

FMH606 Master's Thesis 2022  
Industrial IT and Automation

# **Multiphase flow estimation using accelerometers, ultrasonic sensors, and machine learning**



Ashim Khadka

**Course:** FMH606 Master's Thesis, 2022

**Title:** Multiphase flow estimation using accelerometers, ultrasonic sensors, and machine learning

**Number of pages:** 87

**Keywords:** Machine learning, supervised learning, multiphase flow, accelerometers, ultrasonic sensors, flow regime classification, flow rate

**Student:** Ashim Khadka

**Supervisor:** Håkon Viumdal, Ru Yan, Saba Mylvaganam

**External partner:** Equinor, SINTEF

**Summary:**

The new sensors, accelerometers, and ultrasonic sensors were installed in the existing multimodal sensor suite used in the multiphase flow in the process hall in Campus Porsgrunn of USN. Multiphase flow experiments were conducted, and measurements were recorded using different data acquisition systems.

The collected measurements were preprocessed and analyzed to extract the useful features from the raw signals. To classify the flow regimes, the Classification Learner application was used, and the results from different models were compared based on their accuracies. The feature selection technique was implemented to evaluate the importance of each feature. Then, the models were trained with the new features. As a result, the accuracy of the Ensemble model with Bagged Trees was increased from 94.2 to 97.1%. The Random Forest and Decision Trees techniques were also implemented to classify the flow regimes. The flow rates of oil, water, and gas were successfully predicted using the Regression Learner tool. However, the measurements from ultrasonic sensors were not satisfactory due to technical problems related to wedge. In this case, the new wedge was designed and manufactured for the experiment but was unable to implement and collect measurements for this work due to time constraints.

# Preface

This thesis was written as a part of the FMH606 master thesis course 2022 at the University of South-Eastern Norway (USN). The master's program is conducted under the Department of Electrical Engineering, Information Technology, and Cybernetics at campus Porsgrunn. The work is being done in collaboration with Equinor and SINTEF in the USN process hall's multiphase flow rig. The research work was supervised by Håkon Viumdal, Ru Yan, and Saba Mylvaganam as well as Equinor's Kjetil Fjalestad and SINTEF's Tonni Franke Johansen as external partners.

I would like to express my gratitude to my supervisors for their continuous support and guidance that made it possible to complete it within the given time frame. I would also like to thank my external partners for their encouragement and assistance throughout this master's thesis. I would also like to remember the Department of Electrical Engineering, Information Technology, and Cybernetics for giving me a chance to work on this experiment. I would like to dedicate this thesis to my parents for their immense effort and motivation.

Porsgrunn, May 25, 2022

Ashim Khadka

# Contents

Preface.....	3
Nomenclature .....	6
<b>1 Introduction .....</b>	<b>11</b>
1.1 Report Structure and Outline .....	12
<b>2 Background .....</b>	<b>13</b>
2.1 Previous works .....	13
2.2 Multiphase flow .....	14
2.3 Multiphase flow metering .....	16
2.4 Sensors .....	16
2.4.1 Accelerometer .....	16
2.4.2 Ultrasonic sensor .....	17
2.5 Machine learning.....	20
2.5.1 Decision tree .....	20
2.5.2 Random forest .....	21
2.5.3 Ensemble Method .....	22
2.5.4 Classification and Regression Learner Application .....	22
2.6 Data acquisitions system .....	23
2.6.1 Vibration monitoring system .....	23
2.6.2 US-Key .....	24
2.6.3 Data logging and control program for test rig .....	26
<b>3 Multiphase flow rig.....</b>	<b>27</b>
3.1 New experimental setup .....	27
3.2 Experimental results.....	30
3.2.1 Experiments with air and water .....	30
3.2.2 Experiments with air, water, and oil.....	32
3.2.3 Experiment with the ultrasonic transducer .....	33
3.2.4 Design of wedge .....	34
3.2.5 Final experiment with additional measurements .....	36
3.2.6 Ultrasonic measurements from transparent section.....	37
3.2.7 Single phase experiment for Sailesh's thesis.....	40
<b>4 Pre-processing and feature extraction .....</b>	<b>41</b>
4.1 Pre-processing of collected measurements.....	41
4.2 Signal analysis of acoustic signal .....	44
4.3 Classification with acoustic signals.....	51
4.4 Data preparation .....	52
4.5 Data preparation for image processing .....	53
<b>5 Results and Discussion .....</b>	<b>54</b>
5.1 Classification of flow regimes.....	54
5.1.1 Results from Classification Learner tool .....	54
5.1.2 Results from Decision Tree and Random Forest technique .....	59
5.1.3 Results from image processing .....	60
5.2 Flow rates prediction using Regression Learner tool.....	61
5.2.1 Gas flow rates prediction .....	61
5.2.2 Water flow rates prediction .....	63
5.2.3 Oil flow rates prediction .....	64

<b>6 Conclusion.....</b>	<b>67</b>
<b>7 Future work.....</b>	<b>68</b>
<b>8 Bibliography.....</b>	<b>69</b>

# Nomenclature

AI	Artificial Intelligence
CNN	Convolutional Neural Network
MATLAB	Matrix Laboratory, software from MathWorks
P&ID	Piping & Instrumentation Diagram
ML	Machine Learning
LSTM	Long Short-Term Memory
ECT	Electrical Capacitance Tomography
EIT	Electrical Impedance Tomography
GRD	Gamma-Ray Densitometer
PDT	Pressure Differential Transmitters
RNN	Recurrent Neural Network
SVM	Support Vector Machine
ML	Machine Learning
RMS	Root Mean Square
PT	Pressure Transmitter
DPT	Differential Pressure Transmitter
WC	Water-Cut
USN	University of South-Eastern Norway
EDA	Exploratory Data Analysis
PCA	Principal Component Analysis
RMSE	Root Mean Square Error
MSE	Mean Squared Error
MAE	Mean Absolute Error
SNR	Signal to Noise Ratio

## List of Tables

Table 1 Description of the used sensor. ....	17
Table 2 Variations of acoustic velocity, density, and acoustic impedance on different mediums [9]. ....	19
Table 3 Specification of the system. ....	28
Table 4 Duration of each experiment. ....	30
Table 5 Calculation of the refracted shear angle and distances from the different angles of incidence. ....	35
Table 6 Statistical and frequency features of the signal[20].....	51

# List of Figures

Figure 1 Overview of the system for predicting the multiphase flow rates and flow regimes. .....	13
Figure 2 Flow regime map of two-phase flow [4] .....	14
Figure 3 Plug, slug, annular, stratified, and wavy flow regimes in the horizontal pipe[2].....	15
Figure 4 Monostatic and bistatic transducer [10] .....	18
Figure 5 Ultrasonic sensor with A and B representing diameter and height of sensor.....	19
Figure 6 Supervised and unsupervised machine learning algorithms [11] .....	20
Figure 7 Decision Tree algorithm structure[12]. .....	21
Figure 8 Random Forest algorithm structure[12] .....	21
Figure 9 Bagging and Boosting method of Ensemble method[14].....	22
Figure 10 Different models present in Classification and Regression Learner applications [15] .....	23
Figure 11 Communication between Twave T8 and laptop using a router[8]. .....	24
Figure 12 Main screen of TWave T8 system.....	24
Figure 13 Main screen of US-Key system.....	25
Figure 14 HMI of Multiphase Rig. ....	26
Figure 15 Piping and Instrumentation Diagram(P&ID) with accelerometers and ultrasonic sensors [3] .....	27
Figure 16 Experimental setup for multiphase flow.....	29
Figure 17 Two accelerometer sensors that are installed just before the transparent section (left) and near the bend of pipe (right).....	29
Figure 18 Position of the ultrasonic sensor in the transparent section of the test rig. ....	30
Figure 19 Experiments performed based on the past test matrix [3] .....	31
Figure 20 Ultrasonic testing in metal sheet to measure the thickness of metal. ....	32
Figure 21 text matrix of multiphase flow of air, water, and oil .....	32
Figure 22 Pitch and catch mode of ultrasonic testing. ....	33
Figure 23 Pulse-echo mode of ultrasonic testing. ....	34
Figure 24 Calculations of different distances and angles for new wedge.....	35
Figure 25 A new wedge for ultrasonic measurements.....	36
Figure 26 new test matrix with 173 experimental data.....	36
Figure 27 Distribution of different flow regimes in the final test matrix. ....	37
Figure 28 Wavy flow output from ultrasonic sensor. ....	38

Figure 29 Stratified flow output from ultrasonic sensor.....	38
Figure 30 Plug flow output from ultrasonic sensor. ....	39
Figure 31 Slug flow output from ultrasonic sensor. ....	39
Figure 32 Annular flow output from ultrasonic sensor.....	40
Figure 33 Test matrix for single phase flow. ....	40
Figure 34 Workflow of data processing for classification and regression model.....	41
Figure 35 Acceleration vs time plot of two acoustic signals. ....	42
Figure 36 Output of new measurements with longer time.....	42
Figure 37 Output of acoustic signals after making adjustment.....	43
Figure 38 Time series plot of accelerometer 1 and accelerometer 2 of Stratified flow.....	44
Figure 39 Time series plot of accelerometer 1 and accelerometer 2 of Wavy flow. ....	45
Figure 40 Time series plot of signals from accelerometer 1 and accelerometer 2 for Annular flow. ....	45
Figure 41 Time series plot of signals from accelerometer 1 and accelerometer 2 for Plug flow. ....	46
Figure 42 Time series plot of signals from accelerometer 1 and accelerometer 2 for Slug flow. ....	46
Figure 43 Accelerometer signal output after implementing FFT. ....	47
Figure 44 Output of the signal after applying pwelch. ....	48
Figure 45 Design of low pass filter using designer learner application.....	48
Figure 46 Comparison between original and filtered signal in frequency domain.....	49
Figure 47 Output of peak estimates of Power Spectral Density. ....	49
Figure 48 Output of find peaks function after specifying the conditions. ....	50
Figure 49 extractSignalFeatures function for extracting features of whole data set.....	50
Figure 50 Steps involved in data preparation process. ....	52
Figure 51 Resize architecture of ResNet-50[23]. ....	53
Figure 52 Results from different models on test data set.....	55
Figure 53 Confusion matrix of Ensemble model. ....	56
Figure 54 Feature selection using ANOVA technique. ....	57
Figure 55 Results of different models on a 25- features test data set. ....	58
Figure 56 New confusion matrix of Ensemble model on 25 features test data set.....	59
Figure 57 Results from Decision Tree and Random Forest models. ....	59
Figure 58 Confusion matrix for stratified, wavy, annular, plug, and slug flow.....	60
Figure 59 Confusion matrix for wavy, annular, and slug flow.....	60

Figure 60 Results of training, and test data sets of Ensemble Boosted Trees. ....61

Figure 61 Actual vs predicted gas flow rates by Ensemble Boosted Trees model. ....62

Figure 62 Predicted vs true response of gas flow rates prediction model. ....62

Figure 63 Results of Trees model for training and test data set.....63

Figure 64 Actual vs predicted water flow rates by Trees model. ....63

Figure 65 Predicted vs true response of water flow rates prediction model.....64

Figure 66 Comparision between training and test results of Ensemble model.....65

Figure 67 True vs predicted oil flow rates. ....65

Figure 68 Predicted vs true response of oil flow rates prediction model.....66

# 1 Introduction

Over the past few decades, the proper measurement of multiphase flow has been evident and subjected to a lot of research to better understand the different parameters of multiphase flows. Many traditional devices have been in use for flow measurement for centuries, but with the advancement of technology each day, more efficient systems are being sought. However, the prediction of multiphase flows is one of the most difficult tasks because of the complexity between the flow geometry and phase changes.

It is vital that any industry measures different parameters and monitors the results properly to ensure smooth operation. In particular, in the oil and gas industry, an accurate measurement of flow rate and flow type plays a huge role in improving production. The fluids are extracted from wells, which are later supplied and processed to obtain the final product. The raw fluid contains a mixture of oil, gas, water, sand, and other additives that are present in the well. It is necessary to know the content of each fluid at different stages of production. Understanding the behavior of distinct flows helps improve the overall efficiency of the system by controlling the production more effectively. So, the accurate measurement of different flows plays a crucial role in determining the performance of the system. Thus, it is important to have a better understanding of the multiphase flow to optimize the safety, productivity, and reliability of the system [1].

In the past, the study of multiphase flow was more challenging due to measurement uncertainty, and time-consuming techniques. In the traditional measurement procedure, different phases need to be separated and processed separately. During the testing process, a plant must be partially or completely shut down to monitor the flow parameters in multiphase flows. Thus, the traditional methods of phase separation for multiphase flow measurement are complex and limited in different aspects as compared to modern techniques. In recent years, the development and advancement of technologies have enabled accurate measurement and monitoring of the parameters of multiphase flow without separating each phase. At present, researchers have been working on the development of new methodologies and are using modern software and tools to measure multiphase flow. Of all the methods, machine learning has proved to be the most promising for multiphase flow measurement. As machine learning continues to advance, its abilities for handling large and complex datasets have increased significantly. In machine learning, large as well as small amounts of information can be processed, and both produce impressive results. Over recent years, there have been numerous attempts at using machine learning techniques for multiphase flow estimation, like the prediction of flow regime with long short-term memory (LSTM) and recurrent neural network (RNN) for two-phase flow, and similar information can be found in these documents [2]– [4]. The estimation of multiphase flow parameters can be accomplished using both supervised and unsupervised methods of learning. Flow regimes are already known in the supervised learning process but unknown in the unsupervised one.

In this work, the new accelerometers and ultrasonic sensors will be installed in the existing multiphase test rig in the process hall in Campus Porsgrunn of USN. The data will be collected by running the experiments. After that, the collected data will be preprocessed and analyzed using machine learning techniques. This thesis aims to utilize these data to estimate different parameters of multiphase flow, mainly focused on the estimation of flow rates and flow regimes.

## 1.1 Report Structure and Outline

In this experimental setup, the new sensors, such as ultrasonic sensors, accelerometers, and data acquisition systems, are installed in the existing multimodal sensor suite used in the multiphase flow rig.

### Chapter 2

This chapter presents information about the fundamentals and background of this thesis. Firstly, it gives an overview of the system and describes the different sensors that are used for this experiment. It also provides information related to past work on multiphase flows. Basic concepts of different flow regimes are explained and a short discussion on different machine learning algorithms is given.

### Chapter 3

Experimental planning of measurement using the multiphase test rig is presented in this chapter. It also explained the experimental setup and modifications made to the existing system on the test rig. The procedures to run the experiments and collect the data are described.

### Chapter 4

In chapter 4, the most important parts of the thesis, i.e., data preprocessing and exploratory data analysis on the dataset, are explained. The feature extraction techniques are explained in this chapter.

### Chapter 5

The results are presented and compared in this chapter.

### Chapter 6

The conclusion chapter summarises the results related to flow regimes and flow rates from different machine learning approaches.

## 2 Background

The background chapter provides information regarding the fundamentals of this thesis and briefly describes past work performed in multiphase flow, sensors, different types of flow regimes, and machine learning techniques. In this work, the different parameters related to multiphase flow are predicted using machine learning (ML) techniques. Figure 1 shows the overview of the system for predicting the multiphase flow rates and flow regimes.

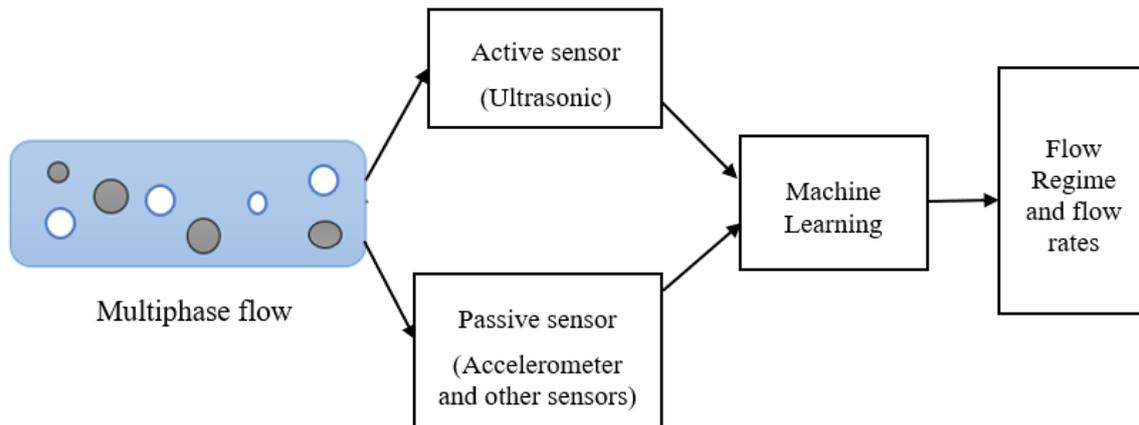


Figure 1 Overview of the system for predicting the multiphase flow rates and flow regimes.

### 2.1 Previous works

Over the past three decades, the multiphase flow rig at the University of South-Eastern Norway, Porsgrunn, has been conducting different research projects related to multiphase flow. It has been a great platform for researchers and students to perform different experimental studies and research. USN has established and improved this research center through continuous collaboration with several companies and the Norwegian Research Council. Over the period, the test rig has been modified and upgraded in accordance with the objectives of the research. In recent years, the companies Equinor, and SINTEF, which are external partners for the thesis, have been working together with USN to conduct research studies regarding multiphase flows in the test rig.

In a previous study at USN, "Machine Learning Algorithms in Multiphase Flow Regime Identification using Electrical Capacitance Tomography" by Rafael [2], a machine learning technique, particularly Convolutional Neural Networks (CNN), was implemented for identifying flow regimes. Images obtained from electric capacitance tomography (ECT) that contain flow dynamics, as well as the specific nature of the fluids, were processed through CNN using image recognition techniques and further optimized using a genetic algorithm. The accuracy of classification was compared among different compositions of pixel strips, and the performance of the model was concluded for different flow regimes.

Aleksander T. P. [3] implemented Long Short Time Memory (LSTM) of deep learning algorithms for evaluating the void fraction, and water-cut to identify the flow regimes. In this study, the combinations of ECT, EIT, GRD, and PDT were processed and used to classify the flow regimes. The neural network parameters need to be adjusted according to the specific

experiment data to extract useful information from the data. Eigenvalues and preprocessing techniques improve the accuracy of flow identification. It also concluded that the accuracy of the model was improved by increasing the acquisition rate of the sensor.

## 2.2 Multiphase flow

In this work, a mixture of oil, gas, and water will be considered, and different parameters related to multiphase flows are estimated based on machine learning techniques. As part of this project, USN plans to upgrade the existing multimodal sensor suite with new sensors and data acquisition systems in the process hall in Campus Porsgrunn. The experimental setup with accelerometers and ultrasonic sensors, along with data acquisition systems, will be established to conduct experiments related to multiphase flows. The main goal of this research project is to identify different parameters that are related to multiphase flow in the horizontal pipe from a new experimental dataset of ultrasonic and accelerometer measurements.

Multiphase flow is a complicated phenomenon that occurs when two or more than two phases flow at the same time. This phenomenon results in the formation of different flow regimes or flow patterns. Flow regimes are mostly determined by fluid properties and flow conditions. In this work, flow regimes occur from the mixing of multiple fluids under different flowing conditions. The flow pattern is influenced by various factors like fluid properties, flow velocity, mixture ratio, pipe geometry, and its orientation in the system. Because of its dependency on several flow characteristics, multiphase flow is difficult to understand and predict. Figure 2 shows the flow regime map of the two-phase flow. Any change in the flow conditions and fluid properties results in a new flow regime map. As the flow rates of the fluids are changed, the flow regime changes from one type to another, as shown in the figure below.

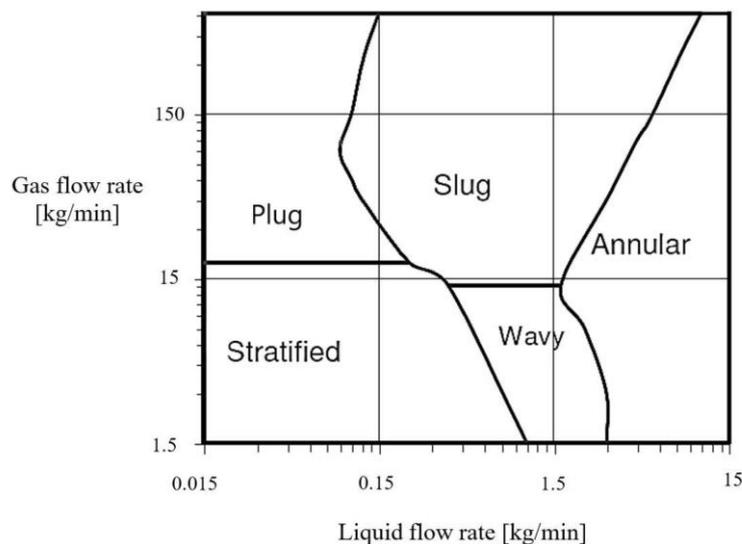


Figure 2 Flow regime map of two-phase flow [4].

This study will mainly focus on estimating the flow rates and flow regimes such as slug, wavy, annular, plug, and annular. Figure 2-3 shows the different flow regimes in the horizontal pipe with different flow rates of liquids and gases. Stratified flow is a simple flow where the liquid and gas phases are completely separated. The gas-phase will be at the top and the liquid phase will be at the bottom of the pipe. This smooth flow type is usually observed at very low liquid and gas velocities. If the gas flow rate is slightly increased without changing the liquid flow rate, it will form small waves at the interface of liquid and gas. As the gas flow rate is further increased, waves will form with higher amplitudes. This flow is called the wavy flow regime. Gas velocity within the channel determines the amplitude of the wave. If the flow rate of the gas is gradually increased, the flow changes from wavy flow to annular flow. In this regime, the flow is not as smooth as compared to the stratified flow. In annular flow, gas flowing at high velocity will form thin liquid films on the walls of the pipe where the gas and droplets are in the center of the pipe [5].

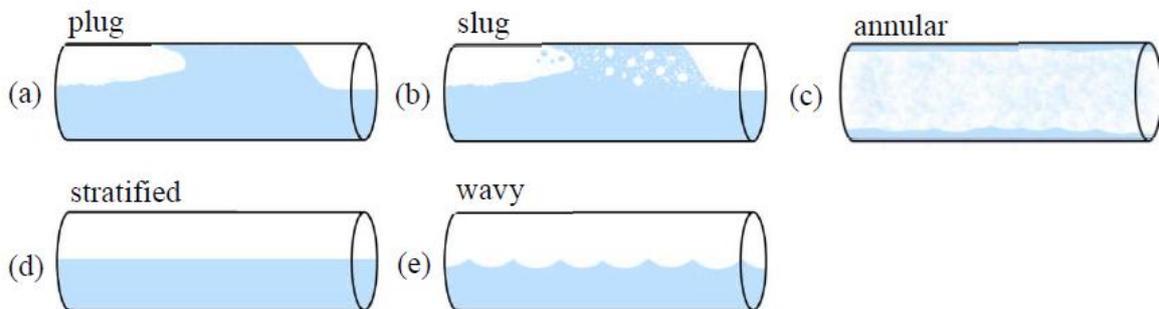


Figure 3 Plug, slug, annular, stratified, and wavy flow regimes in the horizontal pipe[2].

Plug and slug have slightly different characteristics as compared to stratified, wavy, and annular flow. These flow regimes occur at higher liquid velocities and lower gas velocities. In both flow regimes, there will be fluctuations in liquid flow and instant changes in pressure that completely fill the pipe with liquid. Plugs and slugs appear similar during the experiments, but their fluid properties are quite different.

When the velocity of the liquid is high, and the gas is low, small bubbles combine to form a big bubble that separates the liquid at the top of the pipe. whereas there will be a continuous liquid phase at the bottom of the pipe. This phenomenon is known as "plug flow." If the gas flow is increased, it shifts from plug to slug flow. During this flow regime, the liquid phases at the top of the pipe are separated by elongated bubbles followed by small gas bubbles [6]. Figures (a) and (b) of 3 show the main difference between plug and slug flow.

## 2.3 Multiphase flow metering

Multiphase flow meters (MPFMs) are devices that can measure the flow rates of individual phases in a multiphase flow. These meters are commonly used by the oil and gas industry, where they need to continuously monitor each phase, starting from the well all the way to the final oil supply pipelines.

The conventional way of measuring the multiphase flow is single-phase metering, where each of the states is separated for measurement. It requires more processing and needs a complete separation before the measurement. These traditional metering systems require more time and cost, eventually leading to lower production of the plant. To minimize the limitations of these traditional methodologies, many MPFMs have been developed over recent years. There are a lot of new advanced technologies and MPFMs that can measure the flow rates of each phase with any separation [1]. However, selecting an appropriate multiphase flow meter is crucial for optimizing production. So, the MPFMs must be selected based on their applications and requirements for measurement.

Proper installation and regular calibration are also necessary to increase the accuracy of the system. This helps to reduce the measurement uncertainties to some extent. Although MPFM systems are capable of continuous metering, they have some limitations that need to be addressed before installing them in the plant.

## 2.4 Sensors

### 2.4.1 Accelerometer

An accelerometer is a device that can measure the acceleration or vibration within any structure when it is attached to it. Its unit is G, which corresponds to the acceleration due to gravity of 9.81 m/s. It is most often used in systems where vibration levels and related parameters need to be monitored.

There are different types of accelerometers available on the market depending on their sensitivity range and applications. Capacitive accelerometers, piezoelectric accelerometers, and piezoresistive accelerometers are the types of sensors that are commonly used. The piezoelectric accelerometer is the most widely used in industrial applications to measure vibration and shock. It is a spring-mass system that consists of a sensor base, a piezoelectric element, and a sensor mass [7]. The piezoelectric element is located between the sensor mass and the sensor base. In the test section, the acceleration or change in motion causes forces in the piezoelectric element, which produces an electric charge proportional to vibration velocity. It has become more popular because of its high sensitivity rating, wide frequency response, and ease of installation.

