Kent Academic Repository

Abbas, Faisal (2022) *Mass Flow Measurement of Pneumatically Conveyed Solids Through Multi-Modal Sensing and Data-Driven Modelling.* **Doctor of Philosophy (PhD) thesis, University of Kent,.**

Mass Flow Measurement of Pneumatically Conveyed Solids Through Multi-Modal Sensing and Data-Driven Modelling

A Thesis Submitted to the

University of Kent

For the Degree of PhD in

Electronic Engineering

By

Faisal Abbas B.Sc M.Sc

December 2021

Abstract

Bulk material is pneumatically conveyed in many industrial processes such as steel manufacturing, food processing, chemical engineering, cement production, and power generation. On-line continuous measurement of the mass flow rate of solids in pneumatic conveying pipelines is essential to balance the mass and energy and further to control the energy efficiency and raw material consumption. This thesis describes a novel method of mass flow rate measurement of pneumatically conveyed solids using multi-modal sensing incorporating data-driven models. A review of direct and inferential methods and the methods incorporating the data-drivel models is given with the technical requirements and limitations in the field of gas-solid two-phase flow.

The multi-modal sensing system is comprised of an array of ring-shaped electrostatic sensors, four arrays of arc-shaped electrostatic sensors, a differential pressure (DP) transducer, a temperature and humidity sensor and an accelerometer. The ring-shaped and the arc-shaped electrostatic sensors are capable of measuring the cross-sectionally averaged and the localized velocity and mass flow rate of solids, respectively. The DP transducer is capable of measuring the drop in line pressure caused by different air velocities, mass flow rates of solids and pipe orientations. The temperature and humidity sensors are integrated to measure the temperature and RH values inside the pipe under different ambient conditions. The accelerometer is installed to acquire the information related to different orientations of pipe.

The data-driven models, including artificial neural network (ANN), support vector machine (SVM), and convolutional neural network (CNN), are established through training with statistical features extracted from the post-processed data from the sensing system. Statistical features are shortlisted based on their importance by calculating the partial mutual information between the features and the corresponding reference mass flow rate of solids. The SVM model is also applied on the data of each sensor and different combinations of sensors to evaluate the importance of each sensor in terms of relative error in predicted mass flow rate measurements. Experimental work was conducted on a laboratory-scale test rig with different air velocities, pipe orientations and ambient conditions to measure the mass flow rate of solids. The data-driven modelling techniques are also applied for the measurement of coal and biomass ratios in the coal/biomass/air mixture. Experimental results suggest that the measurement system is capable to reduce the measurement errors to within ±12% under all the physical conditions.

Acknowledgements

Contents

Nomenclature

A   Cross-sectional area of the pipe

A_{rms} Root mean square value of the electrostatic sensor signal

a   Input statistical feature

a Bias

α Constant that determines the confidence limits of the outlier.

α General scaling factor

b   Bias

β Firefly attractiveness value

β_s Volumetric concentration of solids

c   Weight of the adaptive neuro-fuzzy inference system

C   Center vector

C Resultant of the continuous wavelet transform

$\bar{C}$ Mean of the continuous wavelet transform

d   Degree of polynomial

d Probability distribution of S

E($\bullet$)   Mean of the signal

f Function to check if the difference is higher than pre-defined threshold

f_x Frequency where the power of the signal is reached x% of the total power of the signal

I_{fit} Fitness function of firefly algorithm

L   Axial distance between upstream and downstream electrode

L_o	Number of nodes in the hidden layer
m	Number of input variables
N	Total number of samples
N_f	Total number of fireflies
N_p	Total number of parameters to be optimized
N_{max}	Maximum number of iterations
n	Number of training samples
p	Probability distribution of x
ρ_s	True density of solids
$q_{m,s}$	Mass flow rate
r	Cross-correlation coefficient
r_{ij}	Distance between fireflies
γ	Media light absorption coefficient
rand	Random number generator
σ	Standard deviation
σ^2	Variance
R_L	Load resistance
R_{Lmax}	Max load resistance
R_{xy}	Cross-correlation function
S_k	Scaling factor of each parameter
S	Power spectrum
$\bar{S}$	Mean of power spectral density

T	Transpose of input matrix
τ_m	Time delay between upstream and downstream electrode
U	Supply voltage of DP transducer
u	Residual of the desired outputs
$\mu(x)$	Generalized bell function
v_c	Cross-correlation particle velocity
ω	Weight
x	Samples of the sensor signal
X	Discrete Fourier transform of x
$\bar{y}$	Mean of the mass flow rate
$\widehat{y_k}$	Predicted solids mass flow rate

List of Abbreviations

AC   Alternating Current

AIC   Akaike Information Criterion

AVL   Average Pooling Layer

ANFIS   Adaptive Neuro-Fuzzy Inference System

AWNN   Adaptive Wavelet Neural Network

ANN   Artificial Neural Network

BNL   Batch Normalization Layer

BP-ANN   Backpropagation Artificial Neural Network

BOP   Basic Operator Panel

CFB   Circulating Fluidized Bed

CL   Convolutional Layer

CNN   Convolutional Neural Network

DAQ   Data Acquisition

DBN   Deep Belief Network

DBM   Deep Boltzmann Machine

DC   Direct Current

DNN   Deep Neural Network

DCSA   Differential Capacitance Sensor Array

DP   Differential Pressure

EMD   Empirical Mode Decomposition

ELM   Extreme Learning Machine

FA   Firefly Algorithm

FDA   Fisher Discriminant Analysis

GP   Genetic Programming

HHT   Hilbert Huang Transform

LED   Light Emitting Diode

MPL   Max Pooling Layer

NRMSE   Normalized Root Mean Square Error

NSTD   Normalized Standard Deviation

PCA   Principle Component Analysis

PMI   Partial Mutual Information

PSO   Particle Swarm Optimization

PSD   Power Spectral Density

PDF   Probability Density Function

RBF   Radial Basis Function

ReLu   Rectified Linear Unit

RNN   Recurrent Neural Network

RSD   Relative Standard Deviation

RH   Relative Humidity

RMS   Root Mean Square

SVM   Support Vector Machine

TDNN   Time Delay Neural Network

VFD   Variable Frequency Drive

List of Tables

List of Figures

xxi

Chapter 1 Technical Requirements for Mass Flow Measurement of Pneumatically Conveyed Solids

1.1 Introduction

Bulk material is pneumatically conveyed in many industrial sectors including power, cement, chemical, agriculture, food and shipping. On-line continuous measurement of the mass flow rate of solids in pneumatic conveying pipelines is essential to balance the mass and energy and further to control energy efficiency and raw material consumption. There has been a long history in developing techniques for the mass flow rate measurement of pneumatically conveyed solids [1]. However, each technique has its own set of benefits and drawbacks depending on the application. Further study is instantly required in terms of the accuracy, robustness, reliability, and the sensitivity of the measuring tools.

This chapter provides an introduction about the gas-solid two-phase flow and the industrial sectors where they can be found; requirement of the online continuous measurement system with industrial perspective including the penalties that can be faced during the harsh situations; challenges which are faced for the measurement of mass flow of pneumatically conveyed solids in a two-phase flow; superiority and necessity of data-driven modelling techniques over direct measurement techniques with different aspects of flow conditions. Running a process with pneumatically conveyed solids under different physical conditions including pipe orientations, ambient conditions and solids velocity and concentration, may have an adverse impact on the sensing system which can severely effect the mass flow measurement. Therefore, this chapter also provides the justification to carry out this research in the way it has been proposed. The novelties of the research in terms of design and implementation of experimental setup and the data-driven modelling based mass flow rate measurement technique is well explained.

1.2 Gas-Solid Two-Phase Flow

Two-phase gas-solid flows can be found in a variety of natural phenomena as well as many industrial processes that include pneumatic solid transportation. Industrial emissions, fluidized beds systems, and especially pneumatic conveying materials in pipes are the principal sources of gas-solid flow in the industrial sectors. Many industrial processes, that involve solids handling, such as agricultural product processing, cement, chemical, food processing, metallurgy, mining, pharmaceutical, plastic, semiconductor, and power generation, have widely adopted pneumatic transportation of solids materials through enclosed pipe. Since the early twentieth century, pneumatic transportation of particulate materials via pipes has been used in a variety of industrial operations.

Pneumatic conveying is the transportation of solids by a gas through a pipeline. Pneumatic conveying systems are highly flexible, yet simple systems that may be used to transport a wide range of powdered and granular materials. A pneumatic conveying along with measurement system may be considered as comprising a source of compressed gas, a device to feed solids into the pipeline, the conveying pipeline and a setup which will contain a set of sensors, data acquisition module, and a processing unit as shown in Figure 1-1. A source of air or any kind of gas is required to transport the solids from one end of the pipe to another. For this purpose, a fan/blower can be used to create a positive pressure vacuum or a suction pump can also be used to create a negative pressure in the pneumatic pipeline. This is the major element of the conveying system which is responsible to control the velocity of solids. The concentration of solids highly depends on the source that is delivering the solids in the pneumatic pipeline. The slight variation in the concentration of solids can introduce large errors in the reference data for mass flow rate measurement, therefore a systematic approach is required that could transfer the consistent amount of solids in the pipeline. One of the most reliable method to transfer the solids into pipeline is based on the vibratory feeder. The solids are stored in the hopper that is placed at one edge of the feeder from where they are transferred into the pipeline through vibrations via metallic plate. In a pneumatic conveying system, the pipeline structure is utilized to confine and regulate the particle flow. The pipeline structure can contain horizontal, vertical and also the inclined pipe sections in pneumatic conveying systems. The bore of the pipeline structure varies depending on the

pneumatic conveying method used, and can range from tens to thousands of millimeters [2]. The pipe orientation and the diameter size play a significant role for the type of flow regime of solids.
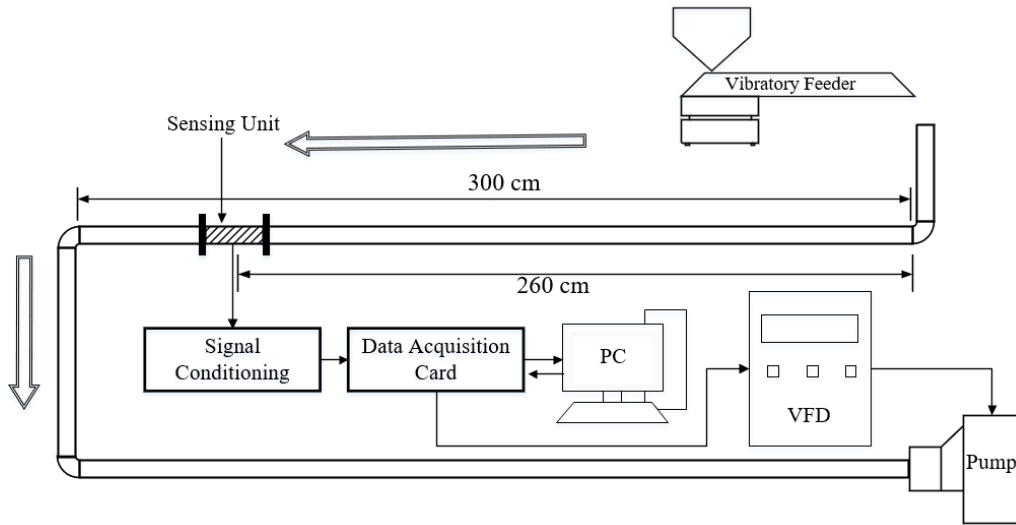
Figure 1-1: Schematic of pneumatic conveying system

1.3 Online Continuous Measurement of Mass Flow Rate of Solids

An accurate, reliable, on-line, continuous and non-invasive measurement of mass flow rate of solids is desirable where the mass and energy balances are required in many industrial processes to achieve the efficient utilization of energy and raw materials. For example, considering the case of producing electricity from a furnace that takes the pulverized fuel as its input. In order to improve the efficiency of the burning process in terms of reduced fuel consumption and reduced carbon emissions of the furnace, online measurement of the velocity, concentration, and mass flow rate of the pulverized fuel is required to control the performance of the furnace. Correspondingly, the measurement of these flow parameters is important from the perspective of maintenance of pulverizing mill since it can be dangerous for the pneumatic pipeline providing pulverized fuel if the mass flow of solids is not the same which is set by the user.

A classical method to measure the flow parameters is based on the separation of two-phase material into two single-phase materials followed by single-phase flow monitoring equipment. This classical method is reasonably accurate and efficient. However, it requires very expensive and bulky equipment to separate the two phases. Online monitoring of large and complex industrial processes in critical situations has become increasingly necessary to improve their reliability and

3

to minimize the catastrophic failure. There are many examples exist in which a sudden breakdown of a process in a plant can result in having a very high financial loss. Failure to do online monitoring can lead towards a catastrophic situation and eventually this will increase the cost in terms of energy consumption, maintenance, and unscheduled shutdown. In many industries, productivity is the main factor in their revenue. Downtime of a process during an un-expected shutdown for few hours can cause a loss in productivity by several orders of magnitude.

Microprocessor-based embedded systems have been developed to predict any kind of abnormal behavior in the ongoing process so that the preventive maintenance can be prearranged in a cost-effective manner well before the time to avoid the cost of an unwanted and unscheduled breakdown. For the case of gas-solid two-phase flow measurement, instrumentation and measurement systems are currently being developed. These systems are continuously monitoring the flow parameters like velocity, flow rate, and concentration through measurement of physical and environmental parameters, such as electrostatic charge, differential pressure, temperature etc. Having such a system in place, any kind of variation in flow parameters can be identified and corrected either by changing the operating strategy or through planned maintenance and repair work. An on-line continuous measurement system and the use of the measured information for the identification of flow parameters can also be a financially attractive proposition if it is to gain the attention of the machine designers, plant managers, and engineers.

1.4 Challenges in Mass Flow Measurement of Pneumatically Conveyed Solids

(a) Non-uniform distribution of solids inside pneumatic pipeline

The distribution of the solids in pneumatic pipeline can be highly inhomogeneous and non-uniform depending upon the orientation of the pipe, solids size/shape, air velocity and so on [3]. In an ideal scenario, the distribution of the solids will be uniform if the friction between solids and pipe wall is assumed to be zero and solids are moving at higher air velocity (> 20 m/s) as shown in Figure 1-2 (a). The most common type of the solids flow is the stratified, in which the solids in large size travel along the bottom of the pipe and face the highest friction as shown in Figure 1-2 (b). However, the smaller solids which remain fully suspended face less friction and hence move with a higher velocity. The movement of the solids, that are transferred from horizontal to vertical pipe

4

section, changes after hitting with the wall of the bend. The higher concentration of solids move over a specific portion of the cross-section of the pipe in the tilted angles that is called roping flow as shown in Figure 1-2 (c)
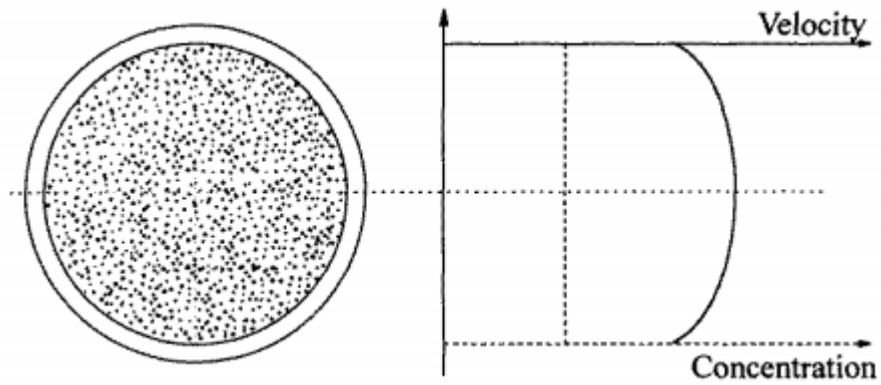
(a) Uniform flow
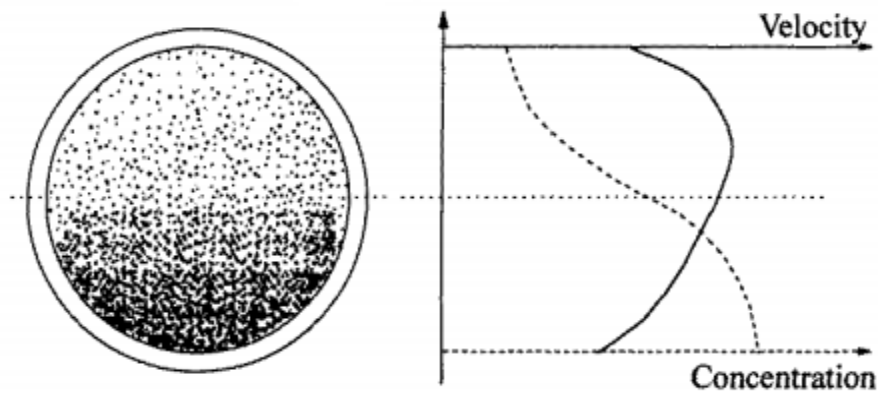
(b) Stratified flow
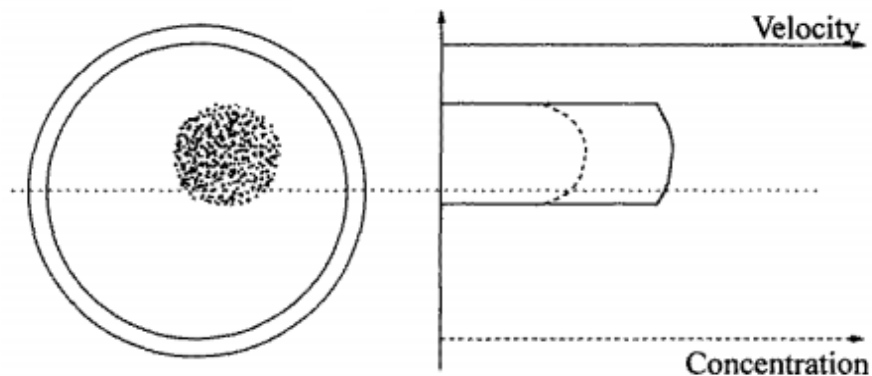
(c) Roping flow

Figure 1-2: Typical solids distribution and velocity profile in horizontal pipe section [3]

(b) Dilute nature of gas-solid flow

Air velocity is used in dilute-phase flow to fully suspend the solids in the conveying system. Almost any powdered or granular substance, regardless of particle size or shape, may be transported using this approach [4]. In a dilute-phase flow, the particle concentration is less than 0.1% by volume [5]. Due to the low solids concentration, the measurement becomes difficult. The air velocity must be in the range of 12 m/s for fine pulverized material and about 16 m/s for fine granular material in order to suspend the solids in the conveying system. The lower air velocity can result in solids being settled at the bottom of the pipe and hence causing blockage in the pipe.

(c) Particle size distribution

The size of the solids varied from few micrometers, in case of pulverized coal, to few centimeters in case of biomass [3]. Distribution of size of the solids also varies within one type of solids. Compared to biomass, pulverized coal has a more consistent size distribution. The reason for this is that, unlike coal, which is mostly composed of carbon, biomass is made up of fibers that are difficult to pulverize. The irregular size of the solids results in the unwanted variations in the flow regime of the solids and hence causes measurement errors [6].

(d) Irregular shape of the solids

The common terms that are used to define the particle shape and size are fine, powder, cubes, coarse, pellets, chips, flakes and granular. It is assumed that the shape of the solids for a given handling system remains fixed. However, a slight variations in particle shape often occur. The shape of the solids can be less a problem in the industrial application where only one type of solids are pneumatically conveyed such as in food processing. However, shape of the solids can significantly impact the flow pattern in a situation where two different types of solids are mixed together. In case of solids mixture made of coal and biomass particles, the shape of coal particles is often more consistent. Biomass particles, on the other hand, are longer due to their fibrous structure. This makes it difficult to measure the individual concentration of two types of solids because the particle spin is more evident on biomass particles than on coal particles [6].

(e) Three-phase flow

Biomass co-firing with pulverized coal is an effective way of reducing CO_2 and other pollutant emissions and is being adopted by many power plants across the world. In order to monitor the fuel conveying process and achieve smooth and balanced fuel distribution, the volumetric concentrations of coal and biomass particles in the fuel pipelines should be measured concurrently on an online continuous basis. Earlier research has been focused on the measurement of pulverized coal or biomass individually [7]. The measurement of coal/biomass ratios in a pneumatic conveying pipeline, that can be further used to determine the individual concentrations of coal and biomass, is still unknown. The development of an instrumentation system for this application exhibits a considerable challenge for scientists and engineers because of the inherently complex nature of coal/biomass/air three-phase flow in a pipeline and the variability of operating conditions of the power plant due to fluctuations in load demand.

1.5 Techniques for Mass Flow Measurement of Pneumatically Conveyed Solids

Based on the principle of measurement, all the techniques proposed in the literature can be divided into direct and indirect categories. A range of measurement techniques based on a variety of sensing principles, such as electrostatic [8], optical [9], acoustic [10], ultrasonic [11], capacitive [12] sensors and nuclear magnetic resonance [13] have been developed. All these types of sensor have the advantage of being non-intrusive and capable of measuring the mass flow rate of solids under certain process conditions. Amongst these techniques, the electrostatic sensors coupled with correlation signal processing algorithms provide a promising practical solution to the measurement of particle velocity due to their advantages over other sensing techniques, such as robustness in a hostile environment, non-intrusiveness in operation, and inexpensive capital cost and low maintenance requirements [8, 14]. In addition, signals from electrostatic sensors are sensitive to moving solids with little influence of the physical properties of solids being accumulated in the pipe that adversely affect other sensing techniques [1]. The conventional method for mass flow rate measurement of solids through electrostatic sensing is inferential, i.e. the mass flow rate of solids is derived from the measured particle velocity and concentration of solids, while the latter is measured through root mean square (RMS) level of the electrostatic sensor output [15, 16]. This inferential method of measuring the mass flow rate of solids is widely deployed [16, 17]. However, despite the advantages of electrostatic sensors for particle velocity measurement, the main problem

in applying such a sensing technique is to relate the solids concentration to the RMS level of the sensor output, which depends on various physical factors, including particle velocity, ambient conditions, pipe orientations and solids size/shape etc [8]. It is worth noting that different particle velocities affect the measurement of solids concentration and hence the mass flow rate of solids in the inferential method [18].

1.6 Data-Driven Modelling Techniques for Mass Flow Measurement of Solids

In order to compensate for the effect of different physical factors on mass flow rate measurement of solids, data-driven models can be trained with different physical conditions such as air velocity, pipe orientations and ambient conditions. The data-driven models can also be applied for the measurement of individual concentrations of two different types of solids with different physical properties such as in coal/biomass/air flow. Integration of data-driven models with rapidly increasing complexity of the sensor data from the two-phase gas-solid flow is becoming essential to analyze the data with higher accuracy. In this case, data-driven models are deployed to establish a relationship that fuses the outputs from a variety of sensors that contain the information about the flow parameters. The iterative nature of the data-driven modelling techniques is important because they are able to adapt independently as the models are exposed to new and unseen conditions of two-phase flow. To generate accurate and repeatable measurements, they learn from previous computations. Although several data-driven models have been around for a long time, a recent breakthrough is the ability to apply complex mathematical equations automatically to a big data. The main advantage of incorporating machine learning models in the measurement system is that the models can be trained off-line and in-situ to provide the mass flow rate measurements of pneumatically conveyed solids under a variety of conditions. Furthermore, the generalization ability of the models can enable the measurement system to predict the mass flow rate under unseen test conditions that lies outside the training data range.

1.7 Aim and Objectives

The research programme aims to develop an integrated system that can do the online continuous measurement of mass flow rate of solids through multi-modal sensing and data-driven modelling. The developed system will be capable of measuring the mass flow rate of solids under different

particle velocity, pipe orientation and ambient conditions. The whole system will be made robust and inexpensive to install to make it worth implementing under full plant conditions. It will be constructed in such a way that it is scalable to any level of the industrial problem without making any significant changes in the core hardware and software algorithm of the system. The primary objectives of this research are:

Review of state-of-the-art measurement techniques – A literature review will be carried out on the current state of the art techniques for flow measurement of pneumatically conveyed bulk solids. This review will be focused on direct and indirect methods of measurements. Direct methods will be based on model-free techniques and a variety of sensor paradigms and indirect methods will be focused on data-driven models along with different sensor paradigms.

Reconfiguration of sensing unit – A sensing unit comprised of ring and arc-shaped electrostatic sensors is already designed and developed. However, the design of existing sensing unit will be modified by integrating a DP transducer, humidity, temperature sensor and accelerometer to obtain the pressure, humidity and temperature and the pipe angle values for the experiments under different physical conditions.

Development of a control system – Testing facilities at the University of Kent provide a fully operational negative pressure bulk solid pneumatic conveying system which is designed for on-line experiments. A negative pressure test rig is comprised of a suction pump, vibratory feeder, and pneumatic pipeline, so the variable frequency drive based inverters will be researched and interfaced to control the suction power and the vibrations of the feeder via computer. A computer programme will be designed to automatically change the power of the suction pump and the magnitude of vibrations for dataset collection in equal intervals under range of air velocities and mass flow rates of solids.

Determination of useful sensors and parameters– It is not straightforward to determine the number of sensors that provides the useful information. Therefore, performance of the individual and different combinations of electrostatic sensors will also be assessed to optimize the number of sensors for mass flow rate measurement. One of the key issue that will also be addressed in this research is the determination of the parameters of the direct method of mass flow rate measurement through firefly optimisation algorithm under different flow conditions.

Design and establishment of data-driven models – As the primary purpose to apply data-driven models is to measure the mass flow rate of solids under different physical factors such as particle velocity, pipe orientation and ambient conditions, therefore a dataset will be collected under a variety of mass flow rates of solids. In order to determine the generalization ability of the model, test conditions will be kept different from the training conditions. Models will also be tested with the conditions which lie partially outside the range of training conditions. A method to shortlist the number of sensors and the number of statistical features from each sensor will be implemented for the effective development of models.

Experimental evaluation of proposed methodology under a variety of conditions – In order to evaluate the effectiveness of the proposed technique, multiple experiments will be conducted to measure the mass flow rates of solids under different particle velocities, pipe orientations and ambient conditions. The performance of the proposed technique will also be evaluated on a complex mixture of different particle types such as coal/biomass/air.

1.8 Thesis Outline

The contributions of this thesis to the state-of-the-art in gas-solid two-phase flow metering include (1) reconfiguration of a sensing system that incorporates variety of sensors for the purpose of dataset collection under different physical conditions (2) investigations into the performance of inferential method of mass flow rate measurement of solids under different physical conditions including air velocity, pipe orientations and ambient conditions (3) Development of data-driven models including Artificial Neural Network (ANN), Support Vector Machine (SVM) and Convolutional Neural Network (CNN), for the compensation of impact of different physical factors on the measurement of mass flow rate of solids (4) Identification of the important sensors and shortlisting of statistical features for the effective development of models.

This thesis is organized in seven chapters as follows:

Chapter 1: Technical Requirements for Mass Flow Measurement Through Data-Driven Modelling

This chapter will introduce the pneumatic conveying system for the transportation of solid. The significance of the online continuous measurement system will be explained. An overview about the challenges in gas-solid flow measurement will be described. Lastly, the gap between this research and the previously done work will be explained followed by the approach to cover the gap.

Chapter 2: Review of Mass Flow Measurement Techniques

This chapter will be focussed on the literature of the techniques to measure the mass flow rate, concentration and velocity of solids. The literature will be divided into direct and indirect methods of measurement. The literature on direct method will cover the techniques based on different sensing systems, whereas the literature on indirect methods will cover the measurement techniques based on data-driven models.

Chapter 3: Test Rig Reconfiguration and Test Conditions

This chapter will introduce the test rig and the test conditions used to conduct the experiments for the dataset collection. The elements of the test rig and their operating principles will be explained. The design of the sensing unit including all the sensors required for different experimental conditions will be discussed. The method to acquire the sensor data and selection of parameters for the control system will also be demonstrated.

Chapter 4: Measurement of Velocity and Mass Flow Rate of Solids with Conventional Techniques

This chapter will be focussed on measurement of particle velocity through cross-correlation based signal processing technique and the mass flow rate of solids through inferential method. Experimental data with different physical conditions, including air velocities, pipe orientations and ambient conditions, will be presented. Selection of useful sensors for the measurement of particle velocity and mass flow rate of solids will be demonstrated.

Chapter 5: Mass Flow Rate Measurement with Data-Driven Modeling under Different Air Velocities

This chapter will present the technique of mass flow rate measurement of solids based on the data-driven models. Pre-processing of sensor data to train the data-driven models will be discussed in

detail. Experimental data and measurement results with different air velocities will be demonstrated.

Chapter 6: Mass Flow Rate Measurement with Data-Driven Modeling under Different Pipe Orientations, Ambient Conditions and Particle Types

This chapter will elaborate the proposed methods and their results for mass flow rate measurement under different pipe orientations and ambient conditions through data-driven modelling. The results of the experiments, carried out with the mixture of coal/biomass/air for the measurement of individual ratio of each type of solids, will also be presented in this chapter.

Chapter 7: Conclusion and Recommendations for Future Work

This chapter will present the conclusions made for the mass flow rate measurement of solids through direct and indirect methods under different air velocities, pipe orientations and ambient conditions. The recommendations for the work, that will be carried out in future, will also be discussed in this chapter.

Chapter 2 Review of Gas-Solid Mass Flow Measurement Techniques

2.1 Introduction

Substantial work has been carried out in the domain of gas-solid two-phase flow measurement [1]. Gas-solid two-phase flows, particularly those solids in the form of powder or grain transported through gases (typically air) in a pipeline, are important in a pneumatic conveying system which has widespread applications across many industries including chemical, food processing, cement, mining, petrochemical, pharmaceutical, semiconductor, and power industry. This variety of applications has led to conducting extensive research on gas-solid two-phase flow systems. Gas-solid flows, such as those in pneumatic conveying systems, are usually classified as dilute, intermediate and dense-phase flow. The dilute-phase flow (which is the case in this study) can be normally achieved with low concentrations of solids (typically below 0.1% by volume) and the simultaneous condition of high gas velocities in the gas-solid flow mixture [5]. Different techniques have been developed to measure the mass flow rate of the pneumatically conveyed solids in the dilute phase. However, there still exists some limitations to meet all the requirements to develop an reliable measurement system. In order to define the state-of-the-art gas-solid flow measurement, fully understanding of the basic measurement strategies and the development of the existing techniques is useful.

This chapter reviews some recent techniques proposed and developed for the mass flow measurement of pneumatically conveyed solids over the past few years. The review is divided into the categories of direct and indirect techniques. Firstly, the chapter begins with direct techniques that are focused on the model-free methods of mass flow measurement. The model-free methods are derived directly from the output from variety of sensors. Secondly, the overview of data-driven modelling techniques and the measurement techniques based on them are explained under the category of indirect techniques. Lastly, the chapter provides the reasoning of the techniques used in this research and highlights the novelties of the research.

13

2.2 Two-phase Flow Measurement Through Direct Methods

Measurement of mass flow of gas-solid two-phase through variety of sensor paradigms, including electrostatic, capacitive sensor, differential pressure transducer, electrical resistance tomography, electrical capacitance tomography, and microwave probes, is referred to direct measurement method because the system, in that case, is designed to measure the desired two-phase flow parameters directly from the sensor signals. In this method, the output of a specific measurement strategy is directly based on the post-processed signals of the sensors.

2.2.1 Electrostatic Sensor

The electrostatic sensor can play a vital role in identifying the flow parameters of solid-gas flow because the solids that are moving with the air in the pipe will gain some value of charge that will be dependent on the particle size, velocity, and air flow pressure. The magnitude of charge is not precisely quantifiable with the charge units because it is unknown that how the charge has been generated. However, the magnitude of charge is high enough to develop an input-output relationship between the intensity of charge and predicted measurement, respectively. A number of techniques have been developed to determine the velocity [19], mass flow rate [20] and concentration [20] of solid particles. The electrostatic sensor is comprised of an electrode that takes the charge value from the charged solids and converts it into a filtered and amplified voltage signal using a signal conditioning circuit. Based on the design of electrode, electrostatic sensors can be categorized in intrusive and non-intrusive type of sensors.

