tools Used in Thesis : Specifications

Software Used :

MATLAB R2022a (9.12.0.1884302)

Laptop Used :

ASUS ROG Zephyrus G14 GA401II

Processor : AMD Ryzen 5 4600HS with Radeon Graphics, 3000 MHz, 6 Core(s), 12 Logical Processor(s)

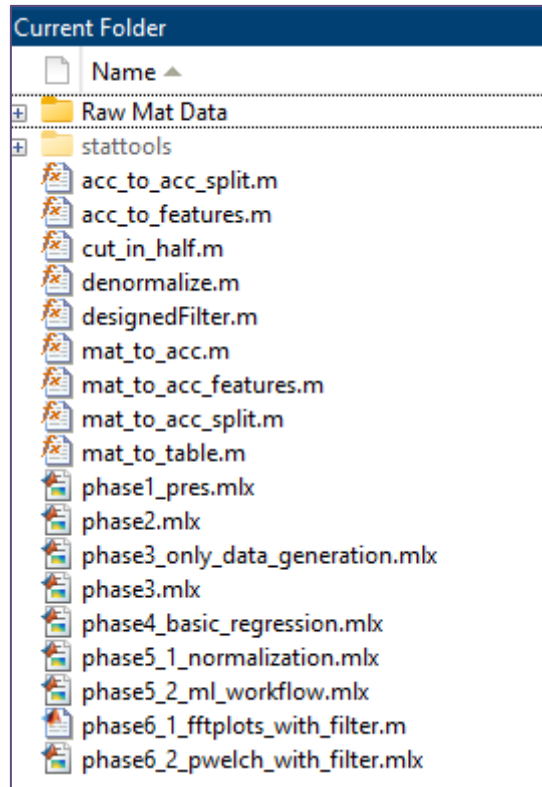
Graphics Processor : NVIDIA GeForce GTX 1650 Ti

Dedicated video memory : 4096 MB GDDR6

Appendix E

Importing Raw Data to MATLAB

The files are arranged in one single folder as shown.



Investigation of the Test Data

Table of Contents

Introduction

Importing the data

- Load everything except accelerometer data

- Inspect accelerometer data

- Load accelerometer features and combine with the other features

Data Exploration

- Histogram and Boxplot

- Descriptive statistics table

- Missing values

- Correlation

- Scatter plots

Introduction

In this report, we investigate the test data to understand its structure and contents. At first, we will convert the structures to a simple mat format. Second, we will show how to reach the data by an example. And finally, we will create descriptive statistics and visuals to better understand the behaviour of the data and detect some potential mistakes.

```
clear, rng default
addpath stattools
mkdir descriptive_figures
```

Warning: Directory already exists.

```
mkdir descriptive_figures\histograms
```

Warning: Directory already exists.

```
mkdir descriptive_figures\boxplots
```

Warning: Directory already exists.

```
mkdir descriptive_figures\scatters
```

Warning: Directory already exists.

Importing the data

Load everything except accelerometer data

In this section, all of the .mat files in the "Raw Mat Data" folder are read and combined into a table ("contents"). Accelerometer data is discarded at this point.

```
fds =
fileDatastore("C:\Users\shail\Documents\Thesis\combined_space_filtered_dataset\R
```

```
aw Mat Data\*.mat", "ReadFcn",
@mat_to_table, "UniformRead", true, "IncludeSubfolders", false);
contents = fds.readall("UseParallel", true) %Use multiple cpus for a quicker
operation
```

contents = 32x51 table

	name	time_start	time_stop	oilRef_q	oilRef_w	...
1	"G02"	10-Feb-2020 09:07:00	10-Feb-2020 09:17:00	6.8961	5.5765	
2	"G03"	10-Feb-2020 09:24:00	10-Feb-2020 09:34:00	6.8904	5.5671	
3	"G04"	10-Feb-2020 09:40:00	10-Feb-2020 09:50:00	6.8850	5.5585	
4	"G05"	10-Feb-2020 09:56:00	10-Feb-2020 10:06:00	6.8813	5.5532	
5	"G06"	10-Feb-2020 10:12:00	10-Feb-2020 10:22:00	6.8761	5.5485	
6	"G07"	10-Feb-2020 10:30:00	10-Feb-2020 10:40:00	6.8729	5.5451	
7	"G08"	10-Feb-2020 10:50:00	10-Feb-2020 11:00:00	6.8738	5.5454	
8	"G09"	10-Feb-2020 11:15:00	10-Feb-2020 11:30:00	6.8731	5.5451	
9	"G10"	10-Feb-2020 11:37:00	10-Feb-2020 11:52:00	6.8747	5.5463	
10	"G11"	10-Feb-2020 11:58:00	10-Feb-2020 12:09:00	0	0	
11	"OT08"	07-Feb-2020 13:10:00	07-Feb-2020 13:25:00	40.0008	32.4139	
12	"OT09"	07-Feb-2020 13:30:00	07-Feb-2020 13:45:00	30.0004	24.2612	
13	"OT10"	07-Feb-2020 11:15:00	07-Feb-2020 11:30:00	10.0841	8.1543	
14	"OT12"	07-Feb-2020 11:00:00	07-Feb-2020 11:15:00	12.1327	9.8119	
	:					

Inspect accelerometer data

A function called `mat_to_acc` converts a raw mat file to a struct that is much easier to work on the accelerometer data. This function is to be used when accelerometer data is to be investigated file by file.

```
acc = mat_to_acc("Raw Mat Data\W01.mat")
```