In this work, two accelerometers, Acc-810367 and Acc-810366, are mounted on the upper part of the pipe to measure the vibration caused by multiphase flow [8]. Acc-810367 is placed just before the transparent section of the test rig, and Acc-810366 is just before the bend of the pipe. The measured data is recorded with the help of the TWave T8 vibration analyzer system [9]. Table 1 describes the specifications of the accelerometer sensor.

Table 1 Description of the used sensor.

Name	Acc-810367& Acc-810366
Manufacturer	TWave
Description	Standard IEPE vibration sensor with M12-4
Model	1001005405
Serial number	810367
Input unit	Acceleration (G)
Sensor type	Dynamic
Sensitivity	0.1 V/(G)
Offset	0
Frequency range	Max. 30k Hz
Limits	-80 to +80 G

#### 2.4.2 Ultrasonic sensor

The ultrasonic sensor is a type of sensor that uses sound waves to measure distances. The ultrasonic sensor transmits sound waves with frequencies above 20 kHz, which are too high for human ears to hear. The ultrasonic transducer is mainly used for object detection and distance calculation based on the propagation of sound waves. For object detection and distance calculation, the desired sound waves are produced based on the calculation [10]. The nature of the material, which can be solid, liquid, or gas, can be determined by observing how sound waves travel to and from the object. Solids, liquids, and gases have different reactions to sound waves; therefore, any changes within the target object can be observed from changes in the sound wave. By observing the characteristics of sound waves, received echo, and time of flight, several parameters, such as distance, liquid level, object detection, flowline parameters, etc. can be measured.

Nowadays, an ultrasonic sensor has gained more popularity due to its wide range of applications. It is also known as a piezoelectric transducer. It converts electrical signals into mechanical signals and vice versa. It consists of a transmitter and a receiver. In this method, the transducer emits an ultrasonic wave which is passed through a target section and reflected from the object. The transmitter and receiver can either be on the same transducer or on a separate transducer, depending on the application. For that reason, a transducer can be classified into two types: monostatic and bistatic. Figure 4 shows a monostatic transducer with a combined transmitter and receiver and a bistatic transducer with a separate transmitter and receiver.

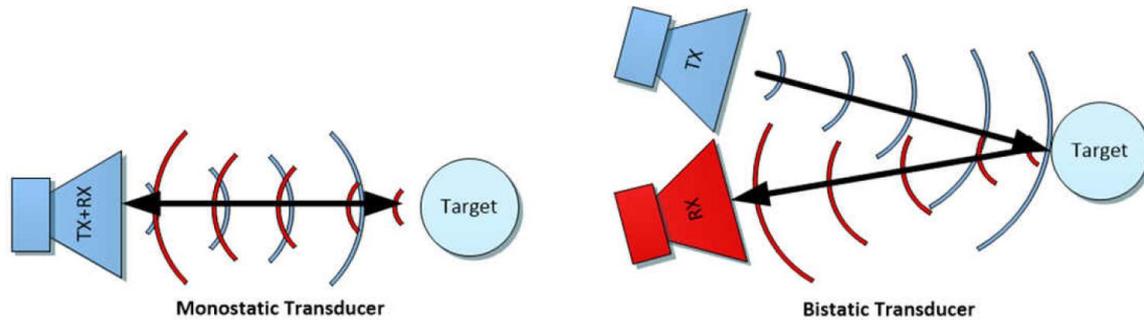


Figure 4 Monostatic and bistatic transducer [10].

The distance depends on the velocity of sound waves and total time. It can be calculated with this formula.

$$d = \frac{c\Delta t}{2} \quad (2.1)$$

Where  $d$  is the distance,  $c$  is the velocity of waves and  $t$  is the time between the emitted and received signal.

Similarly, the relationship between wavelength, velocity and frequency can be determined with the equation below.

$$\text{Wavelength}(\lambda) = \frac{\text{Velocity}(v)}{\text{Frequency}(f)} \quad (2.1)$$

In ultrasonic testing, the pitch-catch and pulse-echo methods have been popular ways of observing the test piece. When the pitch-catch method is implemented, one acts as a transmitter that transmits signals, and the other acts as a receiver to receive the reflected signal. In contrast, pulse-echo can transmit and receive using the same transducer. Although both methods work almost the same, the main difference is in the transducer used to receive the signals.

The performance of the ultrasonic sensor depends on different factors such as transmission medium, acoustic impedance, ambient conditions, and so on. Table 2 shows the variation of acoustic velocity, density, and acoustic impedance in a different medium.

Table 2 Variations of acoustic velocity, density, and acoustic impedance on different mediums [10].

Medium	Acoustic velocity( $\text{ms}^{-1}$ )	Density ( $\text{kgm}^{-3}$ )	Acoustic Impedance ( $\text{kgm}^{-2}\text{s}^{-1}\times 10^6$ )
Air	330	1.3	0.00429
Water	1450	1000	1.45
Aluminum	6320	2700	17.1
Iron	5900	7700	45.43
Steel	5900	7800	46.02
Skin	1540	1109	1.6

In this work, the Panametrics-NDT ultrasonic transducers are used to perform the ultrasonic test. This transducer is a contact type that emits a longitudinal wave in the direction of the test piece. Figure 5 shows the ultrasonic sensor with a frequency range of 5 MHz, A, and B represent the diameter and height of the sensor. It is suitable for testing specimens that need a better signal-to-noise ratio (SNR) [10]. Here, the transducer transmits the pulse through the pipe containing a mixture of air, water, and oil. At first, the signal travels through the outer diameter of the pipe and then into a mixture of fluids. After that, the signal is reflected from the upper diameter of the pipe and received by the transducer. When different fluids at different flow rates flow through a channel, sound waves behave differently, and it is possible to determine the flow pattern by observing the pattern of the sound waves.



Figure 5 Ultrasonic sensor with A and B representing diameter and height of sensor.

## 2.5 Machine learning

ML is part of AI that developed the model for the analysis of data. The machine learning models are capable of learning from input data and have the ability to recognize patterns within the provided data sets. It makes a prediction based on the recognized patterns. The most common types of machine learning models are supervised and unsupervised learning models. In supervised machine learning techniques, the input and output variables are well known before making the prediction. In the end, the outcome of the model is compared with the known output to know the accuracy of the model. However, the unsupervised model makes a prediction based on the input data only. Figure 6 shows the different types of machine learning techniques.

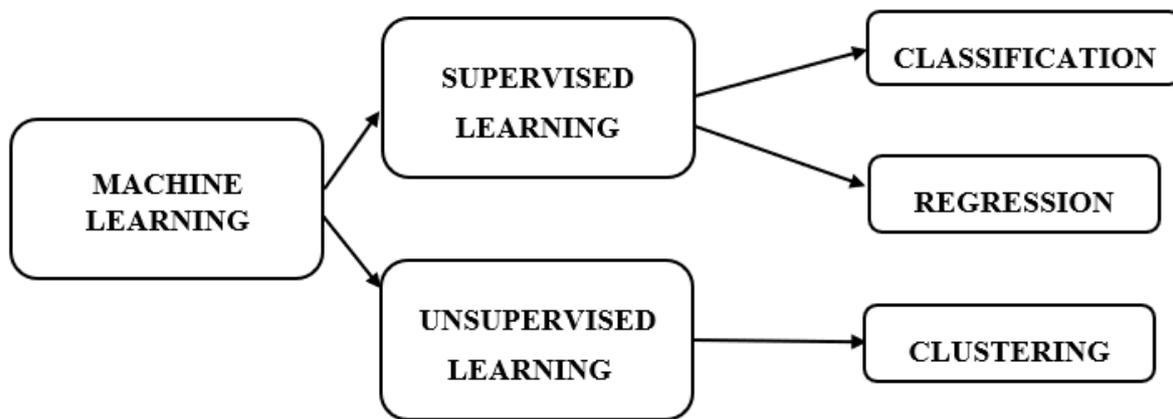


Figure 6 Supervised and unsupervised machine learning algorithms [11].

In this work, supervised machine learning techniques are used to estimate the multiphase flow parameters. The main aims are to classify the flow regimes and predict the flow rates of different fluids. The classification of flow types is performed in MATLAB using decision trees, random forests, and different models present in the classification learner application. In the regression part, the flow rates of the fluids are predicted using the regression learner tool present in the MATLAB software. These models are explained below.

### 2.5.1 Decision tree

A decision tree is one of the supervised learning methods whose structure is similar to a tree. As the name implies, it is made up of leaves, branches, nodes, and roots, each with its own importance and function in making the decision. It can be used both for classification and regression problems, but most typically, it is used for classification. The data is initially imported into the root and internal nodes of the decision tree where it is processed. After processing, the data is analyzed, and decisions are made based on the rules present in each branch of the model. Finally, the model reaches the leaves that represent the outcome of the model. The outcome may be a dependent variable in a regression problem or a category in a classification problem. In our work, the outcome of the classification is the type of flow regime, and the outcome of the regression is the predicted flow rates. Figure 7 shows the general structure of the decision tree algorithm.

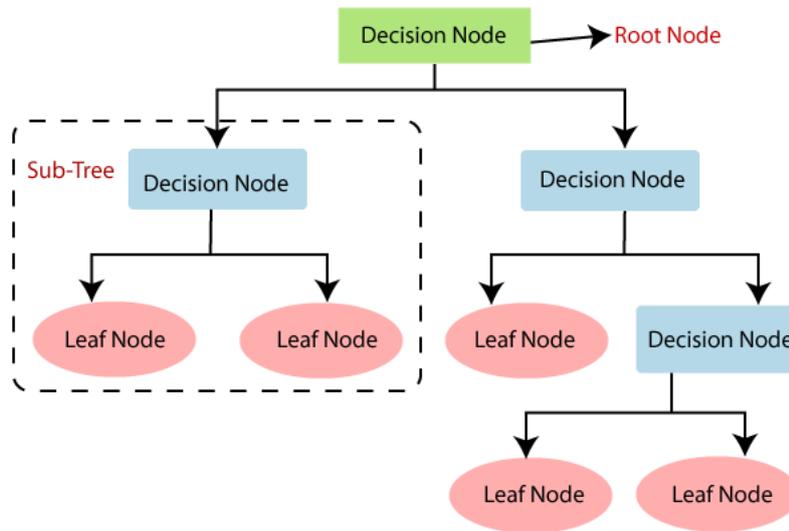


Figure 7 Decision Tree algorithm structure[12].

### 2.5.2 Random forest

A random forest is a tree-based model that consists of multiple decision trees to make independent decisions. Each tree makes its own decision on the provided data, just like in the decision trees algorithm. However, the final decision is made based on the majority or average of the outputs from different trees. The method of predicting the results based on the majority of votes made the model more accurate than the decision tree method. The performance of the model increases with the increase in the number of trees. An overview of Random Forest machine learning algorithms is shown in figure 8.

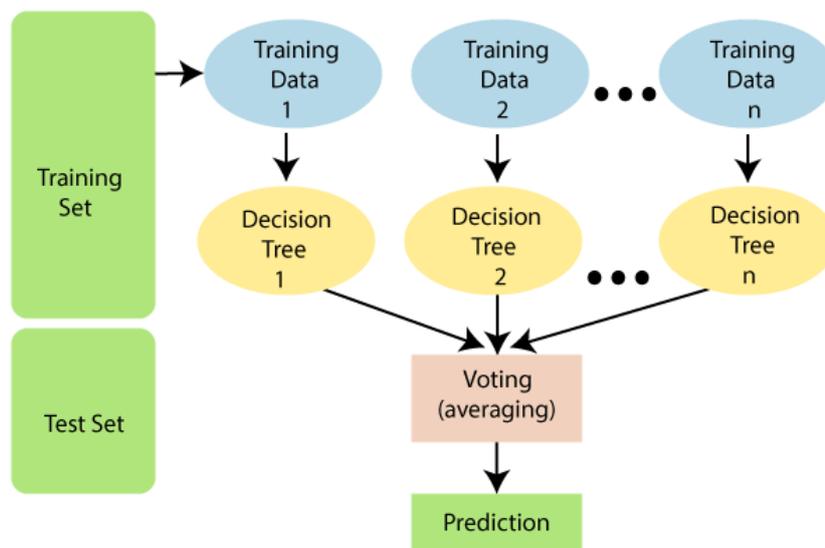


Figure 8 Random Forest algorithm structure[12].

### 2.5.3 Ensemble Method

The ensemble model consists of multiple learners that can solve classification and regression problems. Multiple learners are trained with input variables to predict the target variable. It has better accuracy as compared to individual learning models. The most popular types of ensemble techniques are bagging and boosting. The bagging technique is also known as the bootstrap aggregating technique [13]. In the Bagging technique, multiple models of the same learning algorithm are trained with subsets of the dataset that are randomly picked from the training dataset. The base learners are formed independently, and it reduces the variance and minimizes the overfitting problem. The boosting technique is slightly different from the bagging technique. The main difference is that the base learners are dependent on the previous weak base learner, which helps to improve the performance of the model.

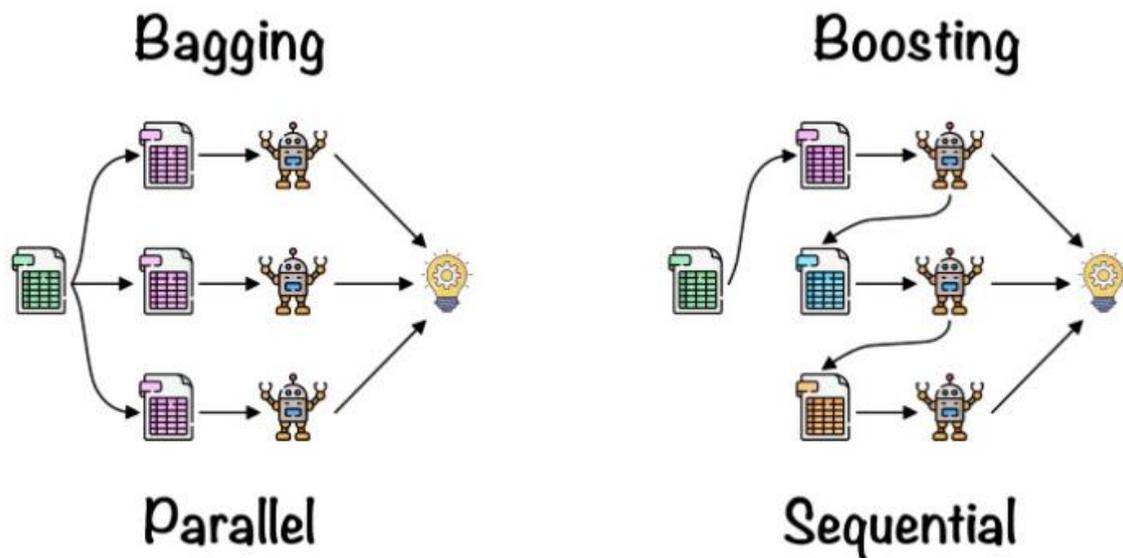


Figure 9 Bagging and Boosting method of Ensemble method[14].

### 2.5.4 Classification and Regression Learner Application

The Classification and Regression Learner tools are inbuilt tools in MATLAB and are present in the machine learning and deep learning sections. These applications consist of different models that can be implemented all at a time. Both applications can be used in machine learning to predict using supervised machine learning. Figure 10 shows the different models that are present in the Classification and Regression Learner tools.

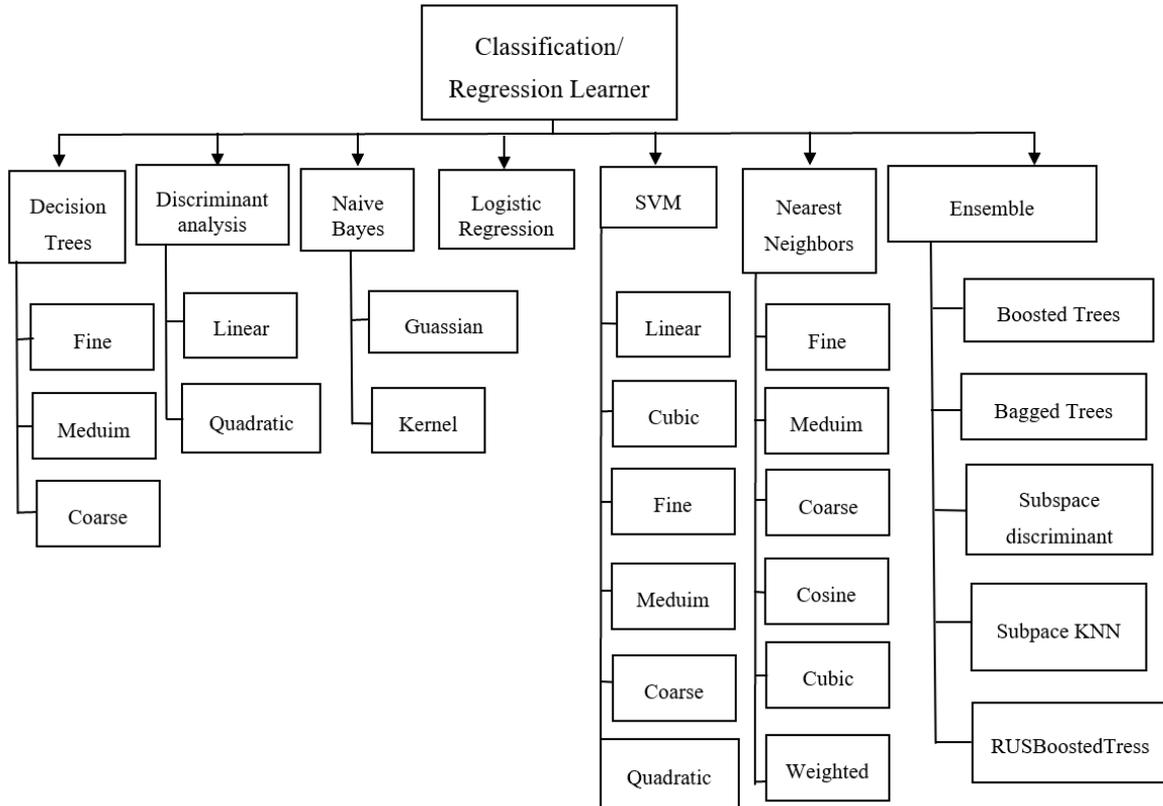


Figure 10 Different models present in Classification and Regression Learner applications [15].

## 2.6 Data acquisitions system

### 2.6.1 Vibration monitoring system

TWave T8 is a monitoring system that is used to analyze the vibrations within different mechanical systems. This monitoring system has been designed to directly monitor the real condition of the system without the need for supportive software or external databases. One of the main advantages of this system is that it has an online monitoring feature that allows getting access to measured data by using a web browser. It has an inbuilt 4G modem/wireless Wi-Fi and an embedded web-based user interface that can be accessed through any operating system. The measured data is stored in a memory space and historical data can be easily accessible at any time via an internet connection. Figure 11 shows the internet connection between T8 and the device.

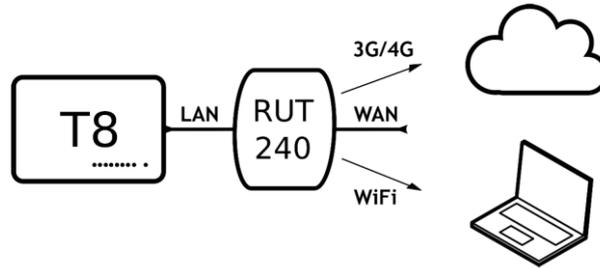


Figure 11 Communication between Twave T8 and laptop using a router[8].



Figure 12 Main screen of TWave T8 system.

It is commonly used to measure the vibration of mechanical systems such as pumps, engines, fans, gearboxes, turbines, compressors, and motors. In this work, the T8 large model is used to measure the vibration that is produced during multiphase flow in the pipe. The T8-L has eight high-speed analog input ports with a range of 20kHz and an additional four input ports for tachometers and other purposes. It has different functions that allow for better visualization of measured data. The two accelerometer sensors, Acc-810367 and Acc-810366, measure the vibration at two different points of the test rig pipe.

### 2.6.2 US-Key

US-Key is manufactured by Lecoer Electronique and is used in non-destructive testing. It manufactures different ultrasound products and is well-known for their precise measurements and ease of use. This US-Key system contains four channels in total, including four transmitter ports for transmitting the signal and four receiver ports for receiving the signal. The system is connected to ultrasonic sensors and a laptop that measures the echo produced by the reflection of the signal. Echoes are shown in the graph in the form of a waveform that indicates their amplitude and time of occurrence. The most important parameter of an ultrasonic test is the acoustic interface, which is primarily determined by the amplitude and time at which it occurs from the pulse transmission point.

The cracks or changes in the medium that cause the interface are determined by the change in amplitude of the echo. After that, other parameters are calculated based on the obtained amplitude and time. Figure 13 shows the main screen of the US-Key data acquisition system.

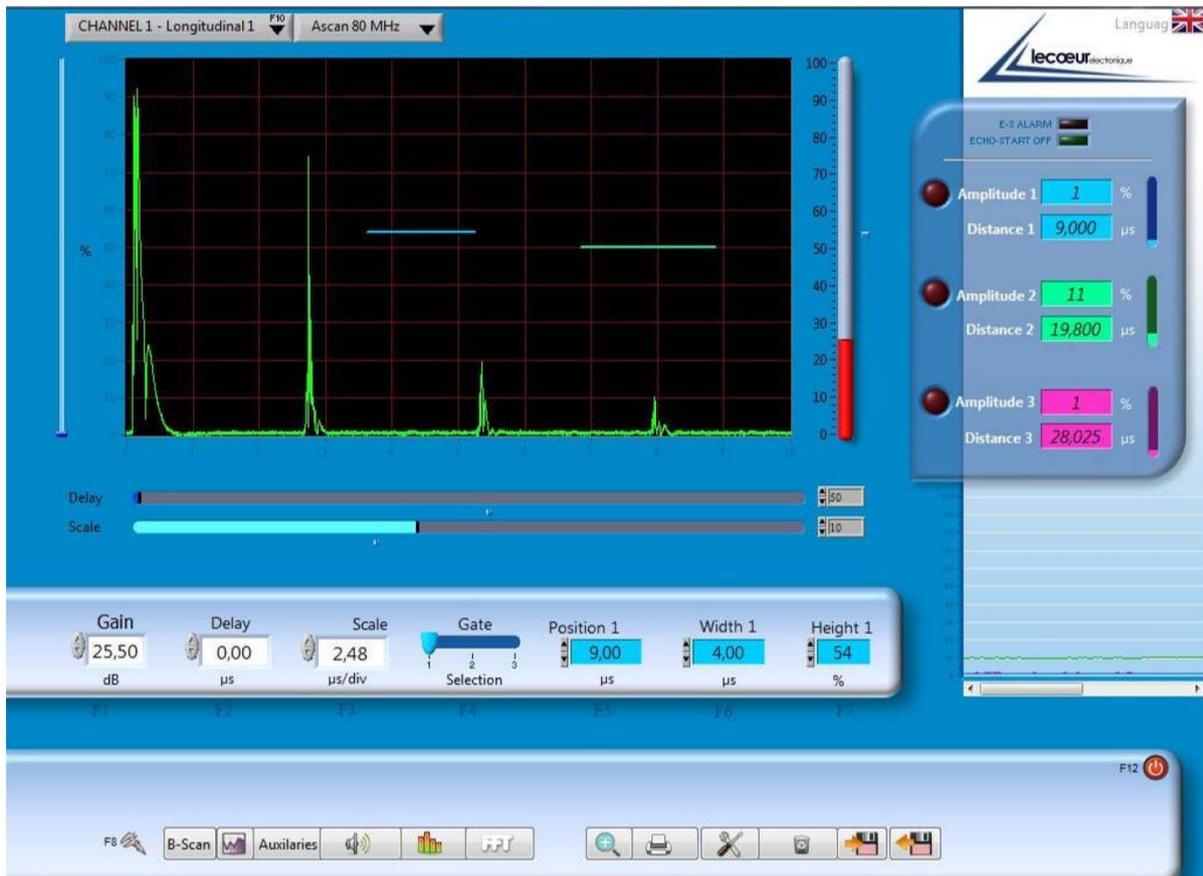


Figure 13 Main screen of US-Key system.

The main screen of the system consists of 4 sections: a graphical display for showing the graph between amplitude and time, another for alarms and amplitudes that shows amplitudes and distances at different gates; the next for adjusting gain, delay, scale, position, width, and height of each gate; and the bottom one for additional settings.



## 3 Multiphase flow rig

In this work, the existing multimodal sensor suite was slightly upgraded, and two additional sensors accelerometers and ultrasonic sensors were installed to perform multiphase flow analysis. The experiments are performed by varying the flow rates of air, water, and oil separately. This chapter discusses a new experiment setup with accelerometers and ultrasonic sensors, and experimental results of both two-phase air-water flow and three-phase air-water-oil flow.

### 3.1 New experimental setup

In this experimentally based thesis, the two accelerometers and ultrasonic sensors are installed in the existing test rig in the process lab of Campus Porsgrunn. The setup for this work is planned in collaboration with the USN team at USN. Figure 15 shows the piping and instrumentation diagram (P & ID) of the test rig with two accelerometers and two ultrasonic sensors added to the original system. In the P&ID diagram, the blue, red, and green lines represent the pipelines that carry water, oil, and air in the system.