2.2.1.1 Intrusive Type of Electrostatic Sensors

Intrusive type of electrostatic sensors are the ones which are placed inside the pipe and come in contact with particles flow. Owing to the reason that particles can directly hit with electrode, these types of sensors provide more accurate details about localized flow parameters in the pipe.

2.2.1.1.1 Mesh-shaped Electrode

Zhang and Yan have developed a system for on-line continuous measurement of particle size using mesh type electrostatic sensors that appearances like a grid [21]. It is quite obvious that the solids with larger size will contain a high magnitude of charge value, therefore to acquire the accurate charge values of all the solids inside the pipe, six electrodes have been used and placed

perpendicular to each other to formulate a grid to cover the major portion of the pipe as shown in Figure 2-1.
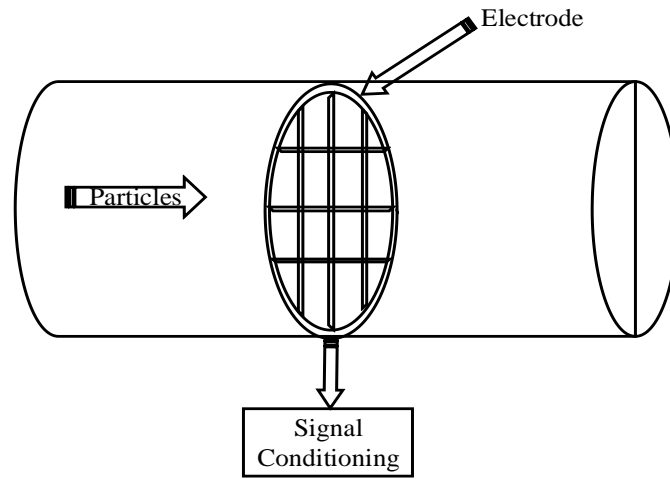
Figure 2-1: Schematic of a mesh type electrostatic sensor [21]

The main advantage of using this mesh type electrodes is that they can cover the whole cross-section of pipe and provide more precise information about the magnitude of charge of the solids in different areas of pipe. On the contrary, it can be seen that electrodes of this mesh can be a source of the hindrance for bigger solids which can potentially disturb the actual velocity of the solids. The measurement of velocity has a huge part to play in the mass flow rate measurement of solids under different air velocities, therefore this mesh type electrostatic sensor cannot be used.

2.2.1.1.2 Rod-shaped Electrode

Another type of electrostatic sensor that can be used to overcome the issue faced in the mesh-type electrostatic sensor is the rod type electrostatic sensor which is shown in Figure 2-2. Design parameters should be considered before installing it inside the pipe. For example, the cross-section of the electrodes should be designed and optimized in such a way that it has least impact on the flow of solids inside pipe.
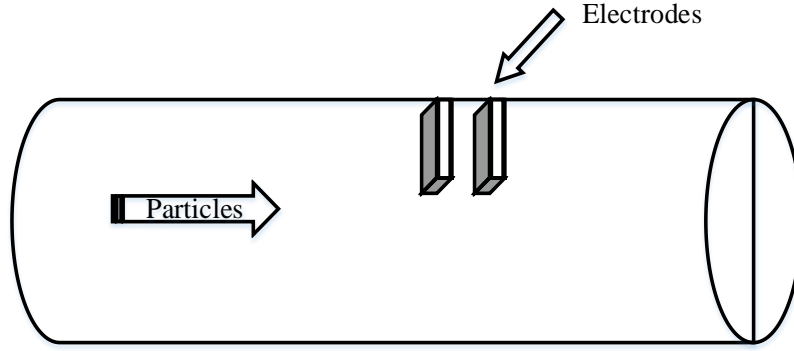
Figure 2-2: Schematic of the rod type electrostatic sensors [22]

Shao in [22] has investigated the optimal design of these electrodes after performing a combination of practical experimentation and offline finite element modeling. It has been discovered from this study that electrodes should cover 30% to 50% of the pipe diameter in order to achieve optimal accuracy value. This proposed study has reduced the amount of error in velocity caused by electrodes acting as a hindrance but does not provide the better accuracy. Likewise, it is evident from Figure 2-2 that still there is a portion inside the pipe that has not been covered by the electrodes, causing the misleading measurements due to particles near the bottom of the pipe.

2.2.1.1.3 Intrusive Sensor Array

A model based on electrostatic sensors has been used to measure the flow parameters of pulverized coal and biomass particles in a two-phase solid-gas material [23]. The intention is to find the velocity and the concentration of the particles which are contained in a pipe and carried by the air. The methodology adopted to measure the velocity of the particles is based on the electrostatic sensor-based system. The particle velocity is determined by applying cross-correlation based signal processing technique on the signals of two consecutively placed electrostatic sensors. The concentration level of the solids has been measured by finding the RMS value of the electrostatic charge. Results of the velocity measurements under different pipe diameters, different air velocities and with two different sensor array orientations have been presented. Two electrodes (downstream and upstream) will be required to measure the particle velocity as shown in Figure 2-3.
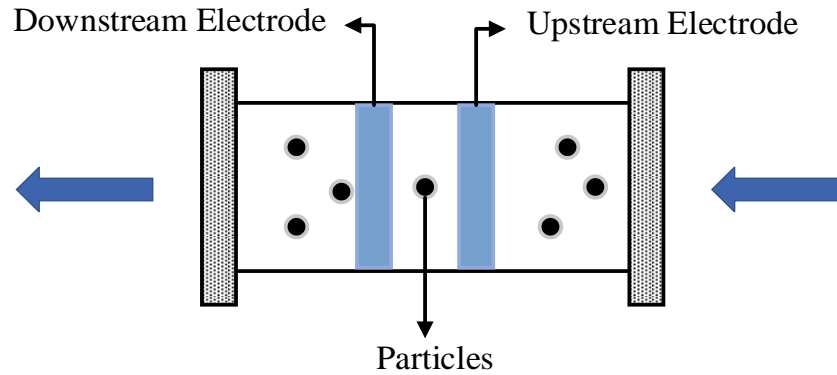
Figure 2-3: Upstream and the downstream electrodes configuration [23]

The velocity can be computed by taking the ratio of the spacing between two electrodes and the time difference between upstream and downstream electrodes. Correlation between two signals obtained from upstream and downstream electrodes have been computed to find the time difference between these two electrodes. In the plot showing the cross-correlation function response, the time from zero to the peak value of the correlation has been recorded as the time difference between two electrodes. In order to find the particle concentration, a measure of root mean square (RMS) has been applied to the charge values of the electrode. The actual electrostatic sensor design of this experiment is based on five independent electrodes in upstream and five in downstream as shown in Figure 2-4.
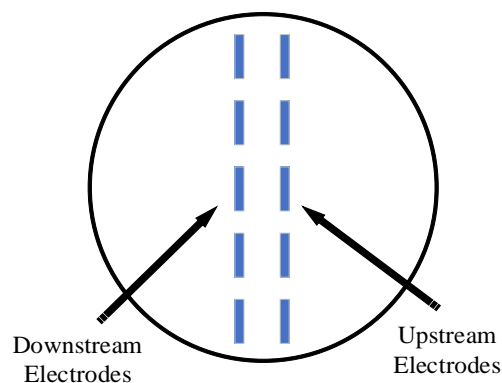
Figure 2-4: Design of an intrusive electrostatic sensor array [23]

2.2.1.2 Non-intrusive Type of Electrostatic Sensors

Unlike intrusive type of electrostatic sensors, non-intrusive type of sensors does not make any contact with particles of two-phase flow and can be flush-mounted inside the walls of pneumatic pipeline. This type of sensor has an advantage over intrusive type of sensor that it does not disturb the ongoing flow in pipeline.

2.2.1.2.1 Ring-shaped Electrode

A technique based on ring-shaped electrostatic sensors has been presented in [16] to measure the particle velocity and mass flow rate of solids. The particle velocity is measured using cross-correlation based signal processing technique, however the mass flow rate is derived from the particle velocity and the RMS charge value of electrostatic sensor as shown in Figure 2-5. The whole instrumentation system including multiple electrostatic sensing heads has been developed and installed on 510-mm bore primary air pipes on the same mill of a 600-MW coal-fired boiler unit for the measurement of pulverized fuel (PF) mass flow and velocity distributions. An array of four electrodes with different axial widths is housed in a sensing head. The axial width of one electrode is slightly greater than the other electrodes to measure the root mean square value of the charged particles. The rest of the three electrodes are used to derive the electrostatic signals for the measurement of PF mass flow rate and velocity. The PF velocity is determined by multiple cross-correlation coefficients of the electrostatic signals from the three narrow electrodes. The measured PF velocity measured from three narrow electrodes was combined with the root-mean-square magnitude of the measured electrostatic signal from the wide electrode for the calibration of the PF mass flow rate. On-plant comparison trials of the developed system were conducted under five typical operating conditions after a system calibration test. Isokinetic sampling equipment is used to obtain reference data to evaluate the performance of the developed system. Experimental data and results displayed in this research show that the developed system is effective and reliable for the online continuous measurement of the mass flow and velocity distributions between the primary air pipes of the same mill.
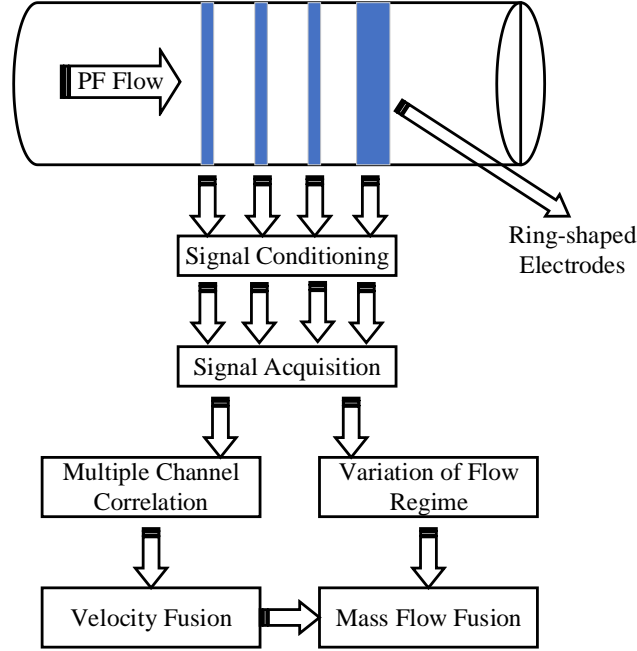
Figure 2-5: Measurement strategy with ring-shaped electrodes [16]

2.2.1.2.2 Arc-shaped Electrode

In view of the fact that the signal obtained by ring-shaped electrodes is the average electrostatic charge of the whole cross-section. Whereas, the distribution of particle velocity and solids concentration in the overall cross-section can be more different in the gas-solid flows. The arc-shaped electrodes have been used in many researches to quantify the variations between the various areas of the pipe. In order to achieve a more detailed view of the induced electrostatic signals in the fluidized beds, Shi et al. [24] has designed and installed an array of arc-shaped induced electrostatic sensors to the outer wall of the fluidized bed. In conjunction with the cross-correlation signal processing technique, the induced electrostatic voltage signals and the correlation velocity of the particles have been determined simultaneously.

The minimum air velocity or the minimum pressure drop in pneumatic pipeline can easily characterize the solids moving from deposition to suspension but it becomes difficult when the mass flow rate is very low. Wang et al. [25] has presented a technique to determine the characterization of the solids moving from deposition to suspension using four arc-shaped electrostatic sensors as shown in Figure 2-6. The combination of Empirical Mode Decomposition

19

(EMD) and Hurst exponent has been used to extract the accurate scale signals that describe the random motion of the particle centered on a four arc-shaped electrostatic sensor with spatial sensitivity. Differences in the particle random motion energy ratio of 4-channel electrode signals has been used to describe the flow-state transformation. The particle random motion energy is proportional to the particle velocity and the distance between the charged particle and electrode. In the gas-solid two-phase flow system trial, it has been found out that the particle random motion energy ratios of 4-channel electrode signals are all increasing with an increase in superficial fluid velocity, and the particle random motion energy ratios of specific electrodes are very different at low superficial gas velocity.
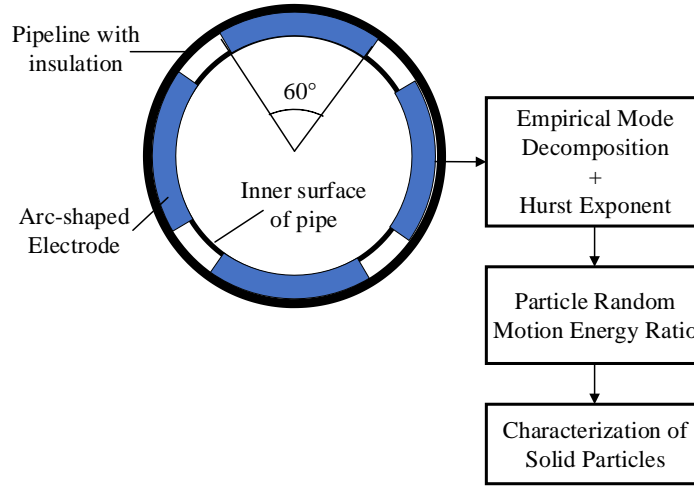
Figure 2-6: Measurement strategy with arc-shaped electrostatic sensors [25]

2.2.1.2.3 Strip-shaped Electrode

Strip-shaped electrostatic sensors, placed in square-shaped pneumatic conveyor pipes, are seen in some manufacturing processes, such as fuel injection systems in coal-fired power plants and moving fluidized industrial processes. The particle dynamic characteristics across the four sharp corners are very complex. Zhang et al. [26] has proposed a novel non-intrusive strip-shaped electrostatic sensor arrays for the measurement of velocity and characterization of pulverized fuel in the square shaped pneumatic pipeline as shown in Figure 2-7. The sensor arrays consist of 12 pairs of strip-shaped electrodes that are uniformly inserted in the four flat pipe surfaces. Experiments have been performed on a laboratory scale test rig under a variety of conditions. This

proposed work describes fuel velocity and flow stability profiles around the entire cross-section of the square pipeline.
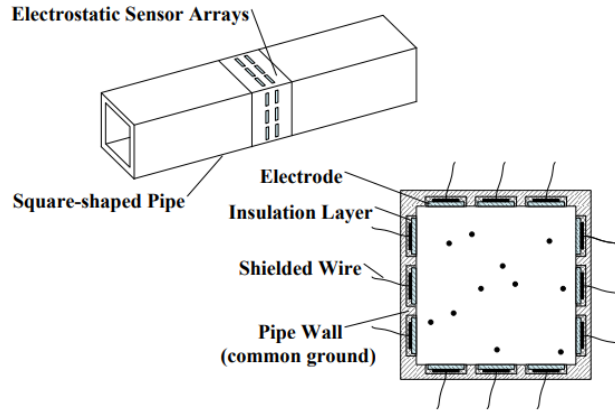
Figure 2-7: Schematic of the sensing head containing the strip-shaped electrodes [26]

The spatial sensitivity of the electrostatic sensor is known as a significant factor influencing the precision of the flow measurement of solids particles in the pneumatic conveying pipeline. However, the spatial sensitivity distribution is usually inhomogeneous owing to the physical configuration of the electrostatic sensor and the intrinsic electrostatic sensing process. Zhang et al. [27] has presented an approach for sensitivity homogenization which is focused on differential measurements by utilizing differential outputs from two electrodes of specific axial widths. The validity of the sensitivity homogenization approach for the square-shaped electrostatic sensing head, consisting of strip-shaped electrode arrays with different widths, is confirmed by statistical study. In addition, the response of the electrostatic sensing head integrating the sensitivity homogenization method to the roping flow regime has been evaluated on the gravity-fed solids flow test rig. The findings of both simulation and experimental experiments show that the homogeneity of spatial response is greatly increased.

Zhang et al. [28] has proposed a technique to improve the spatial sensitivity of electrostatic sensors for gas solid flow measurement. The technique is based on power frequency spectrum of the output of sensor signal. Power frequency distribution has been found by repeatedly decomposing the total output power distribution of the sensor to classify the elementary frequency spectra corresponding

to the relative flow rate. The decomposed elements have then been re-weighted to increase the uniformity of spatial sensitivity and also to ease the flow regime dependent problems.

Wang et al. has identified that the inner flush mounted electrostatic sensor produces a signal which includes the induced charge signal and the transmitted charge signal [29]. According to the induced charge signal band characteristics, an adaptive decomposition technique based on the discrete wavelet transform has been proposed to derive induced charge signal from the measured signals of the sensors. In the calculation of the cross-correlation velocity of the solids, the mean cross-correlation coefficient between the detected induced charge signal of the upstream and downstream electrodes has found to be 0.74, improved from the previous coefficient by 0.48 for the total calculated signals obtained from the electrodes. This large improvement has indicated that the induced charge signal has been suitably adapted to calculation with higher precision. Meanwhile, the mean relative standard deviation (RSD) of the solid velocity has been observed to be decreased from 3.35% to 2.69% after decomposition, resulting in increased stability. In addition, the RMS of the derived transmitted charge signal has indicated a pattern of linear increase as the solid mass flow rate and superficial air velocity grow, suggesting that transmitted charge signal may also function as an efficient tool for predicting the solid mass flow rate as an alternative to induced charge signal.

2.2.2 Capacitive Sensor

Capacitive sensors, for the measurement of flow parameters in a pneumatic bulk solid conveying system, introduce the idea of allowing dielectric swings as particles migrate through the sensor plates (active sensor plate and ground plate) as shown in Figure 2-8. A signal conditioning circuit, such as an AC deflection bridge, can be used to transform the capacitance to a functional output voltage that can be digitized by an analog to a digital converter to make the signal being able to be processed by the computer. Capacitive sensors have been used to measure both the particle size and the particle velocity, enabling them to assess the mass flow rate of pneumatically conveyed bulk solids [30, 12, 31].
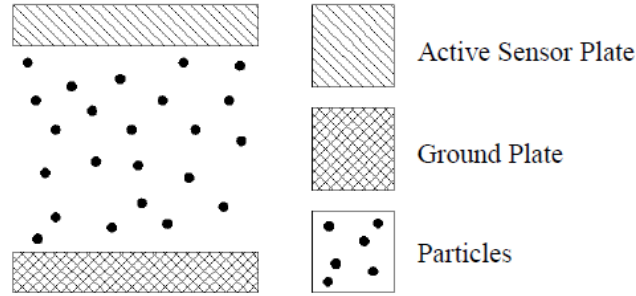
Figure 2-8: Schematic of a capacitive sensor plate configuration [2]

Heming et al. has proposed a novel method of velocity measurement based on a differential capacitance sensor array (DCSA) and spatial filter velocity measurement principle for solids in a gas-solid mixture [30]. First, the DCSA has been designed by the ANSOFT finite-element program and its response characteristics have been learned. In addition, the spatial filtering influence of DCSA has been explored technically, which has validated the viability of DCSA-based particle velocity measurement. The velocity of the local particles in the pipeline have been determined by calculating the frequency bandwidth (cut-off range) of the DCSA output capacitance signal. Figure 2-9 shows the schematic of proposed sensing system which is consisted of electrode array, insulated pipeline and metal shield.
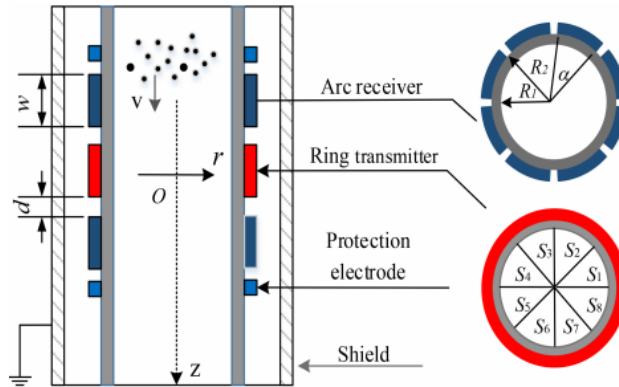
Figure 2-9: Sensor plate configuration of a capacitive sensor [30]

The electrode configuration comprises of 16 arc-shaped receiver electrodes, one ring-shaped transmitter electrode and two axial safety electrodes. The metal mask protects the electrode layer in order to prevent electromagnetic intrusion. The axial security electrode restricts the electrical field line divergence along the pipeline and is used to achieve a well-defined axial sensitive area. The receiver electrodes are symmetrically arranged on both sides of the ring-shaped transmitter electrode, creating two-layer arrays in the axial path of the device. The results indicate that the method has the feasibility for the velocity measurement and its repeatability is within 11% over the velocity range of 1–3 m/s.

2.2.2.1 Electrical Capacitance Tomography

In some applications, 3D image analysis has been proved very helpful for gaining a multi-directional perspective of pictures, e.g. exploring the inside of a bulk of solids in a circulating fluidized bed (CFB) by presenting multi-layer concentration and revealing the distribution of clustered solids in the conical portion of a cyclone separator. Mao et al. has investigated the effect of various excitation techniques on the output of a multi-plane electrical capacitance tomography sensor used to test gas-solid flow in a circulating fluidized bed [32] as shown in Figure 2-10. Three excitation methods, including single-electrode excitation, dual-electrode excitation on the same plane, and dual-electrode excitation on separate planes, have been well-thought-out for assessment. Sands have been used as solid material. The velocity of the airflow has been varied from 0.6 to 3.0 m/s with a step of 0.2 m/s. Capacitance data under various operational conditions and specific excitation methods has been obtained and used to recreate the permittivity distribution using a linear back projection algorithm.
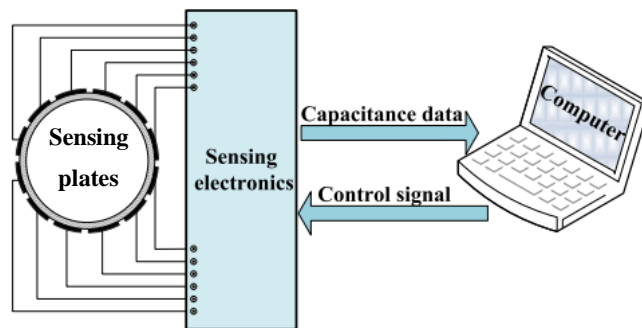
Figure 2-10: Schematic of an ECT system [32]

2.2.3 Optical Sensors

Optical sensors have been widely used to measure the particle velocity mostly with cross-correlation technique. Mahmod et al. has proposed a technique which focusses on the design and construction of an optical sensing module that uses light emitting diode (LED) as a light source and an LED photosensor as a detector [33]. The sensor has been built to determine the particle velocity in a solid-gas flow within a closed container. Various considerations, such as power dissipation, light source wavelength, switching time and expense, have been included in the design phase of this sensor. The cross-corelation based signal processing methodology has been used to assess the flow rate at which small particles have been inserted in a natural gas flow and have passed through two distant sensor module sets. The LED beam source in the first set of the sensor will be dispersed as the particle is crossed and the resulting photodetector will absorb the obtained light signal and produce the pulse signal. The second pulse signal is produced when the particle crosses the second subset of the sensor at a specific time interval. The time period calculated is used to determine the particle velocity. The overall schematic of the measurement system is shown in Figure 2-11. The cross-correlation method cannot provide a time history of gas velocity, because its output is a single-valued estimate of the average interfacial velocity, under the assumption that the temporal fluctuations of gas velocity would be negligible [34].
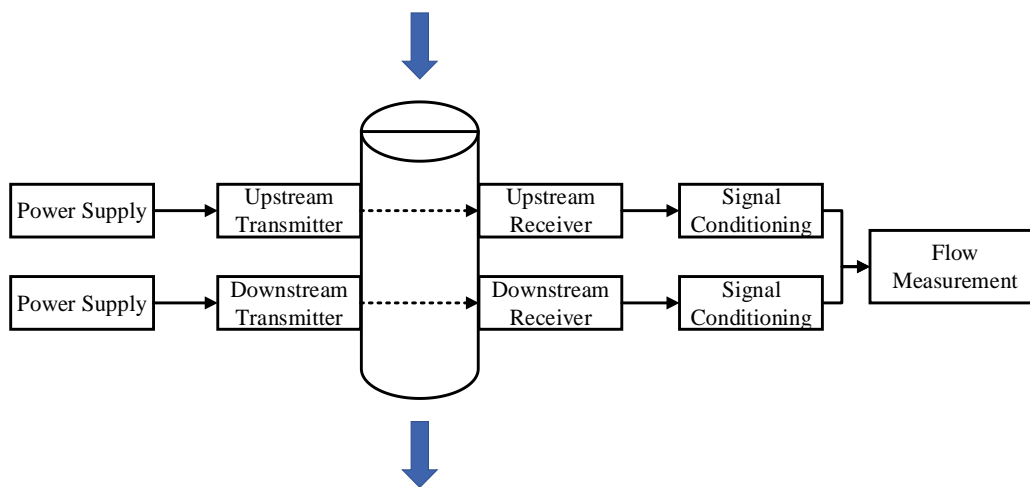
Figure 2-11: Schematic of the measurement system with optical sensors [33]

2.2.3.1 Optical Tomography

The general method of optical tomography includes the usage of a collection of light sources and photo detectors to gain simultaneous views of the tube. In the optical configuration, a light beam is transmitted from one boundary point across a medium, and this light beam is observed at another boundary point. At the receiving point, the photodetector's voltage output is determined and any change of the output is proportional to the position of the source in the pipe, i.e. the optical tomography measures the attenuation of the signal induced by the artifacts to be imaged.

Muji et al. has implemented an optical tomography setup which is capable of implementing fan beam projection in parallel view [35]. The proposed architecture includes a sensor jig primarily developed for concurrent systems that do not require a collimator. The fan beam projections can also be carried out in the same sensor set without difficulties. This approach is a realistic way to resolve the drawbacks of parallel beam projection. While the fan beam has its own drawbacks, the combination of the fan beam method and the parallel beam method is supposed to further improve the picture quality of the optical tomography. The accuracy of the picture has been calculated using the peak signal-to-noise ratio and the normalized mean-square error parameters. The proposed optical tomography device consists of three essential subsystems, namely the sensor array, the signal conditioning unit and the data acquisition unit as shown in Figure 2-12.
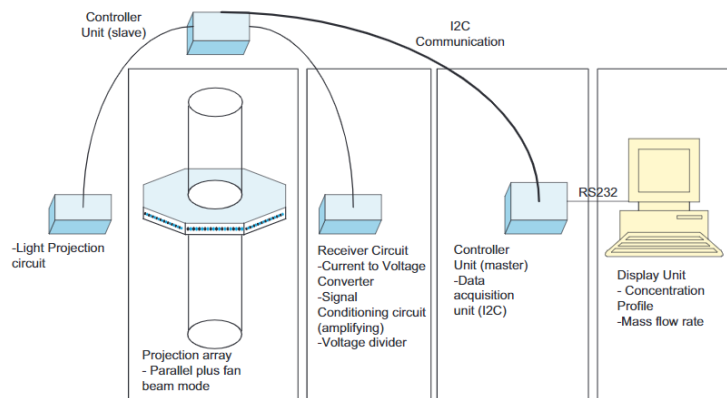
Figure 2-12: Schematic of an optical tomography system [35]

2.2.4 Acoustic Sensor

Passive acoustic emission methodology senses main parameters in the manufacturing cycle by collecting vibration signals and connecting them to the working parameters of the fluidization cycle. The methodology is highly sensitive, effective, environmentally sustainable, non-intrusive and works in real-time. Zhang et al. has presented a new approach for measuring the mass flow rate of pulverised fuel in real time using acoustic emission detection and electrostatic sensing [36]. The development of an integrated sensing head with an AE probe and three electrostatic sensor arrays is undertaken as shown in Figure 2-13. The suggested approach uses multichannel cross correlation of electrostatic signals to measure particle velocity and derives mass flow rate information from the AE signal resulting from particle impacts with a waveguide projecting into the flow as shown in Figure 2-14. The energy of the AE signals, particle velocity, and mass flow rate are all calculated using an analytical model. The findings show that on a vertical pipe, the sensing head should be positioned in any orientation away from the elbow. However on a horizontal pipe, the waveguide should be installed in the horizontal direction. The instrumentation system can measure the mass flow rate of particles in the vertical pipe with a relative error of less than 6.5% regardless of the sensing head's orientation over a mass flow rate of 7 to 25 kg/h and a particle velocity of 12 to 30 m/s. However, the error is less than 5.8% on the horizontal pipe when the sensing head is installed with the waveguide in the horizontal direction under the same flow conditions.
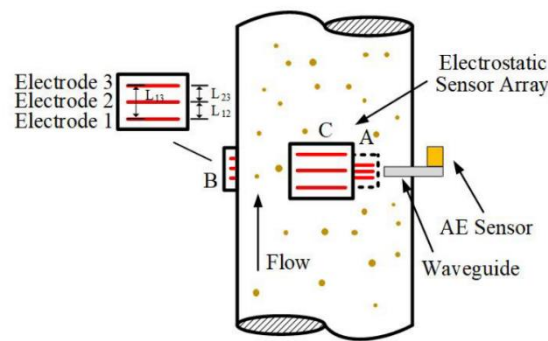
Figure 2-13: Schematic of a sensing head containing electrostatic and acoustic sensors [36]
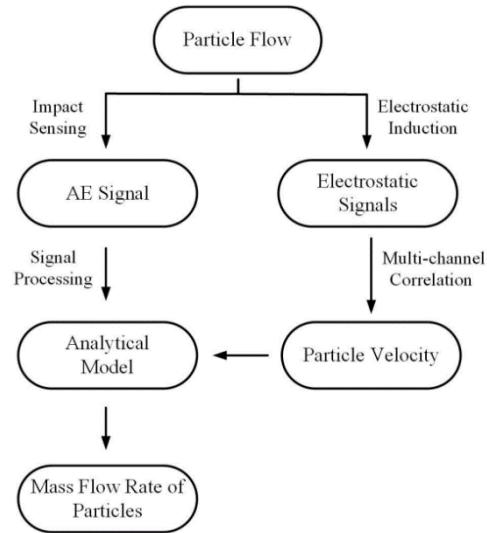
Figure 2-14: Measurement strategy through electrostatic and acoustic sensing [36]

Hu et al. [37] has introduced an experimental instrumentation method for on-line continuous measurement of particle size distribution through acoustic emission detection. The proposed approach derives the information on the scale of the particles from the impulsive AE signals originating from the impacts of the particles with a metallic waveguide protruding through the surge as shown in Figure 2-15. The relationship between the particle size and peak acoustic emission voltage has been defined through mathematical simulation of the particle impact cycle.
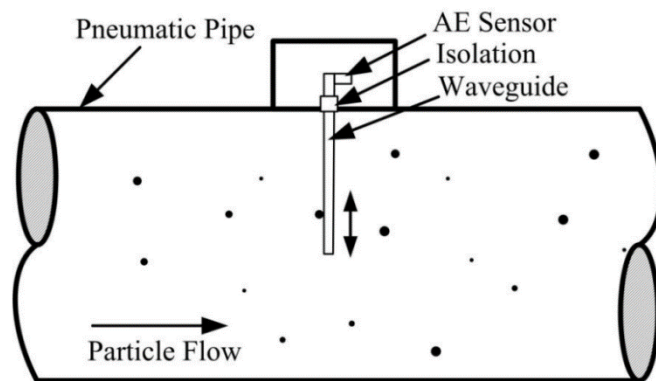
Figure 2-15: Placement of an acoustic sensor in the pipe [37]

A non-intrusive, real-time and environmentally safe acoustic emission technique and an auxiliary pressure fluctuation system has been developed by Zhou et al. [38] to investigate the flow regime transformation process of gas-solid two-phase fluidization. Particle behavior and collision amplitude in the pipeline have been measured by means of a standard deviation analysis. On this basis, it has been observed that acoustic signal measures will more accurately represent the transition velocity between large flow regimes, like from bubbling to turbulent fluidization, from turbulent to rapid fluidization and from fast flow to dense phase pneumatic convection. Multi-scale resolution of acoustic signals throughout regime transitions has been accomplished by Hurst and wavelet analysis.