Accelerometer data can now be reached as `acc.data(:,n)` where `n` is the channel number, from 1 to 3. First channel versus the time is plotted as an example:

```
% plot(acc.time_axis(1:250),acc.data(1:250,1));
% %plot(acc.data(:,1))
% title("1st Ch")
% xlabel("Time")
% ylabel("W01 FFT Measurement")
```


Further operations would also be possible, for instance, one can calculate the magnitude for the acceleration vectors and plot it as well:

```
% magnitude = sqrt(acc.data(:,1).^2 + acc.data(:,2).^2 + acc.data(:,3).^2);
% plot(acc.time_axis(1:250), magnitude(1:250))
% title("Magnitude")
% xlabel("Time")
% ylabel("OT10 Measurement")
```

It is also possible to extract descriptive statistics:

```
% range_of_magnitude = range(magnitude)
% mean_of_magnitude = mean(magnitude)
```

This feature extraction process will be developed with respect to relevant literature and similar projects with operations like smoothing, noise removal,, domain transformation and normalization.

Load accelerometer features and combine with the other features

So far, we combined all the scalar features in a table named "contents" and we opened one .mat file to view it's accelerometer data. In this section, we will add the extracted features from the accelerometer data to the basic features in the "contents" table. Right now, as merely as an example to show how the code works, three features are added to the table, interquartile range (iqr), median and skewness of the magnitude of the accelerometer data.

Following code extracts features for all mat files and combines them in table:

```
fds_a = fileDatastore("Raw Mat Data\*.mat", "ReadFcn",
@mat_to_acc_features, "UniformRead", true);
```

```
Error using fileDatastore
Cannot find files or folders matching: 'Raw Mat Data\*.mat'.
```

```
accelometer_features = fds_a.readall("UseParallel", true)
```

Following code joins the first table we created ("contents") with the feature table we just created. In the end, we have a table with the basic values from the .mat files and the extracted values from the accelerometer data.

```
% Join tables
combined = outerjoin(contents, accelometer_features, "Keys", "name", ...
    "MergeKeys", true);

combined
```

Since the .mat files are now converted to tabular format, we can easily extract it to formats like csv:

```
writetable(combined, "combined1.csv")
```

Data Exploration

Histogram and Boxplot

Histograms and boxplots are common tools in data exploration. We create those for each of the numeric table columns (including the columns generated from the accelerometer data). Results are saved in the folder "descriptive_figures".

```
% descriptiveTableColumnsVisuals(combined);
```

Descriptive statistics table

Again, for all of the columns, some common statistics are reported.

```
% stats_table = descriptiveTableColumns(combined)
```

Missing values

The dataset has very little amount of missing values:

```
column_names = string(combined.Properties.VariableNames)';  
for column = 2:1:width(combined)  
    missing_amount(column,1) = sum(ismissing(combined(:,column)));  
end  
missings = table(column_names, missing_amount);  
missings = sortrows(missings, 'missing_amount', 'descend')
```

It seems that Krohne was not calculated for four experiments.

Correlation

As a part of understanding the data, Pearson correlation coefficient is calculated between all numerical columns.

```
numerical_parameters = combined(:,4:end);  
correlations =  
array2table(corr(table2array(numerical_parameters), "rows", "pairwise", "type", "Pearson"));  
vn = string(combined.Properties.VariableNames);  
correlations.Properties.RowNames = vn(4:end);  
correlations.Properties.VariableNames = vn(4:end)
```

Since it is harder to see which correlation coefficients are bigger (by the means of absolute values), they are also placed in the figures in the following section.

Scatter plots

In addition to the correlation coefficients and other statistical tests, scatter plots of each possible column pair is also created to detect relationships.

```
% for g=1:1:height(correlations)
%   for gg=1:1:height(correlations)
%     if g>gg
%       f=figure;
%       scatter(combined{:,g+3},combined{:,gg+3})
%       lsline
%       xlabel(vn(g+3),"Interpreter","none")
%       ylabel(vn(gg+3),"Interpreter","none")
%       title(vn(g+3) + " vs " + vn(gg+3),"Interpreter","none")
%       legend("R = " + correlations{g,gg})
%
saveas(f,"descriptive_figures"+filesep+"scatters"+filesep+vn(g+3)+"_vs_" +
vn(gg+3) + "_correlation.jpg");
%       close(f)
%     end
%   end
% end
```