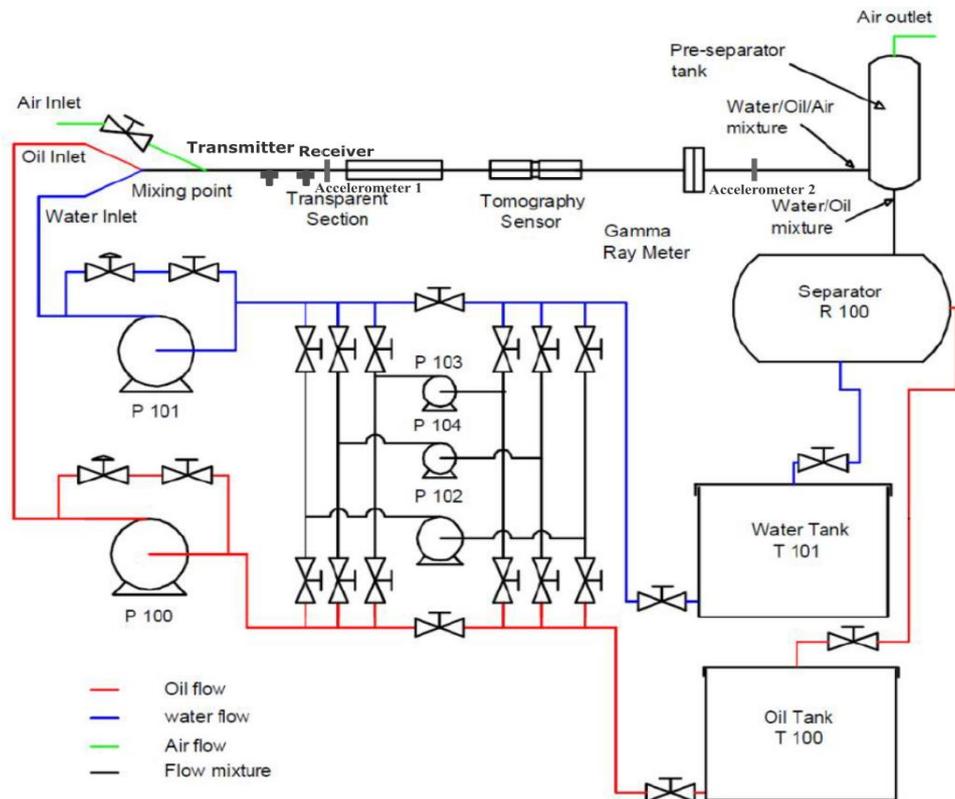


Figure 15 Piping and Instrumentation Diagram(P&ID) with accelerometers and ultrasonic sensors [3].

In this system, there is an air injector to inject the air and pumps to supply oil and water into the channel. Two large pumps and two small pumps are present in the system to achieve different flow rates effectively for both the water and oil supply. The flow rates of air, water, and oil are controlled individually with the help of different flow controllers present in the test rig, and fluids get mixed at the mixing point of the test section. As a result, the various flow

patterns are obtained in the test rig by controlling the supply of fluids from injectors and pumps. In the end, each of the fluids is separated with the help of a pre-separator and separator R100. The different parameters such as flow rates, temperature, pressure, density, and fluid level are monitored with the help of transmitters and Coriolis meters that are installed on the different parts of the test rig.

The test rig consists of different blocks in its configuration, such as pumps, injectors, pipelines, sensors, transmitters, Coriolis meters, and data acquisition systems. It is crucial to have general knowledge of these blocks to perform experimental work effectively. Table 3 provides information about the dimensions of the pipe and the properties of the fluid that are used in the test rig.

Table 3 Specification of the system[16].

Density Exxsol D-60	790 kg/m <sup>3</sup> (25°C)
Viscosity Exxsol D-60	0.00164 Pa s (25°C)
Density water	1000 kg/m <sup>3</sup> (20°C)
Viscosity water	0.00102 Pa s (20°C)
Interfacial tension (o/w)	43 mN/m (25°C)
Inner pipe diameter	0.00563 m
Flow area	0.0024895 m <sup>2</sup>
Length PDT-120	10.21 m
Pipe roughness	1.00×10 <sup>-5</sup> m
Length PDT-121	5.37 m

In this work, all the experiments have been conducted based on the mixture of oil, water, and air, while in the previous works, only air and water were used to study the multiphase flow. The existing test rig setup is slightly changed, and new sensors with data acquisition systems are used to collect the data. For the experiments, two accelerometers are installed on each side of the transparent section. According to the plan, two ultrasonic sensors, one as a transmitter and another as a receiver, will be mounted on the left side of the transparent section. It is not possible to mount the ultrasonic sensors using the existing wedges. For the initial phase, a single ultrasonic sensor, which serves as both transmitter and receiver, is mounted onto the transparent section of the test rig, as shown in figure 18. There are two new data acquisition systems and LabVIEW software to collect the data for this experiment. T wave and US-Key are two data acquisition systems to record the data of accelerometers and ultrasonic sensors, respectively. The LabVIEW software collects data from other existing sensors that are preinstalled on the rig. Figure 16 shows the experimental setup for this experiment. Figure 17 shows the location of two accelerometer sensors in the test rig. Figure 18 indicates the location of the ultrasonic sensor in the test rig.



Figure 16 Experimental setup for multiphase flow.

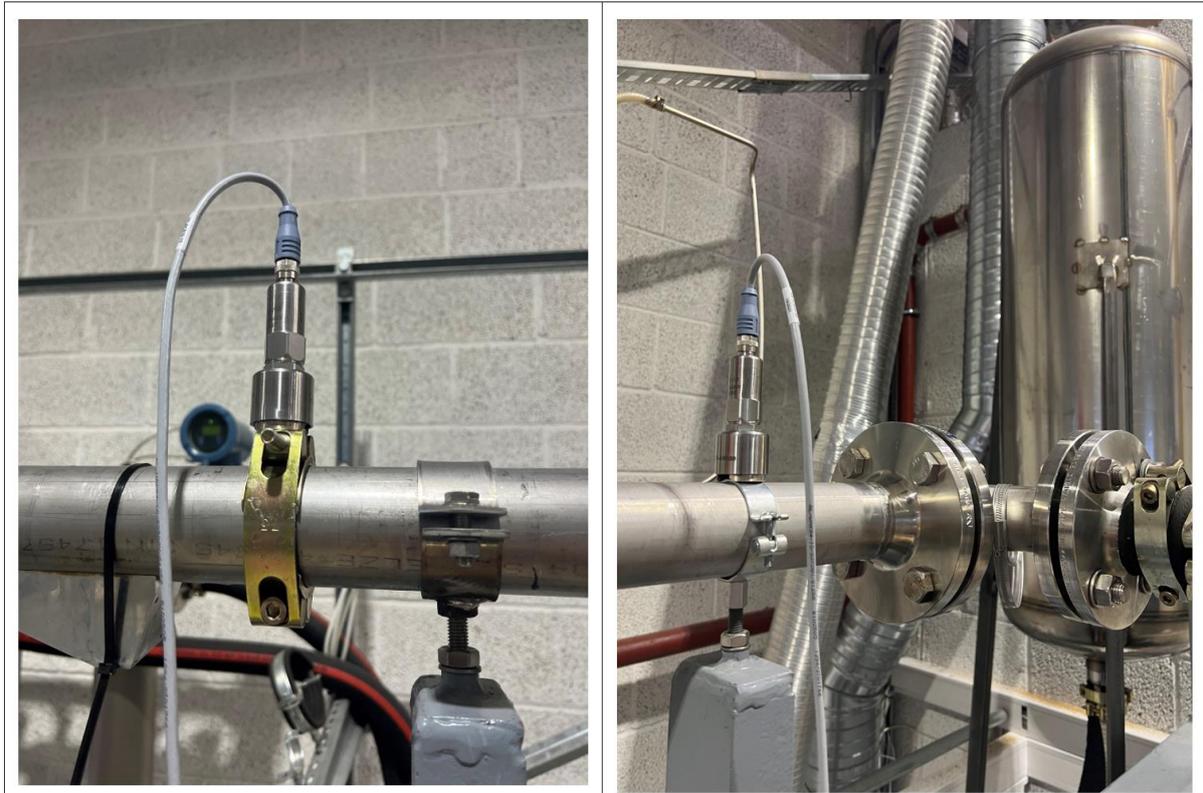


Figure 17 Two accelerometer sensors that are installed just before the transparent section (left) and near the bend of pipe (right).

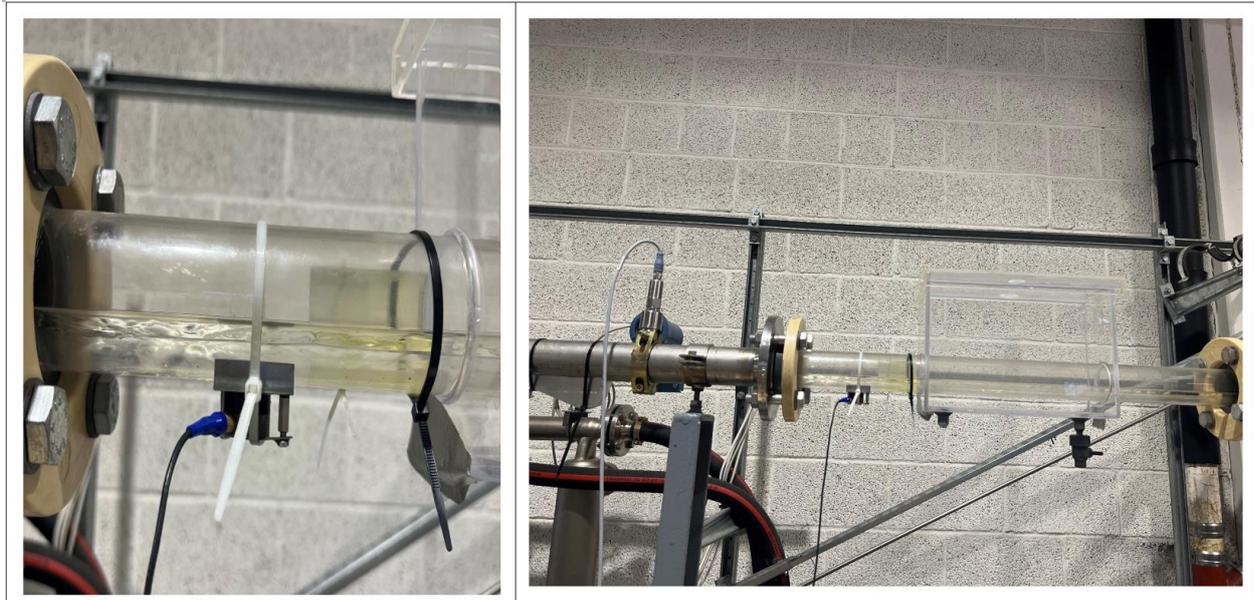


Figure 18 Position of the ultrasonic sensor in the transparent section of the test rig.

## 3.2 Experimental results

The data were collected by performing the physical experiments at the test rig of campus Porsgrunn. In this work, different types of flow patterns are obtained by mixing air, water, and oil at different flow rates. Then, the flow profiles are categorized into stratified, wavy, annular, plug and slug based on the nature of flow patterns. The measurements were recorded several times based on the results obtained from the models.

### 3.2.1 Experiments with air and water

At the beginning of this work, the experiments were performed with a mixture of air and water only. The main purpose of these experiments is to get familiar with the test rig and observe the results of past theses. A total of 86 measurements were conducted based on previous work related to multiphase flows. A flow regime map of air and water was taken as a reference to perform the experiments at the test rig in the process lab. As per a flow regime map, the flow rates of air and water were changed, and the different flow regimes were obtained. From the transparent section of the test rig, different types of flow regimes were inspected to get familiar with the nature of each flow.

Table 4 Duration of each experiment.

Unit	Sample rate	Duration	
Accelerometer	52KHz	2 seconds & 5.12 seconds	T-Wave platform
Non-acoustic sensors	1Hz	60 seconds	LabVIEW program
Ultrasonic sensor	5MHz	-	US-Key





### 3.2.3 Experiment with the ultrasonic transducer

The experimental setup of the ultrasonic test consists of two ultrasonic transducers, wedges, and a data acquisition system to collect the data. The two parametric ultrasonic sensors of video scan type with a frequency of 5MHz were connected to US-Key using the LCM-74-6 cable from Olympus, and the ultrasonic data acquisition system, US-Key, was connected to a laptop. Initially, the ultrasonic sensors were first coupled with two existing wedges and then coupled to the outer diameter of the pipe using a coupling oil. For contact tests like ultrasonic testing, oil, grease, and gel can be used as coupling mediums to prevent an impedance mismatch between the ultrasonic transducer and the test piece. However, a suitable coupling medium must be used based on the nature and purpose of the experiment. To avoid the loss in signal transmission, sensors must be clamped correctly without any air in the joint. For higher sound energy, there should be no air between the transducer and the test piece. As per our plan, the two modes, i.e., pulse-echo and pitch and catch mode, will be used in this work, as shown in figures 22 and 23.

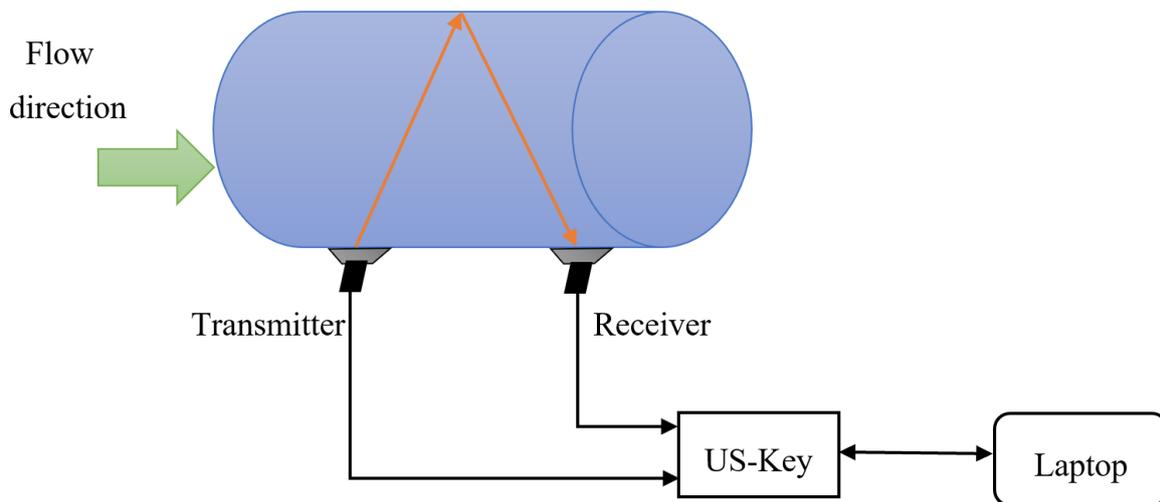


Figure 22 Pitch and catch mode of ultrasonic testing.

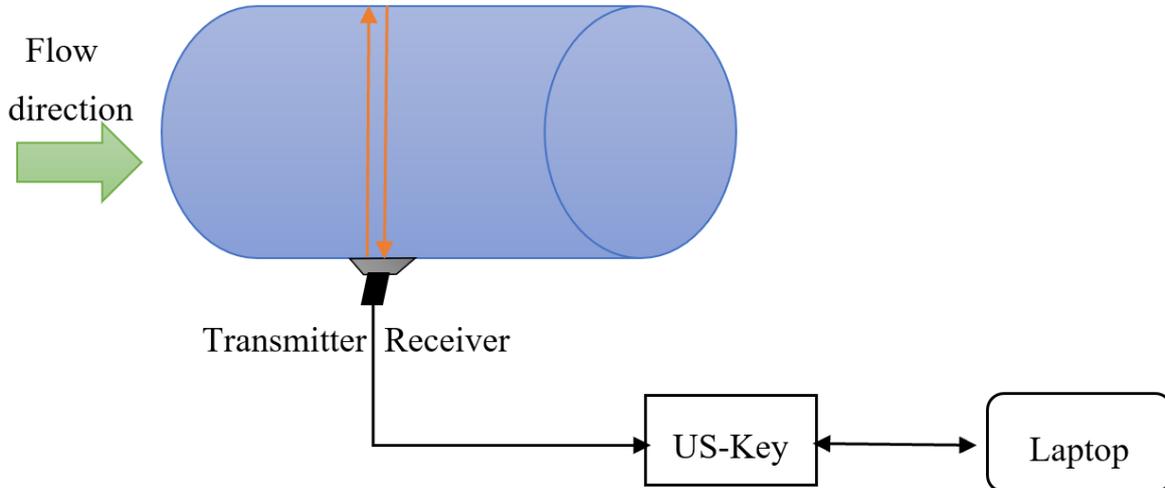


Figure 23 Pulse-echo mode of ultrasonic testing.

However, the experimental setup did not work as expected, and the US-Key was unable to detect the proper signal from the ultrasonic sensors. The location and orientation of the ultrasonic sensors were changed to solve the problem. But the US-key failed to record the signal from ultrasonic sensors in both pitch-catch and pulse-echo mode. The experiment was then conducted in a small aquarium to determine if the system without wedges was functioning properly or not. The system is functioning properly this time, and signals are being recorded as binary files using the US-Key system.

Now, it's confirmed that the problem was caused by the geometry of the wedges. Since the existing wedges have different wedge angles, the signal transmitted by the transmitter was unable to be properly received by the receiver. During the weekly meeting, the problem was discussed, and it was decided to design new wedges for the experiments.

### 3.2.4 Design of wedge

A new wedge was designed in collaboration with the USN team. The wedge was designed to reduce the noise that was previously caused by the geometry of existing wedges. This new wedge was required to perform the pitch and catch mode of ultrasonic measurements. In the initial stage of design, the dimensions of the test section were considered and the signal properties were understood properly. Then, the calculations were made by considering the properties of the materials [17]. The calculation of the wedge for non-destructive testing is given below [18].

According to Snell's law,

$$n_1 \sin \theta_1 = n_2 \sin \theta_2 \quad (3.3)$$

The index of refraction is

$$n = \frac{\text{Velocity of light in a vacuum } (c)}{\text{Velocity of light in the medium } (v)} \quad (3.4)$$

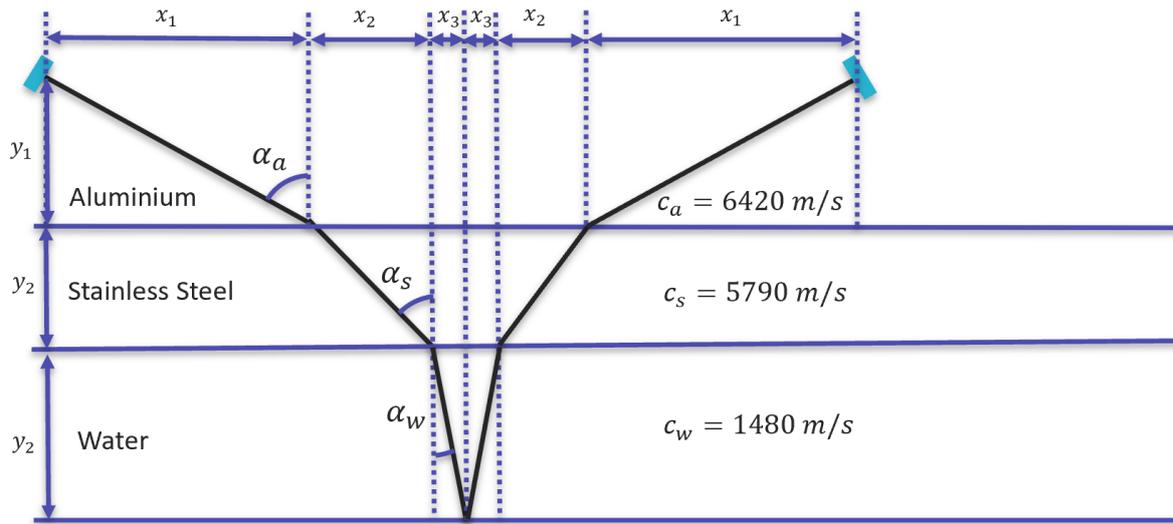


Figure 24 Calculations of different distances and angles for new wedge.

Now,

Choosing  $\alpha_a = 60^\circ$

$$\alpha_s = \sin^{-1} \left( \frac{\sin 60^\circ \cdot 5790}{6420} \right) \approx 51.4^\circ$$

$$\alpha_w = \sin^{-1} \left( \frac{\sin 57.9^\circ \cdot 1480}{5790} \right) \approx 11.5^\circ$$

The detailed calculation of different distances with respect the o different angle of incidence is shown in table 5.

Table 5 Calculation of the refracted shear angle and distances from the different angles of incidence.

$c_a$ [m/s]	$c_s$ [m/s]	$c_w$ [m/s]	$y_1$ [m]	$y_2$ [m]	$y_3$ [m]	$\alpha_a$ [°]	$\alpha_s$ [°]	$\alpha_w$ [°]	$x_1$ [mm]	$x_2$ [mm]	$x_3$ [mm]	$x_1 + x_2 + x_3$	Total distance [mm]
6420	5790	1480	0.03	0.002	0.03	30	26.80	6.62	17.3	1.0	3.5	21.8	43.6
6420	5790	1480	0.03	0.002	0.03	45	39.62	9.38	30.0	1.7	5.0	36.6	73.2
6420	5790	1480	0.03	0.002	0.03	60	51.36	11.52	52.0	2.5	6.1	60.6	121.2
6420	5790	1480	0.03	0.002	0.03	70	57.94	12.51	82.4	3.2	6.7	92.3	184.5

Based on the calculation made, the new wedge was designed using solid works software. Figure 25 shows the detail design of new wedge.



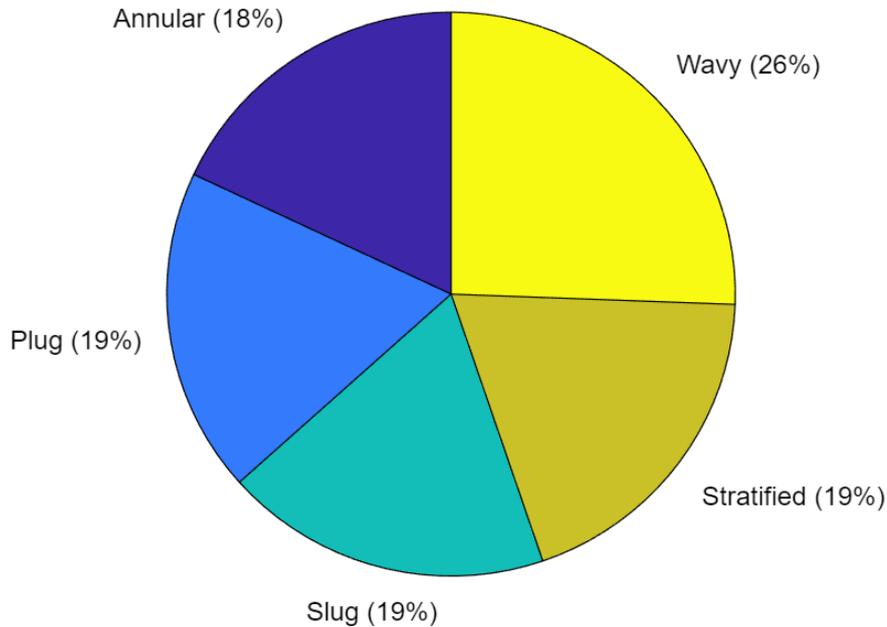


Figure 27 Distribution of different flow regimes in the final test matrix.

### 3.2.6 Ultrasonic measurements from transparent section

The newly designed wedges could not be manufactured at the USN because of time and manpower constraints. The problem was discussed in a meeting, and it was decided to wait for 4 weeks. Unfortunately, the wedges were not ready, so the problem was discussed, and it was decided to concentrate mainly on the acoustic part of this thesis.

However, some measurements were collected using the available wedge. As part of these experiments, a single ultrasonic sensor was installed on the transparent section of the test rig for pulse-echo mode. Installing the ultrasonic sensor is not as realistic in the oil and gas industry as it should be in the pipelines. It is done so that the experimental setup and information can be used for future studies.

The experimental setup was done, and a few experiments were conducted for different flow regimes. The measurements were collected, and the data was saved in binary file format. Then BIN files were converted into JPG files by using the code provided by Tonni. The code is attached in appendix F, and ultrasonic images of different flow regimes are shown below. The x-axis represents the line numbers. The y-axis represents the time traveled by the pulse in microseconds.

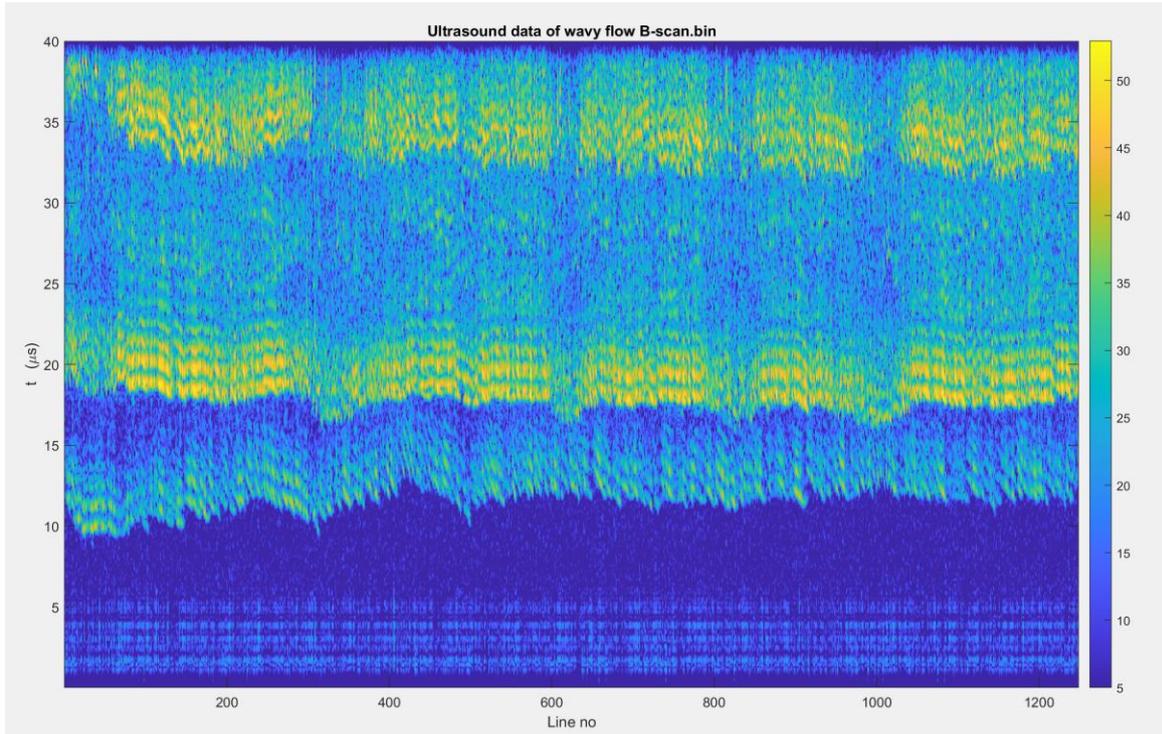


Figure 28 Wavy flow output from ultrasonic sensor.