2.2.5 Critical Evaluation of Sensing Techniques

A number of sensing techniques are presented in the review for the measurement of velocity and mass flow rate of solids. All of them are suitable for the application of this project and have an advantage of achieving good accuracy for the measurements under fixed conditions. However, there are some cases, as listed in Table 2-1, where these techniques may not work very well. In this case, electrostatic sensors have achieved more advantages over other sensing techniques such as low capital and maintenance cost, robustness and fast response time and hence they will mainly be used in this project.

Table 2-1: Comparison between different sensing techniques for the purpose of solids flow measurement

Sensing technique	Advantages	Disadvantages
Electrostatic [8]	• Low capital and maintenance cost in case they are implemented in industrial application • Good accuracy, high resolution and fast response for the application of velocity and mass flow rate measurement • Physically non-intrusive in nature	• Characteristics of the sensor may differ due to different physical conditions such as pipe orientation, ambient conditions, air velocity etc
Capacitive [39]	• Good accuracy and high resolution for the application of	• Highly succesptible to moisture that will affect

	velocity and mass flow rate measurement • Physically non-intrusive in nature	the dielectric properties of the material in a condition where the RH is relatively high
Optical [40]	• Good accuracy and high resolution for the measurement of velocity and mass flow rate measurement • Physically non-intrusive in nature	• Requires a transparent window for the purpose of velocity and mass flow rate measurement
Acoustic [1]	• Good accuracy and high resolution for the measurement of velocity and mass flow rate measurement.	• Succesptible to false signals when used in a harsh and noisy industrial environment • Intrusive approach can effect the flow of particles

2.3 Data-Driven Modelling Techniques

Due to the difficult nature of two-phase flow and complexity of the sensing systems, the applications of direct techniques of two-phase flow measurement have achieved limited success in the industry [41]. Therefore, it is necessary to explore the data-driven modelling based algorithms to compensate the impact of physical conditions that can potentially influence the measurement system. Recently, there are number of data-driven modelling techniques that have gained some attention in the field of two-phase flow measurement.

A comprehensive review of data-driven modelling techniques for multiphase flow metering, with a main focus on the measurement of individual phase flowrates and phase fractions, has been presented in [42]. This particular review explains the sensors which have been used to obtain the physical parameters, the working principle, modeling and the applications of the various data-driven modelling techniques in addition to their advantages and limitations. Correspondingly, the indirect methods incorporating data-driven modelling techniques to measure the individual flow

rate and fraction of different elements of multiphase flow have been discussed. Their detailed study, first of all, explains the data-driven modelling techniques which can be applied possibly in this domain of flow monitoring and then the type of sensors that have been used in the history to collect the data and how that data is driving the models.

2.3.1 Artificial Neural Network

ANN establishes the input-output relationship by inheriting the dataset available to build the model [43]. Each neuron in the ANN's network is taking the input either from the dataset or from any other neuron. There are three types of ANN models that exist including single layer, multi-layer and recurrent. In the single-layer ANN, there is no hidden layer used in the network. In multi-layer ANN, one, two or maybe more layers can be used to run the network. In the recurrent ANN, at least one feedback loop is involved.

ANN can also be categorized based on their learning topologies such as supervised learning and unsupervised learning. These two categories have further been divided into some sub-categories based on their application. Amongst all those learning types of ANN, MLP and RBF are widely used in the previous research works on multi-phase flow metering. Figure 2-16 shows the basic structure of an m-input, single hidden and single output multi-layer perception. With the input vector $x = [x_1 \ x_2 \ x_3 \dots x_m]$, the input-output relationship can be represented as [44]:

$$y = \sum_{j=1}^{L} \omega_j H_j + b = \sum_{j=1}^{L} \omega_j f \left(\sum_{i=1}^{m} \omega_{ij} x_i + a_j \right) + b \tag{2-1}$$

Where m and L are the Number of input variables and the number of nodes in the hidden layer respectively and a_j and b are biases. The ω_j is the weight value that is used to connect the jth neuron with the output layer. Similarly, ω_{ij} is the weight value that connects the ith input variable to jth node of the hidden layer.

31
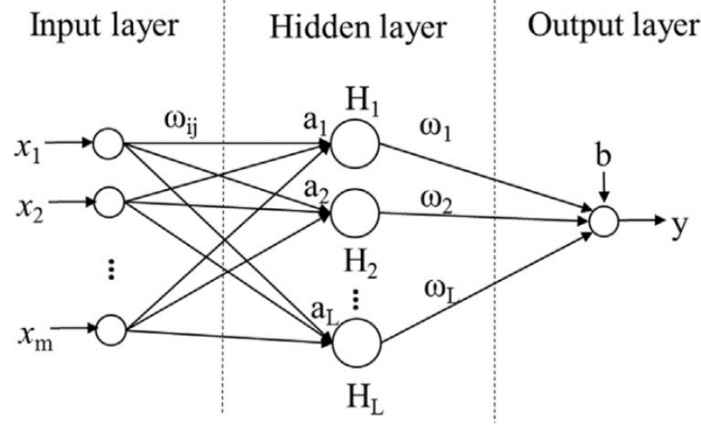
Figure 2-16: Basic structure of a MLP [42]

The number of neurons in the hidden layer must be chosen as per the guideline [45]:

$$L \leq 2m + 1 \qquad (2\text{-}2)$$

$$L \leq \frac{n}{m + 1} \qquad (2\text{-}3)$$

Where m and n are the number of input variables and the number of training samples respectively.

An activation function in the ANN model, performs the complex computation in the hidden layer and then transfers the outcome to the output. Activation functions are mainly used to introduce the non-linearity in the model [46]. Table 2-2 shows the other activation functions that are also considered in this research.

Table 2-2: Activation functions for ANN model

Activation function	Equation
Sigmoid	$\frac{1}{(1 + e^{-x})}$
Hyperbolic tangent	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$
Softmax	$\frac{e^x}{\sum_j e^{x_j}}$
Softsign	$\frac{x}{1 + |x|}$
Rectified linear unit	$\begin{cases} x, & if\ x \geq 0 \\ 0, & if\ x < 0 \end{cases}$

Exponential linear unit	$\begin{cases} x, & if\ x > 0 \\ ae^x - 1, & if\ x \le 0 \end{cases}$

Where x is the input of the activation function which is derived from the statistical features of the sensor data.

2.3.2 Support Vector Machine

Support vector machine can be defined as the system that uses hypothesis space of a linear function in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory. The basic structure of SVM model is shown in Figure 2-17. The SVM models are one of the most popular and widely implemented data driven algorithms which perform linear regression in a high dimensional feature space and tend to reduce model complexity. Initially, the SVM model was developed to solve the classification problem based on the statistic learning theory and structural risk minimization [47]. Later on, this method of classification was extended to the domain of regression and prediction problems [45]. The type of SVM model, implemented in this research, is regression for the measurement of mass flow rate of solids. The output is a real and continuous number, therefore it becomes very challenging to predict the information at hand which has infinite possibilities. In the case of regression, a margin of tolerance is set in approximation to the SVM which depends on the nature of the problem. SVM regression is considered as a non-parametric technique because it relies on kernel functions. The kernel trick is useful to minimize the computational complexity of the input data which is comprised of several statistical features in the original space. Table 2-3 shows the list of kernel functions that are considered for the effective development of SVM model.
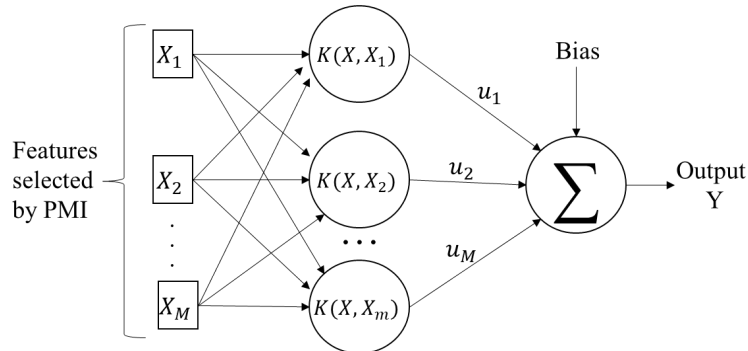
Figure 2-17: Structure of the SVM model.

Table 2-3: Kernel functions for SVM model

Kernel function	Equation
Polynomial	$(a.b + 1)^d$
Gaussian	$e^{-\frac{\|a-b\|^2}{2\sigma^2}}$
Radial basis function	$e^{-\gamma\|a-b\|^2}$
Laplace radial basis function	$e^{-\frac{\|a-b\|}{\sigma}}$
Hyperbolic tangent	$\tanh(ka.b + c)$
Sigmoid	$\tanh(\beta a^T b + c)$

Where a and b are the input statistical features, d is the degree of polynomial, σ is the standard deviation, T is the transpose of matrix, γ, β, k, c are the constants.

2.3.3 Deep Learning Models

Deep learning models have been applied to a wide variety of problems in recent years. In many of these applications, deep learning models have outperformed above all other state-of-art techniques like classical neural networks, SVM and ANFIS. Unlike all of these classical learning techniques, modern deep learning has cracked the code for training stability, generalization and scale on big data [48]. One of the biggest problems in using these classical learning and fuzzy inference techniques is that these techniques require optimization algorithms and a large rule basis to train their classifiers, respectively. Deep learning is a subset of typical machine learning techniques with a large number of hidden hierarchies with great power and flexibility to predict the results more accurately. The advantage of this additionally complex network is that a richer data set can be mapped more accurately from the input vector to the output with a non-linear mapping function. With the increase of computation power and technology of microprocessors, deep learning models have gained much popularity due to its supremacy in terms of accuracy while exposed to a large amount of input data. As per Google's statistics, its popularity trend with time can be seen in Figure 2-18.

Figure 2-18: Growing use of deep learning at Google [49]

As per Andrew Ng, the chief scientist of china's major search engine Baidu and one of the leaders of the Google Brain Project, "The analogy to deep learning is that the rocket engine is the deep learning models and the fuel is the huge amount of data that can be fed into these algorithms". One of the emerging fields of big data has provided a big support to the Deep Learning algorithm to run while taking a big amount of data. One of the big advantages of deep learning model is that its performance in predicting data labels in test cases increases as the amount of input data samples increases during the training session. Unlike all other traditional data-driven models, deep learning models do not make their performance better to some extent and then setting their performance to the saturation state even after increasing the amount of input data samples. This can be well described through Figure 2-19.
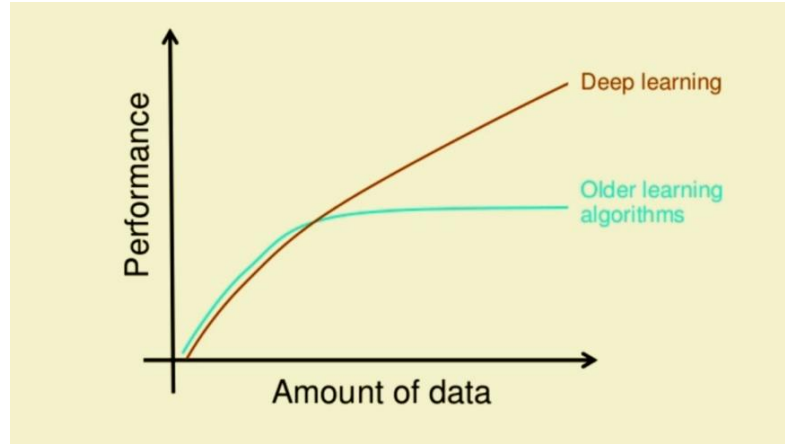
Figure 2-19: Performance comparison of deep learning and older learning algorithms [49]

Another one of the biggest advantages of deep learning model is that it does not require the set of user-defined features for the training purpose, unlike all the other data-driven models for which a user has to determine the suitable features for the development of the model. For example, in the case of identifying a car in an image using traditional data-driven models, it will be pre-defined by the user that what elements of a car a data-driven model has to look for. However, in the case of the deep learning models, it is an in-built functionality of the algorithm to identify the features of an object by detecting the edges and curves in its lower level hidden layer to the final one as shown in Figure 2-20.
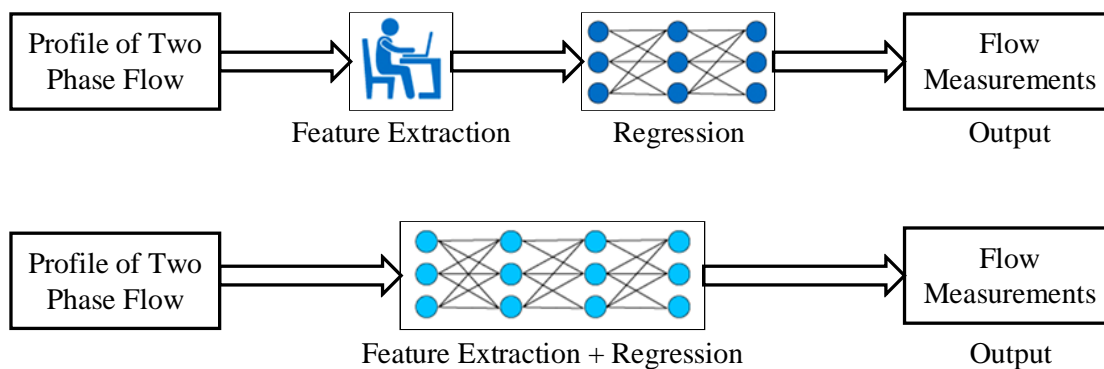
Figure 2-20: Internal hierarchy of classical machine learning and deep learning algorithms

36

Deep learning is a latest technique of data-driven modeling, which has been introduced with the goal of extending machine learning algorithms towards artificial intelligence [50]. The main goal of deep learning models is to increase the abstraction levels in order to better learn the data representations. Almost in all levels, more abstract representations at a higher level are learned by defining them in terms of less abstract representations at lower levels. This type of hierarchical learning process is very powerful as it allows a system to comprehend and learn complex representations directly from the raw data, making it useful in many applications.

There are many deep learning architectures available in the literature such as deep neural network (DNN) [51], recurrent neural network (RNN) [52], deep auto-encoder (DA) [53], deep Boltzmann machine (DBM), deep belief network (DBN). However, the convolutional neural network (CNN) is the one that is widely used in the domain of flow measurement. The CNN model is a multilayer neural network that is inspired by the neurobiology of the visual cortex, which consists of convolutional layers followed by fully connected layers [54]. The convolutional neural network combines three architectural ideas to ensure some degree of shift and distortion invariance. Local receptive fields shared weights and sometimes spatial or temporal subsampling. In between these two types of layers, there may exist subsampling steps. Unlike DNNs which have difficulty in scaling well with multidimensional locally correlated input data, CNN performs better in this situation. Therefore, the main applications of CNN are found in the datasets, where the number of nodes and parameters required to be trained is relatively large similar to image analysis. Exploiting the "stationary" property of an image, convolution filters can train data-driven kernels. Applying such a convolution filter along with a suitable pooling function reduces the features that are supplied to the fully connected network to classify. However, in the case of large datasets, this can be a problem and can be solved using sparsely connected networks. Some of the popular CNN configurations include AlexNet, VGGNet, and GoogLeNet.

Although the deep learning models have been hyped so much nowadays with the rapid growth in the amount of dataset and computational power, the models are not the best solution for every problem. Suitability of these models varies from case to case, depending upon the application and nature of the data obtained from a specific problem domain. One of the possible reasons for deep learning models for not performing better in some cases is that deep learning model is meant to be used for complex domain image data through which the convolutional layer identifies the sharp

edges and curves in terms of data variations by going deep down into the input image data. In that case, the input image data, especially built from sensor data, that need to be fed to the models are not complex enough which could emerge the significance of extracting deep-down features from the data. Practically, deep learning technique is hard and expensive in terms of computation power and time. Therefore, it is an over-killing of computation power and resources if the nature of input data is not complex or if there is an insufficient amount of dataset available to drive the deep learning models.

2.4 Two-Phase Flow Measurement Through Indirect Methods

All the direct and model-free techniques presented in the section 2.1 are simple, computation-inexpensive and easy to implement. However, such direct and model-free measurement techniques, despite their advantages, have gained limited success in industry owing to the difficult nature of the two-phase flow. Indirect methods of flow measurement, which include data driven models, can best accommodate the fusion of sensor data for the instrumentation system comprised of variety of sensors. Recently, data driven models have been deployed to establish the relationship between characteristics of the sensors and the intended measurand to achieve the multiphase flow measurement.

2.4.1 Applications of Data-Driven Models in Two-Phase Gas-Solid Flow Measurement

Two-phase gas-solid flow measurement methods using data-driven models are categorized in the following sub sections based on the sensing technique used to generate the data for the purpose of training and testing the data-driven models.

2.4.1.1 Electrostatic Sensor

Substantial work has been carried out in the history related to particle flow monitoring using the electrostatic sensor in the past one decade. A novel technique of measurement of particle velocity and mass flow rate of fine particles in a pneumatic suspension using electrostatic sensor and neural network is presented in [55]. A single ring-shaped electrostatic sensor has been utilized to measure the charge value of the moving particles as shown in Figure 2-21.
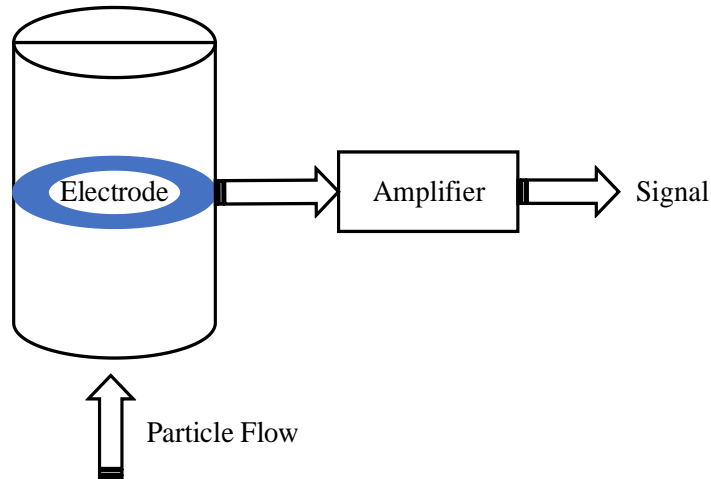
Figure 2-21: Sketch of the ring-shaped electrostatic sensor [55]

A total of nine features (six from the time domain and 3 from frequency domain) have been extracted from the acquired data of the sensor. The DC and AC levels of the signals, the number of zero crossings, variance, skewness, and kurtosis of the sensor signals have been computed for the features in time domain while average fluctuation frequency, entropy and the shape factor are the features in the frequency domain. Before feeding these extracted features to the neural network which is responsible for establishing the relationship between the characteristics of the signal and the target output that needs to be predicted, uncorrelated features have been found out using PCA. Since the first five principal components possessed the 98% of the useful information in the data, therefore the five features were collected from the output of PCA. The resultant features which are completely uncorrelated have been inputted to the neural network having log sigmoid mapping function that eventually translates the data into required mass flow rate and velocity. The Levenberg-Marquardt optimization algorithm has been used to adjust the weight factors of the neural network for training. Figure 2-22 represents the overall software block diagram of the entire system.
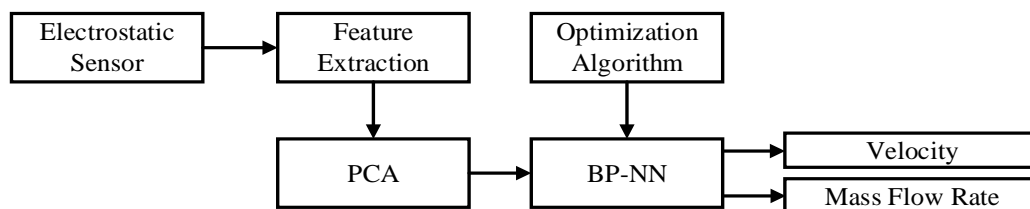
Figure 2-22: Measurement strategy of the flow meter based on a ring-shaped electrostatic sensor and neural networks [55]

Salt particles having a size ranging from 180µm to 330µm were used as solids to evaluate the performance of the presented method. Readings have been taken at 16 different air velocities by keeping the mass flow rate constant. It has been shown in the graph of PSD that all the values of the PSD graph lie between 0 to 500 Hz frequency. Therefore, to minimize the effect of white noise above 500 Hz, the sampling frequency was set equal to 1000 Hz. The relative measurement errors obtained from this proposed technique was within the range of 15%. Standard deviations of these relative errors for velocity and mass flow rate measurements are found to be 7.7% and 6.8%, respectively.

Though the electrostatic sensors are very cost efficient and give very accurate estimation about the flow parameters of solids. However, the voltage induced by the sensors is not only related to the mass flow rate of solids. Instead, the induced voltage can also be altered by velocity, temperature, humidity and other physical conditions which can eventually affect the calibration of sensor data. Kidd et al. [56] has attempted to address this issue by introducing the machine learning models with experimental data and a novel calibration method for the fusion of a gas-solid flow sensor. Sensor Fusion is the use of software that intelligently integrates data from several sensors to boost overall efficiency of the measurement system. This method can be used to measure the mass flow rate of solids in a pipeline using non-intrusive electrostatic sensing techniques. Combining the data from several diverse or unified sensors may resolve the limitations of the individual sensor. This study has proven that the output voltage of the ring-shaped electrode is mainly a function of the solids mass flow rate and the particle velocity in the pipeline while the particle size, properties and the atmospheric conditions stay stable. By integrating particle velocity into the proposed mathematical model, which is developed through machine learning technique, the output voltage could be calculated with superior precision for a broad variety of specific flow parameters from various test conditions. Transposing the model creates a new reference feature that, when used in signal processing applications, allows reliable mass flow rate measurement with compensation of effect of particle velocity. The overall procedure to apply machine learning model to measure the velocity compensated mass flow rate is shown in Figure 2-23.
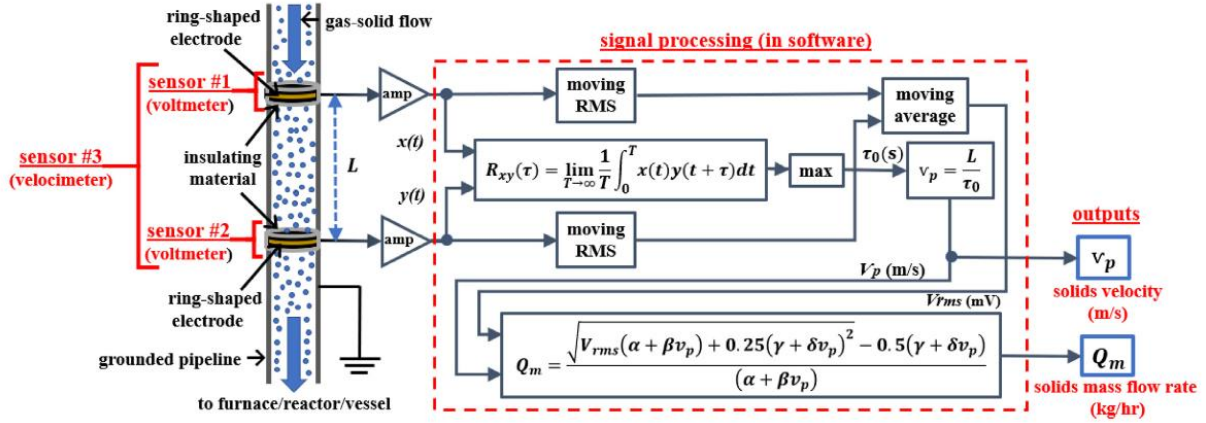
Figure 2-23: Schematic of the measurement system with electrostatic sensors and machine learning [56]

2.4.1.2 Acoustic Sensors

Aminu et. al proposes a novel instrumentation method for on-line measurement of solid flow rate, solid concentration, line pressure drop and gas velocity in a gas-solid multi-phase flow employing acoustic sensing technologies coupled with signal processing techniques and machine learning algorithms [57]. The structure of the overall measurement system is shown in Figure 2-24. The acoustic sensor is used to detect the acoustic wave produced by the impingement of the rigid particles on the turn of the flow line. Signal processing methods are used to extract the necessary impingement characteristics. The integrated, traditional Artificial Neural Network (ANN) is used to capture the distribution of the acoustic feature vectors in order to determine the connection between the measurand and the acoustic signal. However, the modern ANNs are primarily concerned with the detection of systematic trends in the time-fixed distribution of measurements and, in this case, the nature of the produced acoustic signal differs with time. A transition named the time delay neural network (TDNN) is used to catch these dynamics. The proposed framework compares the efficiency of the classic ANN and TDNN versions. The tests conducted indicate that the normalized root mean square error (NRMSE) for the classical ANN is 0.66, 0.29, 0.26 and 0.46 for solid flow rate, solid concentration, line pressure decline, and gas velocity, respectively. For the TDNN model, the NRMSE is 0.18, 0.17, 0.20 and 0.16, respectively, for strong flow rate, strong concentration, line pressure drop, and gas velocity. Compared to the ANN model, the TDNN model has improved results because the NRMSE values are lower than both versions for

41

the measurements. Overall, this research provides the foundation of using signal processing techniques and machine learning algorithms to create a quick, accurate and low-cost, real-time quantitative particulate solid flow monitoring system.
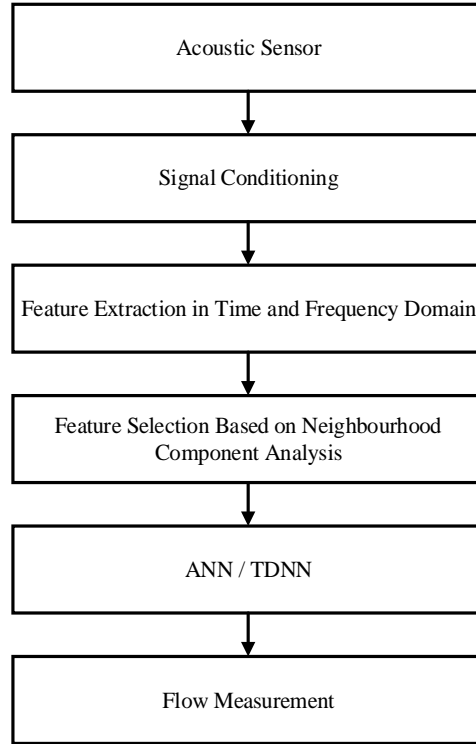
Figure 2-24: Measurement strategy based on acoustic sensing and neural networks [57]

2.4.1.3 Capacitive and Electrostatic Sensors

Some techniques of concentration measurement of three-phase coal/biomass/air are presented in [58] based on multi-sensor data fusion using an adaptive neuro-fuzzy inference system. Two sensors, capacitive and electrostatic, are used collectively to determine the flow condition of three-phase mass flow. Features from two sensors have been extracted independently to feed the ANFIS algorithm under different experimental conditions. Experimental results showed that the mean value of the capacitance has a high dependency on the concentration of the coal or biomass fuel, therefore the mean value has been chosen as the feature from the capacitance sensor. The electrostatic sensor produces a very fluctuating signal, therefore the RMS value of the sensor is the most suitable feature to analyse the magnitude of the electrostatic charge. Two methods, gradient descent and a hybrid model of gradient descent and Kalman filter are used with two

separate rule bases to develop a relationship between input feature values extracted from sensors and the output value of volumetric-concentration of three-phase material by training the ANFIS model. The results of this study show that the ANFIS used with the hybrid model of rule basis outperform the other implemented model. Estimated errors of predicting the biomass and pulverized coal flows are approximately 1.2% and 0.7%. The overall proposed measurement principle is shown in Figure 2-25.
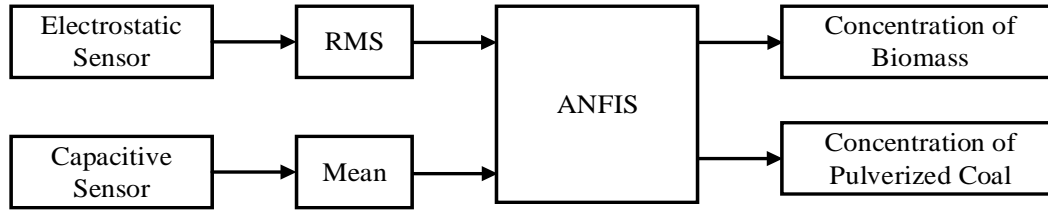
Figure 2-25: Measurement strategy based on ANFIS [58]

It has also been observed that the performance of the hybrid method is better than the gradient descent method in case of RMS error value in the training process. ANFIS trained with gradient descent learning rule can have the RMS error range from 1.7 to 0.0425 after 500 iteration steps and response become almost constant after 80 iterations. On the other hand, training ANFIS with a hybrid learning rule reduces the RMS error from 0.0035 to 0.0019 in 500 iterations.

Another multi-sensor data-fusion technique is presented in [59] for the online volumetric concentration measurement of three-phase coal/biomass/air flow. This technique combines the data of electrostatic and capacitive sensors to establish the adaptive wavelet neural network (AWNN) to predict the concentrations of biomass and pulverized coal flow. A detailed schematic diagram of the whole algorithm is shown in Figure 2-26.
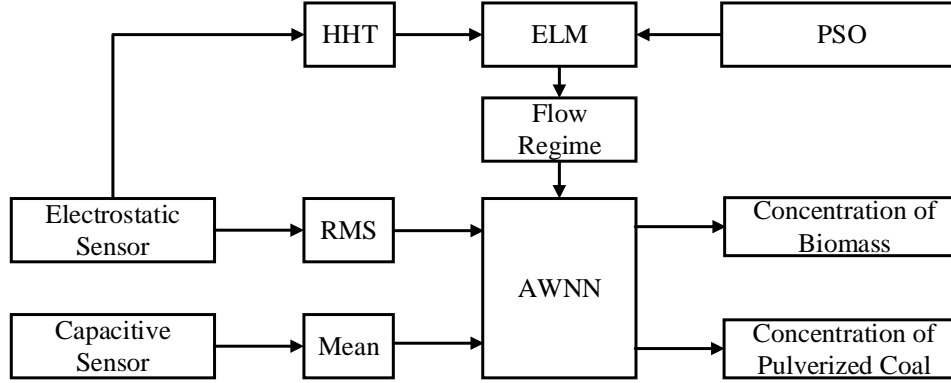
Figure 2-26: Measurement strategy based on ELM and AWNN [59]

The results of this research showed that the accuracy of the predicted results can be increased by first determining the flow regime of the three-phase flow and then applying a separate AWNN related to that particular flow regime. Therefore, as a first step of this technique, the flow regime is identified through an extreme learning machine (ELM). The sub-band energies of the electrostatic signal, Hilbert marginal spectra of different flow regimes are extracted by Hilbert Huang Transform (HHT) and then extracted features are inputted to ELM. Three sets of 100 electrostatic signals are acquired from three separate flow regimes and transformed into the features through HHT. A particle swarm optimization (PSO) algorithm is applied to find the optimized parameters of the ELM algorithm. Mean and RMS values were computed as input features of AWNN from the signals of capacitive and electrostatic sensors, respectively. Based on extracted features of different identified flow regimes, AWNN was trained using two different learning algorithms, gradient descent method, and genetic algorithm. By making the concentration measurement less affected by variations in flow regimes through ELM, the largest fiducial errors of biomass and pulverized coal has been reduced to 2.1% and 1.2%, respectively.