Appendix F

Accelerometer Data Plots : MATLAB code

```
clear, rng default, close all
%% New Example
files = ["W10", "W10"];
filter = [1 0];
combined_fft_plot(files, filter);

%% Old Examples

%Combined FFT Plot
%Combined plot of W10, OT08 & G10 (since they are having same flow rate i.e 40 m3/h
but for differnt flow type)
files = ["W10", "OT08", "G10"];
combined_fft_plot(files);

%All Ws
files = extractBefore(deblank(string(ls("Raw Mat Data\W*"))), ".mat");
combined_fft_plot(files);

%All Gs
files = extractBefore(deblank(string(ls("Raw Mat Data\G*"))), ".mat");
combined_fft_plot(files);

%All OTs
files = extractBefore(deblank(string(ls("Raw Mat Data\OT*"))), ".mat");
combined_fft_plot(files);

function figs = combined_fft_plot(files, filter)

if nargin == 1
    %filter = zeros(1, numel(files));%default behaviour, no filter
    filter = ones(1, numel(files));%default behaviour, filter
end

file_names = "Raw Mat Data" + filesep + files + ".mat";

for channel = 1:1:3 %%%
    figs(channel) = figure;
    hold on;
    ffts = [];
    for file_id = 1:1:numel(files)
        acc = mat_to_acc(file_names(file_id));
        dts(file_id) = acc.dt;

        signal = acc.data(:, channel);
        if filter(file_id) == 1
            signal = designedFilter(signal, 1/acc.dt);
        end

        ffts{file_id} = 2*cut_in_half(abs(fft(signal)))';
        f{file_id} = (0:length(ffts{file_id})-1)*(1/dts(file_id))/length(ffts
{file_id});
        clearvars acc
        lengths(file_id) = length(ffts{file_id});
    end
end
```

```
end
normalized = rescale([ffts{:}]); %note rescale
clearvars ffts
indices = [0 cumsum(lengths)];
for file_id = 1:1:numel(files)
    index_low = indices(file_id) + 1;
    index_high = indices(file_id + 1);
    plot(f{file_id},normalized(index_low:index_high));
end
xlabel("Frequency (Hz)");
ylabel("Normalized Amplitude");
title("Channel " + channel);

legend_text = files;
legend_text(filter==1) = legend_text(filter==1) + " F.";

legend(legend_text,"Location","eastoutside");
hold off
end
end
```

Appendix G

Accelerometer Data Processing

Table of Contents

Data Import.....	1
Import Basic Information and Categorize.....	1
Import Accelerometer Features.....	2
For Example.....	2
For Real.....	5
Combine Accelerometer Data and Basic Information.....	5

Data Import

Import Basic Information and Categorize

We import basic columns (the ones except acc) as usual:

```
clear;
mkdir descriptive_figures
```

Warning: Directory already exists.

```
mkdir descriptive_figures\histograms
```

Warning: Directory already exists.

```
mkdir descriptive_figures\boxplots
```

Warning: Directory already exists.

```
mkdir descriptive_figures\scatters
```

Warning: Directory already exists.

```
addpath stattools\
tic
raw_path = "Raw Mat Data\*.mat";
fds = fileDatastore(raw_path,"ReadFcn", @mat_to_table,"UniformRead",true,"IncludeSubfolders",false);
contents = fds.readall("UseParallel",true); %Use multiple cpus for a quicker operation
```

Extract category from name:

```
contents.category = categorical(extractBefore(contents.name,digitsPattern(1)));
```

Create "average_q" flow rate to be used as target variable:

```
all_average_q = (contents.Endres_q + contents.Krohne_q)/(2);
contents.average_q(~isnan(contents.Krohne_q)) = all_average_q(~isnan(contents.Krohne_q));
contents.average_q(isnan(contents.Krohne_q)) = contents.Endres_q(isnan(contents.Krohne_q));
```

Select the required input with respect to flow category:

```
active_ref(contents.category=="G",1) = contents.gasRef_q(contents.category=="G");
active_ref(contents.category=="OT",1) = contents.oilRef_q(contents.category=="OT");
```



```
active_ref(contents.category=="W",1) = contents.watRef_q(contents.category=="W");
contents.active_ref = active_ref;
```

Select only the columns we will use (either for grouping, predictions or target)

```
useful = contents(:,["category","name","temp_in","temp_out","press_in","press_out","STec_rho",
```

Import Accelerometer Features

For Example

I'll show how the variables are generated by using one example file. At first, we get the usual acc struct that includes accelerometer sensor data. It is not split yet.

```
acc = mat_to_acc("Raw Mat Data\G11.mat")
```

```
acc = struct with fields:
    dt: 1.9531e-05
    time: 10-Feb-2020 11:59:24
    data: [12288000x3 double]
    filename: "G11"
    name: "G11"
    time_axis: [10-Feb-2020 11:59:24    10-Feb-2020 11:59:24    10-Feb-2020 11:59:24    10-Feb-2020 11:59:24
    duration_axis: [00:00:00    00:00:00    00:00:00    00:00:00    00:00:00    00:00:00    00:00:00    00:00:00
```

This new function "acc_to_acc_split" accepts acc structures (as it is generated above), splits the signal with the hard coded duration if 60 seconds and 50% overlap and returns mean signal values for each bin:

```
acc_split_tabular = acc_to_acc_split(acc)
```

```
acc_split_tabular = 94x102 table
```

...

	name	time	peak_value_1_1	peak_value_2_1	peak_value_3_1
1	"G11"	0 sec	939.8540	117.0120	939.8540
2	"G11"	5 sec	935.3531	116.2193	935.3531
3	"G11"	10 sec	930.3504	112.2254	930.3504
4	"G11"	15 sec	923.9186	111.1407	923.9186
5	"G11"	20 sec	923.3273	108.0617	923.3273
6	"G11"	25 sec	917.3268	105.5385	917.3268
7	"G11"	30 sec	913.0665	103.0874	913.0665
8	"G11"	35 sec	907.1930	100.9038	907.1930
9	"G11"	40 sec	120.6305	904.1000	904.1000
10	"G11"	45 sec	133.3369	901.6020	901.6020
11	"G11"	50 sec	114.9123	899.9554	899.9554
12	"G11"	55 sec	901.9232	95.7240	901.9232