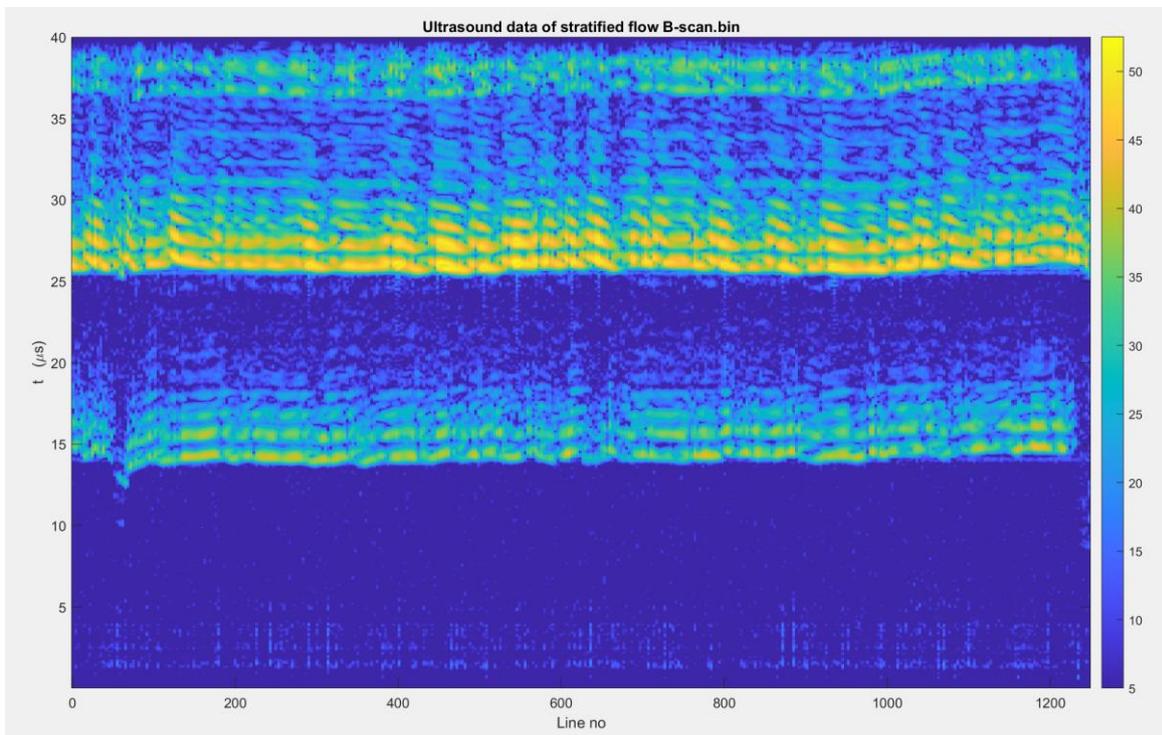


Figure 29 Stratified flow output from ultrasonic sensor.

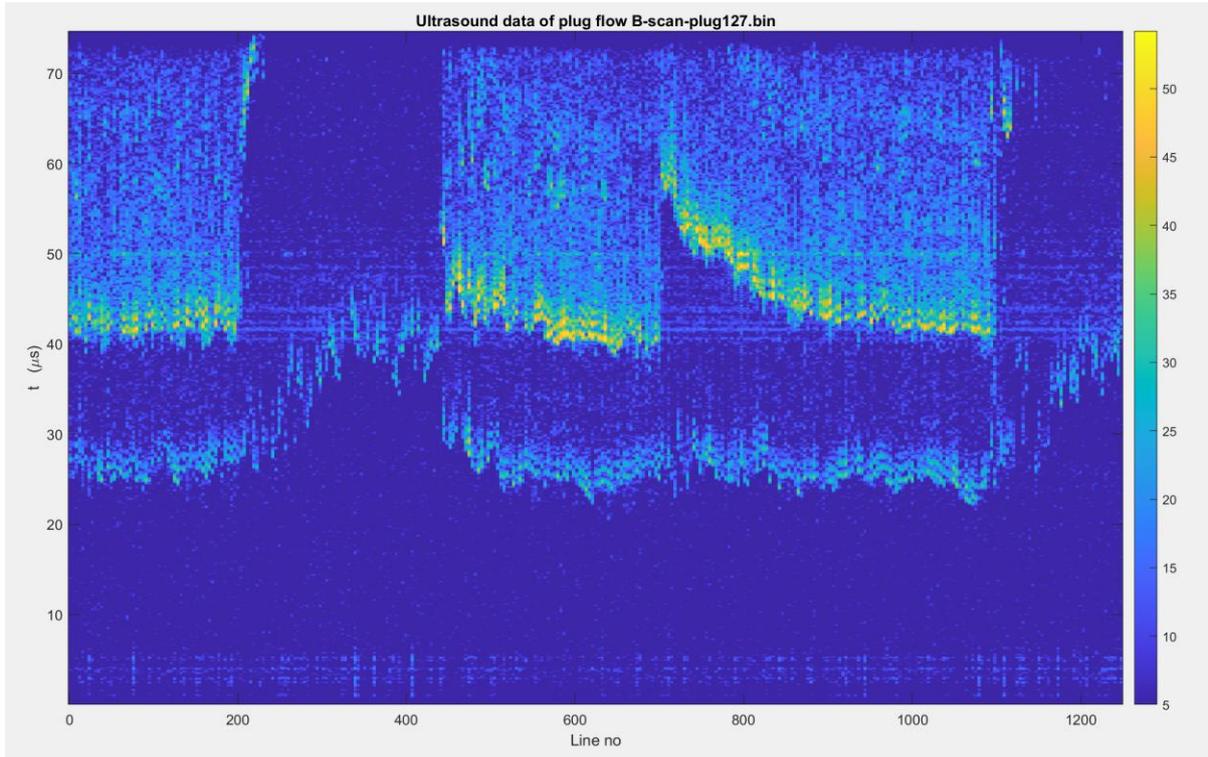


Figure 30 Plug flow output from ultrasonic sensor.

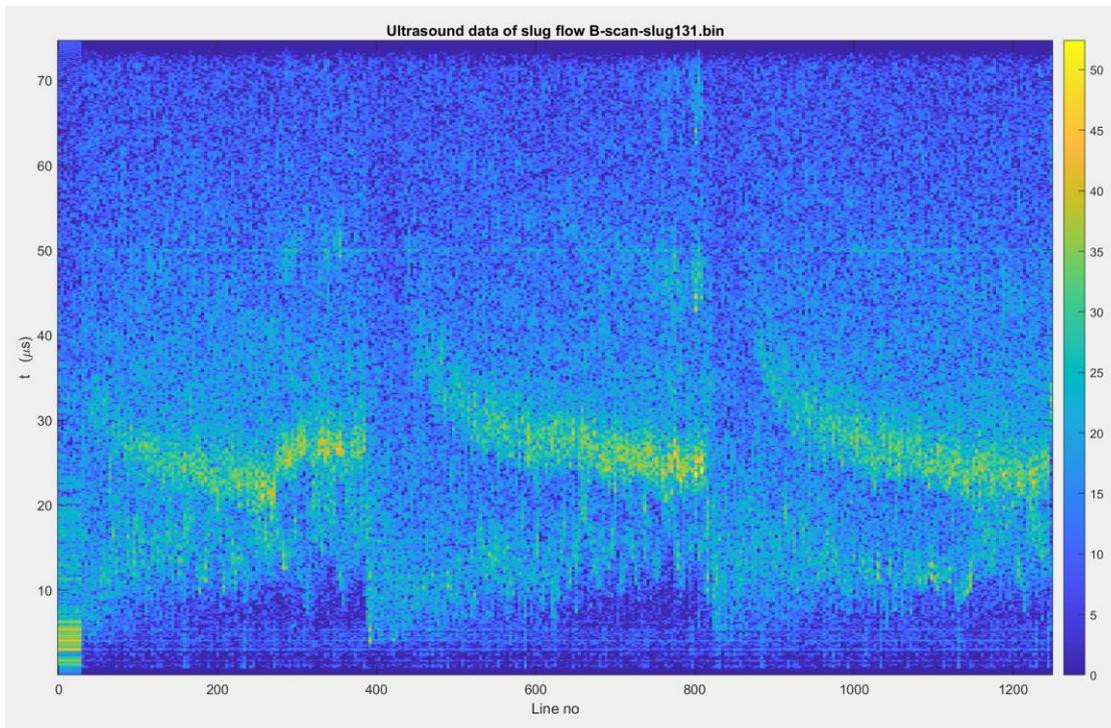


Figure 31 Slug flow output from ultrasonic sensor.

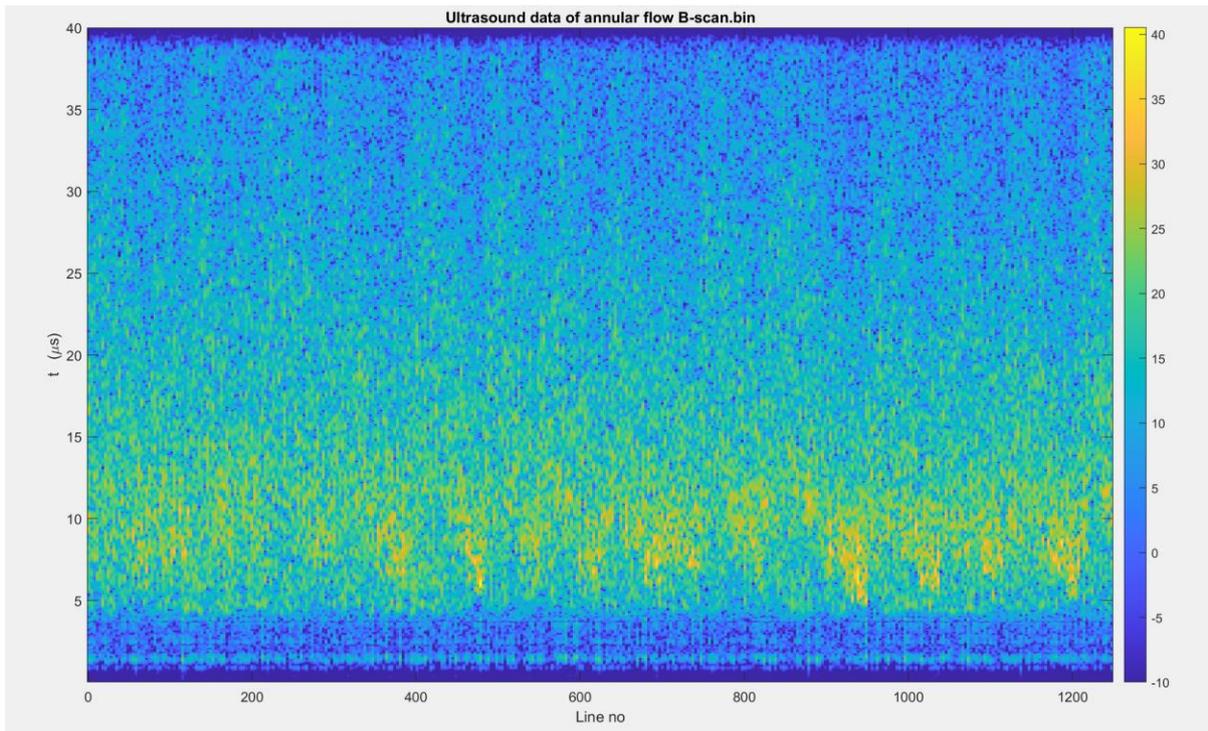


Figure 32 Annular flow output from ultrasonic sensor.

### 3.2.7 Single-phase experiment for Sailesh’s thesis

The data were collected for another thesis student who is doing a similar thesis but focusing on single-phase flow. The acoustic measurements were recorded by flowing one phase at a time in the same experimental setup. Figure 33 shows the test matrix for single-phase flow.

air (Kg/min)	0.2	0.4	0.6	1	1.4	2	2.5							
Water(Kg/min)	2	5	8	10	12	16	25	30	40	45	50			
oil (Kg/min)	2	5	8	10	12	15	18	20	25	30	35	40	45	50

Figure 33 Test matrix for single-phase flow.

## 4 Pre-processing and feature extraction

This chapter explains how different processing techniques were implemented to convert raw data into useful information. The quality of collected data may be affected by external noise. It is therefore necessary to process the measurement data before using it for machine learning. Figure 34 shows the different steps of data preprocessing to obtain the input matrix for machine learning models.

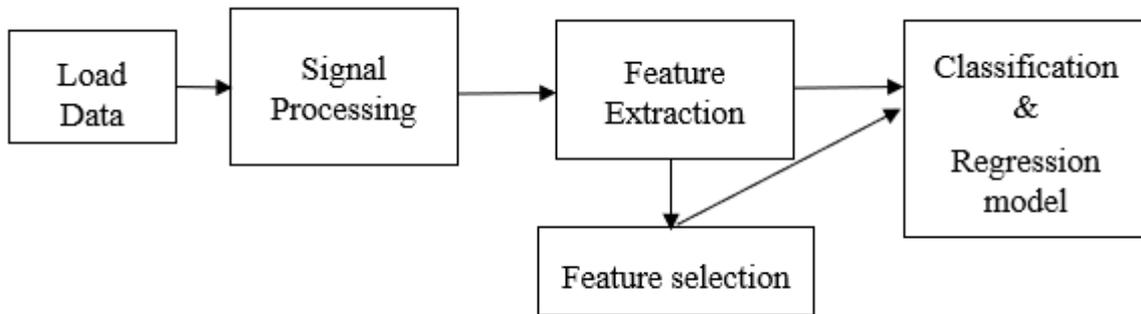


Figure 34 Workflow of data processing for classification and regression model.

### 4.1 Pre-processing of collected measurements

After the data collection, it is necessary to perform preprocessing and exploratory data analysis before developing the data-driven models. During the preprocessing, the signals from the acoustic sensors were plotted in MATLAB to observe the pattern of input signals. From the graph, it was found that there was an overlap in the collected data. Then, the spectrogram in MATLAB was plotted to confirm the nature of the signals. It was found that the signals were noisy. That may be caused by a short sampling period of 2 seconds. The signals from two accelerometers can be seen in figure 35.

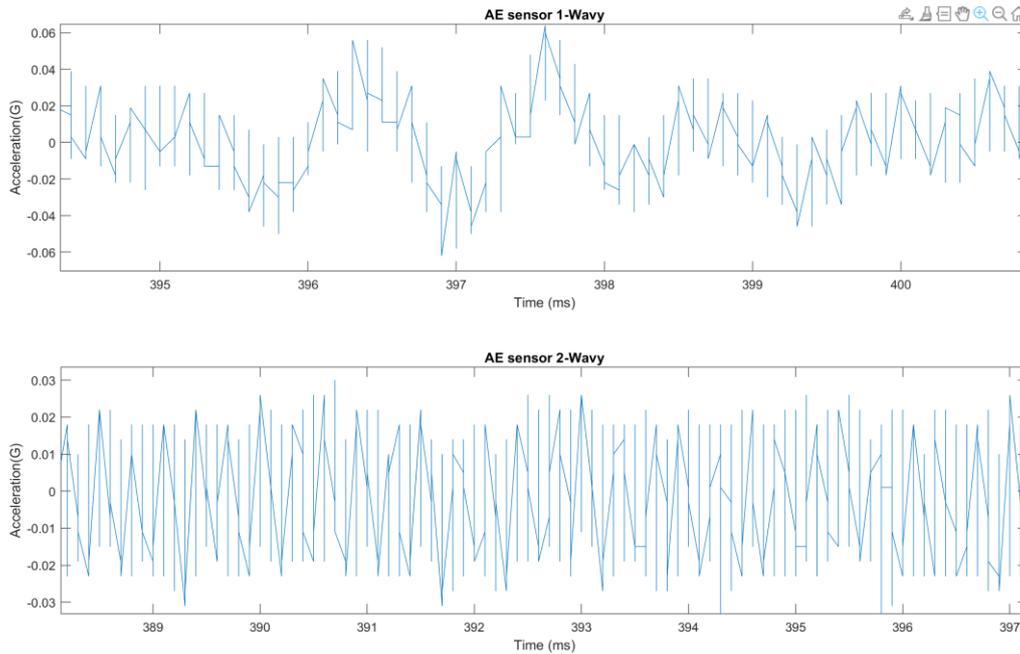


Figure 35 Acceleration vs time plot of two acoustic signals.

From the graph, it can be deduced that the signal has a low resolution and, therefore, the sensors are recording a low level of the signal. Then, the experiment was reconducted with the aim of recording signals for a long time. It was not possible to make longer measurements in our T-Wave data acquisition system because the original purpose of this device was to monitor the vibration in mechanical components for a short period of time. It was found that the maximum time for the measurement of a signal is 5.12 seconds. Then, the measurement of duration was increased from 2 seconds to 5.12 seconds. In the next step, the graph was plotted by combining several measurements, and Figure 4-3 shows the zoomed version of the graph.

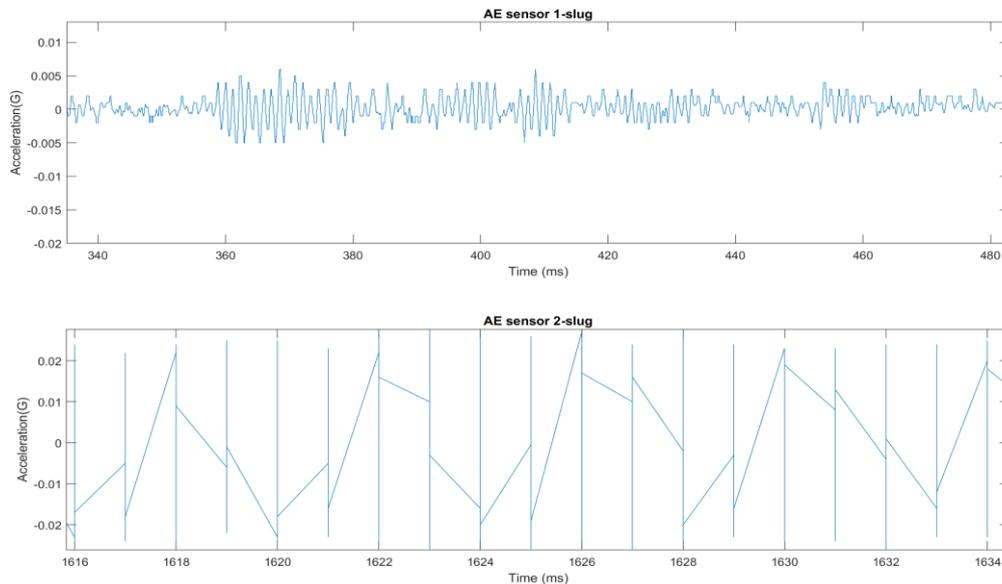


Figure 36 Output of new measurements with longer time.

From the graph, it is evident that the noise level has slightly decreased as compared to previous measurements. But it is still possible to increase the quality of signals by increasing the signal-to-noise ratio. The sample resolution can be increased by increasing the number of bits per measurement. Before making any changes, the raw signals were observed in excel format. While observing, it was found that there were a lot of measurements between 0.001 and -0.001 G. It was observed that the file format suddenly changes when its value is below 0.001G. It was confirmed that the problem was caused by the type of file format. The problem of data loss was not solved by changing the file format as well as saving the signal in terms of a volt.

Different methods were implemented to solve the problem related to recorded signals. Finally, the quality of the signal was improved by changing the sensitivity of the sensors. However, there are still some noises in the acoustic signal which will be removed by filtering it. After that, the decision was made to move forward with the adjustment as described above. It was wise to proceed in our case because the machine learning models perform the analysis based on the extracted features instead of raw signal measurements.

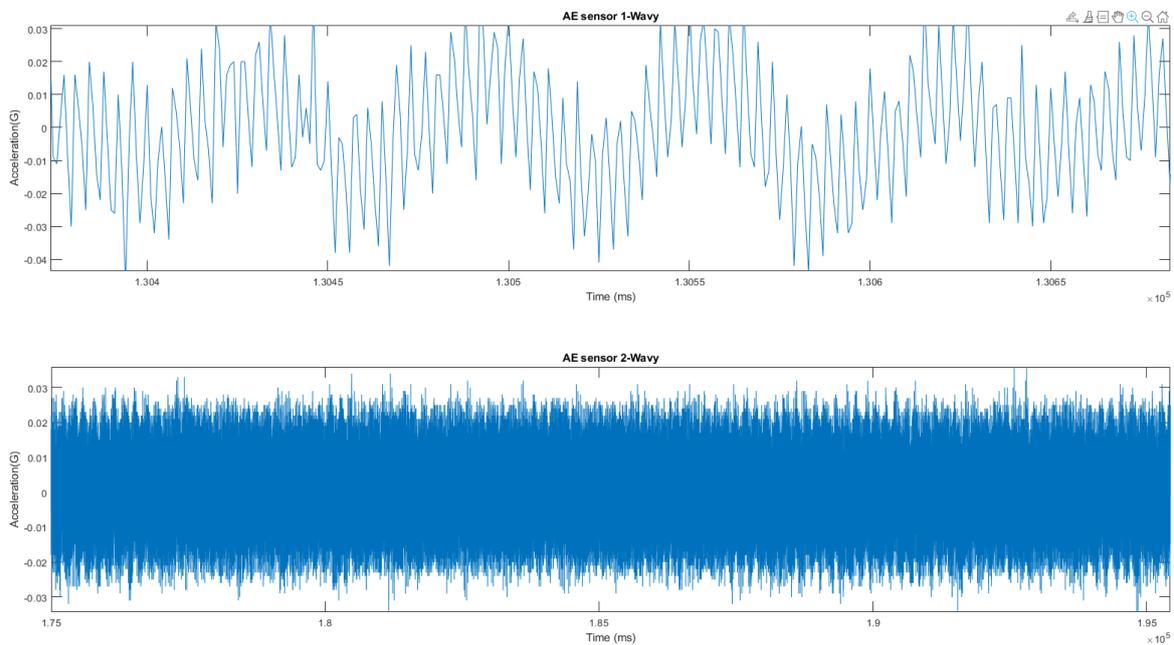


Figure 37 Output of acoustic signals after making adjustment.

## 4.2 Signal analysis of acoustic signal

In this work, the vibration measurements from each acoustic sensor were imported and combined as a single file for the signal processing technique. At first, a single data set was considered and processed to obtain the features from the vibration measurement. The acoustic signals were observed and analyzed using signal processing techniques. After obtaining the results, a new function was created to automate the process for all the experimental datasets. In this way, the raw data from the accelerometer sensors were converted into useful features. This preprocessing step includes data importing, signal processing, and feature extraction. The signal processing was done using a MATLAB program, the source code can be seen in Appendix B.

At first, the acoustic signals were plotted in the time domain to observe how the signal changes over time. The experiments named exp25, exp34, exp57, exp119, and exp124 of the first three-phase test matrix were plotted, which are stratified, wavy, annular, plug, and slug respectively. Figures 4-2, 4-3, 4-4, 4-5, and 4-5 show the time series plots of two accelerometers for stratified, wavy, annular, plug, and slug flow, respectively.

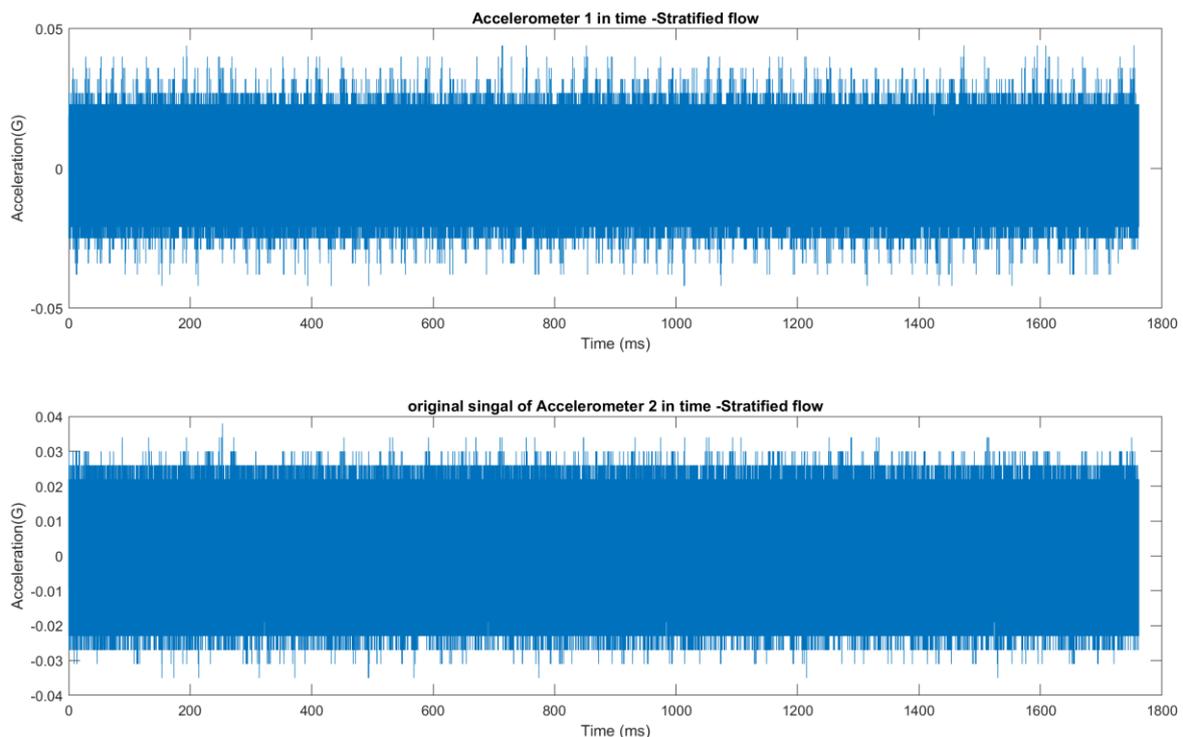


Figure 38 Time series plot of accelerometer 1 and accelerometer 2 of Stratified flow.