2.4.1.4 Pressure, Temperature and Humidity Sensors

Research, presented by Chew and Cocco [60], emphasizes the importance of machine learning approaches in fluidization to advance comprehension and include appropriate predictions with the use of pressure, temperature and humidity sensor data. The objectives of the technique presented were to provide insight into various fluidization phenomena by determining the relative dominance of process variables and to develop a model to provide predictive capability in the apparent lack

of first-principle understanding that remains unknown. Findings from random forest algorithm suggest that the radial direction had the most influential effect on local mass flux and species segregation, the total mass flux was the prevailing factor for local particle concentrations, although no component was especially influential or insignificant for local clustering characteristics. In addition, the Neural Network has been equipped to have strong predictive capabilities, without the requirement of any deterministic interpretation, with the use of a sufficiently broad dataset for training and by compensating the input variables for all the effects at practice. The main purpose of using Random Forest algorithm was to identify the influential variables. After determining the most influential features, a backpropagation neural network has been trained with six input variables, including particle diameter, superficial gas velocity, radial position, particle density, height along the riser and the overall mass flux. Either local mass flux or local concentration or local cluster characteristic has been predicted through neural network as shown in Figure 2-27.
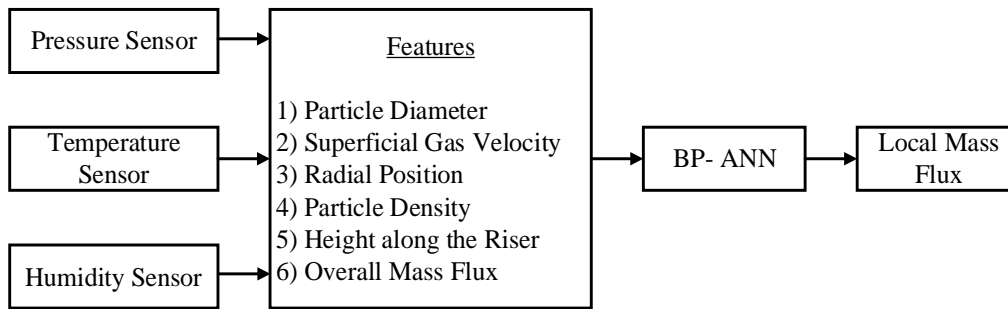
Figure 2-27: Measurement principle with a pressure, temperature and humidity sensor data and neural network [60]

2.4.2 Applications of Data-Driven Modelling Techniques for Multi-phase Flow Measurement

The applications of data-driven modelling techniques in the domain of gas-solid two-phase are limited, therefore the following examples in multi-phase flow demonstrate the other approaches and applications to use the data-driven models for flow measurement.

2.4.2.1 Coriolis Flowmeter

Coriolis flowmeter based two-phase liquid-gas flow measurement methodologies have been presented in [41]. Performance of the three popular computation techniques SVM, Backpropagation ANN (BP-ANN), RBF-ANN and Genetic Programming (GP) has been

compared with different input values of liquid mass rate and gas volume fraction and it has been concluded that the SVM model outperforms all the techniques used while performing the same set of experimentation. Liquid mass flow rate and gas volume fraction has been varied from 700 to 1450 kg/hr and 0% to 30%, respectively. A DP transducer is also used along with the Coriolis flow meter. Signals extracted from the Coriolis flowmeter and DP transducer are recorded and inputted to the computational models as shown in Figure 2-28. Computational models are driven by the four features which were extracted from the Coriolis flowmeter and DP transducer. Three of these features, observed density drop, apparent mass flowrate, and damping are extracted from Coriolis flowmeter while the fourth feature is from DP transducer. Parametric dependencies of individual input feature and combined effect of multiple features on the output of the model have also been observed in this research.
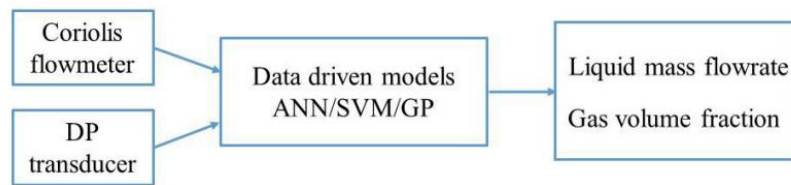
Figure 2-28: Measurement strategy based on a Coriolis flow meter and a DP transducer [41]

2.4.2.2 Ultrasonic sensor

A number of techniques based on ultrasonic sensors have been developed in the domain of multiphase flow measurement. These techniques are non-invasive, non-radioactive, robust and relatively the less expensive and have the capability to provide the required information based on the data collected from the ultrasonic signals. The only drawback which has been faced till now in employing the ultrasonic sensors is that it requires a separate technique to calibrate its data and this drawback can only be eliminated through data-driven modeling techniques. An Artificial Neural Network-based multi-phase flow measurement technique has been presented in [61] using ultrasonic transmitters and receivers. Mainly, the focus of this research is on two processes. Firstly, to present and analyze acoustic attenuation data for vertical, upward oil-continuous multiphase flows in 1-inch and 2-inch acrylic pipes and flow patterns ranging from bubbly flows to annular flows. Secondly, to develop the neural networks for the flow pattern recognition and gas volume fraction (GVF) measurement using the ultrasonic attenuation data as input. Energy ratios have

been computed using the data acquired from four ultrasonic sensors and then these energy ratios are fed into the input of the ANN model. The ANN model is comprised of two hidden layers with five and two hidden neurons, respectively. The output layer generates the identified flow pattern or the estimated gas volume fraction.

2.4.2.3 Differential-Pressure Sensor

A method, using differential pressure sensing technique for the estimation of a multi-phase flow regime and the multi-phase flow rates, has been presented in [34] with the development of backpropagation Artificial Neural Network. Probability density function (PDF) and power spectral density (PSD) has been computed on the pressure values recorded through differential pressure transducer which has been placed between two axially separated wall pressure transducers. In order to determine the two-phase flow rates, the signal from the transducer has been pre-processed by the method of principal component analysis and the independent features have been found through the method of independent component analysis. A total of 12,000 samples have been recorded at a sampling rate of 200 samples/s over the time interval of 60 seconds. The 12,000 samples can be regarded as the features and the dimension of the input data set. The task of dimensionality reduction has been performed through principal component analysis. Furthermore, Independent component analysis is filtering out only those features which are mostly independent of each other. Further on, all of these pre-processed data values are recorded to train the BP-ANN through training set. A flow chart illustrating the main steps of the proposed flow rate measurement method is shown in Figure 2-29.
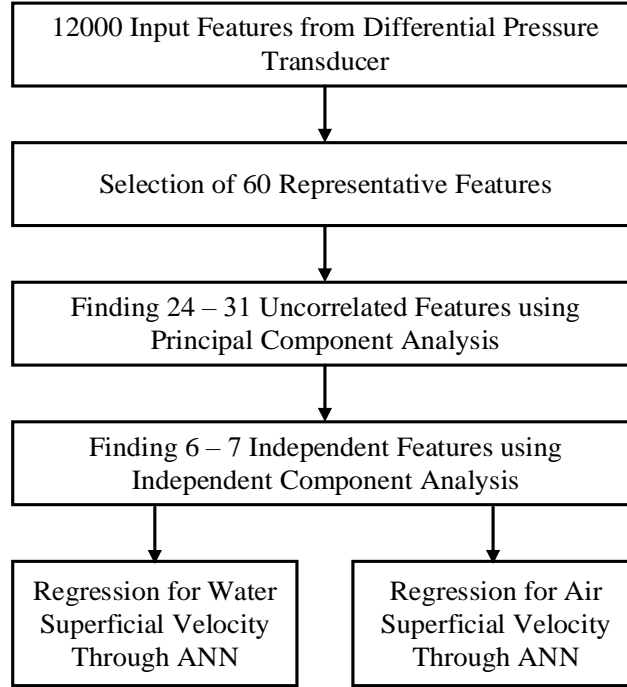
Figure 2-29: Measurement strategy based on a DP transducer and ANN [34]

One of the advantages of this method is that it has been assembled in a modular manner, so that any of its components may be substituted by a different technique, that would be more convenient or accurate under different conditions. In fact, the authors of this paper have tested the signal processing algorithm with input signals provided by different types of instruments. Therefore, in order to find whether the calibration of the method would be suitable in different environments or not, authors are also planning to test the method in several different experimental setups.

2.4.2.4 Conductance Probe

Fan and Yan presented a two-phase air-water slug flow measurement technique in a horizontal pipe using conductance probes and neural networks [62]. Instead of using statistical features which are commonly used for the input of ANNs, five characteristics parameters of the mechanistic slug flow model have been extracted which are directly related to the physical parameters of the system. These parameters are translational velocity, slug holdup, film hold up, slug length, and film length as shown in Figure 2-30. Neural networks have been evaluated with different architectures and also with the different training algorithms for the prediction of the liquid and gas flow rates and there were no significant differences found between their performances. There are no specific rules

48

defined in selecting the best combination of the architecture and training algorithms. It has been shown that the neural network method is able to better relate the characteristic parameters of slug flow to the corresponding gas and liquid flow rates implicitly than the present mechanistic slug closure model which relates the same set of characteristic parameters to the corresponding gas and liquid flow rates explicitly. The reason is that the present mechanistic model relies on a series of assumptions to model this complex phenomenon, while the neural network method learns the correlations from the given examples to avoid the need for explicit mathematical modeling.
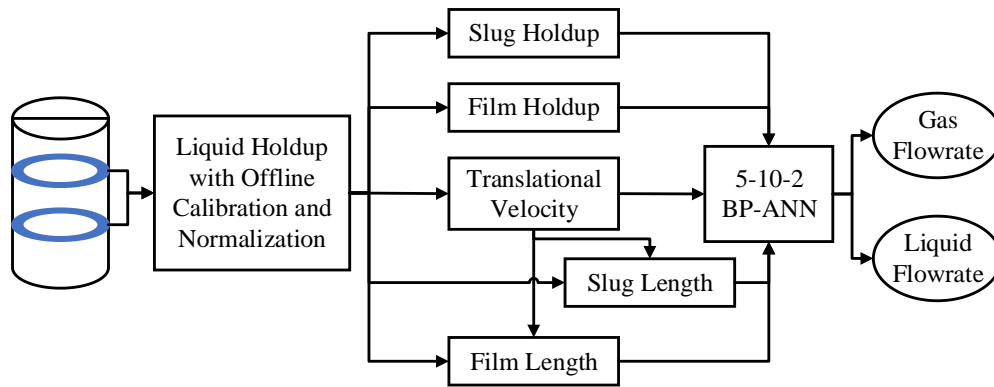
Figure 2-30: Overall schematic diagram of the measurement system [62]

2.4.2.5 Photodiode sensor array

A new void fraction measurement method, based on a laser diode, a 12x6 photodiode array sensor, and machine learning techniques, for gas-liquid two-phase flow in small channels is proposed in [63]. In order to mitigate the effect of the flow pattern on the void fraction measurement, the two-phase flow pattern is first defined by the fisher discriminant analysis (FDA). Furthermore, according to the recognition test, the appropriate null fraction measurement model established by the support vector machine (SVM) is chosen to conduct the null fraction measurement as shown in Figure 2-31. Four typical flow patterns (including bubble flow, slug flow, stratified flow and ring flow) are investigated. Compared to traditional void fraction measurement techniques, the new strategy overcomes the effect of the flow pattern on the void fraction measurement. Implementation of void fraction measurement requires the samples for the training of SVM model. Compared with the other machine learning techniques that can be used for the modeling, SVM has

been considered as a valid machine learning technique and is suitable in terms of establishing a model in the small sample conditions.
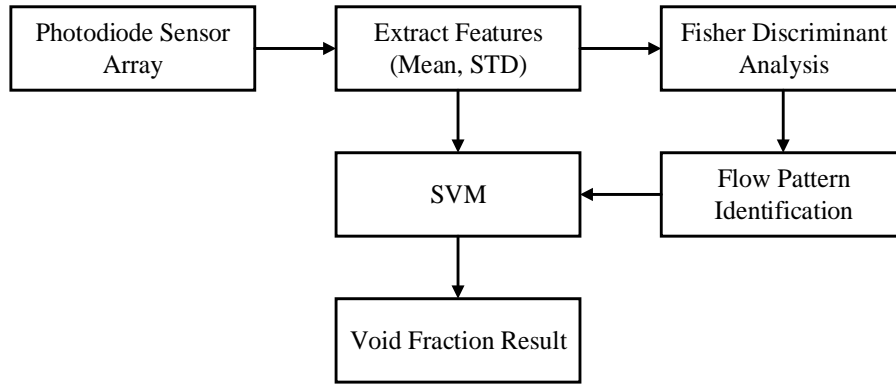
Figure 2-31: Process of void fraction measurement [63]

2.4.2.6 Pressure and Temperature Sensors

Yanfeng et al. [64] has presented a method to measure the gas-liquid two-phase mass flow rates using a slotted orifice couple (a new type of flow sensor) and neural networks. In their research work, they are utilizing a variety of sensors, including temperature, pressure and differential pressure sensors for mass flow rate measurement. They are training and developing a complex relationship between the extracted features from these sensors and the two-phase flow rate through a backpropagation neural network. A total of thirty-one features has been extracted from the sensor values, thirteen from the time domain and eighteen from the frequency domain. As per the statistical theory formulas, the first four moments of the signal represent the mean, variance, skewness and kurtosis. Authors have found it experimentally that these four moments have a very high impact in the case of pressure sensors and differential pressure sensors. Therefore, extracting these four features from one pressure and two differential pressure sensor makes 12 features in the time domain. However, in the case of temperature, only the first moment is more sensitive to gas and liquid flow rate. Therefore, this first moment of the signal from temperature sensor is the 13th feature of the algorithm. Furthermore, it was observed that the reconstruction results of these thirty-one features get weaker for the higher frequency signal. Consequently, the algorithm of PCA was deployed to find nine fully uncorrelated features out of thirty-one as shown in Figure 2-32.
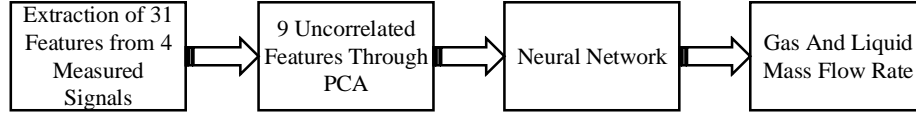
Figure 2-32: Schematic diagram of the signal processing for gas and liquid mass flow rate measurement [64]

2.4.2.7 Combination of Sensors

Previously developed impedance measurement method for two-phase flow measurement does not perform well in case of high water-cut because in that case, two-phase material is neither a good conductor nor a good insulator. Therefore, in order to remove this gap, the impedance measurement technique has been integrated along with an acoustic sensor to improve the accuracy of flow measurement in case of high water-cut within range of 40% to 60% [65]. This study involves the non-radioactive probe which is capable of combining impedance and acoustic measurements that predicts the accurate two-phase flow rate in real-time. Physical properties of the fluid dynamics are measured through capacitance, conductance, ultrasonic, pressure and venture probes to feed the input of feed-forward neural network for pattern recognition. As a first step of this process, the fraction composition of the water has been determined using a neural network with capacitance, ultrasound, conductance, and pressure probe values as an input of this neural network. Later on, in the next step, the flow rate of the mixed fluid has been determined using previously identified water composition along with pressure and venture probe as an input to the second neural network.

2.5 Summary

This chapter has reviewed the existing direct and indirect methods of gas-solid two-phase flow measurement. A number of sensor paradigms, used for the purpose of flow measurement, has been explained while the importance of non-intrusive type of electrostatic sensors has also been elaborated.

Amongst all the techniques presented in the literature, it has been observed that the electrostatic sensors provide the ideal sensing technique for the measurement of particle velocity and mass flow

rate of solids due to their robustness and cost-effectiveness. In addition, signals from electrostatic sensors have been found to be sensitive to moving solids with little influence of the physical properties of solids being accumulated in the pipe that adversely affect other sensing techniques. The conventional method for mass flow rate measurement of solids through electrostatic sensing is inferential, i.e. the mass flow rate of solids is derived from the measured particle velocity and concentration of solids, while the latter is measured through RMS level of the electrostatic sensor output. This inferential method of measuring the mass flow rate of solids is widely implemented. However, despite the advantages of electrostatic sensors for particle velocity measurement, the main problem in applying such a sensing technique is to relate the solids concentration to the RMS level of the sensor output, which depends on various physical factors, including particle velocity, ambient conditions, pipe orientations and solids size/shape etc. It is worth noting that different physical factors affect the measurement of solids concentration and hence the mass flow rate of solids in the inferential method.

The second part of this chapter explains the flow measurement techniques based on variety of sensor paradigms and data-driven modelling. A range of measurement techniques based on a variety of sensing principles, such as electrostatic, optical, acoustic, ultrasonic, capacitive sensors and nuclear magnetic resonance have been developed. All these types of sensor have the advantage of being non-intrusive and capable of measuring the mass flow rate of solids under certain process conditions. Amongst these techniques, the electrostatic sensors coupled with correlation signal processing algorithms provide a promising and practical solution to the measurement of particle velocity due to their advantages over other sensing techniques, such as robustness in a hostile environment, non-intrusiveness in operation, and inexpensive capital cost and low maintenance requirements.

The conventional method for mass flow rate measurement of solids through electrostatic sensing is inferential, i.e. the mass flow rate of solids is derived from the measured particle velocity and concentration of solids, while the latter is measured through root mean square (RMS) level of the electrostatic sensor output. This inferential method of measuring the mass flow rate of solids is widely deployed. However, despite the advantages of electrostatic sensors for particle velocity measurement, the main problem in applying such a sensing technique is to relate the solids concentration to the RMS level of the sensor output, which depends on various physical factors,

including particle velocity, ambient conditions, pipe orientations and solids size/shape etc. It is worth noting that different particle velocities, pipe orientations and ambient conditions affect the measurement of solids concentration and hence the mass flow rate of solids in the inferential method.

With the advent of data driven techniques in the field of instrumentation and measurement, it is worth exploring data driven modelling as a potential method for mass flow rate measurement of solids with a minimized impact of the particle velocity, pipe orientation and inhomogeneous distribution of solids across the pipe cross section. In view of the difficult and dynamic behaviour of solids in a pneumatic pipeline, multi-channel electrostatic sensor arrays are proposed to overcome the limitations of a single sensor for the measurement of mass flow rate of solids. The data driven models, in this case, also offer the methodical way of combining the data from multiple sensors.

Chapter 3 Test Rig Reconfiguration and Test Conditions

3.1 Introduction

This chapter provides the details about the test rig reconfiguration that is required to carry out the experiments for the measurement of gas-solid two-phase flow parameters. For this purpose, a laboratory scale test rig was designed and constructed for the measurement of particle velocity and mass flow rate of solids under different air velocities, pipe orientations and ambient conditions. The whole test rig consists of a sensing unit, a vibratory feeder, a suction pump, and a pneumatic pipeline as shown in Figure 3-1. Solids are collected in the storage tank of the suction pump from where they are separated manually at the end of the experiment. Firstly, the initial design of the test rig, developed by previous researcher at University of Kent [66], is explained. The initial design of test rig was comprised of a sensing unit with arrays of electrostatic sensors and the signal conditioning units. The magnitude of vibrations and the suction power was adjusted by an analog controller.
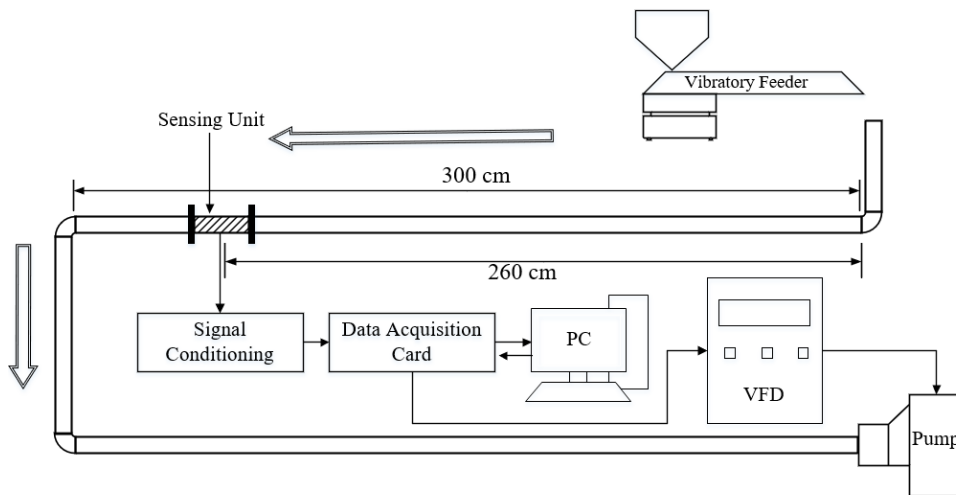
Figure 3-1: Schematic of test rig

Secondly, the modified choice of sensors, including a DP transducer, an accelerometer, a temperature, a humidity and a capacitive sensor, that are added in the sensing unit for the purpose of this research, are explained. The structure of the pneumatic pipeline is also modified in a way that the sensing unit can be adjusted in any orientation. For the effective control of magnitude of vibrations and the suction power, the analog controllers are replaced with variable frequency drives (VFD). The parameters to control the VFDs through computer are also presented in detail. For the safe operation of experiments, the health and safety regulation and the operating procedures of the test rig are elaborated. The software operation to acquire the sensor data and to produce the measurements is also explained at the end of this chapter.

3.2 Sensing Unit

The initial design of sensing unit was comprised of only arrays of electrostatic sensors and it was proposed and constructed by Qian et al. [20]. The simplified schematic of the sensing unit that is modified and proposed for this research is shown in Figure 3-2 which consists of an array of ring and four arrays of arc-shaped electrodes, a differential pressure (DP), relative humidity (RH), temperature sensor, accelerometer and capacitive sensor placed non-intrusively within and outside the walls of sensing head. Each array of electrostatic sensors is comprised of four electrodes. The fusion of data from multiple sensors is able to provide an accurate measurement through data-driven modelling approach in case any of the sensor fail to provide the correct signal. This is due to the reason that the outliers, from the signal of the sensor providing noisy data, are removed before further processing. Furthermore, the data-driven models do not work directly through sensor output. Infact, the statistical features are extracted from the sensor data over a period of time.
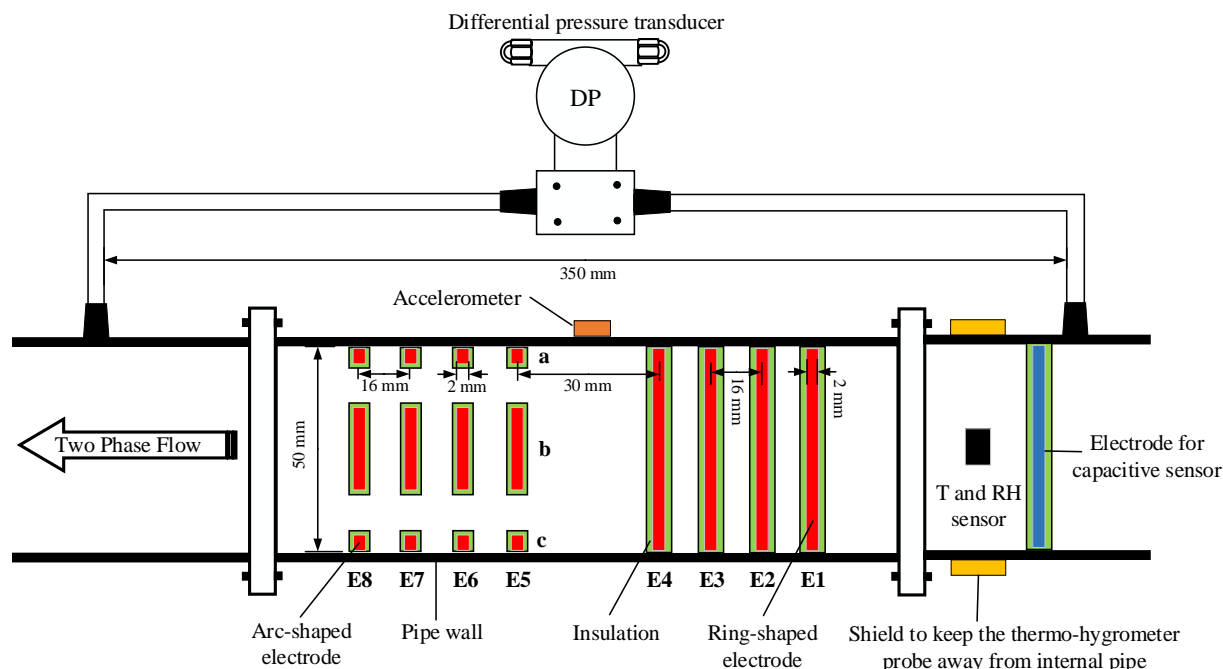
Figure 3-2: Design of sensing unit

3.2.1 Electrostatic Sensors with Ring and Arc-shaped Electrodes

Electrostatic sensors provide a potential solution to the measurement problem due to their strengths over other sensor techniques, such as robustness in a hostile environment, non-intrusiveness in operation, inexpensive capital cost and low maintenance requirements. The proposed work is based on the flush-mounted ring as well as arc-shaped electrostatic sensors, designed and implemented by previous researcher [66], to measure the cross-sectionally averaged and localized flow parameters, respectively as shown in Figure 3-3. Four arrays of arc-shaped electrodes are to cover the top, left, right and bottom area of the pipeline. Instead of using a single circular-shaped electrode, four arc-shaped electrodes are used to produce the four independent voltage signals to increase the number of input features for machine learning models. Four electrodes are placed at an equal distance of 16 mm from each other.

Being completely non-intrusive in nature, this configuration of the electrostatic sensor does not affect the flow pattern of the solids. In this case, the solids moving near the surface of the pipe may have a smaller magnitude of electrostatic charge as compared to those moving in the middle of the pipe. However, in case the measurement strategy is based on data driven models, this factor cannot affect the accuracy of the predicted output as the measurement system is not directly

dependent on the charge value rather it is dependent on the pattern of values that have been acquired from a combination of sensors. Correspondingly, the complexity and the abstraction levels of the data driven models may help in mapping the pattern of sensor values to the actual flow parameter.

Deciding the number of electrostatic sensors and their position in the pipe is a crucial step that needs some research in this area because the distribution of the solids is not uniform in the pipe. The distribution of solids inside the pipe is dependent upon the combination of mass flow rate and air velocity. For example, if the air pressure is low and the volume of solids is high then most of the solids will be moving along the lower surface of the pipe due to the gravitational force in case the pipe is horizontally oriented.
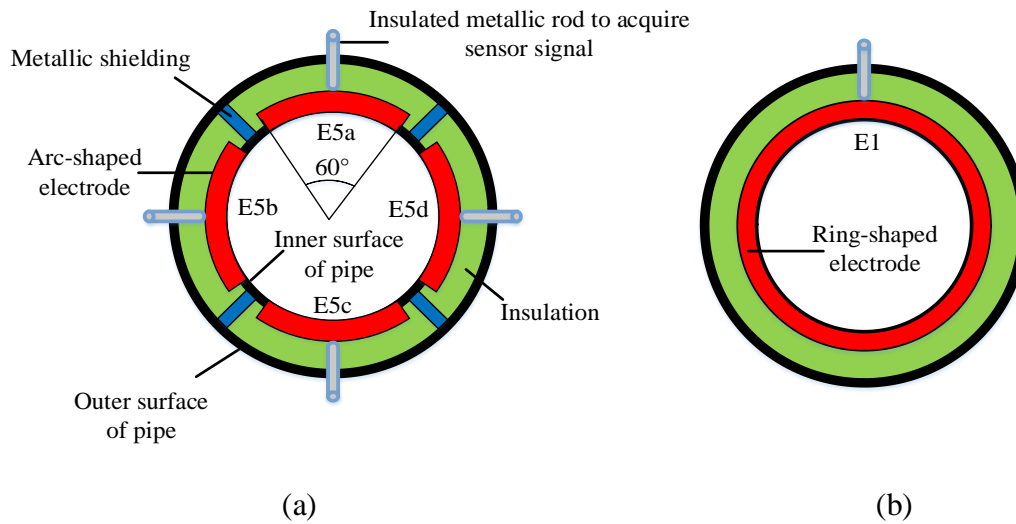
Figure 3-3: Cross-sectional view of electrostatic sensors

(a) Arc-shaped (b) Ring-shaped [66]

3.2.2 Differential Pressure Transducer

The DP transducer can determine the drop in line pressure due to air-solids mixture flow in the pneumatic pipeline. The experiments are focused on different particle velocities and mass flow rates of solids, therefore the DP transducer is used to measure the drop in line pressure due to both particle velocity and mass flow rate of solids. Higher concentration of solids and particles velocity can cause a significant decrease in the line pressure. Therefore, the difference in line pressure between the two ends of the sensing head can provide useful information about the flow parameters of the solids. In the view of range of the differential pressure in the pipeline with given test

57

conditions, Deltabar s PMD75 DP transducer is used that can measure the drop in line pressure from -10 kPa to 10 kPa. The DP transducer measures the drop in line pressure with measurement uncertainity of 0.5% and the transducer is calibrated by

$$DP_{mbar} = \left(\frac{\max pressure}{V_{max} - V_{min}}\right)(V_{out} - V_{min}) * 1000 \tag{3-1}$$

$$DP_{mbar} = \left(\frac{0.1}{9 - 1.8}\right)(V_{out} - 1.8) * 1000 \tag{3-2}$$

Where V_{max} and V_{min} is the maximum and minimum output voltages, V_{out} is the output voltage of the DP transducer and max pressure is 0.1 bar as described in the data sheet.

The difference in pressure is determined through 4 to 20 mA current signal as shown in Figure 3-4 (a). The data acquisition module, used with the test rig, can only take the voltage signal, therefore a load resistor R_L is connected between voltage source U and positive terminal of DP transducer as shown in Figure 3-4 (b). Figure 3-5 shows the ranges of the U and R_L for the safe operation of DP transducer.

(a)   (b)
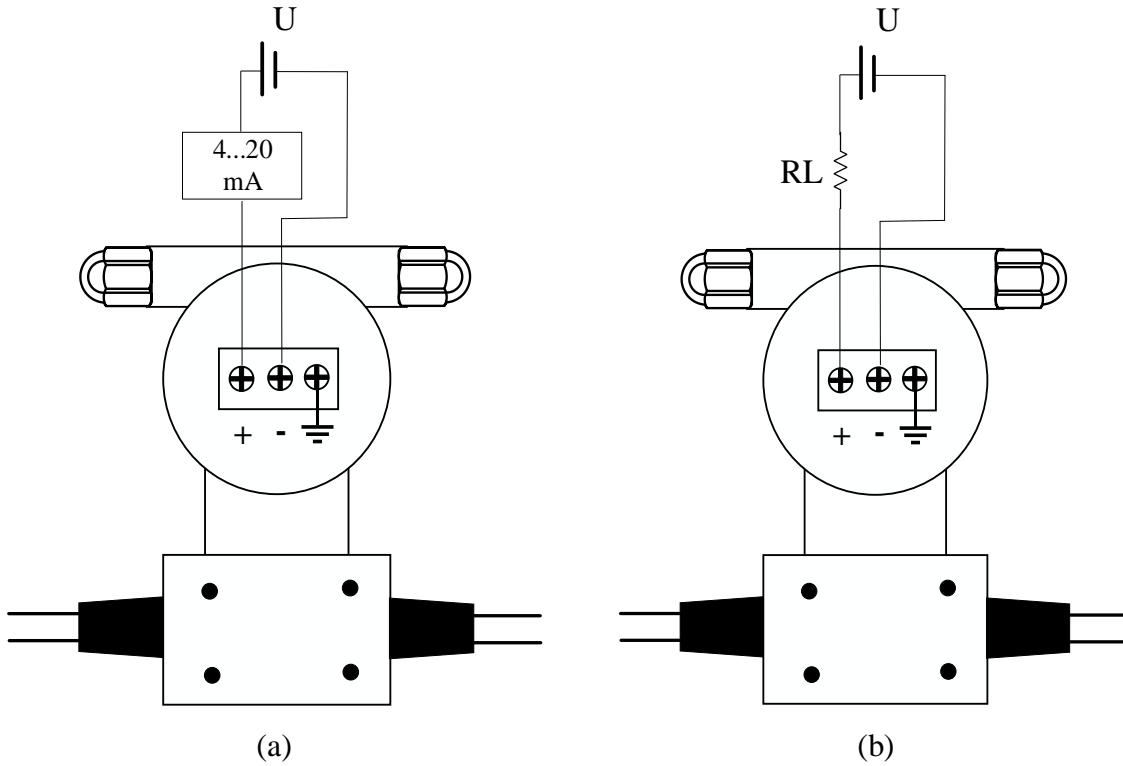
Figure 3-4: Wiring diagram of DP transducer (a) measurement through 4-20 mA signal (b) configuration to convert current signal into voltage
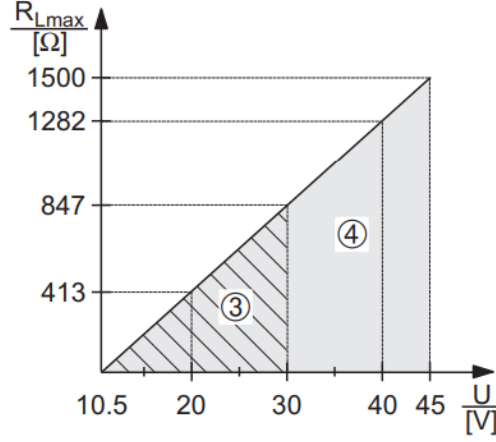
Figure 3-5: Load diagram for the selection of load resistance R_L [67]

By keeping the internal resistance of DP in mind, the value of R_L must be selected in a way that suits with the value of U. The relationship between the maximum value of load resistance and supply voltage is defined by

$$R_{Lmax} \leq \frac{U - 10.5}{23 \ mA} \tag{3-3}$$

Where R_{Lmax} and the U is the maximum load resistance and the supply voltage, respectively.