	name	time	peak_value_1_1	peak_value_2_1	peak_value_3_1
13	"G11"	60 sec	896.4178	95.1190	896.4178
14	"G11"	65 sec	892.7634	95.7652	892.7634
15	"G11"	70 sec	892.5700	93.8898	892.5700
16	"G11"	75 sec	887.7307	94.0705	887.7307
17	"G11"	80 sec	881.8858	99.9805	881.8858
18	"G11"	85 sec	881.4912	98.0483	881.4912
19	"G11"	90 sec	875.3679	101.6665	875.3679
20	"G11"	95 sec	872.2985	101.3658	872.2985
21	"G11"	100 sec	868.4773	102.1048	868.4773
22	"G11"	105 sec	865.6083	105.4448	865.6083
23	"G11"	110 sec	858.6680	106.2777	858.6680
24	"G11"	115 sec	858.4149	108.0058	858.4149
25	"G11"	120 sec	851.2797	109.3169	851.2797
26	"G11"	125 sec	846.4181	110.6482	846.4181
27	"G11"	130 sec	141.2236	844.0637	844.0637
28	"G11"	135 sec	139.3006	842.8309	842.8309
29	"G11"	140 sec	166.2708	837.1658	837.1658
30	"G11"	145 sec	830.4737	115.2311	830.4737
31	"G11"	150 sec	826.4937	119.0370	826.4937
32	"G11"	155 sec	818.3498	121.7659	818.3498
33	"G11"	160 sec	812.8049	123.3467	812.8049
34	"G11"	165 sec	806.1078	122.0055	806.1078
35	"G11"	170 sec	806.7952	121.7574	806.7952
36	"G11"	175 sec	802.1173	125.8531	802.1173
37	"G11"	180 sec	797.7815	126.0031	797.7815
38	"G11"	185 sec	787.1332	127.0848	787.1332
39	"G11"	190 sec	783.8144	127.0983	783.8144
40	"G11"	195 sec	143.0505	778.2587	778.2587
41	"G11"	200 sec	782.3865	130.0403	782.3865
42	"G11"	205 sec	775.6748	130.6173	775.6748
43	"G11"	210 sec	769.7599	133.1360	769.7599
44	"G11"	215 sec	769.4961	133.8544	769.4961
45	"G11"	220 sec	766.0338	133.4076	766.0338

	name	time	peak_value_1_1	peak_value_2_1	peak_value_3_1
46	"G11"	225 sec	762.0754	135.8787	762.0754
47	"G11"	230 sec	757.2138	137.4520	757.2138
48	"G11"	2.5 sec	938.9602	118.6046	938.9602
49	"G11"	7.5 sec	932.9255	112.2592	932.9255
50	"G11"	12.5 sec	926.7545	112.7631	926.7545
51	"G11"	17.5 sec	113.5490	923.7237	923.7237
52	"G11"	22.5 sec	920.2350	107.3312	920.2350
53	"G11"	27.5 sec	914.6670	107.2824	914.6670
54	"G11"	32.5 sec	910.1511	104.1020	910.1511
55	"G11"	37.5 sec	906.3944	102.1886	906.3944
56	"G11"	42.5 sec	148.6048	905.4845	905.4845
57	"G11"	47.5 sec	117.2297	899.9725	899.9725
58	"G11"	52.5 sec	902.3343	96.5751	902.3343
59	"G11"	57.5 sec	899.0494	94.5755	899.0494
60	"G11"	62.5 sec	895.5087	95.9983	895.5087
61	"G11"	67.5 sec	892.2785	93.8178	892.2785
62	"G11"	72.5 sec	889.0020	96.2250	889.0020
63	"G11"	77.5 sec	883.8399	97.1707	883.8399
64	"G11"	82.5 sec	111.8211	881.5444	881.5444
65	"G11"	87.5 sec	878.0424	100.7340	878.0424
66	"G11"	92.5 sec	873.7918	101.2094	873.7918
67	"G11"	97.5 sec	871.6387	102.1223	871.6387
68	"G11"	102.5 sec	864.2257	102.8595	864.2257
69	"G11"	107.5 sec	862.4676	107.0726	862.4676
70	"G11"	112.5 sec	858.6869	110.0589	858.6869
71	"G11"	117.5 sec	853.3808	106.0269	853.3808
72	"G11"	122.5 sec	847.7001	110.7188	847.7001
73	"G11"	127.5 sec	842.0635	112.5980	842.0635
74	"G11"	132.5 sec	136.4733	844.5993	844.5993
75	"G11"	137.5 sec	135.7855	838.4124	838.4124
76	"G11"	142.5 sec	832.5596	114.7667	832.5596
77	"G11"	147.5 sec	828.2279	117.2388	828.2279
78	"G11"	152.5 sec	820.7930	120.6225	820.7930

	name	time	peak_value_1_1	peak_value_2_1	peak_value_3_1
79	"G11"	157.5 sec	816.5324	119.8442	816.5324
80	"G11"	162.5 sec	810.5630	121.6495	810.5630
81	"G11"	167.5 sec	808.0171	122.0627	808.0171
82	"G11"	172.5 sec	803.3905	123.4138	803.3905
83	"G11"	177.5 sec	799.2798	126.5109	799.2798
84	"G11"	182.5 sec	791.3556	125.5407	791.3556
85	"G11"	187.5 sec	784.9528	127.2978	784.9528
86	"G11"	192.5 sec	153.2586	780.7401	780.7401
87	"G11"	197.5 sec	780.9138	131.0519	780.9138
88	"G11"	202.5 sec	779.0349	132.5400	779.0349
89	"G11"	207.5 sec	773.1123	130.1758	773.1123
90	"G11"	212.5 sec	767.2629	134.2479	767.2629
91	"G11"	217.5 sec	768.7841	131.9620	768.7841
92	"G11"	222.5 sec	762.8802	136.0462	762.8802
93	"G11"	227.5 sec	759.1759	134.8933	759.1759
94	"G11"	232.5 sec	755.2138	138.3797	755.2138

The column "time" is the starting time.