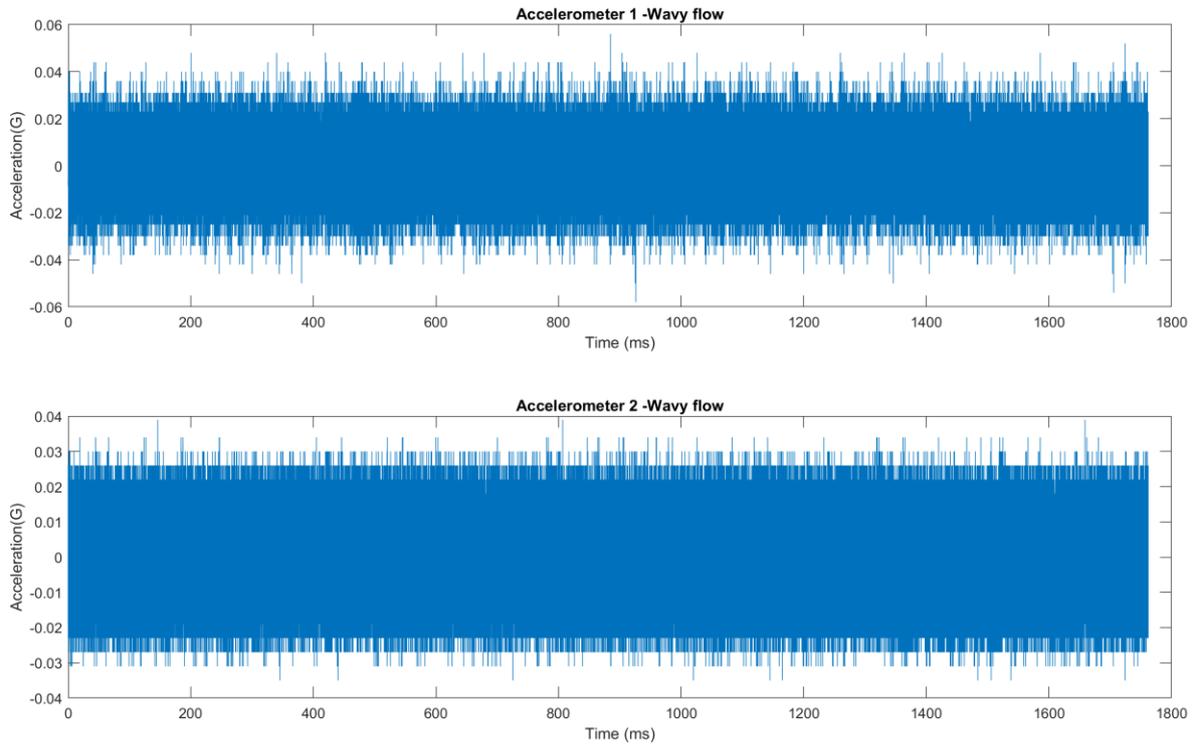


Figure 39 Time series plot of accelerometer 1 and accelerometer 2 of Wavy flow.

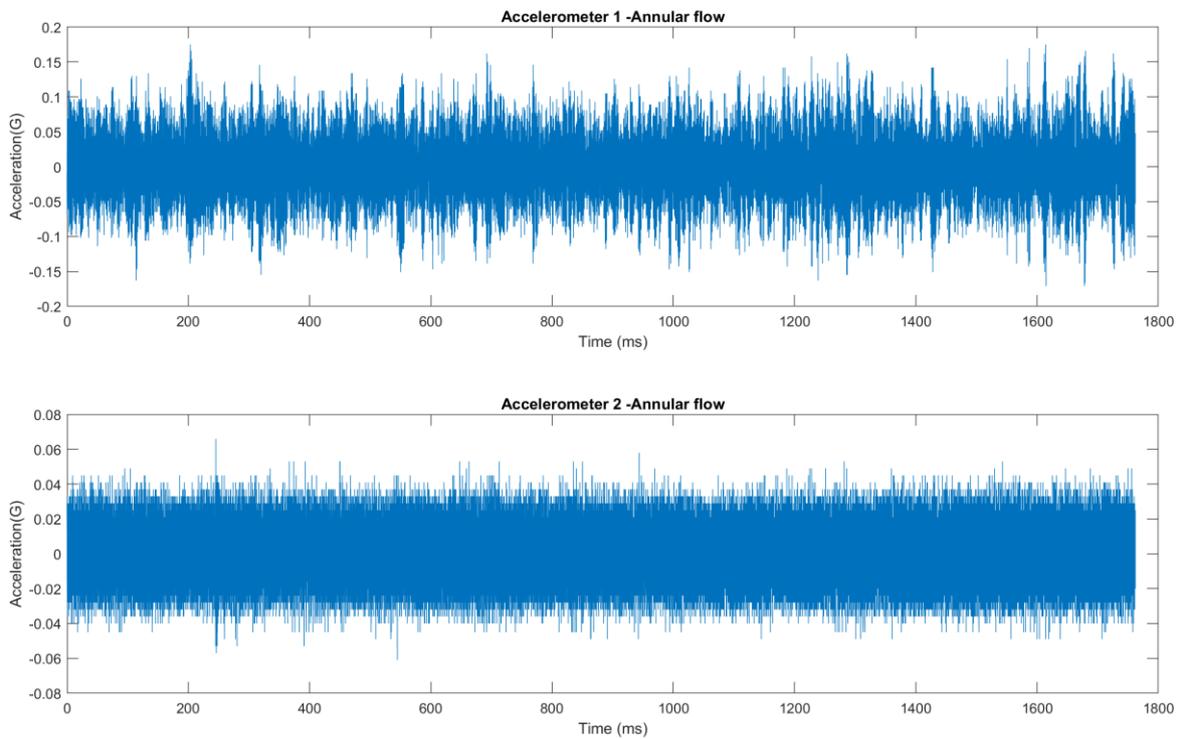


Figure 40 Time series plot of signals from accelerometer 1 and accelerometer 2 for Annular flow.

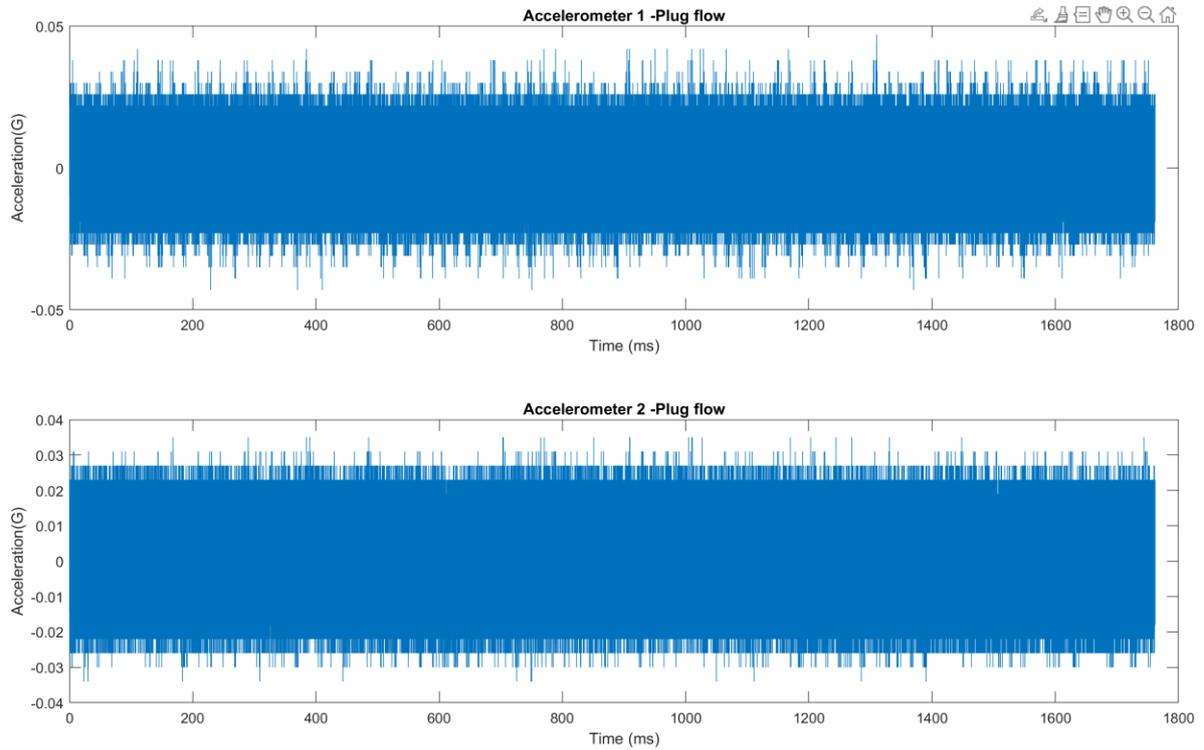


Figure 41 Time series plot of signals from accelerometer 1 and accelerometer 2 for Plug flow.

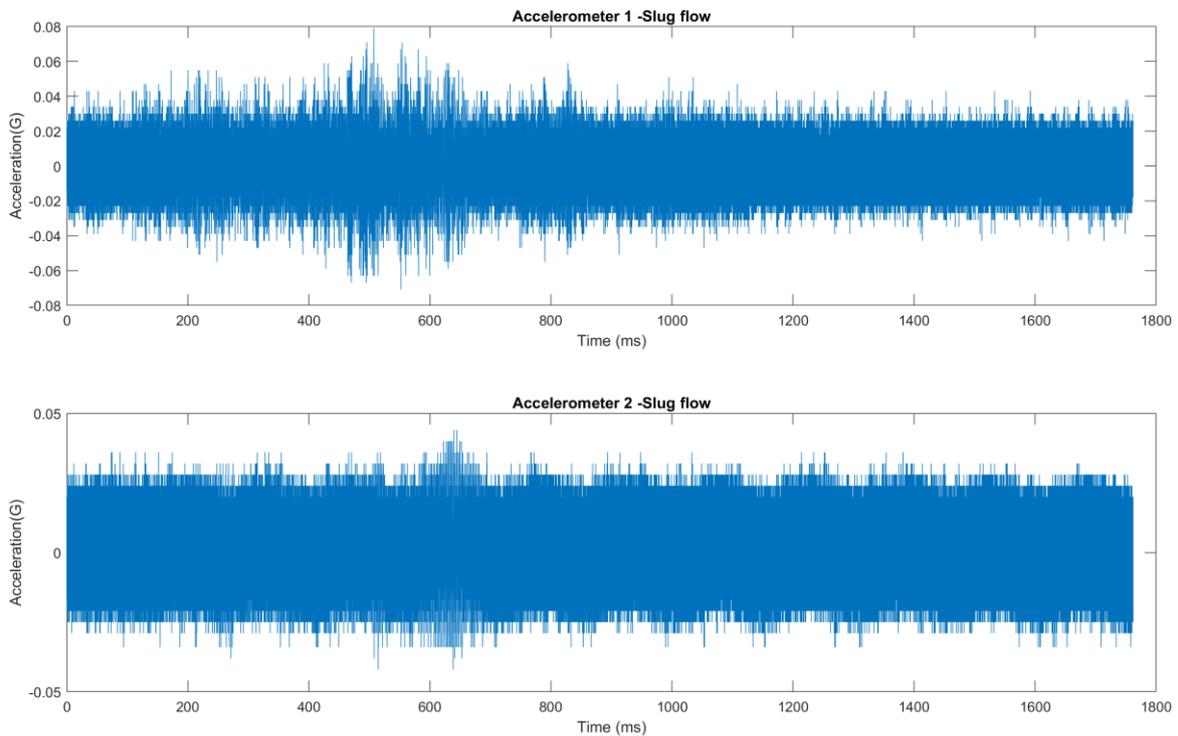


Figure 42 Time series plot of signals from accelerometer 1 and accelerometer 2 for Slug flow.

From the figures above, it's hard to understand the results of accelerometer signals in the time domain. The time series plots can not provide important information about the various flow regimes. As the acoustic sensor records signal based on detected sounds, frequency is a key factor that should be considered when analyzing the acoustic signal. So, the signal was plotted in the frequency domain to better analyze and visualize the signal. The Fourier transform is one of the important methods for analyzing signals in the frequency domain. In this work, the Fast Fourier Transform (FFT) was used to obtain the frequency distribution of the measurement. Figure 43 shows the frequency content of the original signal after applying FFT.

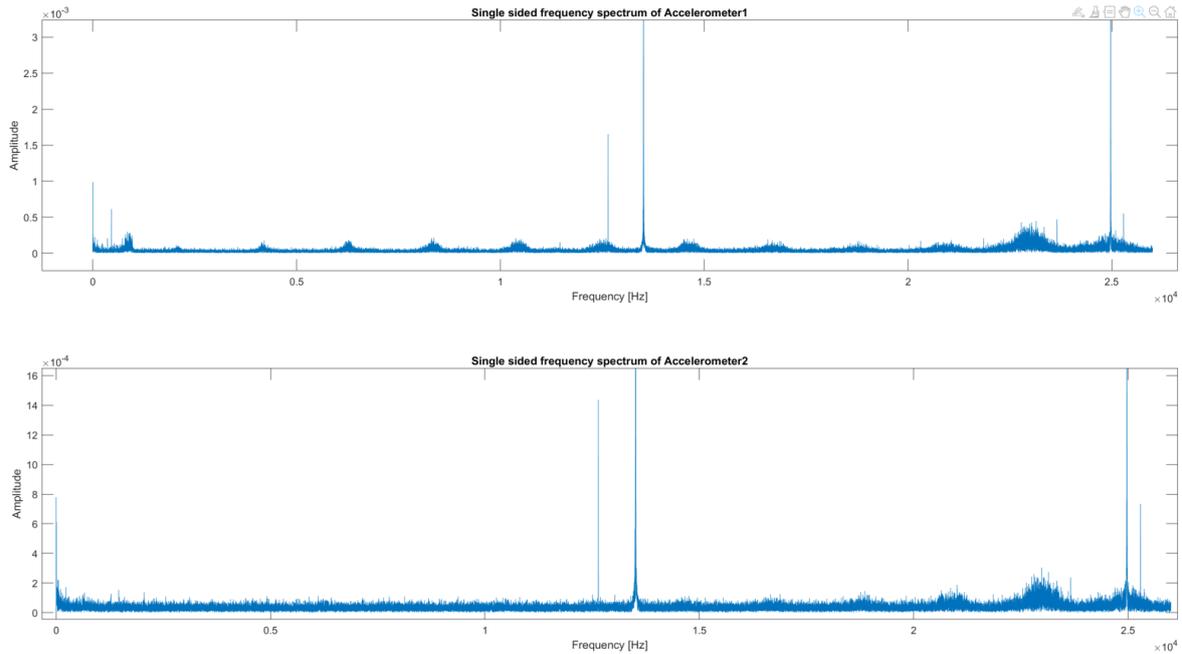


Figure 43 Accelerometer signal output after implementing FFT.

From the above plot of the signal in the frequency domain, it can be observed that the peaks with high amplitudes are located around 13k and 25k Hertz in the frequency spectrum. Similarly, the measurements of each flow regime were plotted in the frequency domain to observe the nature of the curve in the frequency domain. From all the plots, it is observed that there are identical peaks of the same amplitude around 13k and 25k Hertz. Due to the presence of similar peaks in the same frequency range in the plot, it was closely analyzed and concluded that the peaks are caused by external factors like mechanical components of the test rig.

Before designing the filter, it is appropriate to perform spectral analysis and confirm the useful frequency range of the signal in the frequency domain. For that, the power spectral density can be used to understand the distribution of the acoustic signals in the frequency domain. The power spectral density (PSD) measures the distribution of the power signal with respect to its frequency range [19]. In this work, the Welch method was used for estimating the power spectral density (PSD) of the signal. MATLAB has a function called `pwelch` that is used for computing the power spectral density. The `pwelch` function has different options along with FFT and windowing that can generate the single-sided power spectrum of the input signal.

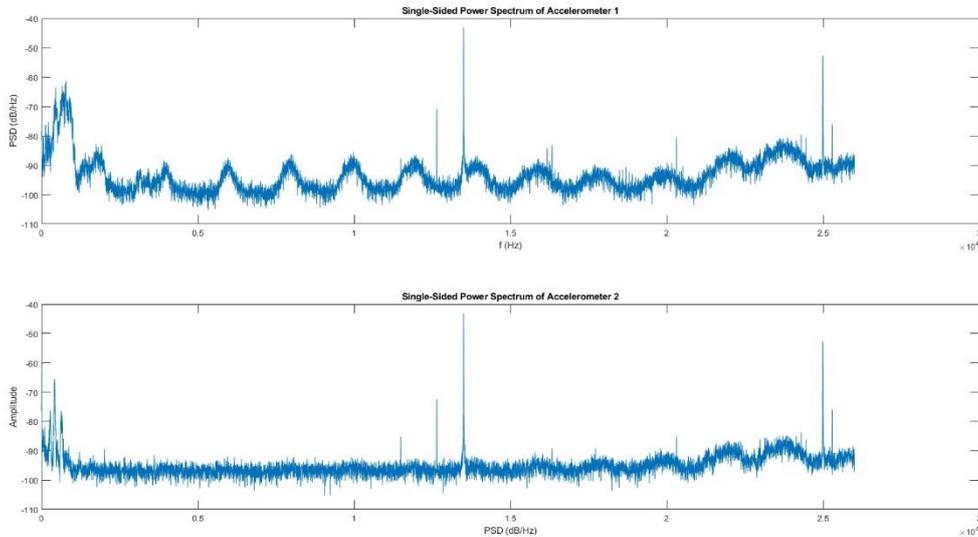


Figure 44 Output of the signal after applying pwelch.

The graph above shows that the same peaks exist within the same range of frequencies. Now, the focus is on the lower frequency range for estimating the multiphase flow. So, the filter needs to be designed to eliminate the peaks at 13k and 25k Hz. For that, the three different types of filters were designed to consider low-frequency signals. All the filters were designed with the help of a filter designer application of MATLAB and imported as a function to filter the unwanted signal present in the raw signal data. With the three filters, the main objective is to consider the signal in different frequency ranges and select the best one for further processing. The first filter is the low-pass filter with a passband frequency of 4500 Hz and a stopband frequency of 5000 Hz. Similarly, the second filter, is also a low-pass filter, with a passband frequency of 10 kHz and a stopband frequency of 11 kHz. For the last filter, the signal was filtered with a bandpass filter from 11kHz to 14kHz to remove the peak present at around 13kHz. Then, the filtered signal was further filtered using the low pass filter, to 24k Hz, to get the appropriate signal. The main screen of filter designer apps can be seen in the figure below.

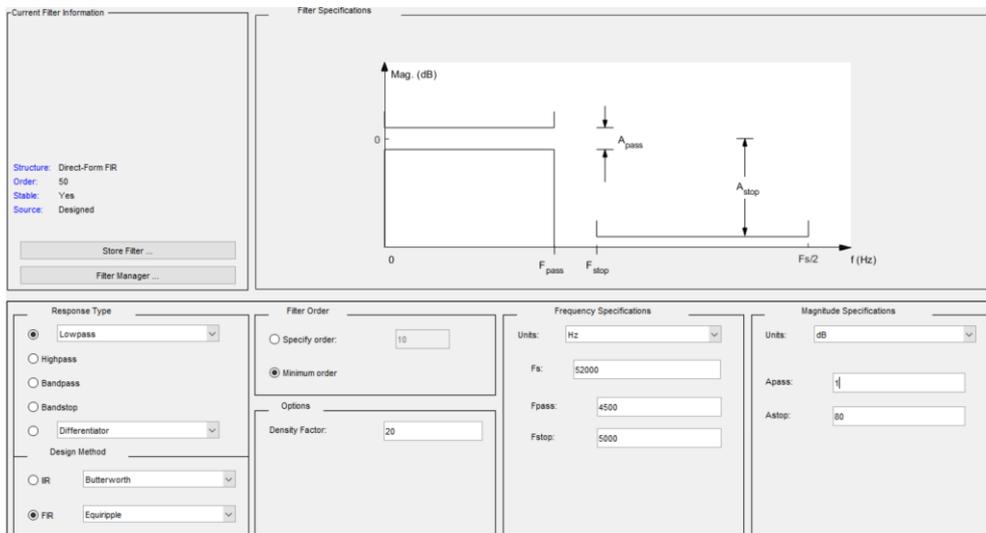


Figure 45 Design of low pass filter using designer learner application.

In the code, the filters were imported as `lpfilter_4.5k`, `lpfilter_10k`, and `filter_24k` functions and used separately to filter the acoustic signals. Then, the filtered signals are compared with the original signal in the frequency domain. Figure 4-13 shows the comparison between the original and filtered signal in the frequency range using the low-pass filter, `lpfilter_10k`, up to a frequency of 10k Hz.

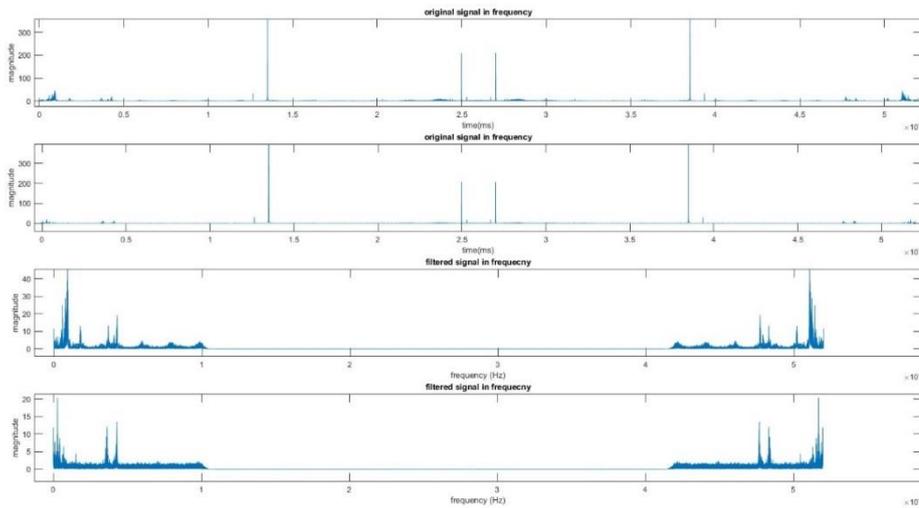


Figure 46 Comparison between original and filtered signal in frequency domain.

For this work, peak analysis can be useful to extract valuable information from input signals. The peaks present in the signal can provide the information needed to uniquely characterize the signals for the estimation of multiphase flow. The next step is to find the peaks and extract the information from the peaks present in the signals. The peaks present in the signal were determined using the `findpeaks` function. For that, the local maxima of the input signal were found using the `find peaks` function available in MATLAB [20]. The peaks of the filtered signals can be seen in figure 47.

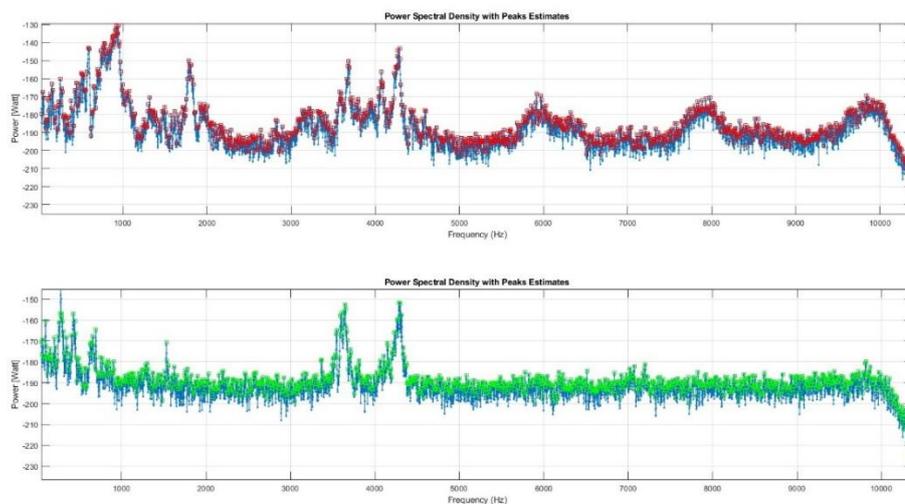


Figure 47 Output of peak estimates of Power Spectral Density.

There are many peaks in the signal and using all of them can lead to overfitting the model. So, the limited peaks were estimated by specifying the conditions such as the number of peaks, and the minimum distance between the peaks [21]. The find peaks function returns the amplitude of the peak and its location in the signal as an output. The output of the find peaks function can be seen in figure 48.

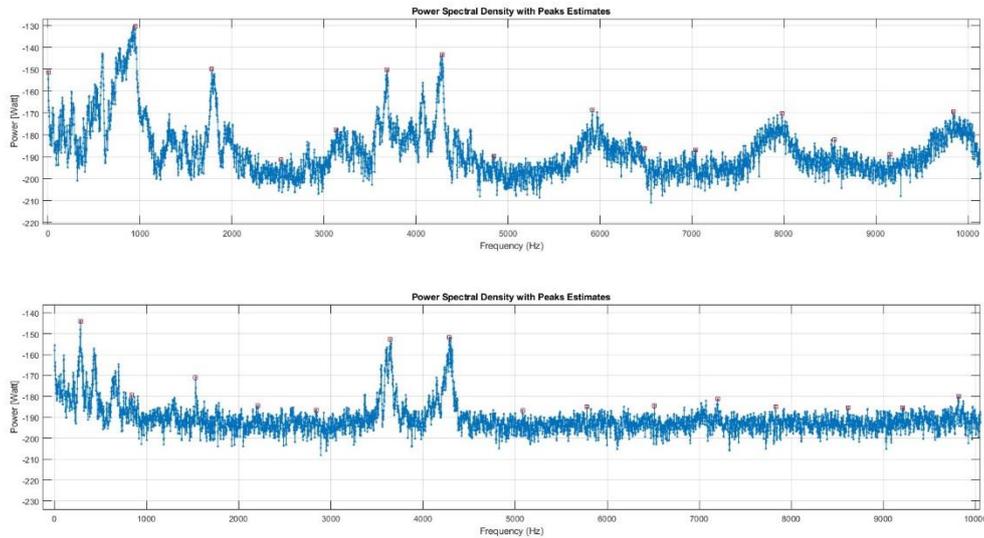


Figure 48 Output of find peaks function after specifying the conditions.