The value of supply voltage can be selected in the range of 10.5 V to 45 V, as shown in Figure 3-5. To keep the output voltage of DP transducer in the range of 0 to 10V, the selected values of R_L and U are 520 Ω and 24 V, respectively, using Equation (3-3).

3.2.3 Temperature and Humidity Sensor

As the solids move inside the pipeline, their inter-particle collision and the collision with the wall of the pipeline may result in heating effect which can influence the internal temperature of the pipeline. This heating effect could be non-linearly proportional to the particle velocity and the solids concentration. At a lower solids concentration, an increase in air velocity can cause a

reduction in temperature because of the rapid exchange of air between the pipeline and the environment. However, at higher solids concentration, a higher air velocity can cause substantial chances for the solids to hit the wall and increase internal temperature. Therefore, it is believed that the temperature sensor may provide a useful information about the mass flow rate of pneumatically conveyed solids. Similarly, a higher moisture content may cause solids to stick around the surface of the electrode and hence resulting in obtaining a weak electrostatic signal. Therefore, in order to determine the level of moisture content and to take necessary action, a relative humidity sensor is required.

For this purpose, a digital thermo-hygrometer Milticomp MP780117 with a thin probe is non-intrusively installed in the sensing head to observe the internal temperature and relative humidity of the pipeline as shown in Figure 3-6. With the help of a supporting ring, it is ensured that the tip of the thermo-hygrometer probe stay away from the cross-section of the pipe to avoid the interference with the gas-solid flow. Temperature and relative humidity values are manually recorded from the screen of thermo-hygrometer and saved in the dataset.
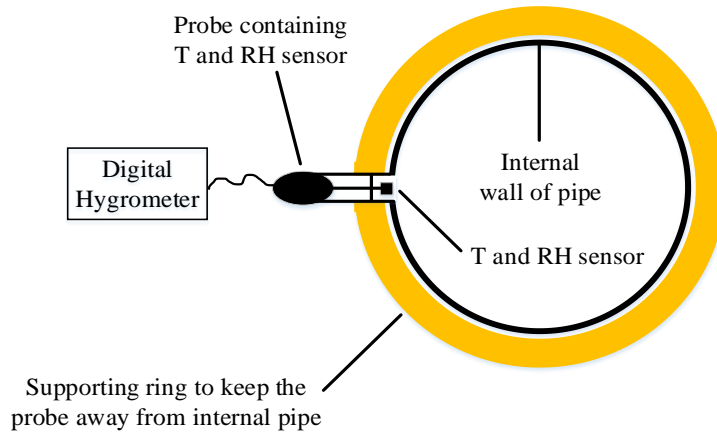
Figure 3-6: Cross-section of the pipe containing a thermo-hygrometer probe.

3.2.4 Accelerometer

In this study, the experiments will also be focused on mass flow rate measurement under different pipe orientations that may affect the flow regime of the gas-solid flow in the pipeline and associated characteristics of sensors and hence causing a large degree of measurement error. Therefore, an accelerometer, in this case, would be helpful to train the data-driven models for intended measurands according to the pipe orientation. It has to be noted that the purpose of

60

installing an accelerometer with the sensing head is to acquire the position related information but not to measure the vibration in the pipeline. Figure 3-7 shows the ADXL326 module that is comprised of XL325b accelerometer and its driving circuit. ADXL326 has the advantage of providing the direct analog signals about x, y and z-axis. The signals from the accelerometer are amplified to bring the magnitude up to a reasonable level for the further signal processing. After taking the amplified signal from xyz axis sensor, demodulator provides the three independent signals for x, y and z axis that are further amplified to a desired level to keep the output signals between 0 and 3V as shown in Figure 3-8. It is intended to place the sensing head in different angles along the z-axis, therefore the signals are only derived from the channel linked with z-axis.

Figure 3-7: Interface diagram of the ADXL326 accelerometer [68].
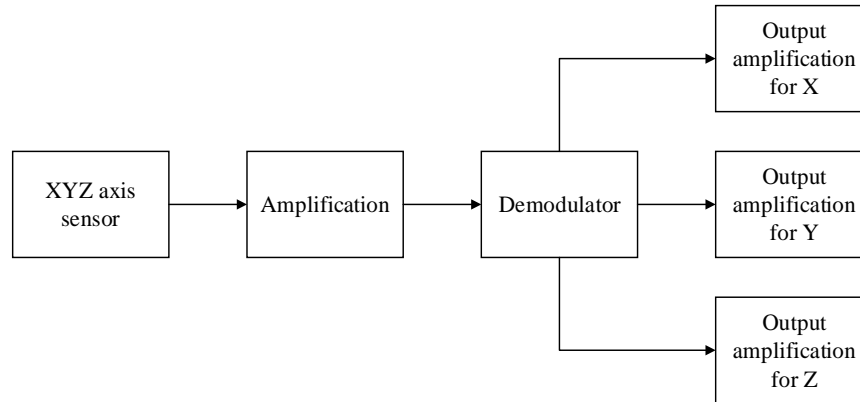
Figure 3-8: Functional block diagram of ADXL326 [68].

3.2.5 Capacitive Sensor

In this thesis, it has also been attempted to measure the ratios of two different particle types by conducting some experiments with the mixture of coal/biomass/air. The different particle types may have different dielectric constants in the gas-solid mixture, therefore the measurement of

capacitance in the spool piece may help to identify the ratio of each particle type. Two electrodes, made of copper, are placed inside the pipe in a non-intrusive way. A range of electrodes with different lengths and widths are tried to determine the specifications of electrodes that provides the reasonably stable signal. The selected electrodes are 4 mm wide and covers the $160°$ circumference of the pipe, as shown in Figure 3-9. The signal from the receiving electrode is obtained using low-impedance coaxial cables. A high resolution and high precision LCR meter (KEYSIGHT E4980AL) is used to measure the capacitance value between the two electrodes.
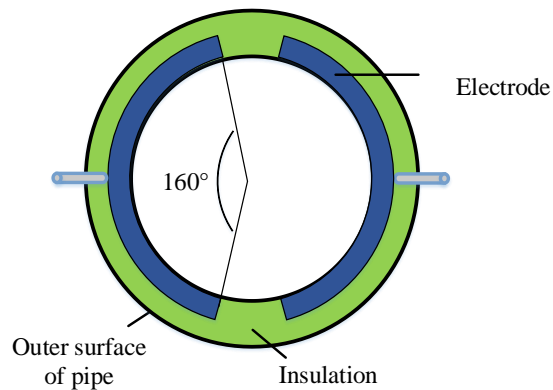
Figure 3-9: Cross-section of pipe containing electrodes of capacitive sensor.

3.3 Mechanical and Electrical Elements of the Test Rig

3.3.1 Design and Implementation of Test Rig

Experiments that are detailed in this thesis were undertaken on a 50 mm bore laboratory-scale pneumatic conveying test rig, which was designed and built by the Instrumentation Research Team at the University of Kent in their earlier research programmes [66, 2]. The initial design of the test rig, as shown in Figure 3-10, consists of a vibratory feeder, stainless steel pipe sections, and a suction pump. In this case, the layout of the pipeline arrangement is structured in rectangular shape with one end connected to a suction pump while another with a vibratory feeder. In the initial design of the test rig, the sensing unit is integrated with the pipeline in such a way that it can be placed and fixed either in horizontal or vertical orientation. The latest version of the test rig, which is modified in this study, is based on the dynamic movement of the sensing unit with the help of a flexible joint as shown in Figure 3-11. In this case, two metallic stands are used to hold the sensing unit in a specific orientation. Flexile joints are used to make the sensing unit moveable in any

direction as shown in Figure 3-12. Flexible joints are made of plastic material (polyvinyl chloride) with spiral-shaped metallic rings outside to avoid the pipe being blocked while bending it in different orientations. It is also made sure to keep the inner surface of the pipe plain to avoid any kind of distortion in the flow regime. The movement of the sensing unit in three different orientations is demonstrated in Figure 3-13 to Figure 3-15. The sensing head will be 50 cm away from the horizontal pipeline when it is placed in the vertical orientation. In this case, the distance between the sensing head and the pipe bend is very small and hence it may experience a stratified or roping flow of solids in vertical pipe orientation.
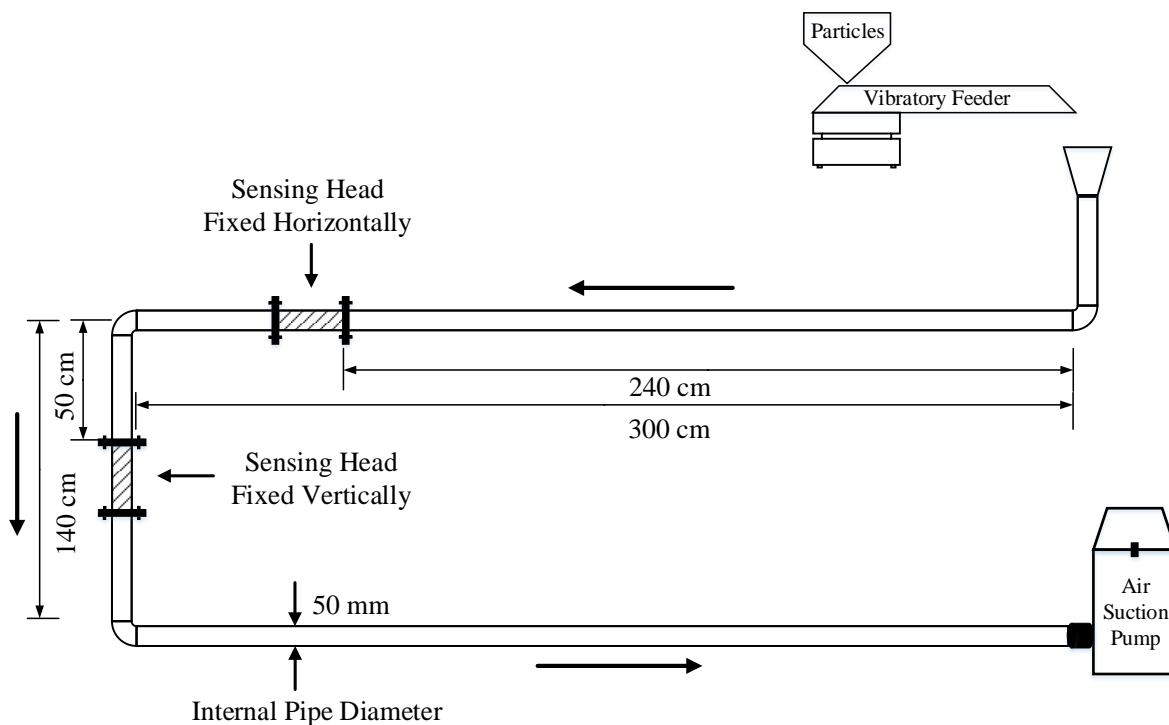
Figure 3-10: Schematic of the test rig with fixed orientation of sensing unit
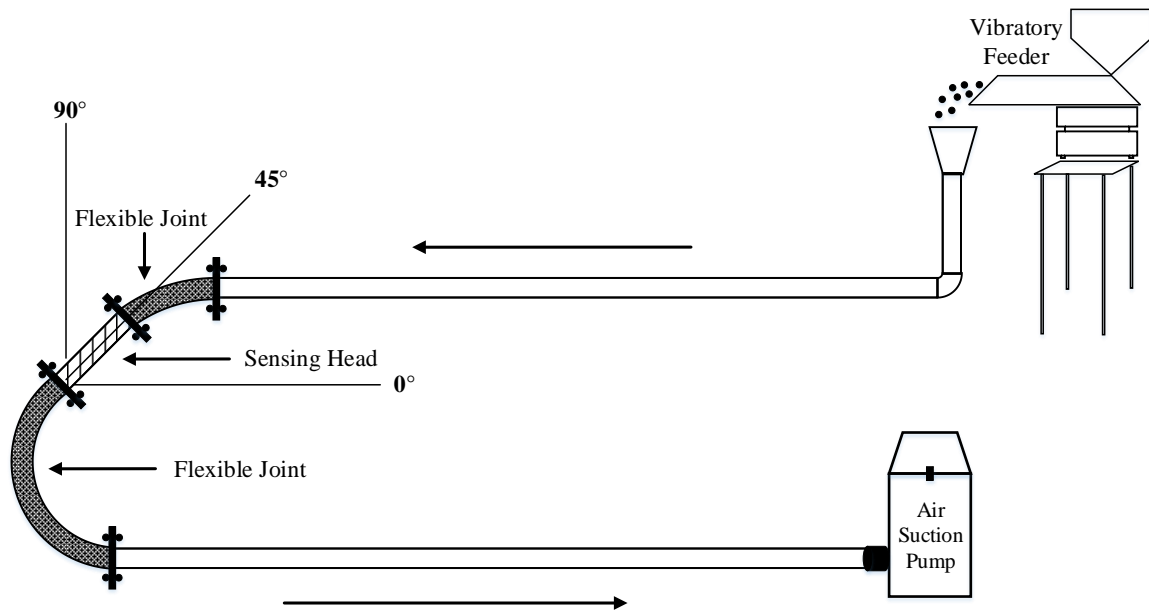
Figure 3-11: Schematic of the test rig with variable orientation of sensing unit
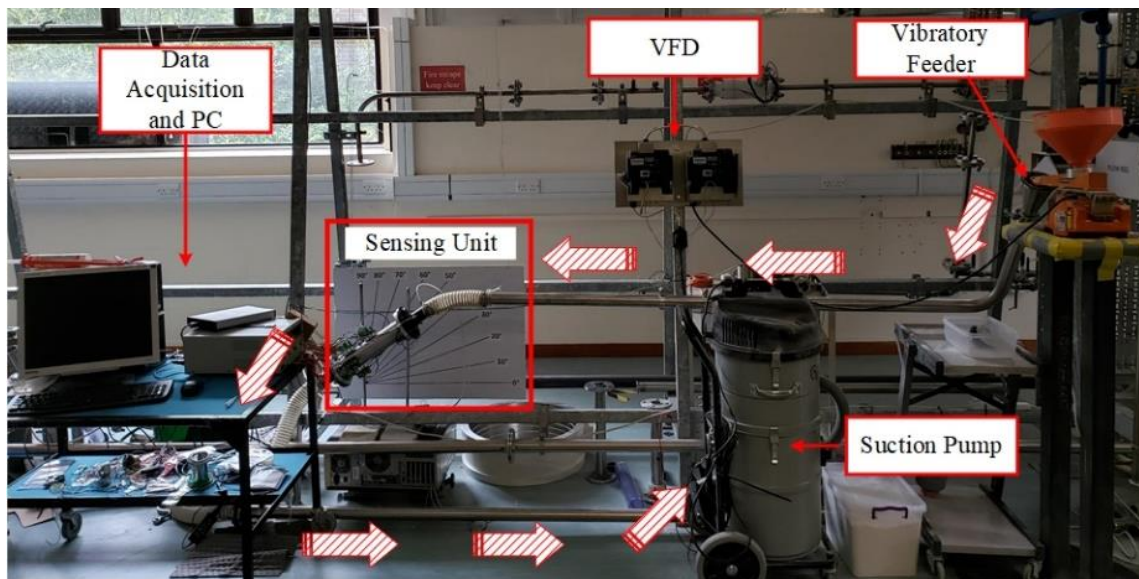
Figure 3-12: Laboratory scale test rig with variable orientation of sensing unit
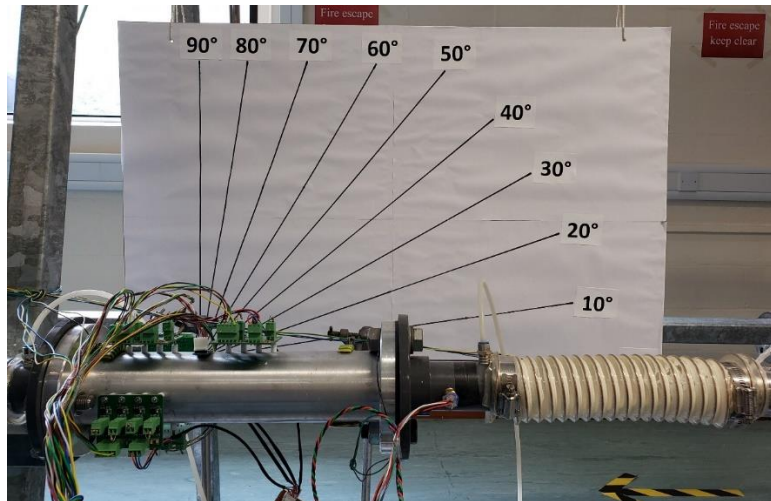
Figure 3-13: Sensing unit at the angle of 0°
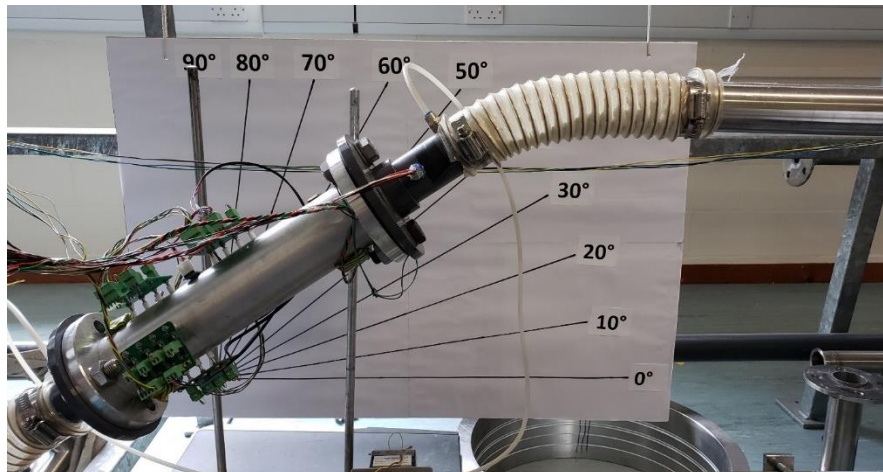
Figure 3-14: Sensing unit at the angle of 40°
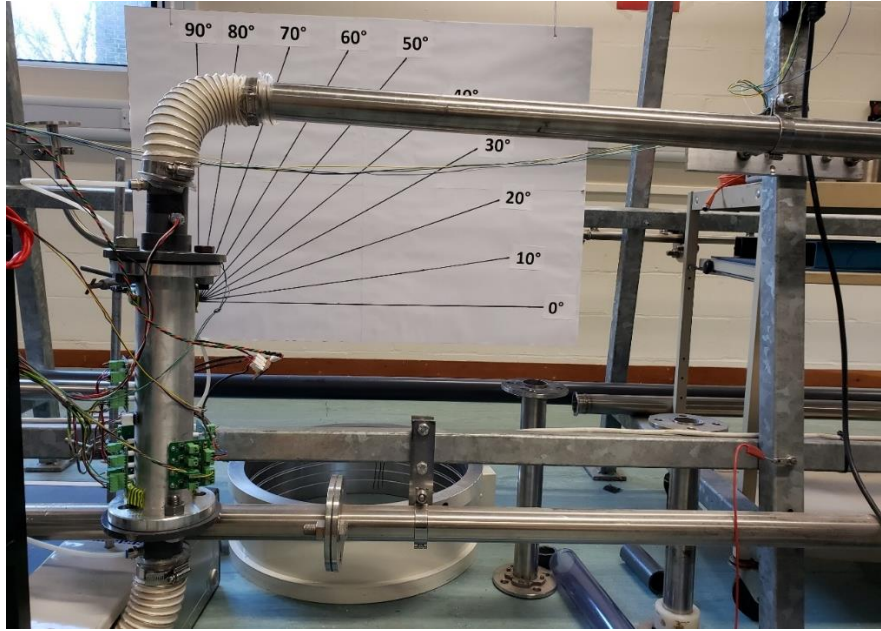
Figure 3-15: Sensing unit at the angle of 90°

As can be seen in Figure 3-10, solids will be placed on the upper surface of the vibratory feeder and then by controlling the flow rate of these particles through the magnitude of vibration, these particles will be directed towards the inlet of the pipe. The solids are pneumatically conveyed through the pipe using air. Therefore, in order to make them travel and pass through the sensing unit, a suction/vacuum pump is used to suck the air and move it in the counter-clockwise direction through pipeline. Air pressure of this suction pump and vibrations of the vibratory feeder can be adjusted externally using a variable frequency drive. Data from the sensing system is acquired with data acquisition module and it is stored in the computer placed near the test rig. Two commercial level air conditioning units are also installed in the lab to record the data under different temperature and humidity set points to determine the effect of different ambient conditions on the accuracy of predicted flow parameters.

3.3.2 Vibratory Feeder

The vibratory feeder regulates the solids concentration that is being transferred to the inlet of the pipeline. By adjusting the vibrations of the feeder, the solids concentration in the vibratory feeder is controlled and maintained. For this purpose, Eriez 16C Electromagnetic Feeder is used that is powered through 220 Vac. The magnitude of the vibrations in the feeder can be controlled with a

variable frequency drive either automatically through computer or manually through user operating panel. Solids are placed in the hopper from where they are moved into the tray of vibratory feeder and then they are transferred into the pipeline with the help of vibrations. Initially, the solids stay on the tray. The solids move in a sequence of tiny hops as the tray vibrates. This sequence of hops combines to produce a steady movement of the solids. The direction of motion is determined by the angle of the springs. The solids rise up from the plate, perpendicular to the position of the springs.

3.3.3 Suction Pump

The suction pump Numatic MV 570 is used that operates on 220 Vac to create a vacuum in the pipeline. Solids are conveyed through pipeline and collected in the storage tank of the suction pump. The velocity of the air is controlled by adjusting the suction power with a variable frequency drive. The suction pump consists of a big storage tank and the head which contains all the electronic arrangements of the pump. The vacuum is created by the fans in the head which are protected by metallic fins. A removable filter, made of fabric, is placed between the head and storage tank to avoid the fans being contaminated by solids.

3.3.4 Variable Frequency Drives

The particle velocity and the solids concentration are dependent on the suction pump and the vibratory feeder respectively, therefore two independent variable frequency drives (VFD) are required to control the suction power and magnitude of vibrations. Based on the electrical specifications of the suction pump and the vibratory feeder, two Siemens Sinamics V20 VFDs were used in this research. The basic working principle of VFD is described in Figure 3-16. VFD takes the AC power as the input and then converts it into DC with a simple rectification process. Smoothening circuit eliminates all the ripples and fluctuations from the signal to make it regulated DC signal. In the inverter part, PWM based switching is implemented to formulate an AC signal of the desired frequency. The controller part, in this case, demands several parameters to be set by the user. All of these parameters, their description, and the selected values are described in Table 3-1. It has to be made sure that the parameters are set as per the electrical specifications of the motors of the suction pump and vibratory feeder. Some of these parameters are also very essential concerning the automatic control of the speed of motors through a computer program. In order to set these parameters in the controller, a basic operator panel (BOP) is used which is shown in

Figure 3-17. Each of the buttons on the BOP offers a range of functionalities based on the way it is pressed as listed in Table 3-2.
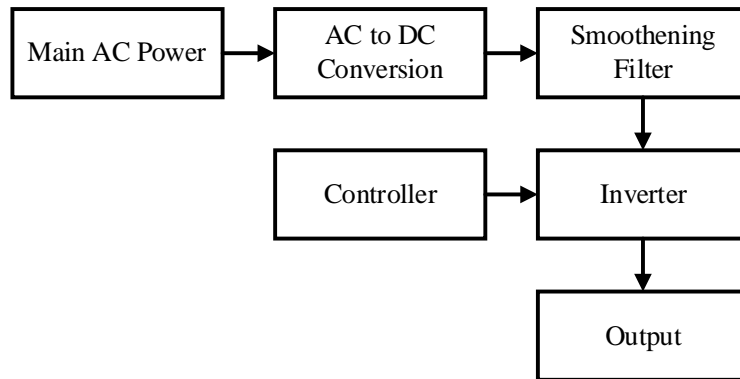
Figure 3-16: Working principle of VFD

Table 3-1: VFD parameters

Parameter	Description	Selected value for pump	Selected value for feeder
P0100	50 / 60 Hz selection [0 2]	0	0
P0304	Rated motor voltage [V]	220	220
P0305	Rated motor current [A]	8	0.5
P0307	Rated motor power [kW]	2200	300
P0308	Rated motor power factor	0.9	0.85
P0309	Rated motor efficiency [%]	Auto	Auto
P0310	Rated motor frequency [Hz]	50	50
P0700	Selection of command source	2	2
P1000	Selection of frequency	2	2
P0701	Function of digital input 1	1	1
P0702	Function of digital input 2	12	12
P0703	Function of digital input 3	9	9
P0704	Function of digital input 4	10	10
P0771	Analog output	21	21
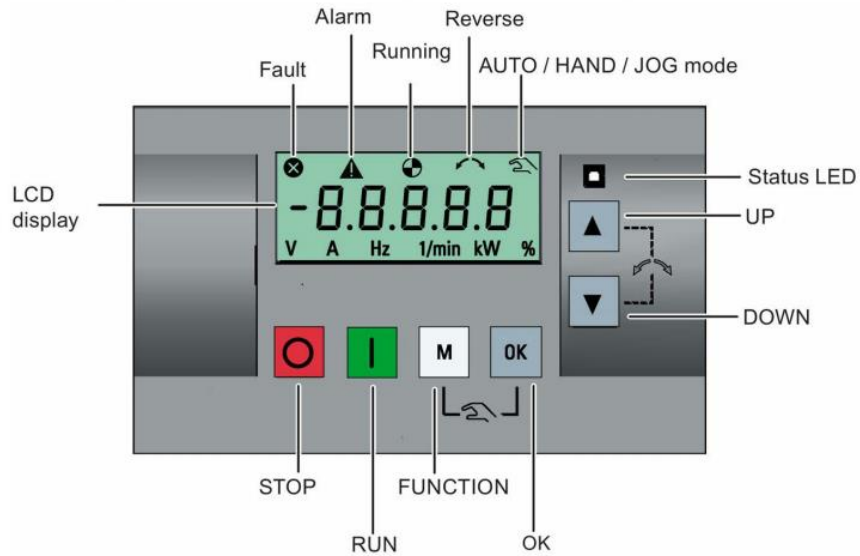
Figure 3-17: Basic operator panel (BOP) [69]

Table 3-2: Functionalities associated with the BOP [69]

Button	Function
	Stops the inverter
	Starts the inverter
	Short press (< 2 s): 1. Enters the parameters menu 2. Restarts the editing on the screen 3. Returns to previous screen if pressed twice Long press (> 2 s): 1. Returns to main screen 2. Enters the setup menu
	Short press (< 2 s): 1. Clears the fault indications

	2. Switches between status values 3. Turns the edit value mode on Long press (> 2 s): 1. Quick parameter value edits 2. Accesses fault information data
M + OK	Press to switch between auto, manual and jog mode
▲	1. Navigates through menu 2. Edits parameters value 3. Increases the output frequency while in run mode
▼	1. Navigates through menu 2. Edits parameters value 3. Decreases the output frequency while in run mode
▲ + ▼	Reverses the movement direction of motor

3.3.5 Signal Conditioning

The magnitude of the electrostatic charge induced on the electrode is substantially weak, therefore a signal conditioning unit is required to boost the signals to a useful level. The whole signal conditioning unit consists of a current to voltage converter, variable gain amplifier and the active low-pass filter to eliminate the high frequency noise as shown in Figure 3-18. The design is based on the idea presented in [66, 2]. The signal conditioning unit is constructed very close to the electrodes to reduce the chances of mixing external interference. The second stage of amplification has the advantage of adjusting the gain of the signals to cover the whole range of the analog to digital converter.
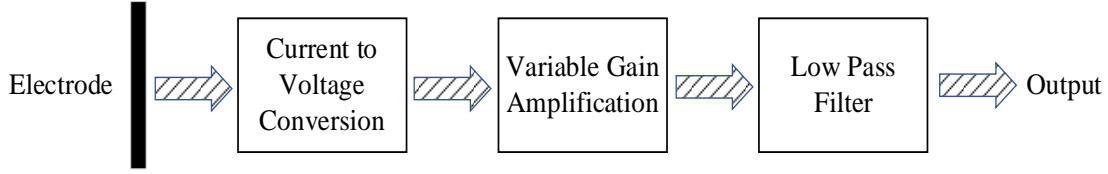
Figure 3-18: Schematic of signal conditioning

The electrostatic charge induced on the electrode can be obtained in terms of current signal. A current signal is more vulnerable to being lost due to the in-built impedance of a complex electronic circuit. Moreover, the magnitude of the charge density on solids is as low as 10^{-7} C/kg [70]. Therefore, the weak current signal from an electrode is initially converted into a voltage signal via a pre-amplifier sitting right next to the electrode with a gain of 5 mV/nA.

The current signal obtained in the first stage of amplification lies in the range of nano amperes. The largest possible value of the feedback resistor still cannot produce a significant change in the gain of output voltage. In order to eradicate this problem, another stage of amplification is implemented to amplify the voltage signal from the range of millivolts to volts. Moreover, based on the experimental conditions in some of the proposed plans, the amplified signal would either be too large reaching the saturation or it would be too small consuming only a few ADC levels of DAQ card. Therefore, the feedback resistor of this particular stage of amplification is kept variable which can be set externally as per the specifications of the sensor signal. Since the first stage of amplification contains a DC component, therefore a decoupling capacitor is used in this stage of amplification to avoid the DC component being amplified.

The electrostatic signal with high-frequency noise can influence the correlation coefficient used in the cross-correlation technique to determine the particle velocity. Therefore, a low pass filter is implemented as the last stage of signal conditioning to remove the high-frequency noise that mainly occurs from external interferences. The bandwidth of the signals from the pre-amplifier is normally less than 1 kHz [55]. Therefore, the cut-off frequency of the low-pass filter is set as 10 kHz to cover the full range of the electrostatic signals [16].