For Real

Now, we will split the accelerometer data for all the available files and combine them in one table:

```
fds2 = fileDatastore(raw_path, "ReadFcn", @mat_to_acc_split, "UniformRead", true, "IncludeSubfolders", true);
accs = fds2.readall("UseParallel", true); %Use multiple cpus for a quicker operation
accs
```

Combine Accelerometer Data and Basic Information

```
% Join tables
combined = outerjoin(useful, accs, "Keys", "name", "MergeKeys", true)
combined = movevars(combined, 'time', 'After', 'name');
```

Now, we finally have our data in a form (table) that can be used by machine learning and correlation test algorithms.

```
save combined combined
toc
```


Appendix H

Manual Separation of Training Data and test data

```
%load combined combined
combined
```

```
combined = 16680x114 table
```

	category	name	time	temp_in	temp_out	press_in	press_out
1	G	"G02"	0 sec	65.0639	63.8343	37.6910	37.1686
2	G	"G02"	1 sec	65.0639	63.8343	37.6910	37.1686
3	G	"G02"	2 sec	65.0639	63.8343	37.6910	37.1686
4	G	"G02"	3 sec	65.0639	63.8343	37.6910	37.1686
5	G	"G02"	4 sec	65.0639	63.8343	37.6910	37.1686
6	G	"G02"	5 sec	65.0639	63.8343	37.6910	37.1686
7	G	"G02"	6 sec	65.0639	63.8343	37.6910	37.1686
8	G	"G02"	7 sec	65.0639	63.8343	37.6910	37.1686
9	G	"G02"	8 sec	65.0639	63.8343	37.6910	37.1686
10	G	"G02"	9 sec	65.0639	63.8343	37.6910	37.1686
11	G	"G02"	10 sec	65.0639	63.8343	37.6910	37.1686
12	G	"G02"	11 sec	65.0639	63.8343	37.6910	37.1686
13	G	"G02"	12 sec	65.0639	63.8343	37.6910	37.1686
14	G	"G02"	13 sec	65.0639	63.8343	37.6910	37.1686
			⋮				

```
[~,index] = unique(combined.name);
cases = combined(index, ["category", "name", "average_q"])
```

```
cases = 32x3 table
```

	category	name	average_q
1	G	"G02"	182.3850
2	G	"G03"	169.0726
3	G	"G04"	154.3336
4	G	"G05"	135.7957
5	G	"G06"	116.3570
6	G	"G07"	96.4461
7	G	"G08"	75.0746
8	G	"G09"	57.0950
9	G	"G10"	36.8624
10	G	"G11"	23.7425
11	OT	"OT08"	40.2297

	category	name	average_q
12	OT	"OT09"	30.1061
13	OT	"OT10"	10.0102
14	OT	"OT12"	12.0815

⋮

```
%combined(ismember(combined.name,["OT24","OT26","OT28","OT30"]),:) = []
[~,index] = unique(combined.name);
cases = combined(index,["category","name","average_q"])
```

cases = 32x3 table

	category	name	average_q
1	G	"G02"	182.3850
2	G	"G03"	169.0726
3	G	"G04"	154.3336
4	G	"G05"	135.7957
5	G	"G06"	116.3570
6	G	"G07"	96.4461
7	G	"G08"	75.0746
8	G	"G09"	57.0950
9	G	"G10"	36.8624
10	G	"G11"	23.7425
11	OT	"OT08"	40.2297
12	OT	"OT09"	30.1061
13	OT	"OT10"	10.0102
14	OT	"OT12"	12.0815

⋮

```
gs = groupcounts(cases,"category")
```

gs = 3x3 table

	category	GroupCount	Percent
1	G	10	31.2500
2	OT	15	46.8750
3	W	7	21.8750

```
% Compute group summary
summary = groupsummary(cases,"category",["mean","median","max","min","range",...
    "std","var"],vartype("numeric"))
```

summary = 3x9 table

	category	GroupCount	mean_average_q	median_average_q	max_average_q
1	G	10	104.7164	106.4015	182.3850
2	OT	15	19.0355	20.1247	40.2297
3	W	7	22.4829	20.0144	50.2890

```
test_cases = ["G04", "G06", "OT09", "OT22", "W03", "W09"];
test_data = combined(ismember(combined.name, test_cases), :)
```

test_data = 3600x114 table

	category	name	time	temp_in	temp_out	press_in	press_out
1	G	"G04"	0 sec	69.1188	68.2849	37.0036	36.7077
2	G	"G04"	1 sec	69.1188	68.2849	37.0036	36.7077
3	G	"G04"	2 sec	69.1188	68.2849	37.0036	36.7077
4	G	"G04"	3 sec	69.1188	68.2849	37.0036	36.7077
5	G	"G04"	4 sec	69.1188	68.2849	37.0036	36.7077
6	G	"G04"	5 sec	69.1188	68.2849	37.0036	36.7077
7	G	"G04"	6 sec	69.1188	68.2849	37.0036	36.7077
8	G	"G04"	7 sec	69.1188	68.2849	37.0036	36.7077
9	G	"G04"	8 sec	69.1188	68.2849	37.0036	36.7077
10	G	"G04"	9 sec	69.1188	68.2849	37.0036	36.7077
11	G	"G04"	10 sec	69.1188	68.2849	37.0036	36.7077
12	G	"G04"	11 sec	69.1188	68.2849	37.0036	36.7077
13	G	"G04"	12 sec	69.1188	68.2849	37.0036	36.7077
14	G	"G04"	13 sec	69.1188	68.2849	37.0036	36.7077
⋮							

```
training_data = combined(~ismember(combined.name, test_cases), :)
```