As a result, the obtained matrix contains the peaks and their location as a feature to estimate the multiphase flow. However, these procedures can only extract the features from single measurement. For whole data sets, the new function, extractSignalFeatures, is defined in such a way that it will automatically perform all above steps to extract the features from raw measurements[20].

```
function features = extractSignalFeatures1(ae,Fs,npeaks,minpeakdist)
[~,n] = size(ae);
fhp = lpfilter_10k;

ae = filter(lpfilter_10k,ae);

for k = 1:n
    [p,f] = pwelch(ae(:,k),[],[],[],Fs);

    fmindist = minpeakdist; % Minimum distance in Hz
    N = 2*(length(f)-1); % Number of FFT points
    minpkdist = floor(fmindist/(Fs/N)); % Minimum number of frequency bins

    [pk,loc] = findpeaks(p,'npeaks',npeaks,'minpeakdistance',minpkdist);

    peaks(:,k) = pk;
    freq(:,k) = f(loc);
end
features(1:n) = mean(ae,1); % Mean values
features(n+1:2*n) = median(ae,1); % Median
features(2*n+1:3*n) = mode(ae,1); % Mode
features(3*n+1:4*n) = std(ae,1); % Standard Deviation
features(4*n+1:5*n) = rms(ae,1); % RMS value in signal
features(5*n+1:(5*n+n*npeaks)) = peaks(:); % Peak estimates
```

Figure 49 extractSignalFeatures function for extracting features of whole data set.

In addition, this function can extract the statistical features such as mean, median, mode, standard deviation, and RMS value of the input signal. All the extracted features of acoustic signal are listed in Table 6.

Table 6 Statistical and frequency features of the signal[22].

S. N	Name	Description
1.	Mean	The average value of the signal
2.	Median	The middle value of the signal
3.	Mode	The most repeating value of the signal
4.	Standard Deviation	Measures the variation of signal from its mean value
5.	RMS	Calculates the average value of signal amplitude over time
6.	Peaks and its location	Estimates the number of peaks and their location in the signal

### 4.3 Classification with acoustic signals

The extracted features from acoustic signals were used as an input for flow regime classification. The flow regimes were classified using supervised machine learning models. For that, the extracted features and the label data were combined, and the integrated matrix was used as an input matrix for MATLAB's classification learner apps to classify the flow regimes. The results obtained from different models were unstable, and the accuracies of different models were below 65%. It indicates that the acoustic signals were not sufficient for developing models that could classify flow regimes based on the extracted features. To improve the model, the input data needs to be improved by further processing of acoustic signals and including more parameters in it.

## 4.4 Data preparation

In this step, the features from the acoustic signals of accelerometers, pressure data from PT 131, and differential pressure transmitter data from PDT 120 and PDT 121 were combined to form input matrices for machine learning models. The measurements from accelerometers were combined into a single CSV file for each experiment. Then, the `extractSignalFeatures` function was used to extract the features from all the experimental datasets. On the other hand, the data from PT131, PDT120, and PDT121 were also extracted from LabVIEW measurement datasets and considered as additional features. An overview of the different steps that are involved in the data preparation process is shown in Figure 50.

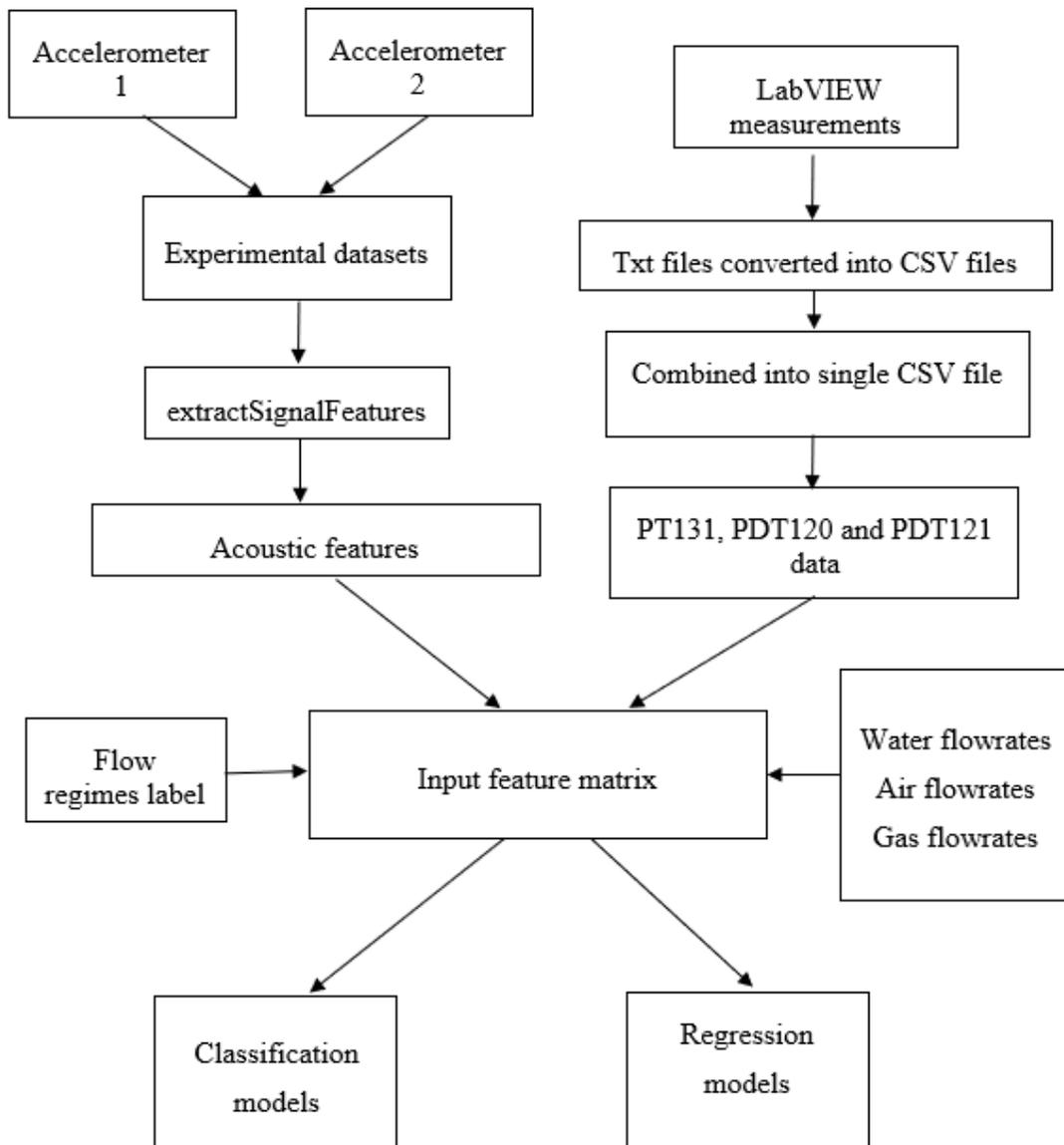


Figure 50 Steps involved in data preparation process.

In the first step, a total of 60 experiments, 15 measurements from each flow type, were considered from the first three-phase test matrix. The acoustic signals were processed, and

features were extracted by following the steps of the data preparation process for all 60 measurements. The models were trained with training and test data sets. The results with these data sets were not so promising because of limited measurements.

In the next step, 80 new experiments were retaken and combined with previously taken measurements to make a total of 173 measurements. Similarly, the features were extracted by following the same procedures. In this case, 15 peaks were selected, with a minimum distance of 550 between them. As a result, 43 features for each data were extracted for 173 input signals. In each measurement, there are 62 measurements that are averaged across 31 samples to obtain 2 data points from each experiment. After averaging, 346 data points were obtained in total as output. The code is attached in Appendix E. In the end, these 346 measurements were used as an input matrix to different models.

## 4.5 Data preparation for image processing

In the beginning, the images of 5 different types of flow regimes obtained from ultrasonic sensors were imported from image data stores. For this method, only 20 images from each category were considered because more measurements could be made due to technical errors in ultrasonic measurements. Generally, a total of 20 images from each flow regime is not sufficient for image processing. But, in this case, the aim of this task is to get familiar with image processing and classify the flow regimes with collected measurements without worrying about the accuracy.

In this case, the pre-trained convolutional neural network was used to perform image processing. ResNet 50, which is available in MATLAB, was installed on the laptop. The ResNet 50 has 1000 object categories, as well as 1.2 million training images loaded, and its architecture is visualized as shown in figure 51.

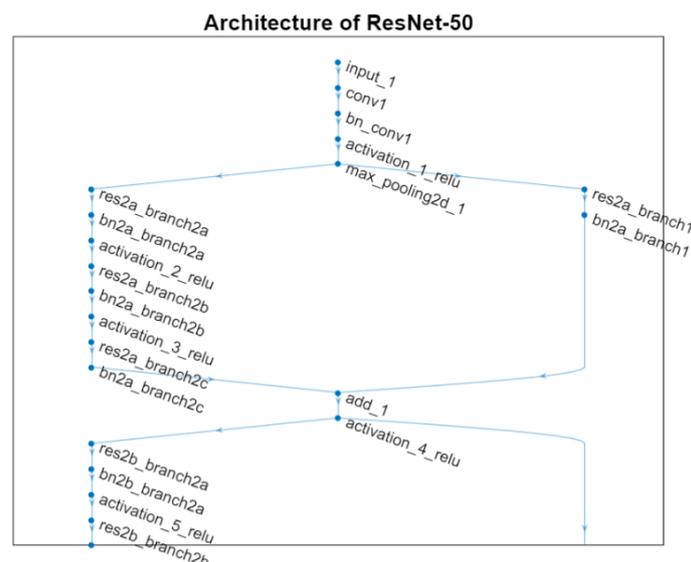


Figure 51 Resize architecture of ResNet-50[23].

The images were converted into size of  $224 \times 224$  with RGB image and imported as input to the first input layer of ResNet-50.

## 5 Results and Discussion

In this chapter, the different models for both classification and regression models were built, and the parameters such as flow regimes and flow rates of different fluids were estimated based on the processed data. This section includes the results from different classification models that classify the different flow regimes in multiphase flow. The classifications made by different models were compared based on their accuracy. It also presents the results of the regression model that estimates the flow rates of gas, water, and oil in multiphase flow.

### 5.1 Classification of flow regimes

Different machine learning models were used to classify the flow regimes into stratified, wavy, annular, plug, and slug flow. For classification, MATLAB has a Classification Learner tool with different models that are based on supervised machine learning. In this work, the flow regimes were first classified using the Classification Learner tool, and the best model was discussed in detail. Besides the Classification Learner tool, the decision trees and random forest model were implemented to separate the flow regimes. In addition, the image processing technique was used for flow regime classification with images obtained from the ultrasonic sensor.

In the model, the extracted features were used as an input for classification. Then, the total data is divided into 80% for training and the remaining 20% for testing the final accuracy of the model. The results for both training and test data are discussed below.

#### 5.1.1 Results from Classification Learner tool

The classification Learner tool consists of different supervised classification algorithms that can be implemented all at a time. It is one of the simple and quick ways to find the best model suitable for different datasets. Multiple models can be run simultaneously in this application, and the best models were categorized according to their accuracy.

In the first case, the input matrix consists of 43 features that were extracted from the acoustic measurements with 14 peaks and data from pressure transmitters. There are five target variables which are represented as 0,1,2,3, and 4, representing stratified, wavy, annular, plug, and slug respectively. Since this is supervised learning, the target variables were also imported along with 43 features as an input to machine learning models. Then, all the models were selected to train the training dataset. During the training, an Ensemble model with the bagging technique was the best model for flow regimes classification. Similarly, various models were tested with test data set. Figure 52 shows the outputs of the different models on the test data set.

Models		
Sort by:	Accuracy (Test) ▼	↓ ↑ 🗑️
☆ 2.21	Ensemble	Accuracy (Test): <b>94.2%</b> Last change: Bagged Trees 43/43 features
☆ 2.20	Ensemble	Accuracy (Test): 91.3% Last change: Boosted Trees 43/43 features
☆ 2.10	SVM	Accuracy (Test): 88.4% Last change: Cubic SVM 43/43 features
☆ 2.26	Neural Network	Accuracy (Test): 87.0% Last change: Medium Neural Network 43/43 features
☆ 2.24	Ensemble	Accuracy (Test): 87.0% Last change: RUSBoosted Trees 43/43 features
☆ 2.19	KNN	Accuracy (Test): 87.0% Last change: Weighted KNN 43/43 features
☆ 2.27	Neural Network	Accuracy (Test): 85.5% Last change: Wide Neural Network 43/43 features
☆ 2.9	SVM	Accuracy (Test): 84.1% Last change: Quadratic SVM 43/43 features
☆ 2.28	Neural Network	Accuracy (Test): 84.1% Last change: Bilayered Neural Network 43/43 features
☆ 2.7	Naive Bayes	Accuracy (Test): 84.1% Last change: Kernel Naive Bayes 43/43 features
☆ 2.11	SVM	Accuracy (Test): 82.6% Last change: Fine Gaussian SVM 43/43 features
☆ 1	Tree	Accuracy (Test): 82.6% Last change: Fine Tree 43/43 features

Figure 52 Results from different models on test data set.

These are the top 12 models with an accuracy of more than 82% for the test dataset, arranged in descending order. In this case, too, the ensemble model with the technique of the bagged tree has the highest accuracy for classification. It concludes that the ensemble model was able to classify both the verification dataset and the test dataset correctly. In figure 53, the confusion matrix shows the estimated flow regimes using an ensemble with bagged trees.

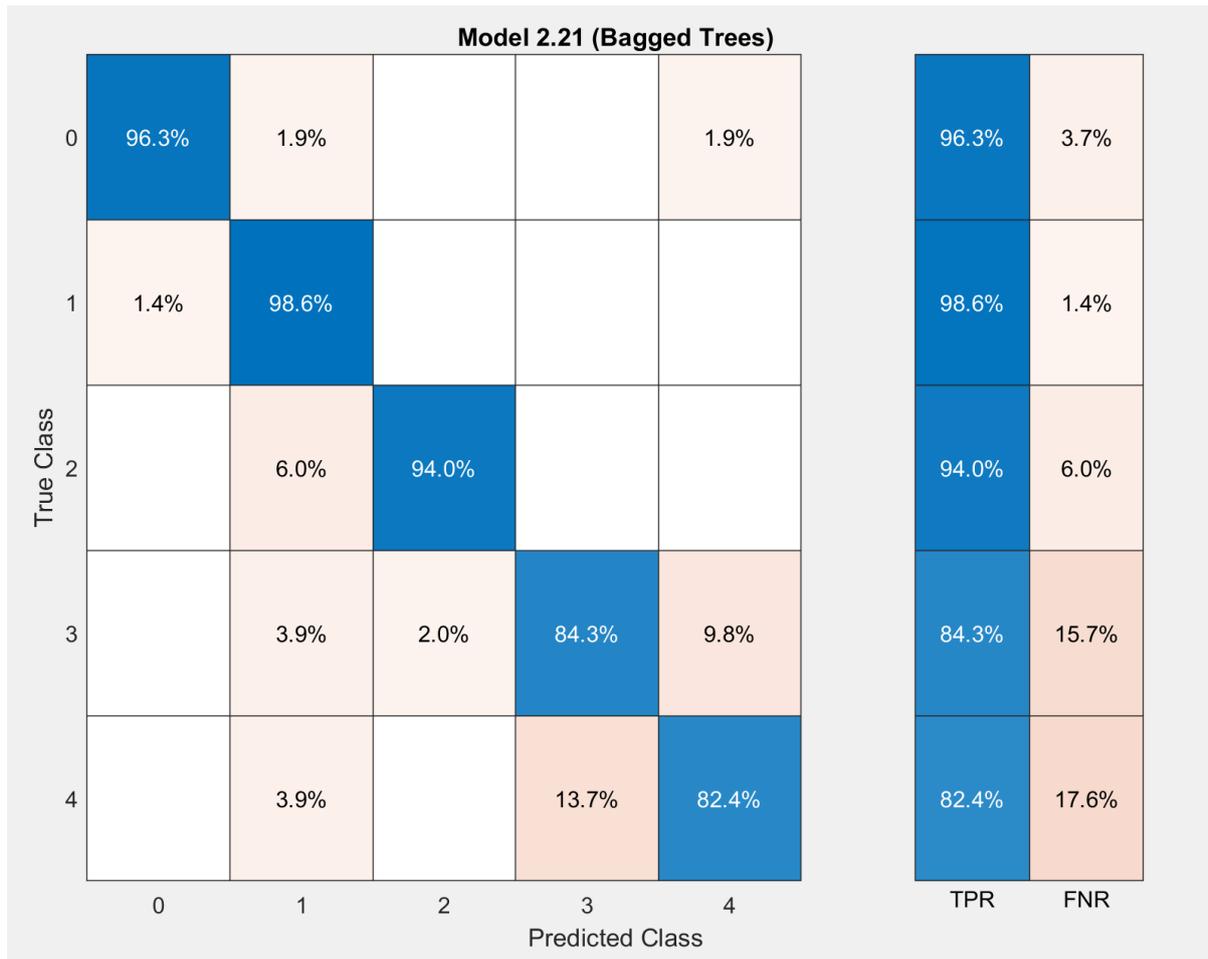


Figure 53 Confusion matrix of Ensemble model.

The confusion matrix shows how different flow regimes were classified into different categories by the ensemble model. It is clearly indicated that there is a misclassification of plug and slug flow as compared to other flow regimes. The model misclassified them because plugs and slugs have somewhat similar properties and natures.

The next step is to select the important features only to improve the model. So, the feature selection has been performed to determine the score of each feature in the classification model. The Classification Learner application has different options for feature selection. Initially, the PCA technique was implemented to select the relevant features. The results from PCA were compared and it was found that the outcomes were not promising, so the ANOVA technique was applied for evaluating the features. This evaluation determines the significance of individual features in classifying flow regimes. Then, the features were selected by observing the graph between features and importance scores. This graph ranks the important features that help to classify the flow regime. Based on the ranking of the features, the relevant and irrelevant features can be selected for the classification model. At first, the irrelevant features were removed to avoid the overfitting of the model. Then, the useful features were selected to increase the accuracy of the models. Figure 54 displays the results of features sorted using the ANOVA algorithm.

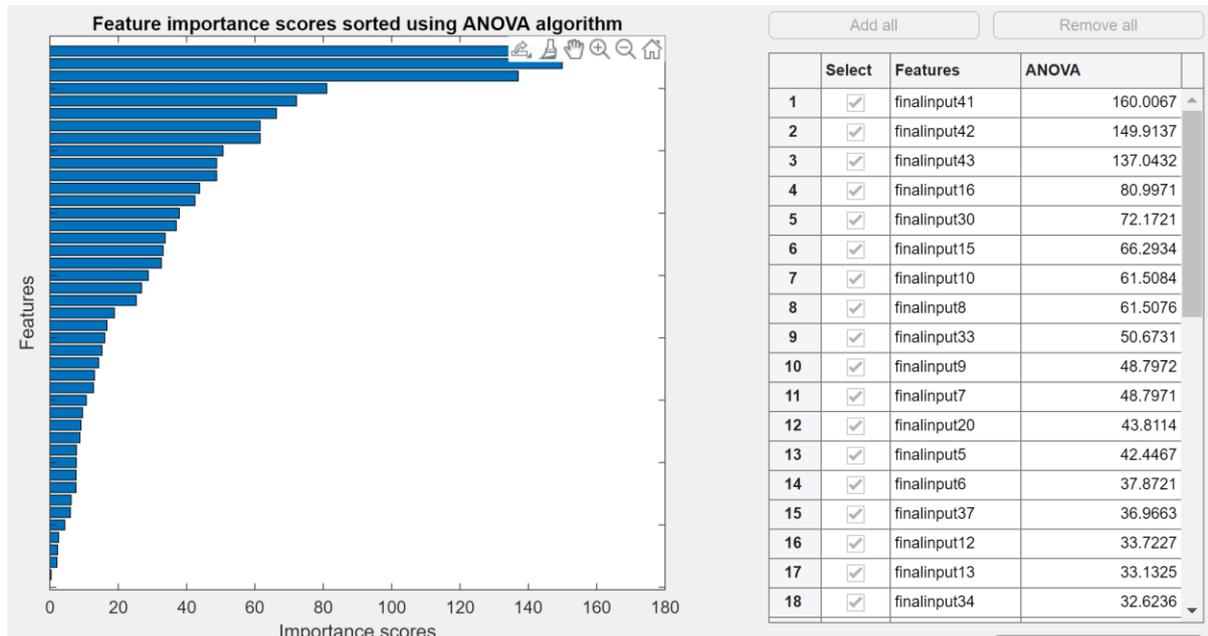


Figure 54 Feature selection using ANOVA technique.

Since the feature selection was done in the classification learning application, it was not possible to edit and add the actual name of each feature. So, the variables were named as the series of the final input files. But the result from the ANOVA technique indicates that PT 131, PDT 120, and PDT 121 are the top three features in this model. It was found that there were some features of the frequency and statistical domains that are not so important in this classification model. Then, the statistical features such as median, and standard deviation were removed, and only 6 peaks were considered.

In the next step, a total of 25 features were considered, and other irrelevant features were removed. Then, the models were trained with a training data set, and the models were tested with a test data set. Figure 55 shows the results of the top 12 models for the test dataset with 25 input features. Figure 56 shows the new confusion matrix of the Ensemble model with the Bagged Trees technique.

Models		
Sort by:	Accuracy (Validation) ▼	↓ ↑ 🗑️
☆ 2.21 Ensemble	Accuracy (Test): 97.1%	▲
Last change: Bagged Trees	25/25 features	
☆ 2.20 Ensemble	Accuracy (Test): 97.1%	
Last change: Boosted Trees	25/25 features	
☆ 2.10 SVM	Accuracy (Test): 89.9%	
Last change: Cubic SVM	25/25 features	
☆ 2.24 Ensemble	Accuracy (Test): 97.1%	
Last change: RUSBoosted Trees	25/25 features	
☆ 2.27 Neural Network	Accuracy (Test): 89.9%	
Last change: Wide Neural Network	25/25 features	
☆ 2.14 KNN	Accuracy (Test): 89.9%	
Last change: Fine KNN	25/25 features	
☆ 2.26 Neural Network	Accuracy (Test): 87.0%	
Last change: Medium Neural Network	25/25 features	
☆ 2.9 SVM	Accuracy (Test): 88.4%	
Last change: Quadratic SVM	25/25 features	
☆ 2.28 Neural Network	Accuracy (Test): 94.2%	
Last change: Bilayered Neural Network	25/25 features	
☆ 1 Tree	Accuracy (Test): 88.4%	
Last change: Fine Tree	25/25 features	
☆ 2.1 Tree	Accuracy (Test): 88.4%	
Last change: Fine Tree	25/25 features	
☆ 2.2 Tree	Accuracy (Test): 88.4%	
Last change: Medium Tree	25/25 features	

Figure 55 Results of different models on a 25- features test data set.

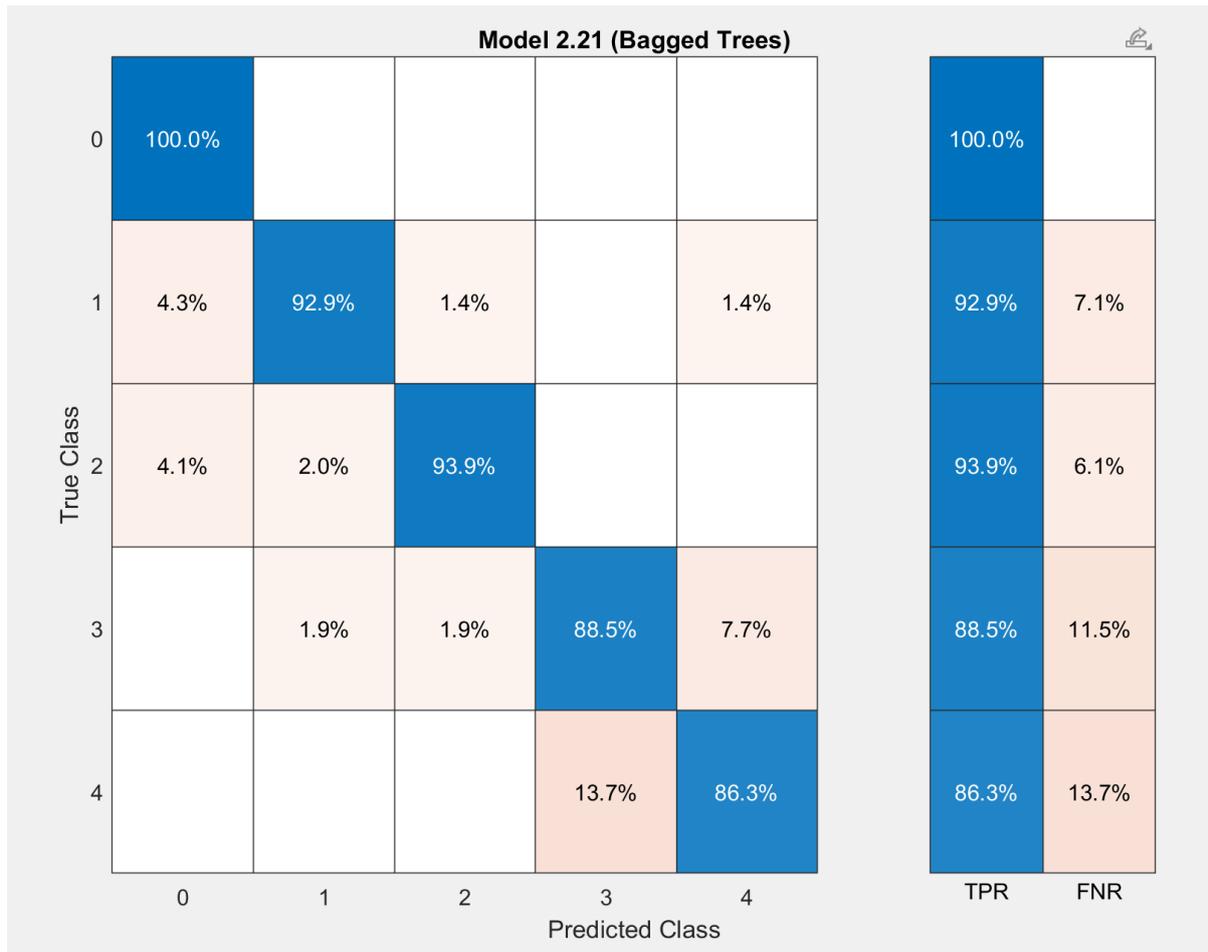


Figure 56 New confusion matrix of Ensemble model on 25 features test data set.