3.3.6 Data Acquisition and Control System

In order to fetch and store the sensor data in the computer, a data acquisition (DAQ) USB-6351 of National Instruments (NI) is used which performs the multiplexing and analog to digital conversion on post-processed sensor data. The data from various sensors can only be transferred

to the computer via serial communication, therefore the data is first multiplexed to set all the information on a single line. An analog to digital converter is required to binarize the multiplexed data to make it readable by the computer as shown in Figure 3-19.
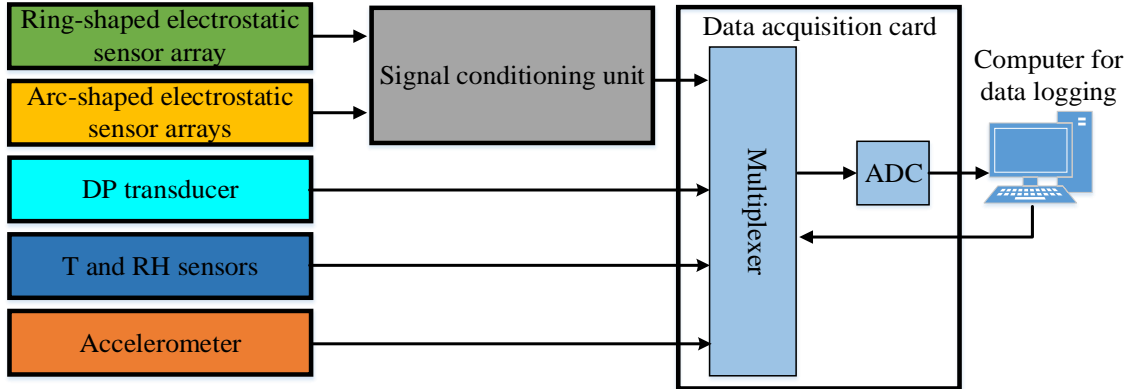
Figure 3-19: Signal conditioning combined with data acquisition

The core elements of the test rig and the connections between them are shown in Figure 3-20. Upon receiving the command from a computer-based program, the DAQ unit acquires all the post-processed sensor data from the signal conditioning unit and transfers it to the computer in the form of a matrix. The powers of two motorized devices in the test rig are also controlled through DAQ. A value, between 0 and 10, needs to be set in a computer program which will be taken by the DAQ unit to produce the respective DC signal between 0 to 10 V. VFD, on the other side, takes these 0 to 10 V signals to generate 0 to 50 Hz AC signal of desired specifications based on the parameters setting defined in the previous section.
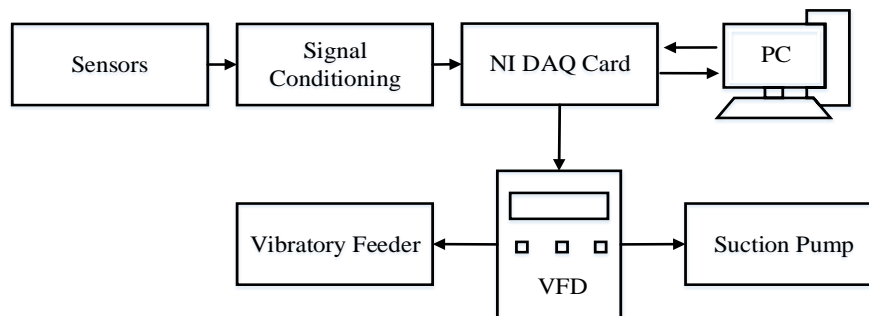
Figure 3-20: Data acquisition and control system

3.4 Considerations for the Operation of the Test Rig

3.4.1 Health and Safety Regulations

In order to run the test rig in a safe and healthy environment, a risk assessment is carried out to highlight the factors that can potentially cause physical damages to the health of a person who is running the test rig. Following measures are taken as per the health and safety regulations of University of Kent:

1) The solids moving at high air velocity can be heavily charged, therefore all the pipes and the equipment are properly grounded to avoid any sort of electric shock or explosion.

2) Running the suction pump at full power for a long period can damage the hearing capability. Therefore, an industrial noise cancellation headset is used to provide the protection from noise.

3) The size of the solids is significantly small (< 1 m), therefore at higher concentrations of solids, breathing problems can be experienced due to air being polluted by solids. The dust mask (moulded unvalved P1) is used to protect the lungs by filtering the air for breathing purposes.

4) Protective glasses are used to prevent the irritation in the eyes caused by the tiny solids in the air.

5) Long over-coats are worn to protect the clothes of the users operating the test rig.

6) Tiny solids on the floor can increase the chances of a person being slipped due to the rolling effect of solids. Therefore, a vacuum cleaner with a brush is used to wipe the solids away from the floor after each experiment.

3.4.2 Issues to be Considered for the Effective Operation of the Test Rig

The datasets, used in this thesis, were recorded at different time intervals under different experimental conditions. There are number of parameters that can influence the characteristics of the sensors. Following are the operating procedures that are defined to ensure the consistent recording of datasets from sensors:

1) **Cleaning of filter:** In order to have consistent vacuum pressure in the pipeline, it is made sure that the filter of the suction pump is clean of any solids to prevent the blockage in the motors of the suction pump before the start of each data logging session.

2) **Cleaning of pipes and storage:** Once the data logging is done for one experimental condition, all the pipes are cleaned by running the pump at maximum suction power without any solids from the feeder side. This was observed to be effective through measurement of repeatability of the results.

3) **Volume of solids in the hoper:** The hoper is the main source of solids for the vibratory feeder, therefore it is ensured to keep it full with the solids whenever the level of solids reduces down to 80% of the total storage. Owing to the gravitational effects, a lesser volume of solids in hoper can cause inconsistency for the solids that are dropped into the plate of vibratory feeder.

4) **Stirring of solids in hoper:** The outlet area of the hoper is small compared to the storage place, therefore solids are continuously stirred to avoid the blockage and to have the consistent flow of solids from hoper to the vibratory feeder. The solids are stirred manually with a long round shaped stick.

5) **Ambient conditions:** The characteristics of the electrostatic sensor can be influenced by the ambient conditions, therefore the temperature and humidity conditions in the lab are continuously monitored and kept constant during the procession of an experiment with the assistance of two commercial level air conditioning units.

6) **Data logging:** The main constituents of the test rig, including the suction pump and vibratory feeder, are controlled from the computer program via DAQ unit and VFD. Therefore, the process of data logging is started immediately with an absolute time delay after the pump and feeder are instructed to run at desired powers. The reason to add delay is to give some time to the solids to settle down with a desired solids concentration and particle velocity. For each of the iteration of experiments, the computer program varies the mass flow rate with eleven different values each with a time delay of 30 seconds. Therefore, the data is logged for 330 seconds in each of the experimental iterations.

3.4.3 Operating Conditions

The focus of this research is to measure the mass flow rate of solids under different physical factors, such as air velocity, ambient conditions and pipe orientations, therefore the test rig is run at the following operating conditions:

1) **Mass flow rate:** The mass flow rate conditions are varied by changing the magnitude of vibrations through VFD. The minimum possible frequency of VFD that can start the vibratory feeder to throw a significant volume of solids is 30 Hz. Therefore, eleven mass flow rate conditions from 3.2 g/s to 35.8 g/s are set up through frequency variations from 30 Hz to 50 Hz with a difference of 2 Hz. The data is calibrated through weiging scale and the timer.

2) **Air velocity:** In order to overcome the torque on fan blades, the suction pump, in the beginning, is provided with the maximum power from VFD. Suction power is started from 50 Hz and reduced down to 10 Hz with a gap of 5 Hz. In this way, the particle velocities are varied with nine different air velocity conditions starting from 10.1 m/s to 36.0 m/s. The reference air velocity is measured through hot-wire anemometer.

3) **Ambient conditions:** Depending upon the nature of the experiment, these conditions are either kept constant or varied in the range of 16 °C to 26 °C and 25% to 50% with the use of 2 commercial level air conditioning units.

4) **Pipe orientation:** Sensing unit is installed in a flexible orientation with a test rig. Based on the nature of the experiment, pipe orientation is either kept fixed in a horizontal position or made variable between 0° and 90°.

5) **Sampling rate:** DAQ module is capable to capture the sensor signal at a sampling rate up to 50 kHz. A higher sampling rate can bring good accuracy in the data. However, the excessive number of samples can consume substantial computation power and memory space. Therefore, the sampling rates for all the sensors are kept at a reasonable level and different from each other based on the bandwidth of the signal. As the electrostatic sensor data exist at a very high frequency, therefore the sampling rate for the electrostatic sensor is chosen to be 20 kHz compared to all the other sensors with a sampling rate of 1 kHz.

6) **Recording duration:** As the data-driven models combine several samples from sensor data to create a sample for training and testing purposes, therefore the data is recorded for 30 seconds for each combination of the test conditions mentioned above.

3.4.4 Test Materials

As the experiments are conducted in a laboratory environment, baking flour is used as test solids for health and safety reasons for all the experiments except one where two different types of particles are mixed in different ratios. The particles of amino plastic, having similar true density

as coal particles, were used to perform the experiments with the mixture of amino/biomass/air. The properties of the particles, used in the different trials, are listed in Table 3-3. An off-line digital imaging system was utilized to measure the particle size and shape. The digital imaging system to measure the particle size and shape is comprised of a USB imaging scanner and the BioMscan software developed by Robert Carter at the University of Kent [71]. The shape and size distribution of flour, amino and sawdust particles, determined through particle imaging system, is shown in Figure 3-21 to Figure 3-23, respectively. The relative permittivity constant was sourced from [72] and the true density was measured through weiging scale and the volume measuring jar.

Table 3-3: Properties of test materials

	Flour	Amino plastic	Sawdust
Aspect ratio	1:1 – 1:3	1:1 – 1:2.5	1:1 – 1:6.5
Particle size (μm)	1800 - 2000	500 - 1000	1400 - 4750
True density (kg/m^3)	596	1500	450
Permittivity at 20 °C and RH 30%	2.5	3.2	2
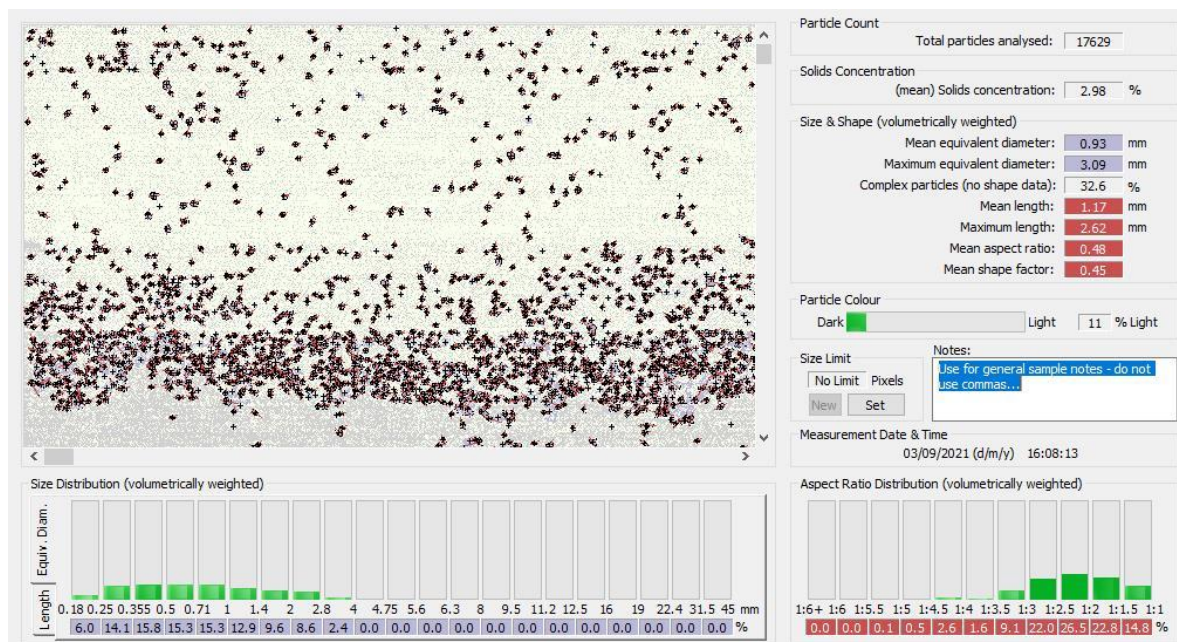
Figure 3-21: Particle size distribution of flour particles using particle imaging system
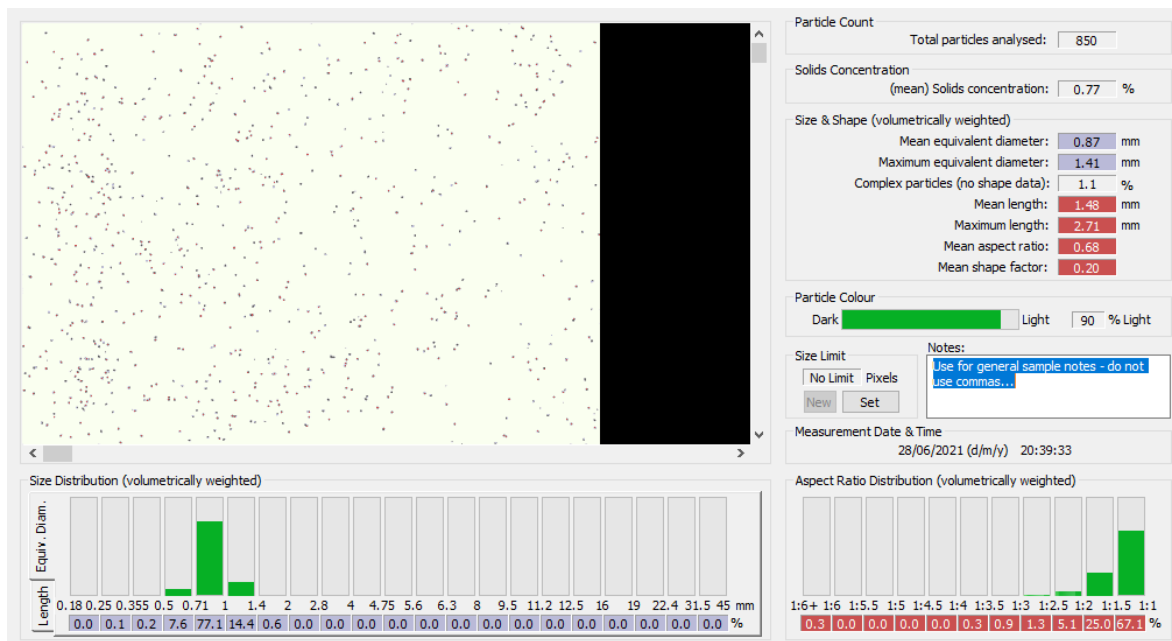
Figure 3-22: Particle size distribution of amino particles using particle imaging system
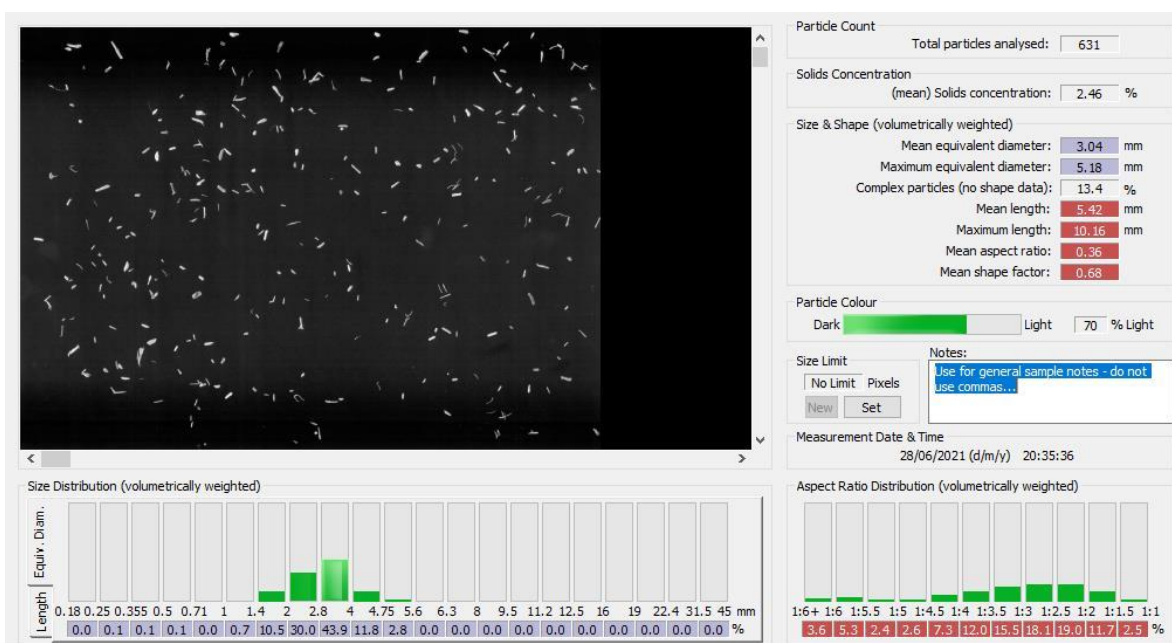
Figure 3-23: Particle size distribution of sawdust particles using particle imaging system

3.5 Programme of Operation

3.5.1 Data Acquisition and Control

Figure 3-24 shows the block diagram of the program that is used to acquire and collect the sensor data. The acquisition of data from all the sensor and the control of VFDs is done through high speed NI DAQ unit in conjunction with program written in MATLAB 2020b. After connecting all the devices with DAQ unit, the MATLAB program also requires the assignment of input and output channels to the sensors and VFDs. The designated channels are set either input or output with an estimated voltage range 0 to 10 V. Data acquisition program captures the data from all the sensors in one single attempt for a specific time that is pre-defined by the user. After setting a new frequency to control the suction power or the magnitude of vibrations, the test rig is run idly for 30 seconds to stabilize the gas-solids flow. Firstly, the program records the data with all the pre-set mass flow rates of solids and then it repeats the process with a different physical parameter, for example air velocity, pipe orientation and ambient condition. At the end of the program, all the useful variables are saved in a MATLAB readable format.
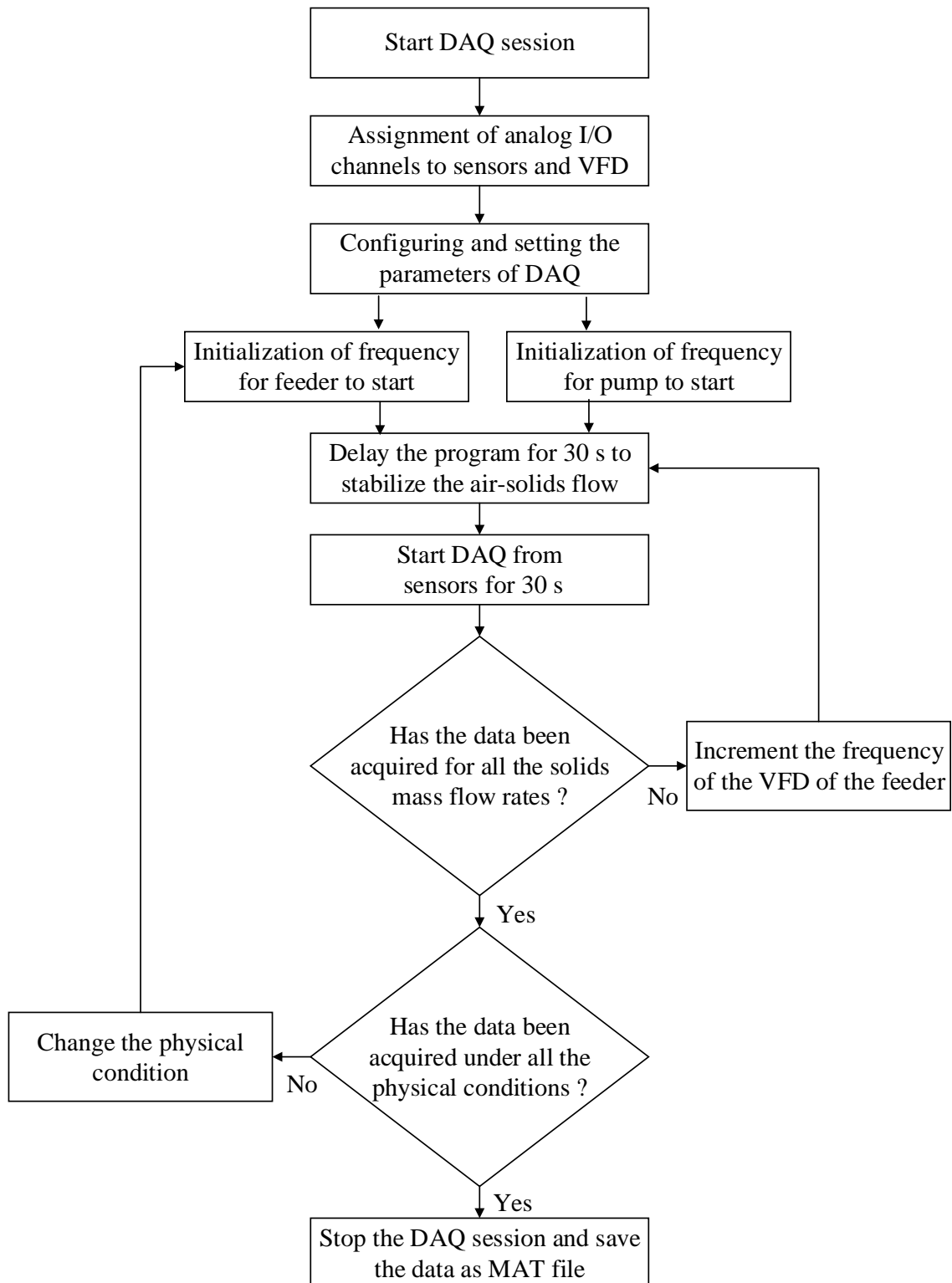
Figure 3-24: Block diagram of the program of data acquisition

3.5.2 Software for Mass Flow Rate Measurement

Once the measurement strategy is developed and validated with the collected data, the measurements are produced in real time by following the steps shown in Figure 3-25. The start button triggers the program to initialize the DAQ session by assigning the channels to input and output devices. The DAQ parameters, including sampling rate and ranges of the signals, are pre-defined by the user. Upon receiving the user-defined window size, DAQ unit collects the samples from the sensors for the same time duration as the window size. The number of samples collected is the product of sampling rate and the window size. The measurement strategy (either based on direct method that will be presented in Chapter 4 or based on data-driven modelling that will be explained in Chapter 5 and 6) is then used in the program to produce the measurements in real time. The whole process repeats until the stopping call is received from user.
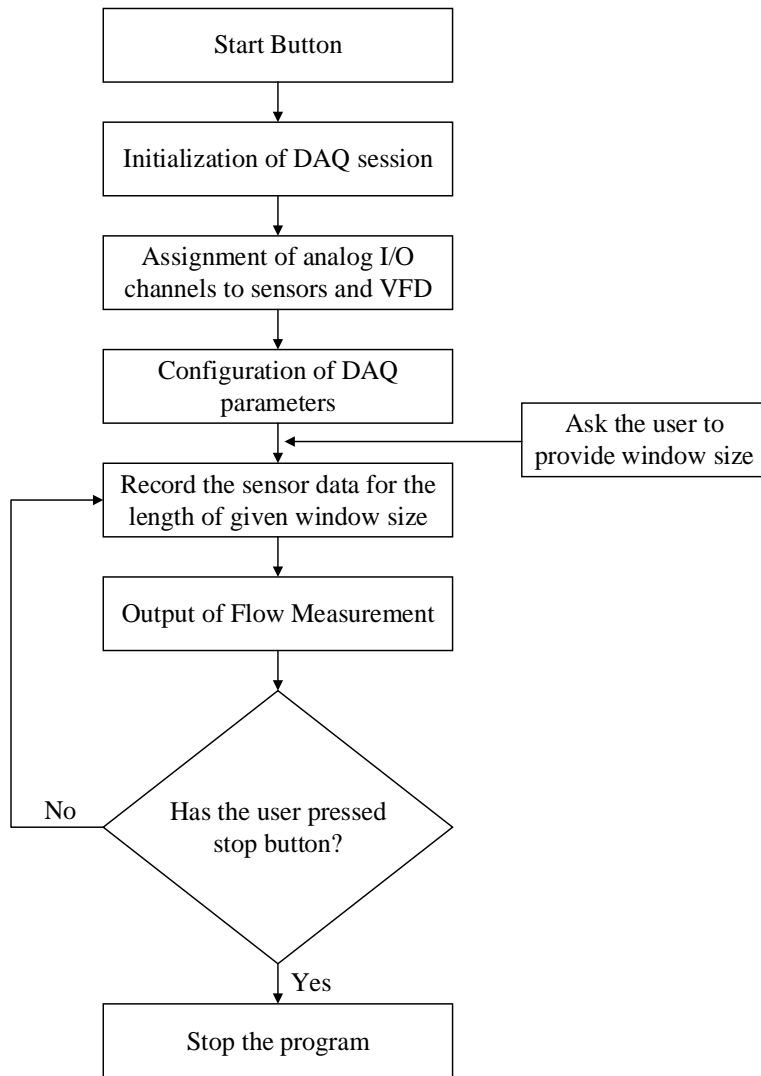
Figure 3-25: Implementation of software for mass flow rate measurement

3.6 Summary

This chapter details the reconfiguration of the test rig and the test conditions used to carry out all the experimental work. Firstly, the major components of the test rig with a fixed pipe orientation are discussed. Then the physical design and the implementation of the test rig with the variable orientation of the sensing unit is described in detail. The design of the sensing unit along with the need for multi-modal sensing including an array of electrostatic sensors, DP transducer, temperature, and humidity sensor, and accelerometer is presented. Since, in order to maintain consistency in the data collection process, it is necessary to run the different mass flow rate conditions continuously with a gap of specific recording interval. For this purpose, the parametric configuration of VFDs is described to control the feeder and pump through a computer program. In order to bring the signals from ring and arc-shaped electrostatic sensors to a computer-readable level and make them free from noise, the stages of the signal conditioning circuit are also reported. As the proposed methodology for measurement is based on the data-driven models, therefore the training and the test conditions are defined to collect the data by following the rig operating procedures.

Chapter 4 Measurement of Velocity and Mass Flow Rate of Solids with Conventional Techniques

4.1 Introduction

Velocity and mass flow rate measurement of pneumatically conveyed solids is essential for the effective control of many industrial procedures. Electrostatic sensors have been widely used for the measurement of particle velocity and solids mass flow rate. However, the magnitude of the electrostatic charge depends on various physical factors and the impact of different pipe orientations and ambient conditions on the measurement of particle velocity and the solids mass flow rate is still unknown. This chapter presents the experimental outcomes of velocity and mass flow rate measurement of solids under different physical factors through electrostatic sensors and the inferential technique. The electrostatic sensing system is comprised of an array of ring and four arrays of arc-shaped electrodes to investigate the averaged and localized flow parameters, respectively (as presented in Chapter 3). The particle velocity is measured, in the range of 10.1 m/s to 36 m/s, through suitable combinations of upstream and downstream electrodes using cross-correlation based signal processing technique. Performance of individual and different combinations of electrostatic sensors is assessed to optimize the number of sensors for mass flow rate measurement. One of the key issue that is addressed in this chapter is the determination of the parameters of the direct method of mass flow rate measurement through firefly optimisation algorithm under different flow conditions. The inferential technique is tested with the mass flow rate of solids in the range of 3.2 g/s to 35.8 g/s under different physical factors, including air velocities, pipe orientations, ambient temperatures and ambient relative humidity (RH) ranging from 10.1 m/s to 36 m/s, 0° to 90°, 16 °C to 26 °C and 25% to 50%, respectively.

4.2 Electrostatic Sensing System

The schematic of the sensing unit used for the measurement of velocity and mass flow rate of solids is shown in Figure 4-1. The 50 mm bore sensing unit is comprised of an array of ring-shaped and four arrays of arc-shaped electrodes, where each array includes four ring/arc shaped electrodes non-intrusively flush-mounted in the pipe. Ring-shaped electrodes are mainly installed to acquire the averaged electrostatic charge of the entire cross-section of the pipe. However, the arc-shaped electrodes are used to obtain the localized electrostatic charge at the top, front, back and the bottom of the pipe (as explained in Chapter 3).
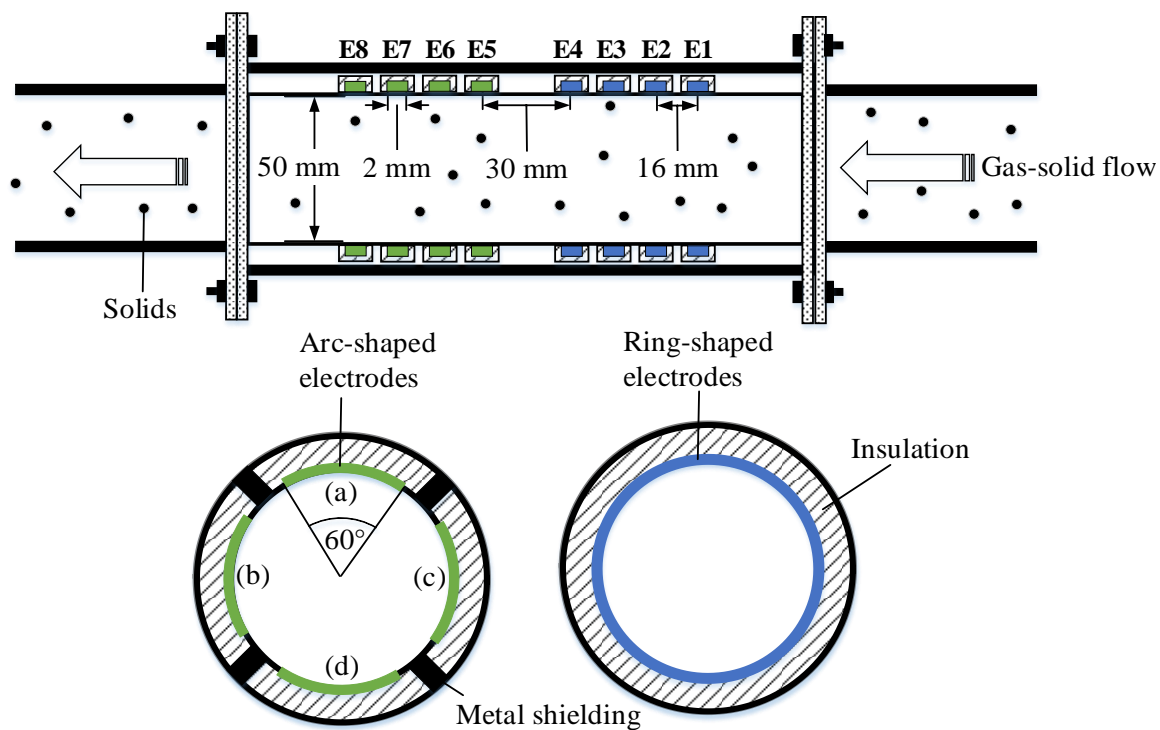
Figure 4-1: Schematic of sensing unit

4.3 Direct Method of Velocity Measurement

Although there are many methods for the measurement of velocity of pneumatically conveyed solids, the cross-correlation method remains the most practical and effective. Therefore, the cross-correlation based signal processing technique is applied on the signals of electrostatic sensors to measure the particle velocity [16]. The signals from two identical electrostatic sensors, with a predetermined axial distance between them, are used to apply the cross-correlation technique. The

distance between the upstream and the downstream electrodes is important because a small separation produces a higher correlation coefficient but requires fast data acquisition and processing, while a larger separation produces a lower correlation coefficient [73]. The distance between the upstream and downstream sensors was set to 16 mm for both ring and arc-shaped electrodes. Having a high-speed data acquisition card, however, a lower value of axial distance can also be used. With the known axial distance between upstream and the downstream electrodes, velocity of the solids moving inside the pneumatic pipe can be measured from

$$v_c = \frac{L}{\tau_m} \tag{4-1}$$

Where L defines the axial distance between upstream and the downstream electrodes and τ_m gives the time solids take to move from upstream to the downstream electrode which can be calculated by locating the dominant peak of the cross-correlation function given by

$$R_{xy}[m] = \frac{1}{N} \sum_{n=1}^{N} x[n]y[n+m] \tag{4-2}$$

Where x and y are the signals from upstream and the downstream electrodes, respectively and N is the total number of samples obtained from the sensors.

Based on the individual velocities measured from each array of the electrostatic sensors, the fused average particle velocity is calculated by [16]

$$v = \frac{r_{12}v_{12} + r_{23}v_{23} + r_{34}v_{34} + r_{13}v_{13} + r_{14}v_{14} + r_{24}v_{24}}{r_{12} + r_{23} + r_{34} + r_{13} + r_{14} + r_{24}} \tag{3}$$

Where r_{12}, r_{23}, r_{34}, r_{13}, r_{14} and r_{24} are the cross-correlation coefficients, v_{12}, v_{23}, v_{34}, v_{13}, v_{14}, v_{24} are the individual particle velocities using individual electrodes in an array as shown in Figure 4-1.