training_data = 13080x114 table

	category	name	time	temp_in	temp_out	press_in	press_out
1	G	"G02"	0 sec	65.0639	63.8343	37.6910	37.1686
2	G	"G02"	1 sec	65.0639	63.8343	37.6910	37.1686
3	G	"G02"	2 sec	65.0639	63.8343	37.6910	37.1686
4	G	"G02"	3 sec	65.0639	63.8343	37.6910	37.1686
5	G	"G02"	4 sec	65.0639	63.8343	37.6910	37.1686
6	G	"G02"	5 sec	65.0639	63.8343	37.6910	37.1686

	category	name	time	temp_in	temp_out	press_in	press_out
7	G	"G02"	6 sec	65.0639	63.8343	37.6910	37.1686
8	G	"G02"	7 sec	65.0639	63.8343	37.6910	37.1686
9	G	"G02"	8 sec	65.0639	63.8343	37.6910	37.1686
10	G	"G02"	9 sec	65.0639	63.8343	37.6910	37.1686
11	G	"G02"	10 sec	65.0639	63.8343	37.6910	37.1686
12	G	"G02"	11 sec	65.0639	63.8343	37.6910	37.1686
13	G	"G02"	12 sec	65.0639	63.8343	37.6910	37.1686
14	G	"G02"	13 sec	65.0639	63.8343	37.6910	37.1686

⋮

```
save dataset_to_be_used_in_ml test_data training_data
```

Appendix I

Normalization of data

```
clear
```

Introduction

It makes sense to normalize same features using same scales. (meanfreq_1 and meanfreq_2 should be normalized together) to not lose their spatial relationship.

For example, imagine these values for meanfreq_1 and meanfreq_2. Note that first three elements are the same.

```
meanfreq_1_example = [1 2 3 3 5 6 2 2];  
meanfreq_2_example = [1 2 3 9 11 12 14];
```

If we normalize them separately, we get different values for first three elements although they have the same unit and magnitude:

```
normalize(meanfreq_1_example)
```

```
ans = 1×8  
-1.1832 -0.5916 0 0 1.1832 1.7748 -0.5916 -0.5916
```

```
normalize(meanfreq_2_example)
```

```
ans = 1×7  
-1.2087 -1.0207 -0.8327 0.2955 0.6715 0.8595 1.2356
```

To solve this, I will combine signal features in one vector, normalize that vector, and then split it back into 4 features. Continuing with the example:

```
meanfreq_all = [meanfreq_1_example, meanfreq_2_example]
```

```
meanfreq_all = 1×15  
1 2 3 3 5 6 2 2 1 2 3 9 11 ...
```

```
meanfreq_all_normalized = normalize(meanfreq_all);  
meanfreq_1_normalized = meanfreq_all_normalized(1:8)
```

```
meanfreq_1_normalized = 1×8  
-0.9384 -0.7076 -0.4769 -0.4769 -0.0154 0.2154 -0.7076 -0.7076
```

```
meanfreq_2_normalized = meanfreq_all_normalized(9:end)
```

```
meanfreq_2_normalized = 1×7  
-0.9384 -0.7076 -0.4769 0.9076 1.3691 1.5999 2.0614
```

Now we got same normalized values for the first three elements.

Normalization of training features

```
load dataset_to_be_used_in_ml.mat training_data  
training_data_normalized= training_data;  
clear training_data
```

Available features:

Note that, in our data set, features that should be scaled together ends with the term "out" or "_4". I'll use this fact to programatically handle this problem, instead of manually writing code for each variable to be normalized together:

```
available_features = string(training_data_normalized.Properties.VariableNames) ;  
variables_ending_with_out = available_features(endsWith(available_features,"out","IgnoreCase"),true);
```

```
variables_ending_with_out = 1x3 string  
"temp_out" "press_out" "MPP_TOut"
```

```
variables_ending_with_4 = available_features(endsWith(available_features,"4","IgnoreCase"),true);
```

```
variables_ending_with_4 = 1x25 string  
"peak_value..." "peak_value..." "peak_value..." "statelevels_fd_l..." "statelevels_fd_hi..."
```

```
temp_all = [training_data_normalized.temp_in; training_data_normalized.temp_out];  
press_all = [training_data_normalized.press_in; training_data_normalized.press_out];  
MPP_Tall = [training_data_normalized.MPP_TIn; training_data_normalized.MPP_TOut];  
[temp_normalized, normalization.temp_in.mu, normalization.temp_in.sigma] = normalize(temp_all);  
[press_normalized, normalization.press_in.mu, normalization.press_in.sigma] = normalize(press_all);  
[MPP_normalized, normalization.MPP_TIn.mu, normalization.MPP_TIn.sigma] = normalize(MPP_Tall);
```