It can be observed that optimizing the features increases the accuracy of different models. It was found that the accuracy of the Ensemble model with Bagging Trees has increased from 94.2% to 97.1%. There is an improvement in the performance of the model for classifying each flow regime, as it is also observed in the confusion matrix above.

### 5.1.2 Results from Decision Tree and Random Forest technique

The models were built for Decision Tree and Random Forest with Bagged Trees with 150 trees. The models were trained and tested separately. In both cases, the model's results and the results obtained through the classification learner are of the same accuracy. It verified that the classifications made by the Classification Learner application were accurate.

```
accuracy_Decisiontree = 94.2000
accuracy_Randomforest = 97.1000
```

Figure 57 Results from Decision Tree and Random Forest models.

### 5.1.3 Results from image processing

At first, the image dataset was randomly divided into 80% for the training and 20% for the validation by the `splitEachLabel` function. The image processing was performed by the `augmentedImageDatastore` function, which converts the greyscale image into an RGB image. Next, the features were extracted, and the model was trained by using the `Fit-Class Error-Correcting Output Codes (ECOC)` function. The `Fitcecoc` function uses a  $k(k-1)/2$  binary support vector machine model using a 1 versus 1 coding design. In our case, the learner was linear classification with one versus all coding instead of a 1 versus 1 coding design. Using test features, the model predicts the labels, and they are compared with the actual labels to find the accuracy of the classifier. Then, the confusion matrix was plotted to evaluate the performance of the classifier. The code is attached in Appendix G. Figure 58 shows the confusion matrix of all flow regimes.

```
confMat = 5x5
    0.9375    0    0    0.0625    0
    0    0.5000    0.5000    0    0
    0    0    1.0000    0    0
    0.0625    0    0    0.6875    0.2500
    0    0    0    0.0625    0.9375
```

```
mean(diag(confMat))
```

```
ans = 0.8125
```

Figure 58 Confusion matrix for stratified, wavy, annular, plug, and slug flow.

Due to the quality of the input data set, the results obtained from the model was not so good. So, in the next step, the wavy, annular and slug flows were only considered. The result obtained from new dataset is shown below.

```
confMat = 3x3
    1.0000    0    0
    0    1.0000    0
    0.0625    0.2500    0.6875
```

```
mean(diag(confMat))
```

```
ans = 0.8958
```

Figure 59 Confusion matrix for wavy, annular, and slug flow.

It indicated that the wavy and annular flow were classified accurately, whereas slug flow was slightly misclassified. Then, the single image of each flow was tested, and the model accurately predicted the image of wavy and annular flow.

## 5.2 Flow rates prediction using Regression Learner tool

The Regression Learner tool that is available in MATLAB was used to predict the flow rates for each phase of the multiphase flow. The Regression Learner application also has a variety of models that can be automatically trained with different models at a time. Just as in the Classification Learner application, the same procedures were followed, and the models were trained for the training and test datasets.

### 5.2.1 Gas flow rates prediction

For the prediction of gas flow rates, the Ensemble Boosted Trees is the best model that has more accuracy than other models. The R squared values for validation and test were 0.95 and 0.97, respectively. It indicates that this model has fitted the values very close to the original measurements. Figure 60 shows the training and test results of the Ensemble Boosted Trees model.

Model 2.14: Ensemble	
Status: Trained	
<b>Training Results</b>	
RMSE (Validation)	0.27424
R-Squared (Validation)	0.95
MSE (Validation)	0.075207
MAE (Validation)	0.16338
Prediction speed	~1200 obs/sec
Training time	12.594 sec
<b>Test Results</b>	
RMSE (Test)	0.20512
R-Squared (Test)	0.97
MSE (Test)	0.042073
MAE (Test)	0.14227

Figure 60 Results of training, and test data sets of Ensemble Boosted Trees.

The graph between true and predicted measurements was observed to better analyze the performance of the gas flow rate prediction model. From the graph, the gas flow rates were correctly predicted, and the patterns of actual and predicted flow rates were similar. Figure 61 shows the graph between actual and predicted gas flow rates. Figure 62 shows the relationship between actual and predicted responses for gas flow rates.

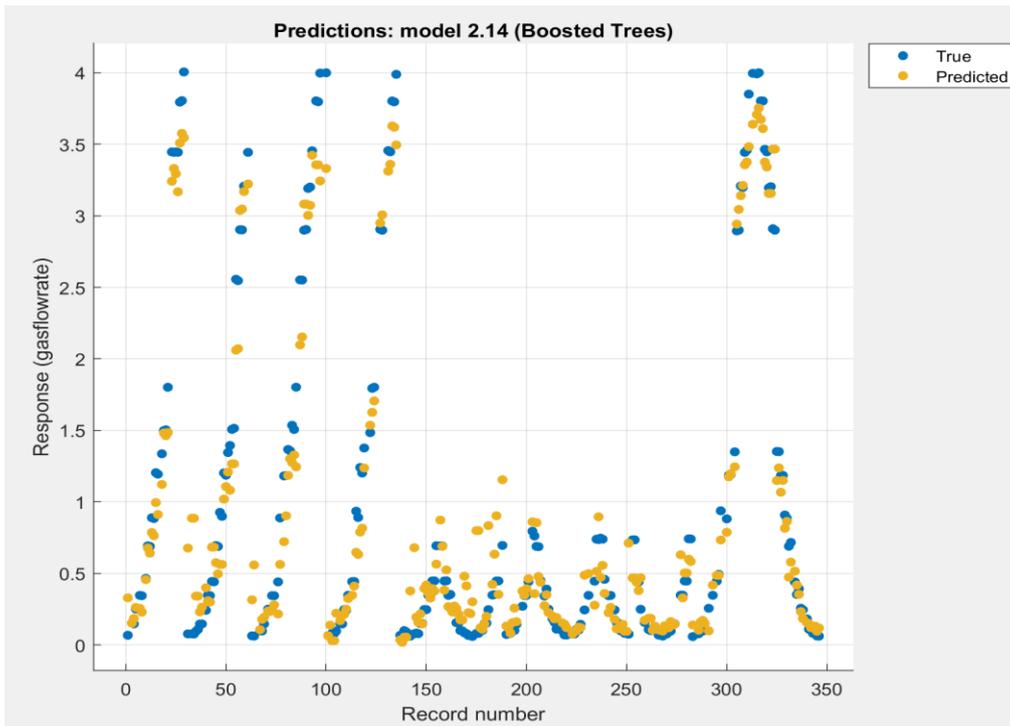


Figure 61 Actual vs predicted gas flow rates by Ensemble Boosted Trees model.

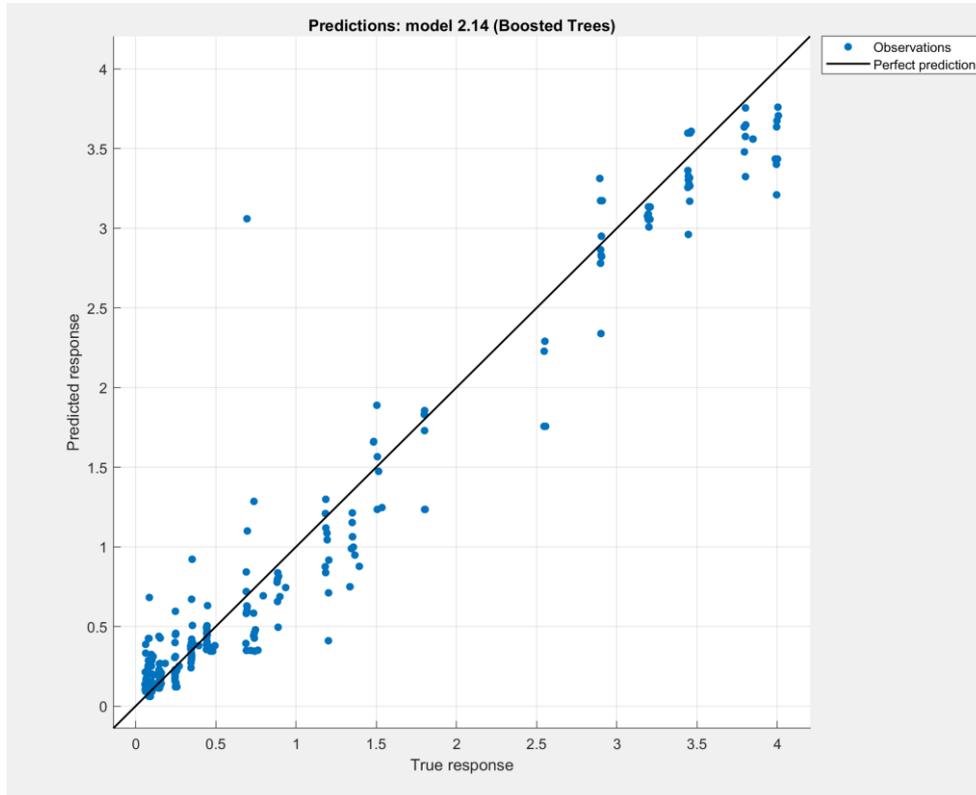


Figure 62 Predicted vs true responses of gas flow rates prediction model.

### 5.2.2 Water flow rates prediction

Similarly, different models were trained and tested to predict the flow rates of water in multiphase flow. In this case, the flow rates of water were accurately predicted by the trees model with a fine tree technique. The results of RMSE, R-Squared, MSE, and MAE for the training and test data sets can be seen in figure 63.

<b>Model 1: Tree</b>	
Status: Trained	
<b>Training Results</b>	
RMSE (Validation)	0.11081
R-Squared (Validation)	0.99
MSE (Validation)	0.012279
MAE (Validation)	0.06481
Prediction speed	~1200 obs/sec
Training time	7.9327 sec
<b>Test Results</b>	
RMSE (Test)	0.21634
R-Squared (Test)	0.97
MSE (Test)	0.046802
MAE (Test)	0.13903

Figure 63 Results of Trees model for training and test data set.

In the graph below, the yellow and blue points indicate the actual and predicted flowrates of water.

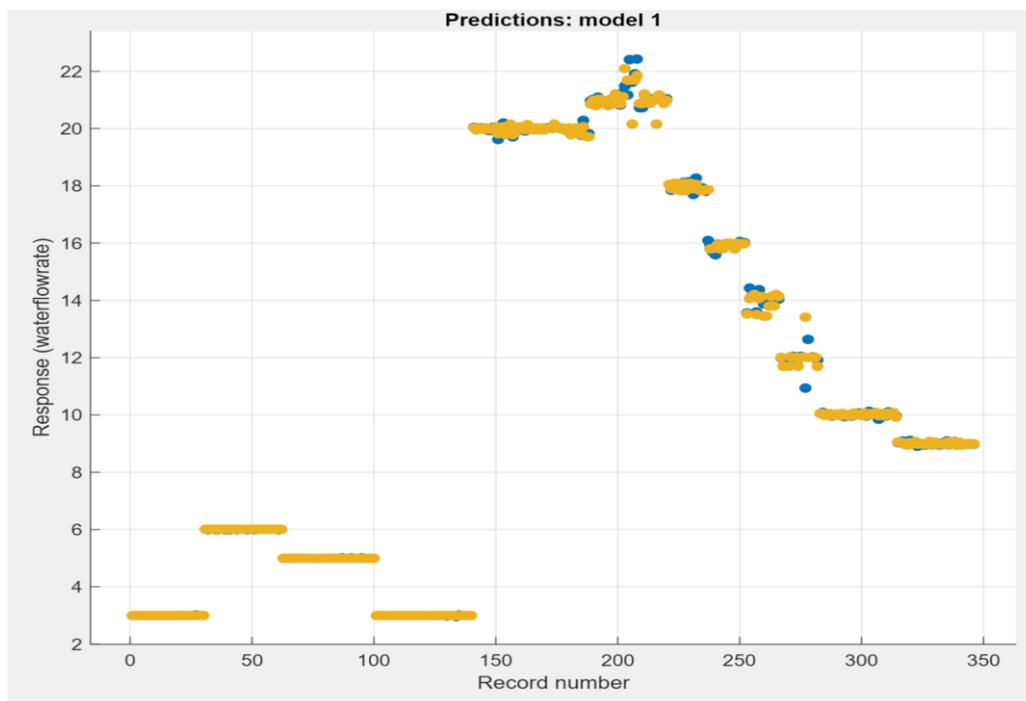


Figure 64 Actual vs predicted water flow rates by Trees model.

In the training data set, the R-squared value was 0.99, while the test data set had a value of 0.97. In both cases, the R-squared value is very close to 1. This means that the model can predict the water flow rates more accurately as compared to the gas flow rate prediction model. Based on the MSE values, it is concluded that the model is more accurate for training data than for test data. Figure 65 shows the graph between the true and predicted responses of the Trees model with a fine tree technique.

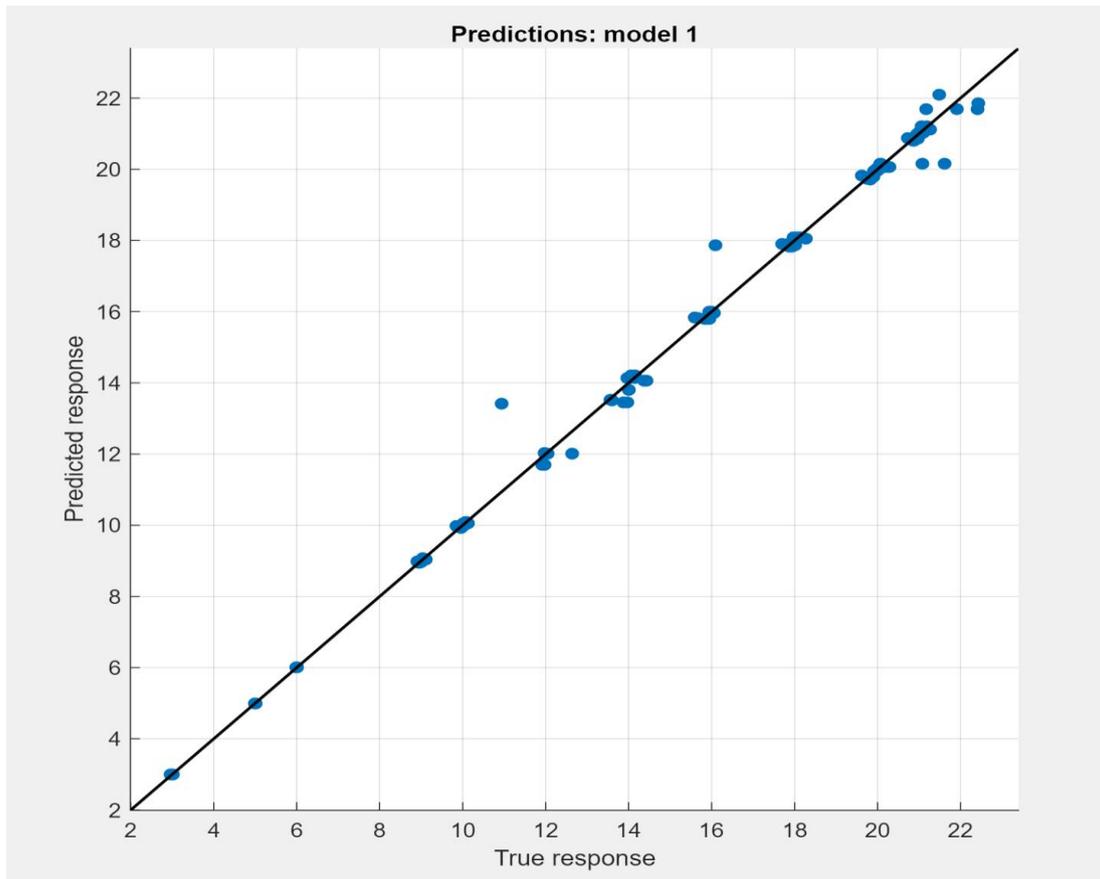


Figure 65 Predicted vs true responses of water flow rates prediction model.

### 5.2.3 Oil flow rates prediction

After training and testing, an ensemble model with boosted trees proved to be the best model for predicting oil flow rates. The R-squared values for the training and test datasets were 0.96 and 0.91. The model was acceptable and the predicted oil flow rates were useful. Figure 66 shows the results obtained after performing training and testing of the model.

<b>Model 2.14: Ensemble</b>	
Status: Trained	
<b>Training Results</b>	
RMSE (Validation)	1.2972
R-Squared (Validation)	0.96
MSE (Validation)	1.6828
MAE (Validation)	0.94554
Prediction speed	~5800 obs/sec
Training time	7.0401 sec
<b>Test Results</b>	
RMSE (Test)	1.9876
R-Squared (Test)	0.91
MSE (Test)	3.9504
MAE (Test)	1.5385

Figure 66 Comparison between training and test results of Ensemble model.

Figure 67 shows the comparison between the actual and predicted oil flow rates.

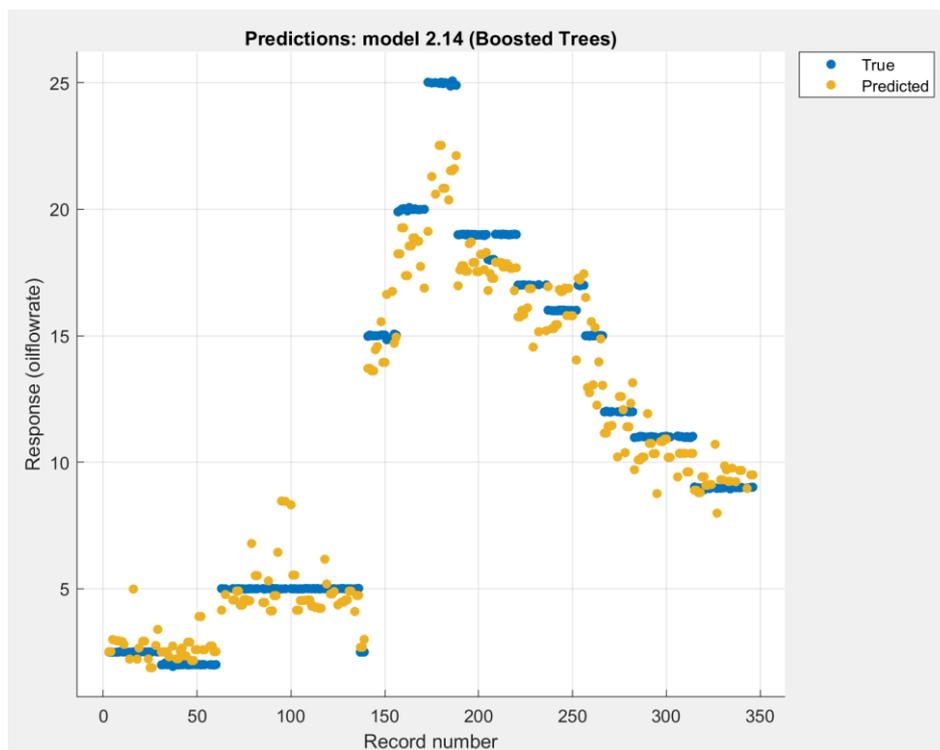


Figure 67 True vs predicted oil flow rates.

The plot between predicted and true responses indicates that the model is correct for predicting oil flow rates. Since there is no any outliers, the accuracy of the model is not affected by them. Figure 68 presents the graph between actual and predicted oil flow rates.

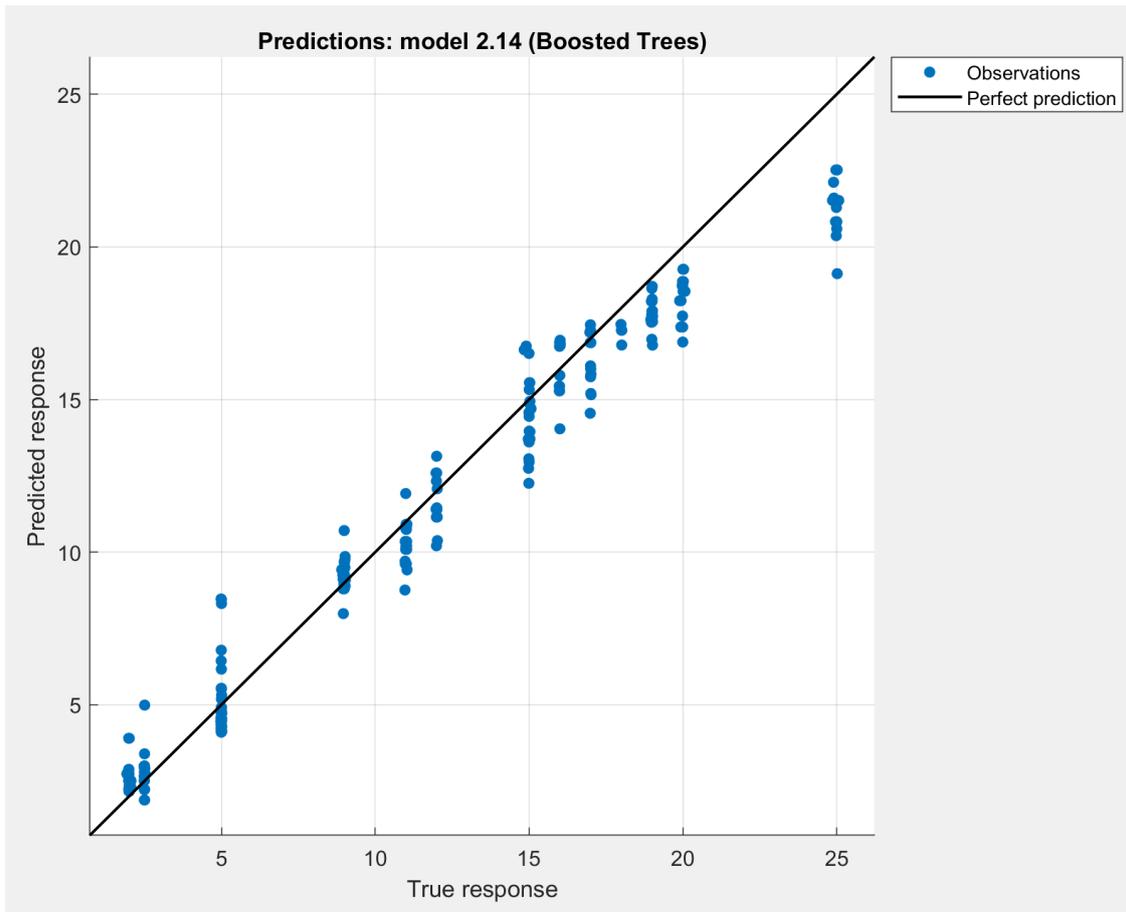


Figure 68 Predicted vs true responses of oil flow rates prediction model.

## 6 Conclusion

It has been demonstrated that machine learning models can estimate multiphase flow from different measurements taken from different sensors present in the test rig. The flow regimes have been successfully categorized into five different types based on the collected measurements. In addition, the flow classification has been performed by using a pre-trained convolutional neural network (CNN). The flow rates of different fluids in multiphase flow were predicted using the Regression Learner application.

The new sensors, accelerometers, and ultrasonic sensors were installed in the existing test rig in the process lab of Campus Porsgrunn. A series of experiments for single-phase as well as multiphase flow were performed, and the measurements were collected using different data acquisition systems. The measurements were observed through the transparent section of the test rig, and they were labeled based on visual inspection. The measurements were analyzed, and different feature extraction techniques, including signal processing techniques, were implemented to extract the useful features from the raw data. Next, the extracted features were divided into training and test datasets and used for training different machine learning models.

The Classification Learner application was used, and different supervised models present in the tool were trained and tested to evaluate the accuracy of each model. The results obtained from various models were compared, and the best model was chosen and discussed in detail. It has been observed that the performance of the model can be increased by optimizing the features. It concludes that the accuracy of the model could be improved by processing and extracting useful features from the dataset. In this case, the ANOVA technique helps to increase the accuracy of the Ensemble Model with Bagged Trees from 94.2 to 97.1%. Then, the Decision Trees and Random Forest techniques were implemented to classify the flow regimes. For the regression part, the flow rates of different fluids such as gas, water, and oil in multiphase flow were predicted using different models present in the Regression Learner tool of MATLAB software. Out of all, the best model for each fluid was selected, and the results were analyzed based on the confusion matrix. The accuracy of each model that predicts the flow rates of oil, water, and oil has an accuracy of more than 90%.

Initially, it was planned to install two ultrasonic sensors and collect the measurements using pulse-echo mode and pitch and catch mode. However, because of the technical problem related to the wedge and time constraints, this task could not be performed properly. But the literature review was performed, and some data was collected by mounting the ultrasonic sensor on the transparent section of the test rig, which is not practical in a real-life scenario. After that, the collected measurements were processed and used for the classification of flow regimes by using image processing. In this case, the new wedge was designed and manufactured for future studies.

## **7 Future work**

A few tasks that are recommended for future work are as follows:

1. Study of bubble formation, droplets formation, classification of flow regimes, etc. using spectrogram plots of acoustic signals and machine learning.
2. Predicting multiphase flow using ultrasonic sensors and image processing techniques.