4.4 Direct Method of Mass Flow Rate Measurement

The mass flow rate of solids is measured through the signals from array of ring-shaped electrostatic sensors. The principle of mass flow rate measurement of solids is based on the inferential approach that can be derived from [16]

$$q_{m,s} = A\rho_s v_c \beta_s \qquad (4)$$

where A is the cross-section area of the pipe, ρ_s is the true density of the solids, v_c is the particle velocity of the solids and β_s is the volumetric concentration of the solids. The root mean square (RMS) of the signal from electrostatic sensor provides the volumetric concentration of the solids, therefore the concentration is measured by taking the average of the RMS values of the signals from four ring-shaped electrostatic sensors [74, 3]. The mass flow rate of solids is measured from the RMS of electrostatic signal and the cross-sectionally averaged velocity by

$$q_{m,s} = a v_c^b A_{rms} \qquad (5)$$

where a and b are the parameters to be tuned with the help of reference data. The value of a relates the properties of the solids, whereas the value of b shows the dependence of the RMS charge value on the particle velocity v_c, respectively. Inappropriate choice of the parameters can significantly impact the accuracy of measured mass flow rate of solids. Furthermore, a little change in the physical conditions including air velocity, pipe orientation, ambient conditions, solids size/shape etc., can significantly affect the RMS charge level of electrostatic signal and hence the values of the parameters a and b. A_{rms} is the averaged RMS of the signals from four ring-shaped electrostatic sensors and it can be computed as

$$A_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} x[i]^2} \qquad (6)$$

where x is the signal from the electrostatic sensor and N is the total number of samples.

4.5 Determination of Parameters

As described in Section 4.4, parameters a and b defines the relationship between the mass flow rate, velocity and concentration of the solids. Unsuitable values of the parameters can negatively influence the accuracy of the measured mass flow rate of solids. Therefore, an optimisation algorithm is needed to regulate the values of the parameters. It should be stressed that the signals of the electrostatic sensors may exhibit different characteristics under different physical factors, such as air velocities, pipe orientations, ambient conditions etc., that may introduce large relative

error in mass flow rate measurement of solids. Therefore, the parameters a and b are selected in a way to keep the relative error optimized by covering as many physical factors as possible. The following steps are implemented to find the optimal value of the parameters a and b.

1) Obtain the sensor data of different mass flow rates taken under variety of physical conditions.
2) Apply the optimization algorithm to find the optimized values of parameters a and b that can produce the measurements with optimized value of relative error for all the physical conditions.
3) Repeat (2) if the physical condition changes.

The firefly algorithm (FA) is considered to be very competitive with multi-modal and global/local optimisation problems among the modern metaheuristics. Yang first suggested the FA in 2008 [75]. This algorithm employed the firefly swarm behavior to create a multi-optimal functional optimisation algorithm. Precisely, it uses the concept of how the brightness of individual fireflies draw the whole swarm together and a randomness factor to encourage exploration of the solution space. Figure 4-2 depicts the steps that are required to implement the FA for the desired purpose.
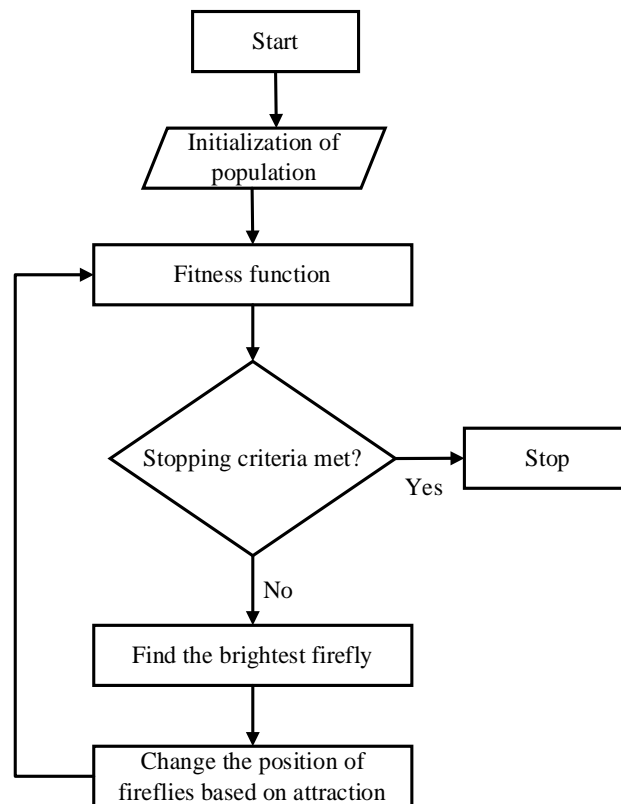
Figure 4-2: Flow chart of FA

The entire optimisation process begins with the random initialization of the population, which includes the parameters a and b. The fitness function, which is used to calculate the fitness value of each set of parameters, is given by

$$I_{fitness} = 6.4 \times 10^{10} e^{-11 \sum_{i=1}^{N_c} a_i e_i} + 0.7 \qquad \text{(4-7)}$$

The fitness function produces the value based on the error between the measured and the reference mass flow rate of solids. All the other constants in the fitness function are adjusted in a way to keep the fitness value in between 0 and 1. Each firefly makes a step towards the firefly with the highest intensity value, based on the intensity values calculated by the fitness function. The position of the ith firefly that represents the value of kth parameter of the model can be updated through j^{th} firefly of highest intensity by

$$x_{ik} = x_{ik} + \beta_o e^{-\gamma \cdot r_{ij}} (x_{ik} - x_{jk}) + \alpha S_k (rand_{ik} - 0.5) \qquad \text{(4-8)}$$

where r_{ij} can be computed from,

$$r_{ij} = \sqrt{\sum_{k=1}^{N_p} (x_{ik} - x_{jk})^2} \qquad \text{(4-9)}$$

where x_{ik} and x_{jk} are the positions of ith and j^{th} fireflies which represent the values of kth parameter.

The entire algorithm continues updating the values of parameters a and b until either the maximum number of iterations is achieved or the intensity of the firefly exceeds the threshold.

4.6 Experimental Conditions

Experiments were carried out at eleven mass flow rates where each condition were also run with variety of air velocities, pipe angles and ambient conditions as shown in Table 4-1. For the purpose of analyzing the effect of one particular experimental condition on the mass flow rate measurement, all the other physical factors were kept constant as underlined and highlighted in Table 4-1. For each pair of the mass flow rate with other experimental condition, a data is recorded for 30 seconds at the sampling rate of 20 kHz.

Table 4-1: Experimental conditions

Experimental conditions	Value
Mass flow rate (g/s)	3.2, 5.4, 7.7, 8.6, 11.3, 14, 17.6, 21.4, 25.7, 30.8, 35.8
Air velocity (m/s)	10.1, 14.5, 18.6, **22.1**, 25.3, 28.2, 31.1, 33.0, 36.0
Pipe angles (°)	**0**, 10, 20, 30, 40, 50, 60, 70, 80, 90
Ambient temperature (°C)	**16**, 18, 20, 22, 24, 26
Relative humidity (%)	**25**, 30, 35, 40, 45, 50

Figure 4-3 shows the nine air velocities where each mass flow rate condition was tested. The experiments were repeated by keeping the pipe at an angle of 0° and by setting the ambient temperature and relative humidity as 16 °C and 25%, respectively, as explained in Chapter 3. It has to be noted that at the higher mass flow rate of solids, the lower air velocities cannot be run in the given test rig due to imbalance ratio of the solids concentration and air that can cause the blockage of the solids in the pipe. Figure 4-4 shows the combinations of mass flow rates tested with various pipe angles. For all the experiments with different pipe angles, velocity of solids was kept constant at 22.1 m/s. Furthermore, the ambient temperature and relative humidity were maintained at 16 °C and 25%, respectively. Figure 4-5 shows the different ambient temperatures where the data for different mass flow rates of solids were recorded. The data for the mass flow rates were also gathered under six different ambient RH as shown in Figure 4-6, while the air velocity and pipe angle were kept fixed at 22.1 m/s and 0°, respectively. Table 4-2 shows the amount of moisture content that the flour has absorbed under different ambient RH measured with heating of flour for 1 hour and measuring the weight before and after the heating process. In common practice, it is hard to control and maintain such large range of ambient conditions under industrial environmental conditions. However, all the data for this experiment were collected in a laboratory where the ambient conditions were controlled using 2 commercial level air conditioning units.
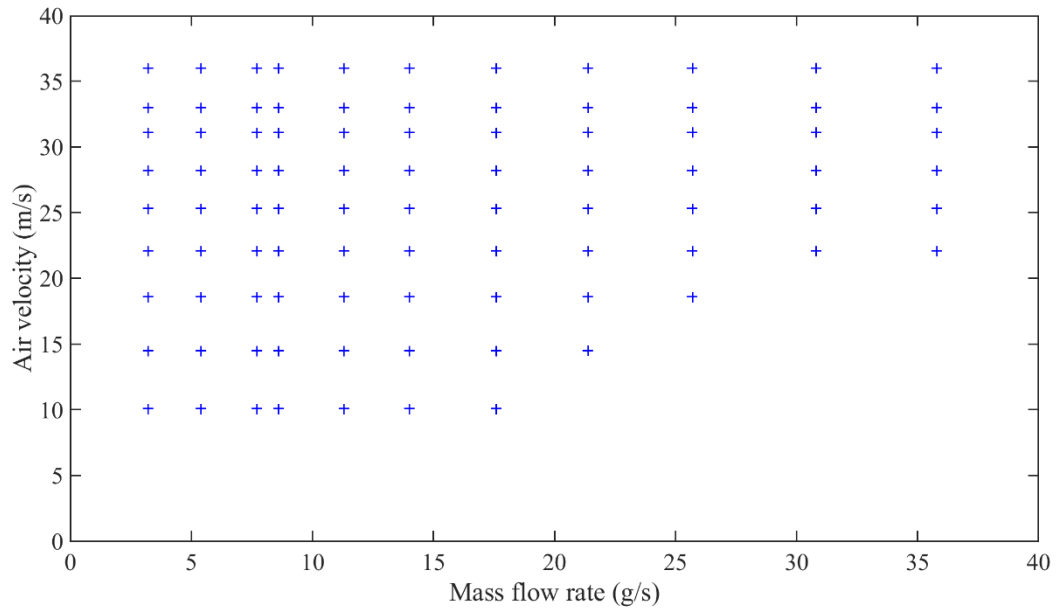
Figure 4-3: Test conditions for mass flow rate measurement under different air velocities
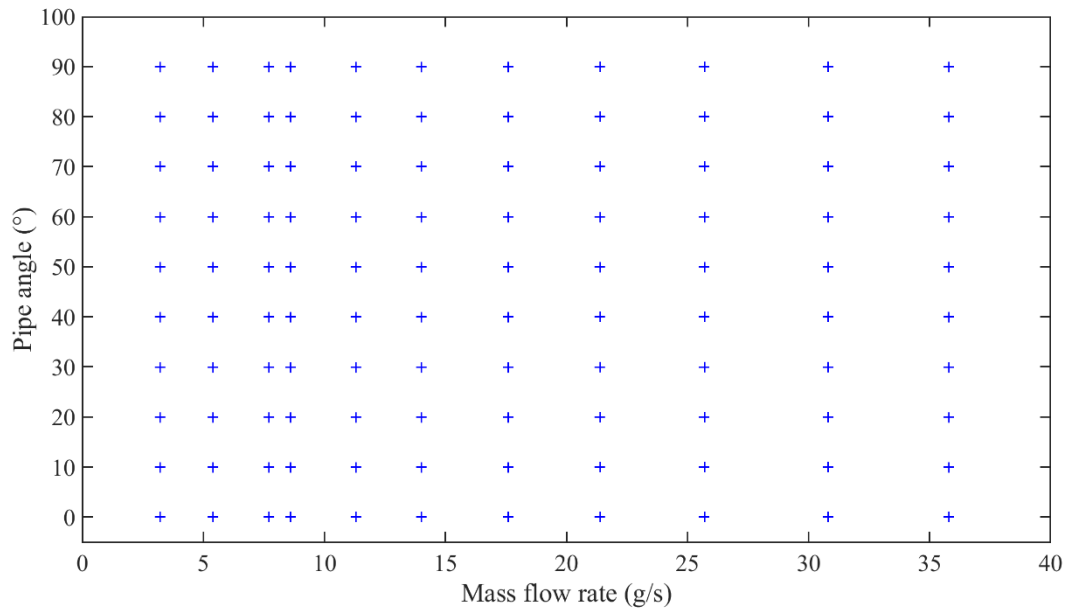
Figure 4-4: Test conditions for mass flow rate measurement under different pipe orientations

90
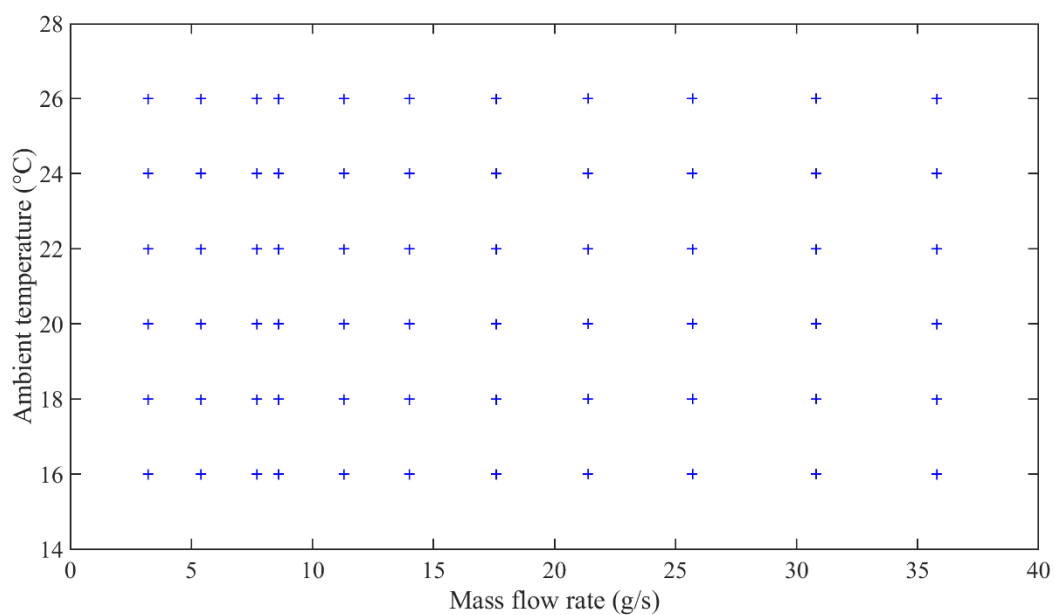
Figure 4-5: Test conditions for mass flow rate measurement under different ambient temperatures and constant RH 25%
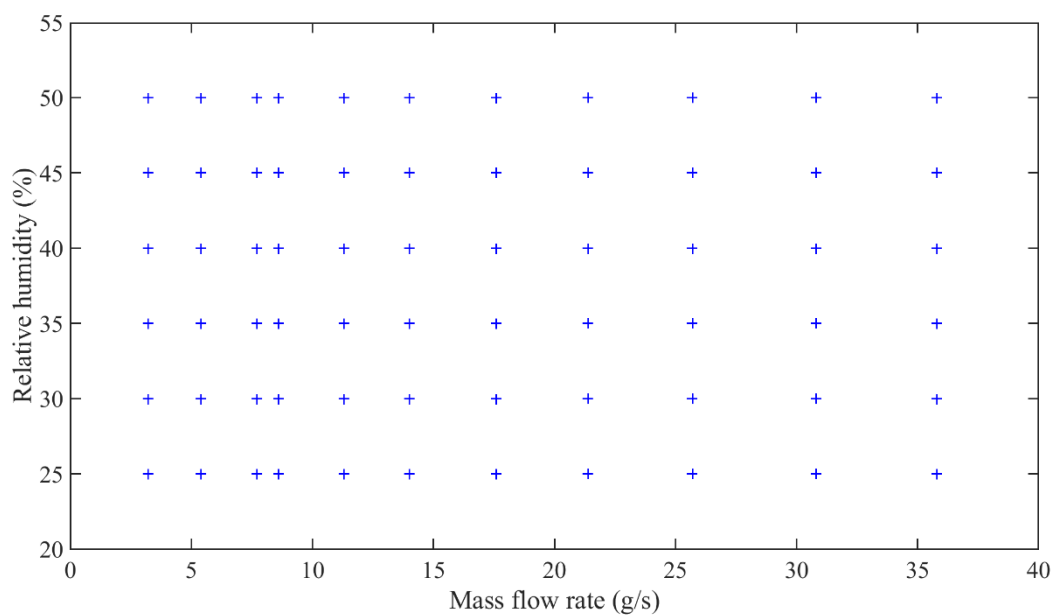
Figure 4-6: Test conditions for mass flow rate measurement under different RH and constant ambient temperature 16 °C

91

Table 4-2: Amount of moisture content in flour under different ambient RH measured with heating of flour for 1 hour

Ambient RH (%)	Weight of flour before heating (kg)	Weight of flour after heating (kg)	Moisture in flour (%)
25	2	1.975	1.26
30	2	1.971	1.47
35	2	1.965	1.78
40	2	1.960	2.04
45	2	1.953	2.41
50	2	1.946	2.77

4.7 Measurement of Particle Velocity

4.7.1 Analysis of the Signals from Electrostatic Sensors

Figure 4-7 shows the typical upstream and the downstream signals from ring-shaped electrostatic sensors E1 and E2 (as shown in Figure 4-1), respectively. The signals are recorded with mass flow rate of 11.3 g/s and the air velocity of 22.1 m/s. Both of these signals look similar to a higher extent with a specific time delay that the solids take to move from upstream electrode to downstream. Figure 4-8 illustrates the output of cross-correlation function when applied on given upstream and the downstream electrostatic signals. The peak of the function is representing the time delay between upstream and the downstream signals which can be used to calculate the cross-correlation solids velocity.
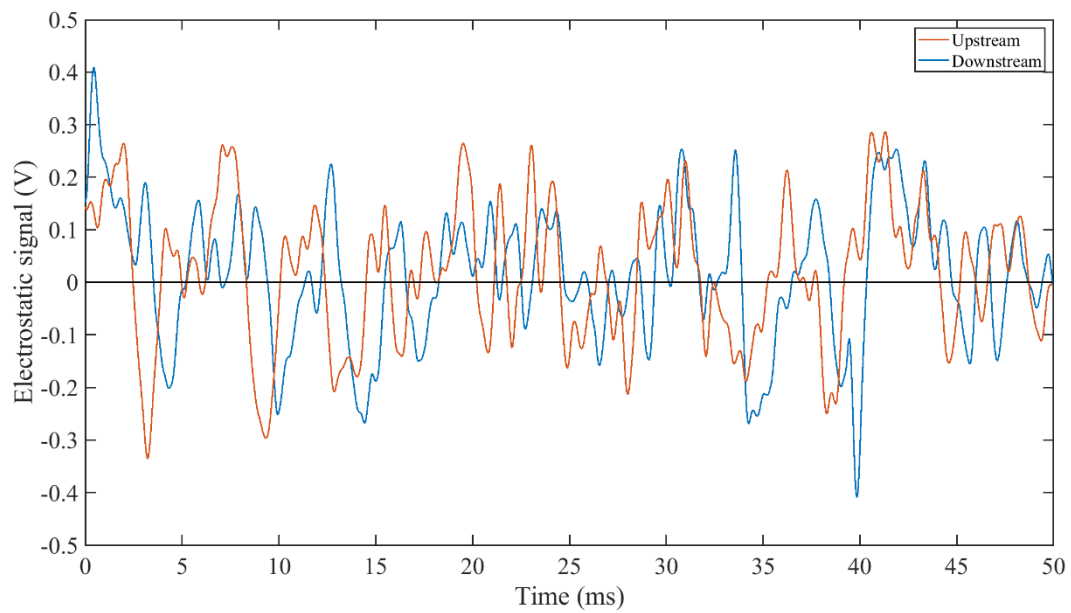
Figure 4-7: Upstream and Downstream electrostatic signals from E1 and E2, respectively
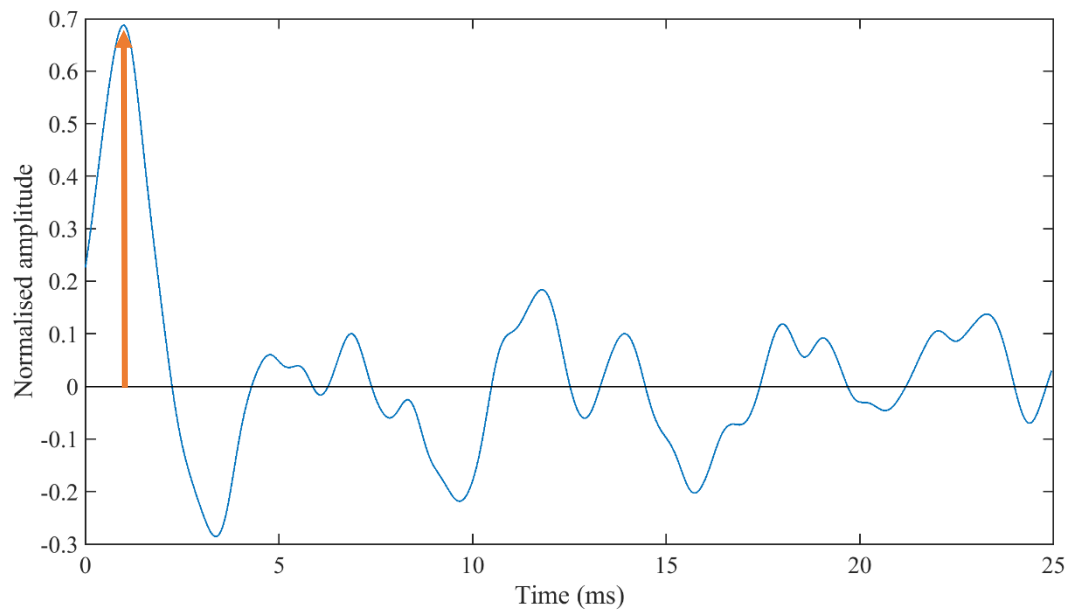
Figure 4-8: Resulting cross-correlation function applied on the upstream and downstream

sensor signals

93

4.7.2 Choice of Suitable Pairs of Upstream/Downstream Electrodes

The normalized root mean square error (NRMSE) is used to evaluate the importance of different pairs of upstream/downstream electrodes for the measurement of particle velocity as shown in Figure 4-9. Each bar in the plot represents the averaged NRMSE of nine measured particle velocities as listed in Table 4-1. The solids movement inside the pipe is highly inhomogeneous and irregular. The magnitude of the signal from one electrostatic sensor can slightly differ from another largely spaced electrostatic sensor. Therefore, the NRMSE of cross-correlation particle velocity is increasing proportionally by selecting the upstream and downstream pairs of electrodes with larger spacing between them. The error bars show the minimum and the maximum error in the cross-correlation particle velocity with each combination of upstream and downstream electrodes. The particle velocity is also measured with the combination of different electrodes to perform the data fusion. Table 4-3 shows the combination of different sensors and their associated NRMSE for the measurement of particle velocity. The combinations are started from the pair of upstream/downstream electrodes that gives the best NRMSE as presented in Figure 4-9. In this process of combining the different sensors, a new pair of upstream/downstream electrodes is only selected if it causes a drop in NRMSE than the previous combination. Meanwhile, none of the sensor combination has resulted in decreasing the NRMSE due to their inefficiency for the measurement of a number of particle velocities. Figure 4-10 shows the graphical representation of NRMSE and the standard deviation of measurement error for each of the sensor combination.
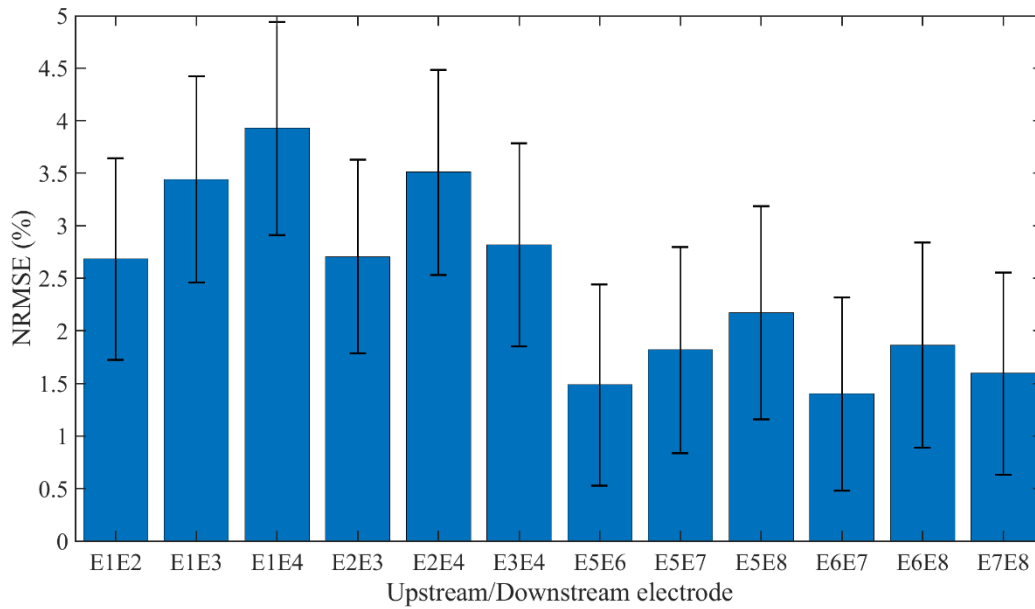
Figure 4-9: Analysis of NRMSE with different combinations of upstream and downstream electrodes

Table 4-3: Combination of sensors for data fusion and their resulting NRMSE for the measured particle velocity

Combination name	Sensor combination	NRMSE (%)
i	E5E6	1.48
ii	E5E6, E6E7	1.49
iii	E5E6, E7E8	1.51
iv	E5E6, E5E7	1.52
v	E5E6, E6E8	1.52
vi	E5E6, E5E8	1.53
vii	E5E6, E1E2	2.01
viii	E5E6, E2E3	2.01
ix	E5E6, E3E4	2.01
x	E5E6, E1E3	2.32
xi	E5E6, E2E4	2.32
xii	E5E6, E1E4	2.33
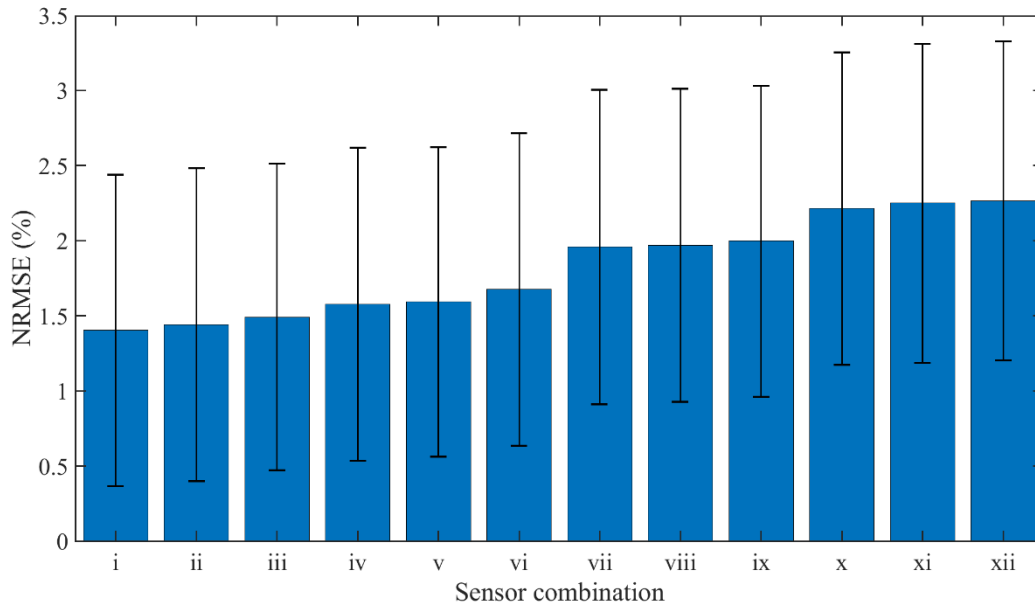
Figure 4-10: Analysis of NRMSE of the measured particle velocity with sensor fusion

4.7.3 Measured Particle Velocity in Horizontal Pipe Orientation

Figure 4-11 shows the measured velocity results with the fusion of data from all the consecutive pairs of ring-shaped electrostatic sensors and cross-correlation based signal processing technique [16]. Measurements are repeated for six different mass flow rate conditions to observe the impact of higher solids concentrations on particle velocity. Straight line represents the reference where the velocity of solids is expected to be a little lower than the actual velocity of air. As the cross-correlation technique of velocity measurement is related with the time delay between upstream and the downstream signals, therefore all the disordered variations in the magnitude of electrostatic signals which occur due to different mass flow rate conditions, have not significantly affected the velocity measurements. However, an imbalance mixture of gas-solid, caused by high mass flow rate and low air velocity conditions, has affected the measurements at lower velocity conditions as shown in Figure 4-11.

Figure 4-12 shows the normalized standard deviation (NSTD) associated with each velocity condition that describes the repeatability of the velocity measurements. Lower velocity conditions under lower mass flow rates of solids exhibit the good repeatability. However, a significant rise can be seen for the higher particle velocities measured with lower mass flow rates due to supplementary dilute mixture of solid and gas. Furthermore, higher velocity conditions with higher

mass flow rate values formulate a balance combination of gas-solid mixture that can be processed easily by the sensing system to produce accurate and more repeatable measurements. Likewise, lower velocity conditions at higher mass flow rate values do not produce promising results owing to the reason that old solids do not get enough time to leave pneumatic pipeline whereas the new solids keep coming in and start settling around the electrodes. Generally, the uncertainty is higher for the measurements taken at higher mass flow rate conditions and this is due to inhomogeneous gas-solid mixture at higher mass flow rates.