Note that I saved normalisation mean and std to be able to replicate the same process on test data. We will use same parameters for _out versions as well.

```
normalization.temp_out = normalization.temp_in;  
normalization.press_out = normalization.press_in;  
normalization.MPP_TOut = normalization.MPP_TIn;
```

```
training_data_normalized.temp_in = temp_normalized(1:(end/2));  
training_data_normalized.temp_out = temp_normalized(((end/2)+1):end);  
  
training_data_normalized.press_in = press_normalized(1:(end/2));  
training_data_normalized.press_out = press_normalized(((end/2)+1):end);  
  
training_data_normalized.MPP_TIn = MPP_normalized(1:(end/2));  
training_data_normalized.MPP_TOut = MPP_normalized(((end/2)+1):end);
```

```
training_data_normalized
```

```
training_data_normalized = 13080x114 table
```

	category	name	time	temp_in	temp_out	press_in	press_out
1	G	"G02"	0 sec	-1.6227	-2.3303	1.6186	1.1113

	category	name	time	temp_in	temp_out	press_in	press_out
2	G	"G02"	1 sec	-1.6227	-2.3303	1.6186	1.1113
3	G	"G02"	2 sec	-1.6227	-2.3303	1.6186	1.1113
4	G	"G02"	3 sec	-1.6227	-2.3303	1.6186	1.1113
5	G	"G02"	4 sec	-1.6227	-2.3303	1.6186	1.1113
6	G	"G02"	5 sec	-1.6227	-2.3303	1.6186	1.1113
7	G	"G02"	6 sec	-1.6227	-2.3303	1.6186	1.1113
8	G	"G02"	7 sec	-1.6227	-2.3303	1.6186	1.1113
9	G	"G02"	8 sec	-1.6227	-2.3303	1.6186	1.1113
10	G	"G02"	9 sec	-1.6227	-2.3303	1.6186	1.1113
11	G	"G02"	10 sec	-1.6227	-2.3303	1.6186	1.1113
12	G	"G02"	11 sec	-1.6227	-2.3303	1.6186	1.1113
13	G	"G02"	12 sec	-1.6227	-2.3303	1.6186	1.1113
14	G	"G02"	13 sec	-1.6227	-2.3303	1.6186	1.1113
⋮							

```

for variable_id = 1:1:numel(variables_ending_with_4)
    current_variable_4 = variables_ending_with_4(variable_id);
    current_variable = extractBefore(current_variable_4, "_4");
    current_columns = available_features(startsWith(available_features,current_variable+"_"));
    current_table = training_data_normalized(:,current_columns);
    current_all = current_table{:,:}(:);
    [current_normalized_all, current_mu, current_sigma] = normalize(current_all);
    for column_id = 1:1:4
        modified_variable = current_columns(column_id);
        normalization.(modified_variable).mu = current_mu;
        normalization.(modified_variable).sigma = current_sigma;
        current_normalized = current_normalized_all((1+height(training_data_normalized))*(column_id));
        training_data_normalized.(modified_variable) = current_normalized;
    end
end
end

```

training_data_normalized

training_data_normalized = 13080x114 table

	category	name	time	temp_in	temp_out	press_in	press_out
1	G	"G02"	0 sec	-1.6227	-2.3303	1.6186	1.1113

	category	name	time	temp_in	temp_out	press_in	press_out
2	G	"G02"	1 sec	-1.6227	-2.3303	1.6186	1.1113
3	G	"G02"	2 sec	-1.6227	-2.3303	1.6186	1.1113
4	G	"G02"	3 sec	-1.6227	-2.3303	1.6186	1.1113
5	G	"G02"	4 sec	-1.6227	-2.3303	1.6186	1.1113
6	G	"G02"	5 sec	-1.6227	-2.3303	1.6186	1.1113
7	G	"G02"	6 sec	-1.6227	-2.3303	1.6186	1.1113
8	G	"G02"	7 sec	-1.6227	-2.3303	1.6186	1.1113
9	G	"G02"	8 sec	-1.6227	-2.3303	1.6186	1.1113
10	G	"G02"	9 sec	-1.6227	-2.3303	1.6186	1.1113
11	G	"G02"	10 sec	-1.6227	-2.3303	1.6186	1.1113
12	G	"G02"	11 sec	-1.6227	-2.3303	1.6186	1.1113
13	G	"G02"	12 sec	-1.6227	-2.3303	1.6186	1.1113
14	G	"G02"	13 sec	-1.6227	-2.3303	1.6186	1.1113

⋮

% Normalize Data

```

data_variables = ["STec_rho", "MPP_pIn", "MPP_dp", "active_ref"];
[training_data_normalized, centerValue, scaleValue] = normalize(training_data_normalized, ...
    "DataVariables", data_variables);
for data_variable_id = 1:1:numel(data_variables)
    variable = data_variables(data_variable_id);
    normalization.(variable).mu = centerValue.(variable);
    normalization.(variable).sigma = scaleValue.(variable);
end
training_data_normalized

```

training_data_normalized = 13080x114 table

...

	category	name	time	temp_in	temp_out	press_in	press_out
1	G	"G02"	0 sec	-1.6227	-2.3303	1.6186	1.1113
2	G	"G02"	1 sec	-1.6227	-2.3303	1.6186	1.1113
3	G	"G02"	2 sec	-1.6227	-2.3303	1.6186	1.1113
4	G	"G02"	3 sec	-1.6227	-2.3303	1.6186	1.1113
5	G	"G02"	4 sec	-1.6227	-2.3303	1.6186	1.1113
6	G	"G02"	5 sec	-1.6227	-2.3303	1.6186	1.1113
7	G	"G02"	6 sec	-1.6227	-2.3303	1.6186	1.1113
8	G	"G02"	7 sec	-1.6227	-2.3303	1.6186	1.1113
9	G	"G02"	8 sec	-1.6227	-2.3303	1.6186	1.1113

	category	name	time	temp_in	temp_out	press_in	press_out
10	G	"G02"	9 sec	-1.6227	-2.3303	1.6186	1.1113
11	G	"G02"	10 sec	-1.6227	-2.3303	1.6186	1.1113
12	G	"G02"	11 sec	-1.6227	-2.3303	1.6186	1.1113
13	G	"G02"	12 sec	-1.6227	-2.3303	1.6186	1.1113
14	G	"G02"	13 sec	-1.6227	-2.3303	1.6186	1.1113