## 8 Bibliography

- [1] “HANDBOOK OF MULTIPHASE FLOW METERING,” 2005. [Online]. Available: [www.tekna.no](http://www.tekna.no)
- [2] R. Johansen, S. Mylvaganam, and A. Dupré, “Title: Machine Learning Algorithms in Multiphase Flow Regime Identification using Electrical Capacitance Tomography.”
- [3] H. Viumdal, “Title: Flow-Analytics using Multiphase Flow Rig with Multimodal Sensor Suite-with focus on Void Fraction, Water-Cut and Flow Regimes.” [Online]. Available: [www.usn.no](http://www.usn.no)
- [4] J. M. Mandhane, G. A. Gregory, and K. Aziz, “A flow pattern map for gas—liquid flow in horizontal pipes,” *International Journal of Multiphase Flow*, vol. 1, no. 4, pp. 537–553, Oct. 1974, doi: 10.1016/0301-9322(74)90006-8.
- [5] H. B. Arteaga-Arteaga *et al.*, “Machine learning applications to predict two-phase flow patterns,” *PeerJ Computer Science*, vol. 7, 2021, doi: 10.7717/peerj-cs.798.
- [6] U. (Ugo) Maione, B. Majone, Lehto, and Rossella. Monti, *New trends in water and environmental engineering for safety and life : eco-compatible solutions for aquatic environments : proceedings of an international conference, Capri, 3-7 July 2000*. Balkema, 2000.
- [7] “(27) ACCELEROMETER - What it is and how it works - YouTube.” <https://www.youtube.com/watch?v=To7JagpPDwY> (accessed May 30, 2022).
- [8] “T8 + MOBILITY CASE QUICKSTART GUIDE.” [Online]. Available: [www.teltonika.lt](http://www.teltonika.lt)
- [9] “[11]Twave-brochure”.
- [10] M. Toa and A. Whitehead, “Ultrasonic Sensing Basics,” 2021. [Online]. Available: [www.ti.com](http://www.ti.com)
- [11] “Machine Learning in MATLAB - MATLAB & Simulink.” <https://www.mathworks.com/help/stats/machine-learning-in-matlab.html> (accessed May 14, 2022).
- [12] “Machine Learning Random Forest Algorithm - Javatpoint.” <https://www.javatpoint.com/machine-learning-random-forest-algorithm> (accessed May 14, 2022).
- [13] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug. 1996, doi: 10.1023/A:1018054314350.
- [14] “Ensemble Learning: Bagging & Boosting | by Fernando López | Towards Data Science.” <https://towardsdatascience.com/ensemble-learning-bagging-boosting-3098079e5422> (accessed May 25, 2022).

- [15] N. Almalki and W. H. Ahmed, “Prediction of two-phase flow patterns using machine learning algorithms,” in *International Conference on Fluid Flow, Heat and Mass Transfer*, 2020, pp. 182-1-182–9. doi: 10.11159/ffhmt20.182.
- [16] “[4]More info about rig (1)”.
- [17] “Solids and Metals - Speed of Sound.”  
[https://www.engineeringtoolbox.com/sound-speed-solids-d\\_713.html](https://www.engineeringtoolbox.com/sound-speed-solids-d_713.html) (accessed May 14, 2022).
- [18] G. A. Ibe, “The Design and Simulation Patterns in Ultrasonic Wedges for Non-Destructive Testing,” 2014.
- [19] M. Zainuddin, L. Politeknik, N. Batam, and H. Manik, “SIGNAL PROCESSING FOR POWER SPECTRAL DENSITY (PSD)”, doi: 10.13140/RG.2.1.2106.2006.
- [20] G. 1e, “APPLICATION OF NEURAL NETWORK IN TWO PHASE FLOW MEASUREMENT TO DETERMINE FLOW REGIMES.” [Online]. Available: [www.usn.no](http://www.usn.no)
- [21] A. L. Lemme, R. Poudel, R. Yan, H. Viumdal, and N.-O. Skeie, “Title: Anomaly Detection and Maintenance for Chips Packaging Machines using Machine Learning.” [Online]. Available: [www.usn.no](http://www.usn.no)
- [22] A. Zheng and A. Casari, “Feature Engineering for Machine Learning PRINCIPLES AND TECHNIQUES FOR DATA SCIENTISTS.”
- [23] “(14) Convolutional Neural Network (CNN) Image Classification in Matlab - YouTube.” <https://www.youtube.com/watch?v=j8taOvhHcoU&t=2718s> (accessed May 21, 2022).

# Appendices

## Appendix A Task description

### FMH606 Master's Thesis

**Title:** Multiphase flow estimation using accelerometers, ultrasonic sensors and machine learning

**USN supervisor:** Håkon Viumdal, Ru Yan, Saba Mylvaganam

**External partner:** Equinor by Kjetil Fjalestad and SINTEF by Tonni Franke Johansen

**Task background:**

The multiphase flow rig at USN, Porsgrunn, has been built and later modified several times with fundings from the industries and Research Council of Norway. It has been used in various CFD studies, testing different flowmeters and predicting *real time* flow regimes for various compositions of multiphase flows, i.e. water/oil /air, oil/water, air/oil and air/water for more than three decades. Based on extensive experimental work, USN has developed a set of test matrices with details of different compositions of air, water and oil flows for generating different types of flow regimes for the multiphase flow loop in the process lab of Campus Porsgrunn.

Along with conventional measurements such as temperature, pressure, flow, and absorption of gamma rays, tomographic measurements using electrical resistance and capacitance tomographic equipment have been also used in these studies. Recently, in close collaboration with SINTEF, initial measurements campaigns using ultrasonic, acoustic emission and acceleration sensors have been used to interrogate multiphase flows.

For this project, USN has procured new instrumentation and data acquisition systems for ultrasonic and accelerometer based measurements. New measurement campaigns should be planned for including these sensors and data acquisitions systems in the existing multimodal sensor suite used in the multiphase flow rig in the process hall in Campus Porsgrunn of USN.

The master students will gain practical experience by running the experiments in collaboration with the team at USN. After the experiments, the process of preprocessing and analyzing the data will start. Data analytics will include machine learning techniques introduced in the IIA study programme, as well as additional techniques.

USN has been collaborating nationally and internationally, in studying multiphase flows using process tomographic techniques. This project aims to build upon these results and has focus on the fusion of data from these different sensor modalities to estimate different parameters of interest in multiphase flow, e.g. composition of the components in the flow (oil, water and gas/air), flowrates of these components, flow regimes, study of bubble and droplets formations and detection of specific events.

**Task description:**

1. Brief literature survey on multiphase flow and multiphase metering, especially emphasizing the usage of accelerometers and ultrasonic sensors
2. Experimental planning for the planned series of measurements using the multiphase test-rig in the process hall in Campus Porsgrunn
3. Running experiments on the test-rig according to the plan (item 2 above)

4. Pre-processing of data, i.e. cleaning the data acquired for achieving homogeneity, removing obvious outliers etc.
5. Performing exploratory data analysis on the dataset from the experiments, model building, selecting suitable machine learning algorithms etc.
6. Data driven modelling, where flowrates of the various phases are to be estimated
7. Data driven modelling, where various flow regimes are to be estimated (both supervised and unsupervised learning techniques may be considered).
8. The points 2-7 may be repeated several times based on the results obtained in 5,6, and 7 above to gain insight into the models and their behaviour
9. Submitting and presenting 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 students

**Is the task suitable for online students (not present at the campus)?** Yes/No

This task is not suitable for students that are not present at campus, as the student should be able to run several experiments at campus Porsgrunn.

**Practical arrangements:**

The student will be trained to run the multiphase test-rig at campus Porsgrunn, and will have access to the test-rig throughout the project period. The student will be part of a team consisting related to the ongoing SAM-project (SAM Self Adapting Model-based system for Process Autonomy - SINTEF), where Equinor and SINTEF are central partners. There might be short visits to the multiphase rig facilities of EQUINOR, at Herøya. Students may be able to participate in some of the technical meetings in the SAM-project group with external partners in the project.

**Supervision:**

As a general rule, 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:**

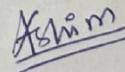
Supervisor (date and signature):



Student (write clearly in all capitalized letters): ASHIM KHADKA

Student (date and signature):

1/02/2022



## Appendix B Data processing of single experiment

```
% Importing Data
ae = importdata('exp18.csv',',',3);
t = ae.data(:,1);
ae1 = ae.data(:,2);
ae2 = ae.data(:,3);
figure
subplot(2,1,1)
plot(t,ae1)
title('Accelerometer 1 -Slug flow')
xlabel('Time (ms)')
ylabel('Acceleration(G)')
subplot(2,1,2)
plot(t,ae2)
title('Accelerometer 2 -Slug flow')
xlabel('Time (ms)')
ylabel('Acceleration(G)')
```

```
%Fs1 = 52000;
%t = 0:1/Fs1:5.12;
%figure
%subplot(2,1,1)
%spectrogram(ae1,1560,1300,1560,Fs1,'yaxis')
%title('Spectrogram of accelerometer 1-wavy flow')
%subplot(2,1,2)
%spectrogram(ae2,1560,1300,1560,Fs1,'yaxis')
%title('Spectrogram of accelerometer 2-wavy flow')
%applying FFT to original signal
N1 = length(ae1);
Fs1 = 52000;
Ts1 = 1/Fs1; %Sampling Time
T1 = N1*Ts1; % Total Time
t1 = 0:Ts1:T1-Ts1; % Time array for plotting
t1 = t1';
fft_1 = fft(ae1);
fft_1 = abs(fft_1(1:length(fft_1)/2 + 1));
fft_1(2:end-1) = 2*fft_1(2:end-1);
f1 = (0:length(fft_1)-1)/T1;

N2 = length(ae2);
Ts2 = 1/Fs1; %Sampling Time
T2 = N2*Ts2; % Total Time
t2 = 0:Ts2:T2-Ts2;
t2 = t2';
fft_2 = fft(ae2);
fft_2 = abs(fft_2(1:length(fft_2)/2 + 1));
```

```

fft_2(2:end-1) = 2*fft_2(2:end-1);
f2 = (0:length(fft_2)-1)/T2;

figure
subplot(2,1,1)
plot(f1,2/N1*fft_1);
title('Single sided frequency spectrum of Accelerometer1');
ylabel('Amplitude');
xlabel('Frequency [Hz]');
subplot(2,1,2)
plot(f2,2/N2*fft_2);
title('Single sided frequency spectrum of Accelerometer2');
ylabel('Amplitude');
xlabel('Frequency [Hz]');

%applying FFt to original signal
X1=fft(ae1);
X2=fft(ae2);
%observing the effect of noise in both time and frequency domain
Fs = 52000; %Sampling frequency
f= linspace(0,Fs,length(ae1));
figure
subplot(2,1,1)
plot(f,log(X1))
title('original signal in frequency')
xlabel('frequency(Hz)'); ylabel('magnitude')
subplot(2,1,2)
plot(f,abs(X2))
title('original signal in frequency')
xlabel('frequency(Hz)'); ylabel('magnitude')

% pwelch is used to comfime the useful frequency range
Fs = 52000; %Sampling frequency
T = 1/Fs;

[P1,f1] = pwelch(ae1,[],[],[],Fs);
[P2,f2] = pwelch(ae2,[],[],[],Fs);

figure
subplot(2,1,1)
plot(f1,10*log10(P1))
title('Single-Sided Power Spectrum of Accelerometer 1')
xlabel('f (Hz)')
ylabel('PSD (dB/Hz)')
subplot(2,1,2)

```

```

plot(f2,10*log10(P2))
title('Single-Sided Power Spectrum of Accelerometer 2')
xlabel('PSD (dB/Hz)')
ylabel('Amplitude')

%%low pass filter with passband frequency of 4500 Hz
%flp = lpfilter_4.5k;
%%low pass filter with passband frequency of 10k Hz
flp = filter_10k;
%%low pass filter with passband frequency of 24k Hz
%flp = lpfilter_24k;

FilteredSignal1 = filter(flp,ae1);
FilteredSignal2= filter(flp,ae2);
%filtered signal converted into frequency domain for analysis
FilterSignalTransform1 = fft(FilteredSignal1);
FilterSignalTransform2 = fft(FilteredSignal2);

%observe the original signal and filtered in time
figure
subplot(4,1,1)
plot(t,ae1)
title('Original signal of accelerometer 1 in Time')
xlabel('time (ms)'); ylabel('magnitude')
subplot(4,1,2)
plot(t,ae2)
title('Original signal of accelerometer 2 in Time')
xlabel('time (ms)'); ylabel('magnitude')
subplot(4,1,3)
plot(t,FilteredSignal1)
title('Filtered signal of accelerometer 1 in Time')
xlabel('time (ms)'); ylabel('magnitude')
subplot(4,1,4)
plot(t,FilteredSignal2)
title('Filtered signal of accelerometer 2 in Time')
xlabel('time (ms)'); ylabel('magnitude')

%observe the effect of filter on the signal in both time domain and
%frequency domain
figure
subplot(4,1,1)
plot(f,abs(X1))
title('original signal in frequency')
xlabel('time(ms)'); ylabel('magnitude')
subplot(4,1,2)
plot(f,abs(X2))

```

```
title('original signal in frequency')
xlabel('time(ms)'); ylabel('magnitude')
subplot(4,1,3)
plot(f, abs(FilterSignalTransform1))
title('filtered signal in frequency')
xlabel('frequency (Hz)'); ylabel('magnitude')
subplot(4,1,4)
plot(f, abs(FilterSignalTransform2))
title('filtered signal in frequency')
xlabel('frequency (Hz)'); ylabel('magnitude')
```

### Appendix C extractionSignalFeatures function to extract the feature

```
function features = extractSignalFeatures1(ae,Fs,npeaks,minpeakdist)
[~,n] = size(ae);
fhp = lpfilter_10k;

ae = filter(lpfilter_10k,ae);

for k = 1:n
    [p,f] = pwelch(ae(:,k),[],[],[],Fs);

    fmindist = minpeakdist; % Minimum distance in Hz
    N = 2*(length(f)-1); % Number of FFT points
    minpkdist = floor(fmindist/(Fs/N)); % Minimum number of frequency bins

    [pk,loc] = findpeaks(p,'npeaks',npeaks,'minpeakdistance',minpkdist);

    peaks(:,k) = pk;
    freq(:,k) = f(loc);
end
features(1:n) = mean(ae,1); % Mean values
features(n+1:2*n) = median(ae,1); % Median
features(2*n+1:3*n) = mode(ae,1); % Mode
features(3*n+1:4*n) = std(ae,1); % Standard Deviation
features(4*n+1:5*n) = rms(ae,1); % RMS value in signal
features(5*n+1:(5*n+n*npeaks)) = peaks(:); % Peak estimates

%features(1:n) = mean(ae,1); % Mean values
%features(n+1:2*n) = mode(ae,1); % Mode
%features(2*n+1:3*n) = rms(ae,1); % RMS value in signal
%features(3*n+1:3*n+n*npeaks) = peaks(:); % Peak estimates
```

## Appendix D Converting fast and slow files from labview to excel files

```
import pandas as pd
import os

# pd.set_option('display.max_columns', None)
# pd.set_option('display.max_rows', None)

def find_title(f_url):
    """
    return the row number of not data
    :param f_url: file path
    """
    container = []
    with open(f_url) as f:
        lines = [line for line in f]
        for i, item in enumerate(lines):
            if '%' in item:
                container.append(lines[i])
            pass
        else:
            break
    return i, container

def clean_csv_file(f_url):
    """
    Clean the csv file into a readable format
    :param f_url: file url
    :return: csv file in the folder
    """
    # How many rows are not data:
    num_skip_raw, col_list = find_title(f_url)
```

```
print('rows to skip: ', num_skip_raw)

# First 1 or 2 rows are datafile general info:
file_info = col_list[0].replace('logging:', ';').split(';')[1].replace('\n', '').replace(' ', '')
print(file_info)

# Remaining rows are variables descriptions:
temp = col_list[2:]
# print(temp)

# Text processing:
col_list = [temp_c.replace('%', '').replace(';\\n', ';').replace(' ', '') for temp_c in temp]
col_list = [temp_c.replace(';Boolean', '-Boolean').replace(']n', '];').replace(';All', ' All')
for temp_c in col_list]
col_list = [temp_c.replace('.\\n', '.;') for temp_c in col_list]
# print(col_list)

# Merge items in list and separated by ';':
col_list = ".join(col_list).split(';')
new_col_list = list(filter(None, col_list))

# print('header info: ', len(new_col_list))

for ind, t in enumerate(new_col_list):
    try:
        if ind != int(t[0:2])-1:
            print(ind, t)
    except ValueError:
        # print(ind, t)
        pass

df = pd.read_csv(f_url, header=None, skiprows=num_skip_raw, sep='\t')
df.columns = new_col_list
```

```
df.index.names = [file_info]
print(df.head(1))
df.to_csv(f_url+'_updated.csv')

# Slow
folder_url_slow = r"D:\thesis\code\lates\plug and slug\w2.5,O3"
g = os.walk(folder_url_slow)
for path, dir_list, file_list in g:
    for file_name in file_list:
        #print(os.path.join(path, file_name))
        clean_csv_file(os.path.join(path, file_name))

# Fast:
#folder_url_fast = r"D:\thesis\code\lates\plug and slug\w13,o9"
#g = os.walk(folder_url_fast)

#for path, dir_list, file_list in g:
#    for file_name in file_list:
#        print(os.path.join(path, file_name))
#        clean_csv_file(os.path.join(path, file_name))
```

## Appendix E Data preparation

```

%% To combine the experiments data into single csv file
clear all
clc
warning off
rootdir='D:\thesis\code\lates\original\copyfile\labview';
outname='allfiles'; % a new name for the output file
d=dir(fullfile(rootdir,'*.csv')); % retrieve the files
fido=fopen(fullfile(rootdir,'newout.csv'),'w'); % open output file to write
allCsv = readtable('D:\thesis\code\lates\original\copyfile\lab-
view\8_supdated.csv', 'HeaderLines', 1);
for i=2:length(d)
    fidi=fopen(fullfile(rootdir,d(i).name)); % open input file
    fullfile(rootdir,d(i).name)
    csv = readtable(fullfile(rootdir,d(i).name), 'HeaderLines', 1);
    allCsv =[allCsv;csv]; % Concatenate vertically
end
header = readtable('D:\thesis\code\lates\original\copyfile\lab-
view\8_supdated.csv').Properties.VariableNames;
allCsv.Properties.VariableNames = header;
fido=fclose(fido);
clear fid*

```

```

%averaging per 31 samples
csvfinal= table2array(allCsv)
N = 31;
movingAverageA = movmean(csvfinal,[(N-1) 0]); % Trailing average of N rows
outputfinal = movingAverageA(N:N:end,:);

```

```

%%PDT and PT as a input variables
x = outputfinal(:,4:6)

```

```

Fs = 52000;
npeaks = 15;
mindistance=550;

expp1 = importdata(append('exp8.csv'))
exp1 = expp1.data;
x1 = [exp1(:,2),exp1(:,3)];
X(1,:) = extractSignalFeatures1(x1,Fs,npeaks,mindistance);

```

```

for i = 9:101
    exp1 = importdata(append('exp', int2str(i), '.csv'))
    exp1 = exp1.data;
    x1 = [exp1(:,2),exp1(:,3)];
    X2(1,:) = extractSignalFeatures1(x1,Fs,npeaks,mindistance);
    X =[X; X2]
    x2=[]
end
%%
for i = 135:213
    exp1 = importdata(append('exp', int2str(i), '.csv'))
    exp1 = exp1.data;
    x1 = [exp1(:,2),exp1(:,3)];
    X2(1,:) = extractSignalFeatures1(x1,Fs,npeaks,mindistance);
    X =[X; X2]
    x2=[]
end

```

```

N=2;
%Z = array2table(X)
Z= kron(X,ones(N,1))
%Z = X
finalinput = [Z x]

```

```

class = array2table(finalinput)
label = table2array(readtable("labelfinal.csv"))

label1= array2table(kron(label,ones(N,1)))

new_table = [class label1]

%corrplot(new_table);

cv = cvpartition(size(Z,1),'HoldOut',0.2);
idx = cv.test;
dataTrain=new_table(~idx,:);
dataTest=new_table(idx,:);
testing=dataTest(1:end,1:end-1);
b = new_table(:,[end])

```

```
%Decision Tree Classifier Code:
model = fitcensemble(dataTrain,'Var1')
%model=fitcensemble(dataTrain,'Var1','Bag',100,'Tree','Type','classification');
prediction=predict(model,testing);
accuracy_Decisiontree =round((sum(prediction==table2array(dataTest(:,end)))/size(dataTest,1))*100,2)
%Random Forest Code:
model=fitcensemble(dataTrain,'Var1','Bag',150,'Tree','Type','classification');
prediction=predict(model,testing);
accuracy_Randomforest=round((sum(prediction==table2array(dataTest(:,end)))/size(dataTest,1))*100,2)
```

```
%water flowrate in kg/min
%y1 = table2array(allCsv(:,11));
%y2 = table2array(allCsv(:,12));
%y1 = outputfinal(:,11);
%y2 = outputfinal(:,12);
%for i = 1:length(y1);
    %if y1(i)> 1
        %a(i) = y1(i)
    %elseif y2(i)> 1
        %a(i) = y2(i)
    %end
%end
%waterflowrate = a'

%%gas flowrate in kg/min
%gasflowrate =table2array(allCsv(:,13));
gasflowrate =outputfinal(:,13);
```

```
%water flowrate in kg/min
%y1 = table2array(allCsv(:,11));
%y2 = table2array(allCsv(:,12));
y1 = outputfinal(:,11);
y2 = outputfinal(:,12);
for i = 1:length(y1);
    if y1(i)> 1
        a(i) = y1(i)
    elseif y2(i)> 1
        a(i) = y2(i)
    end
end
```

```
end
Gasflowrate = a'

%oil flowrate in kg/min
%y3 = table2array(allCsv(:,8));
%y4 = table2array(allCsv(:,9));
y3 = outputfinal(:,8);
y4 = outputfinal(:,9);
for i = 1:length(y3);
    if y3(i) > 1
        b(i) = y3(i)
    elseif y4(i) > 1
        b(i) = y4(i)
    end
end
oilflowrate = b'

%predict = trainedModel.predictFcn(dataTest)
%accuracy
%accuracy_E = round((sum(predict==table2array(dataTest(:,end)))/size(dataTest,1))*100,2)
%incorrect = predict == cell2mat
```

## Appendix F Converting Bin files to JPG files.

```

%% get file names

[file_pc,path_pc] = uigetfile('.bin','Bscan file');

%% Ultralyd data

fid = fopen([path_pc,file_pc]);           % data file
data_pc = fread(fid,2e7,'uint16','b');
fclose(fid);
data_info = importdata([file_pc(1:end-3),'txt'],' ',0); % additional info file

%... get the data
dt = data_info(2)*1e-9;
fs = 1/dt;
N_samp = data_info(3);
N_line = length(data_pc)/N_samp;
rfpc = reshape(data_pc,N_samp,N_line);
t = (1:N_samp)*dt;

%...filtering
[bpc,apc]=butter(2,[1e5 1e7]*dt/2);
rfpc_f = rfpc-mean(rfpc,2)*ones(1,N_line,1);
rfpc_f = filtfilt(bpc,apc,rfpc_f.*tukeywin(N_samp,0.1));

%...get the echo envelope (abs(hilbert()), display in dB)
rfpc_fh = hilbert(rfpc_f);
figure
imagesc(1:1248,t*1e6,20*log10(abs(rfpc_fh)))
colorbar

ylabel('t (\mus)')
xlabel('Line no')
title(['Ultrasound data ',strrep(file_pc,'_','\_')])
set(gca,'YDir',"normal")
[cmi,cma]=caxis;
caxis([round((cma-50)/5)*5, cma])           % use e.g.,50dB dynamic range

[maxval,indpc_max] = max(sum(abs(rfpc_fh)));
[minval,indpc_min] = min(sum(abs(rfpc_fh)));

%...alternative plot of the raw data, RF data plot, logarithmic coding (dB)
figure, plot_wvf_db(1:1248,t*1e6,rfpc_f,60)
set(gca,'YDir',"normal")
title(['Ultrasound data ',strrep(file_pc,'_','\_')])
xlabel('Line no')

```

## Appendix G Image processing for ultrasonic measurements

```

file = fullfile("flowtypes")
catagories = {'wavy', 'annular', 'slug'};
%imagedatastore
imds = imageDatastore(fullfile(file,catagories), 'LabelSource', 'foldernames');
%counting number of images in each catagories
tb1 = countEachLabel(imds)

plug = find(imds.Labels=='plug',1);
slug = find(imds.Labels=='slug',1);

%figure
%subplot(2,1,1);
%imshow(readimage(imds,slug));
%subplot(2,1,2);
%imshow(readimage(imds,plug));
net = resnet50();
figure
plot(net)
title('Architecture of ResNet-50')
set(gca, 'Ylim', [150,170]);

net.Layers(1)

net.Layers(end)
[trainingSet, testSet]= splitEachLabel(imds,0.2, 'randomized');

imagesize = net.Layers(1).InputSize;

augmentedTrainingSet = augmentedImageDatastore(imagesize, trainingSet,"Color-
Preprocessing", "gray2rgb");
augmentedTestSet = augmentedImageDatastore(imagesize, testSet,"ColorPrepro-
cessing", "gray2rgb");

w1 = net.Layers(2).Weights;
w1 = mat2gray(w1);

figure
montage(w1)
title('first convolutional Layer weight')

```

```
%extrating feature from last layer
featureLayer = 'fc1000';
trainingFeatures = activations(net,augmentedTrainingSet, featureLayer, 'Mini-
BatchSize',32, 'OutputAs','columns');

trainingLables = trainingSet.Labels;
classifier = fitcecoc(trainingFeatures, trainingLables,'Learner','Line-
ar','Coding','onevsall','ObservationsIn','Columns');

testFeatures = activations(net,augmentedTestSet, featureLayer, 'MiniBatch-
Size',32, 'OutputAs','columns');

predicLabels =predict(classifier, testFeatures, 'ObservationsIn','Columns');
testLables = testSet.Labels;
confMat = confusionmat(testLables, predicLabels);

confMat = bsxfun(@rdivide, confMat,sum(confMat,2))

mean(diag(confMat))
```