⋮

```
%denormalized = denormalize(training_data, normalization);
```

```
%a = load("dataset_to_be_used_in_ml.mat","training_data");
%isequal(a.training_data, denormalized)
```

```
load dataset_to_be_used_in_ml.mat test_data
test_data_normalized = normalize_custom(test_data,normalization);

save dataset_to_be_used_in_ml_normalized normalization test_data_normalized training_data_normalized
```


Appendix J

Designed Filter

```
function [filtered,Hd] = designedFilter(signal, Fs)
%DESIGNEDFILTER Returns a discrete-time filter object.

% MATLAB Code
% Generated by MATLAB(R) 9.12 and DSP System Toolbox 9.14.
% Generated on: 27-Apr-2022 05:51:23

% Butterworth Bandpass filter designed using FDESIGN.BANDPASS.

% All frequency values are in Hz.
%Fs = 51200; % Sampling Frequency

N = 4; % Order
Fc1 = 10; % First Cutoff Frequency
Fc2 = 15000; % Second Cutoff Frequency

% Construct an FDESIGN object and call its BUTTER method.
h = fdesign.bandpass('N,F3dB1,F3dB2', N, Fc1, Fc2, Fs);
Hd = design(h, 'butter');

filtered = filter(Hd,signal);

% [EOF]
end
```

Appendix K

USN Data Processing

```

clear
clear csv_to_acc
%get a list of csv files
csv_fds = fileDatastore("Raw Mat Data\New accelerometer
data\","IncludeSubfolders",true,"FileExtensions",".csv","ReadFcn",@csv_to_table,
"UniformRead",true)

```

csv_fds =

FileDatastore with properties:

```

Files: {
' ...\Raw Mat Data\New accelerometer data\water_25_acc_1.csv';
' ...\Raw Mat Data\New accelerometer data\water_25_acc_2.csv';
' ...\Raw Mat Data\New accelerometer data\water_35_acc_1.csv'
... and 7 more
}
Folders: {
' ...\Thesis\usn_data_combined_space\Raw Mat Data\New
accelerometer data'
}
UniformRead: 1
ReadMode: 'file'
BlockSize: Inf
PreviewFcn: @csv_to_table
SupportedOutputFormats: ["txt" "csv" "xlsx" "xls" "parquet" "parq"
"png" "jpg" "jpeg" "tif" "tiff" "wav" "flac" "ogg" "mp4" "m4a"]
ReadFcn: @csv_to_table
AlternateFileSystemRoots: {}

```

```

%a = csv_to_table(d("Raw Mat Data\New accelerometer
data\water\water_35_acc_1.csv"))

```

%read them

```

csv_test_data = csv_fds.readall("UseParallel",false) %this errors because of
one broken experment we talked about, I ma

```

csv_test_data_water_2 = 11617x54 table

	name	active_ref	category	time	peak_value_1_1	...
1	"W_water_25_acc_1"	25	"W"	0 sec	8.4792	
2	"W_water_25_acc_1"	25	"W"	1 sec	11.4459	
3	"W_water_25_acc_1"	25	"W"	2 sec	8.7447	
4	"W_water_25_acc_1"	25	"W"	3 sec	8.9975	
5	"W_water_25_acc_1"	25	"W"	4 sec	8.6052	
6	"W_water_25_acc_1"	25	"W"	5 sec	8.1794	
7	"W_water_25_acc_1"	25	"W"	6 sec	11.1384	
8	"W_water_25_acc_1"	25	"W"	7 sec	10.3960	

	name	active_ref	category	time	peak_value_1_1	...
9	"W_water_25_acc_1"	25	"W"	8 sec	9.6348	
10	"W_water_25_acc_1"	25	"W"	9 sec	8.7550	
11	"W_water_25_acc_1"	25	"W"	10 sec	9.3803	
12	"W_water_25_acc_1"	25	"W"	11 sec	13.5163	
13	"W_water_25_acc_1"	25	"W"	12 sec	10.2929	
14	"W_water_25_acc_1"	25	"W"	13 sec	9.2914	
						:

```

save csv_test_data csv_test_data
function out = csv_to_table(file)
%Make sure that current dataset_to_be_used_in_ml_normalized.mat file in the
%folder is up to date.

% Even though normalization of usn test data is mentioned in this function.
% It is commented out below i.e (out = normalize_custom(r,normalization))
% It was removed later due to it giving incorrect values for testing on
% Equinor trained model. The reasons are explained in report.
% So direct utilization of accelroemter features is done using directly code
% out = r
persistent normalization
if isempty(normalization)
    load("dataset_to_be_used_in_ml_normalized", "normalization");
end
try
    if ~endsWith(file,"1.csv")
        out = [];
        return
    end
[acc] = csv_to_acc(string(file));
r= acc_to_acc_split(acc);
% out = normalize_custom(r,normalization);
out = r;
catch er
    disp(file)
    disp(er.message)
    out = [];
end

if width(out) ~=54 && width(out) ~=0
    disp(string(file) + "has weird amount of columns. " + string(width(out)));
    out = [];
end
end

```

end