Figure 4-13 shows the relative discrepancy between velocity of air and particle velocity. A higher mass flow rate causes more friction due to gravitational effect and hence a higher relative discrepancy can be observed.
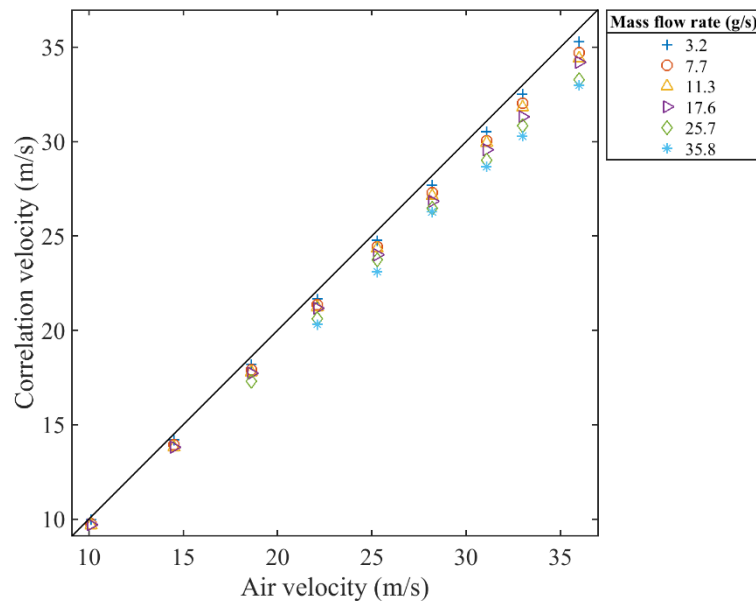
Figure 4-11: Measurement of particle velocity with best combinations of ring-shaped electrodes under horizontal pipe orientation

97
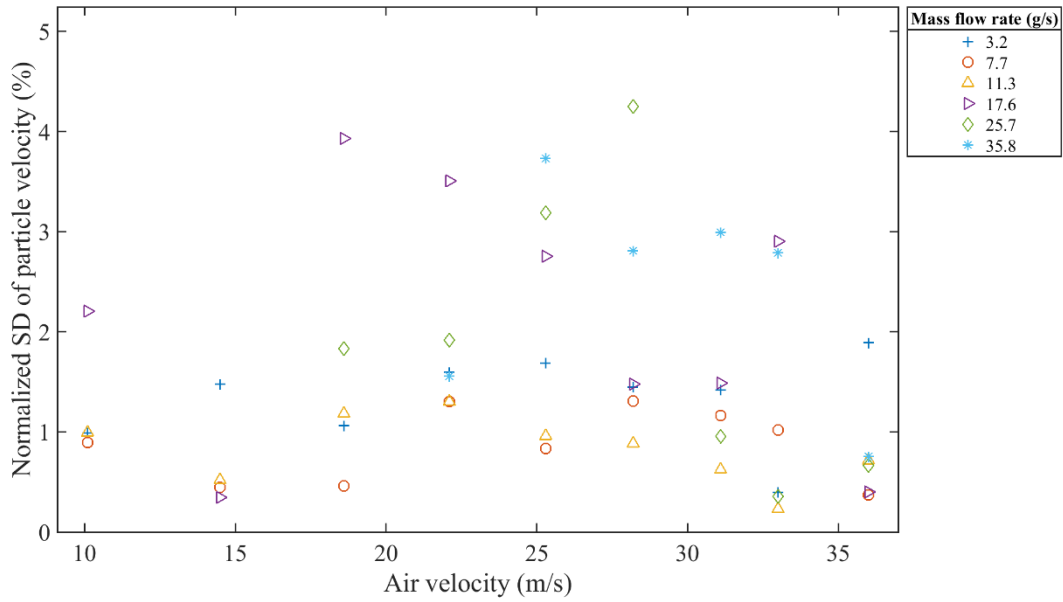
Figure 4-12: NSTD in measured particle velocity with best combinations of ring-shaped electrodes under horizontal pipe orientation
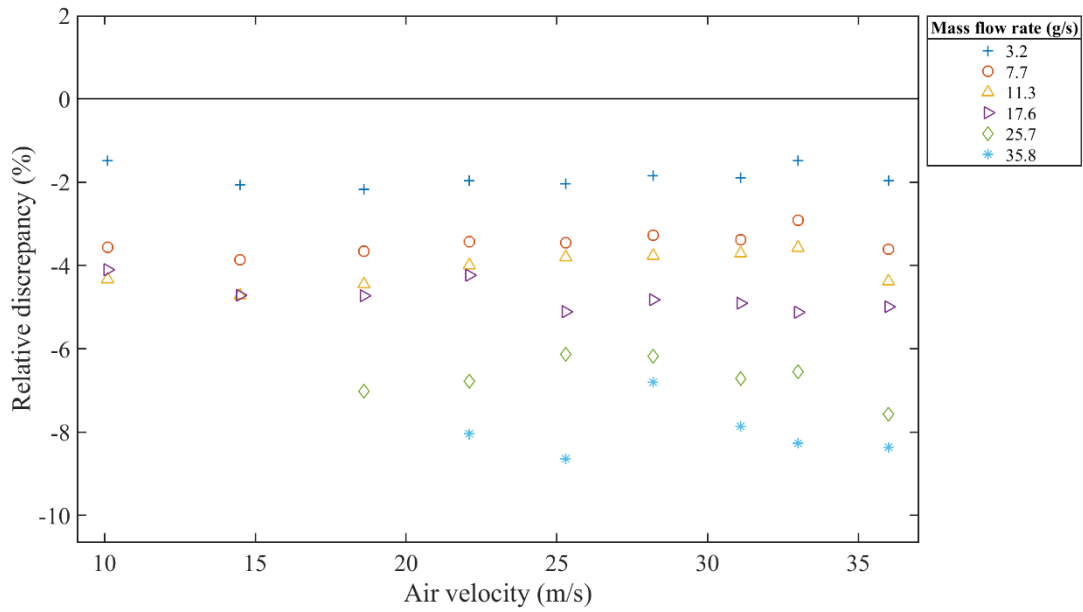
Figure 4-13: Relative discrepancy in measured particle velocity with best combinations of ring-shaped electrodes under horizontal pipe orientation

Figure 4-14 shows the localized particle velocity at the top, front, back and bottom of the pipe measured with four arrays of arc-shaped electrostatic sensors. All the velocity measurements are

reliable in terms of repeatability which is evaluated through NSTD as can be seen in Figure 4-15. The difference between the NSTD of the measured particle velocity through front and back of the pipe could be because of the performance of front and back electrostatic sensors. In the horizontal pipe orientation, a higher volume of solids concentrates around the bottom of the pipe due to gravitation effect and owing to this reason higher relative discrepancy can be seen between the velocity of air and solids flowing through the bottom of the pipe as shown in Figure 4-16. However, the solids flowing through the front and back of the pipe generate less friction with the pipe and hence lower discrepancy between the velocity of air and solids. The relative discrepancy for the solids moving along the top of the pipe is moderate due to very low solids concentration with higher solids-wall friction.
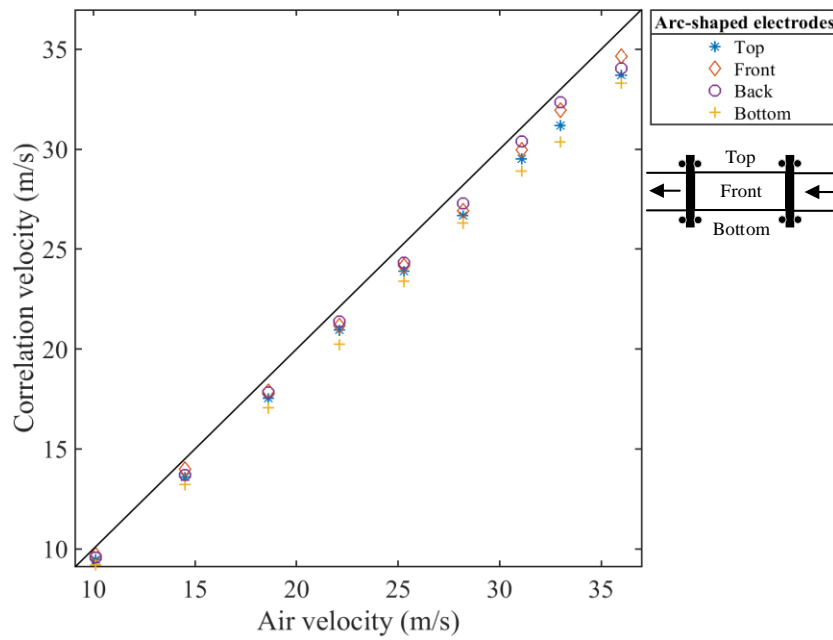
Figure 4-14: Measurement of particle velocity with best combinations of arc-shaped electrodes under horizontal pipe orientation

99
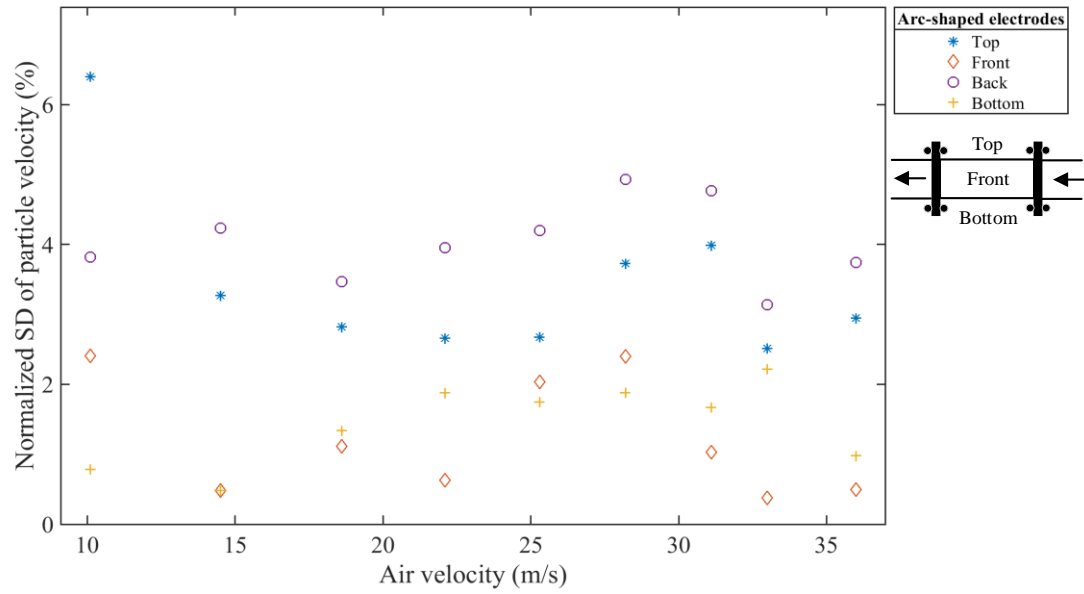
Figure 4-15: NSTD in measured particle velocity with best combinations of arc-shaped electrodes under horizontal pipe orientation
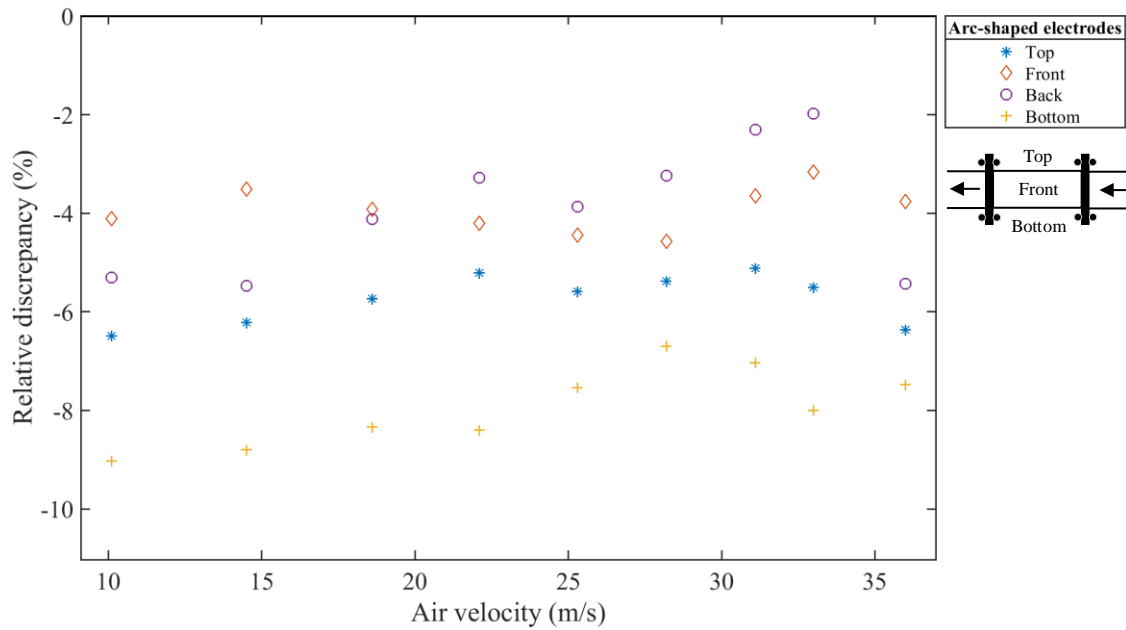
Figure 4-16: Relative discrepancy in measured particle velocity with best combinations of arc-shaped electrodes under horizontal pipe orientation

4.7.4 Measured Particle Velocity in Vertical Pipe Orientation

Figure 4-17 shows the measured particle velocities using the combinations of consecutively placed upstream and the downstream ring-shaped electrodes under vertical pipe orientation. Since the ring-shaped electrodes are used to obtain the averaged charge of the entire cross-section of pipe. Therefore, there is no significant variation seen in the averaged particle velocity measurement under vertical pipe orientations. However, compared to horizontal pipe orientation, measured particle velocities under vertical pipe orientations are more repeatable due to consistent movement of solids in vertical section of the pipe as shown in Figure 4-18. The relative discrepancy between the velocity of air and the solids has also not been significantly influenced under vertical pipe orientation as shown in Figure 4-19.

Figure 4-20 and Figure 4-21 shows the measurement of particle velocity and the NSTD, respectively, using the combinations of consecutively placed upstream and the downstream arc-shaped electrodes in top, front, back and the bottom of the pipe under vertical pipe orientation. Orientation of the pipe can adversely affect the localized behavior of solids. Higher concentration of the solids moving across the horizontal section of pneumatic pipeline hit with the wall of pipe bend and start moving along the left area of the vertical pipe while experiencing the high friction with the wall. Owing to this frictional effect, a higher relative discrepancy can be observed between the velocity of air and the solids moving along the left side of the wall as shown in Figure 4-22. Velocity of the solids moving across the front and back of the pipe generate less friction and hence stays nearly around the velocity of air.
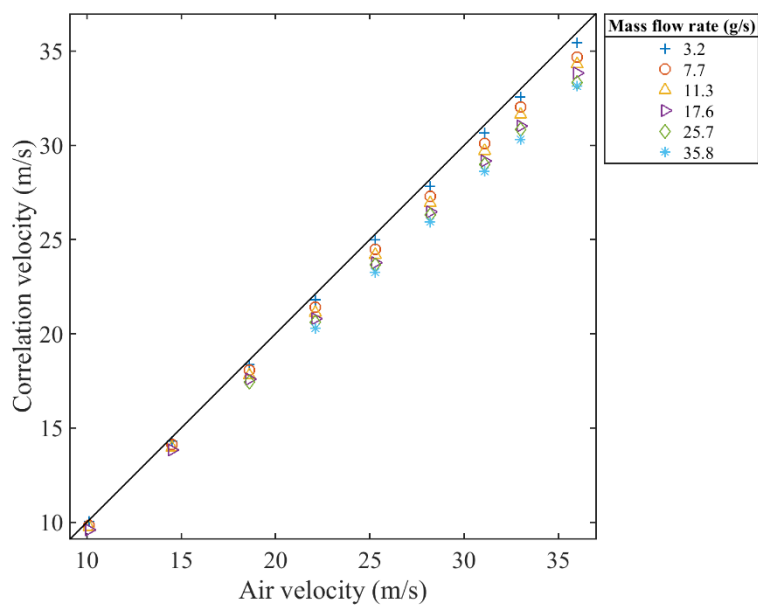
Figure 4-17: Measurement of particle velocity with best combinations of ring-shaped electrodes under vertical pipe orientation
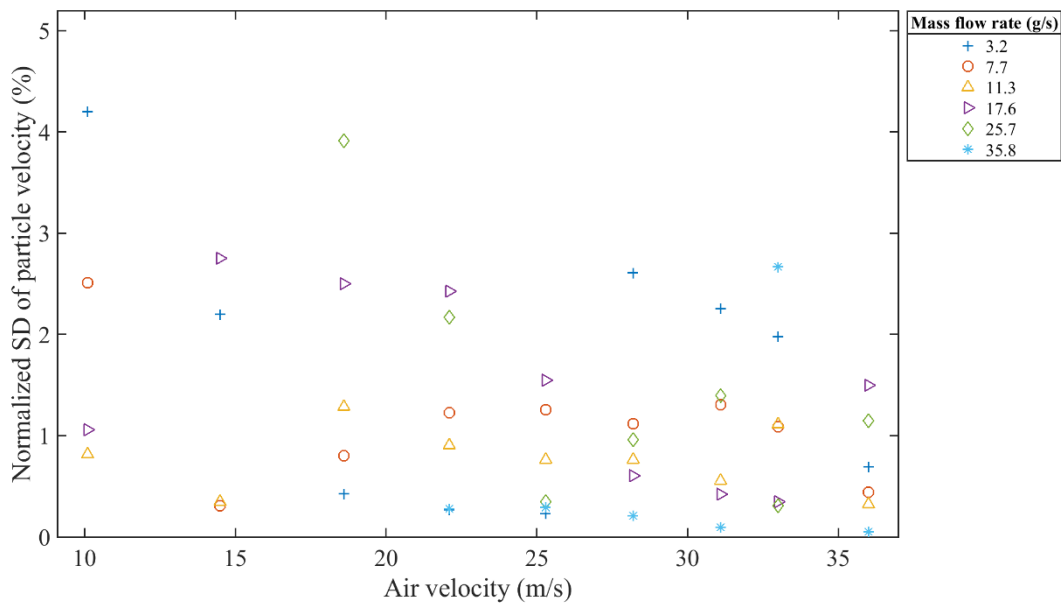
Figure 4-18: NSTD in measured particle velocity with best combinations of ring-shaped electrodes under vertical pipe orientation
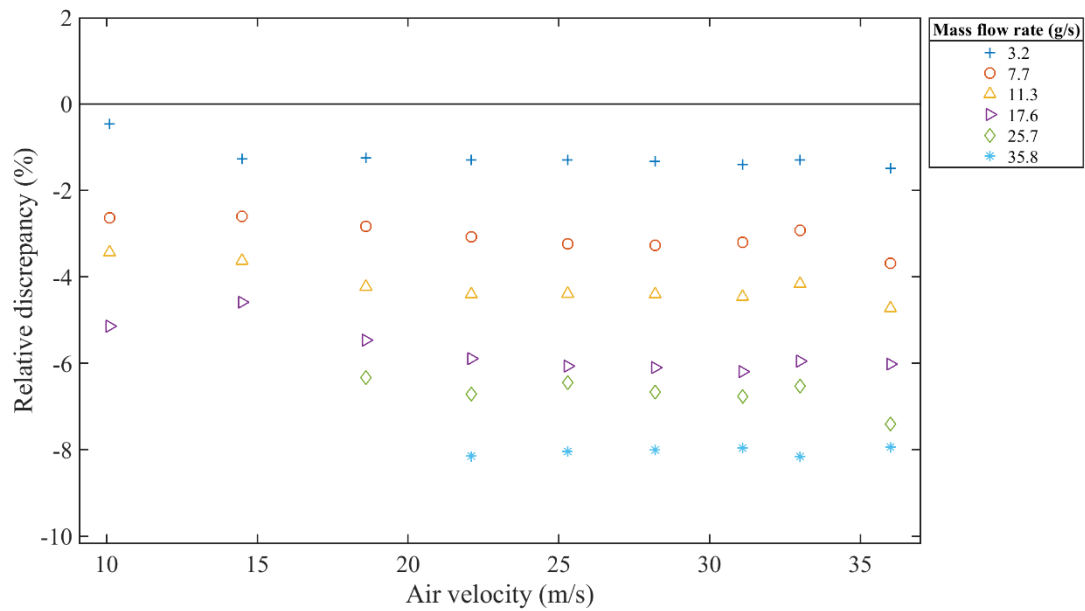
Figure 4-19: Relative discrepancy in measured particle velocity with best combinations of ring-shaped electrodes under vertical pipe orientation
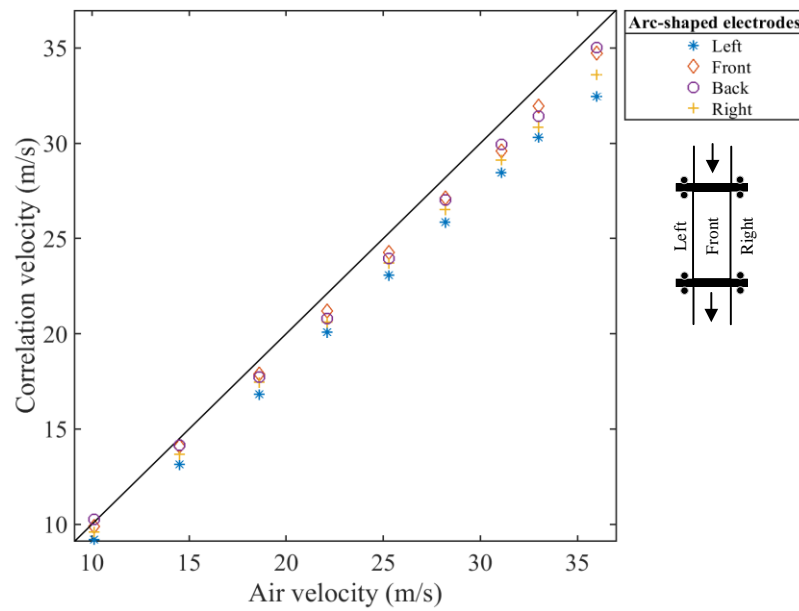
Figure 4-20: Measurement of particle velocity with best combinations of arc-shaped electrodes under vertical pipe orientation
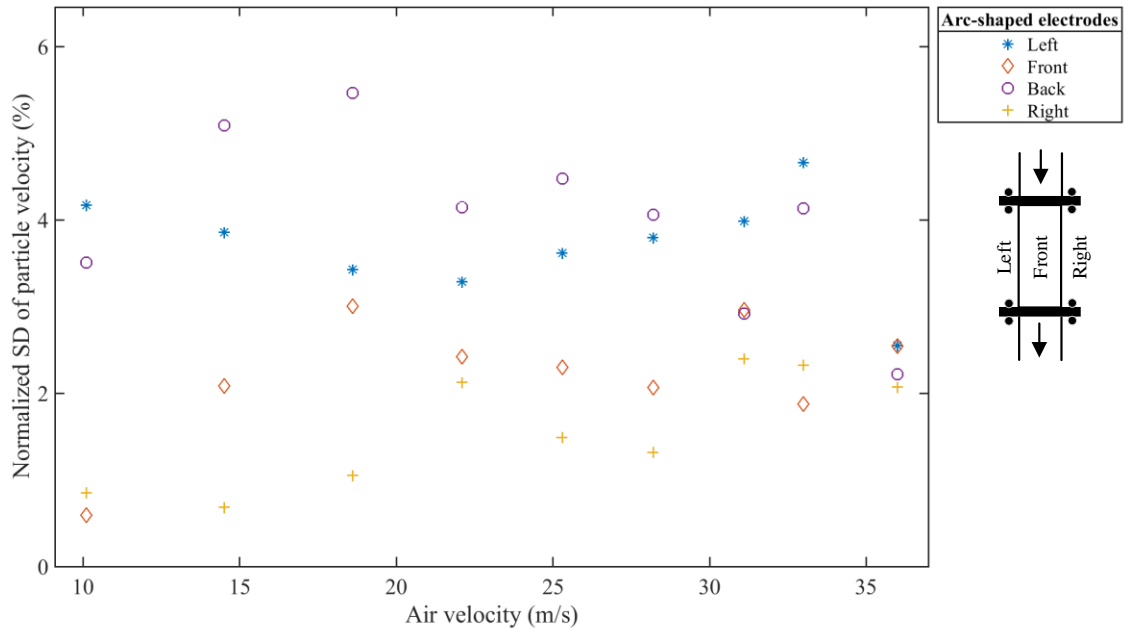
Figure 4-21: NSTD in measured particle velocity with best combinations of arc-shaped electrodes under vertical pipe orientation
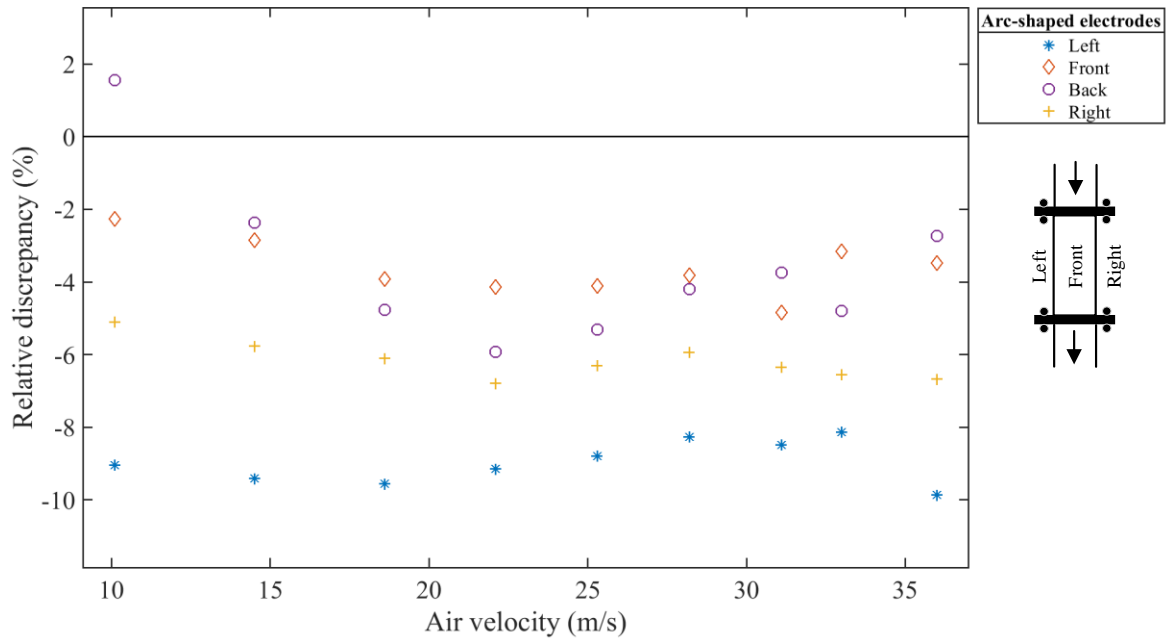
Figure 4-22: Relative discrepancy in measured particle velocity with best combinations of arc-shaped electrodes under vertical pipe orientation

4.8 Measurement of Mass Flow Rate of Solids

4.8.1 Analysis of Sensor Data

Figure 4-23 shows the relationship between the RMS of the signal from the E1 electrode of the ring-shaped electrostatic sensor and the mass flow rate of solids under different air velocities. Each data point on the graph also indicates the uncertainty in both mass flow rate and RMS of electrostatic signal that is calculated from the reference data of mass flow rate of solids and the recorded electrostatic signal under different conditions. The magnitude of electrostatic signal increases nonlinearly by increasing the mass flow rate and the velocity of solids. Since a high volume of large solids (less surface charge) is concentrated at the bottom of the pipe due to gravitational effect, the RMS of the signal, obtained from the arc-shaped electrode at the bottom of the pipe, is lower than others, as shown in Figure 4-24. Nevertheless, the RMS values from the electrodes at the front and back are comparatively higher than that at the bottom owing to the higher number of small particles in the suspension. The RMS of the signal from the sensor at the top is lower than those at the front and the back due to the lower solids concentration along the top of the pipe. However, at much lower mass flow rates (<10 g/s), the outputs from the top, front and the back electrodes yield very similar RMS magnitudes due to the fully suspended flow pattern except some large particles moving relatively slow at the bottom causing a lower RMS.
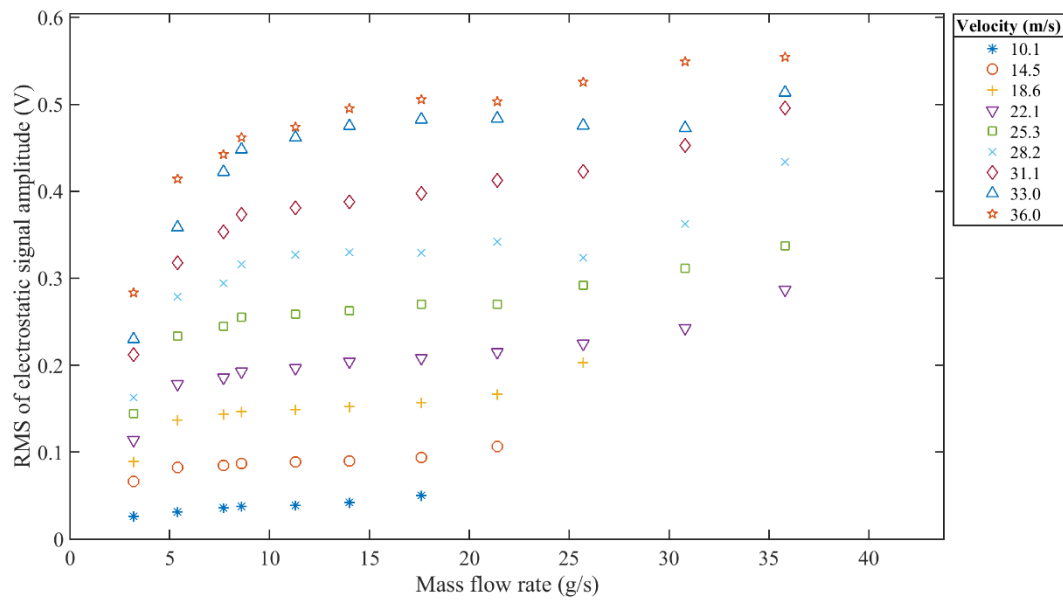
Figure 4-23: Relationship between RMS of ring-shaped electrostatic sensor (E1) and mass flow rate of solids under different air velocities
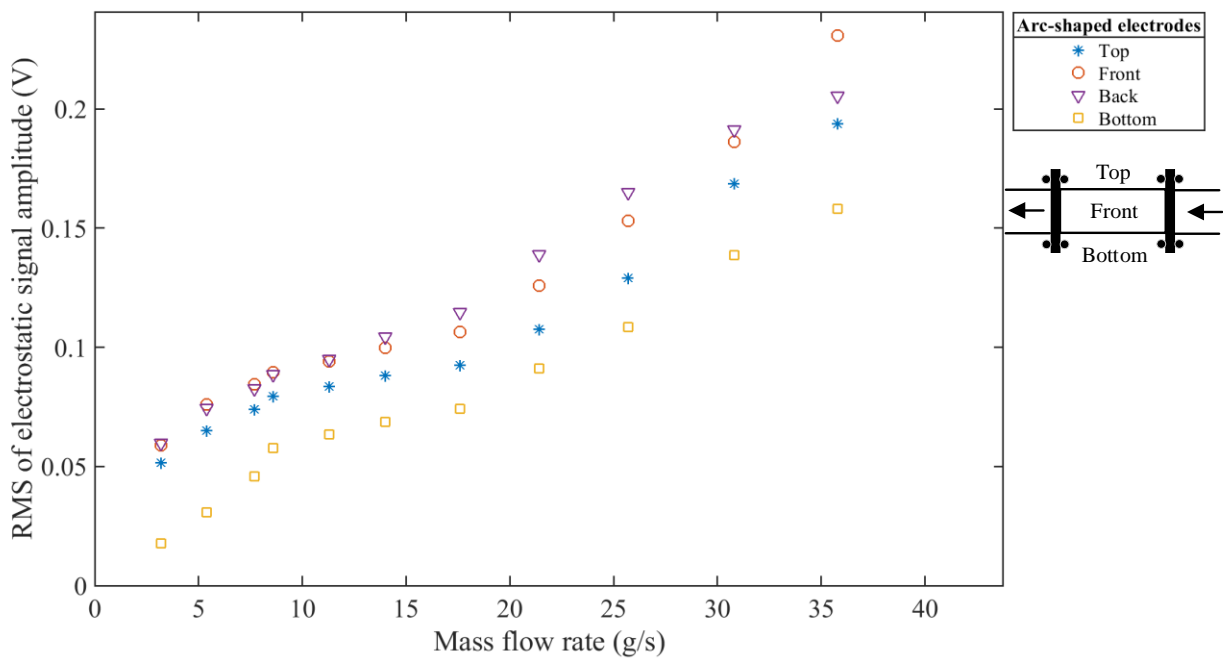
Figure 4-24: Relationship between RMS of four arcs of E5 electrostatic sensor and mass flow rate of solids

Figure 4-25 shows the variations in the magnitude of the signals from the ring-shaped electrostatic sensor E1 with the different mass flow rates and pipe angles. As the velocity of air is kept constant throughout all the experiments with different pipe orientations, concentration of solids increases with the mass flow rate of solids. The signals from electrostatic sensors were collected with different mass flow rates of solids under different pipe angles. The magnitude of the signal increases with the mass flow rate of solids. The placement of the sensing head at tilted orientations causes the solids to hit with the walls of the flexible joint, which can result in a possible reduction in solids energy and the charge magnitude. Therefore, the RMS value decreases as the pipe angle increases at the same mass flow rate condition. The rate of change of RMS value with increasing mass flow rate conditions is higher in lower pipe angles. However, the rate of change decreases dramatically as the pipe is being tilted towards the vertical orientation. Furthermore, there is comparatively a lower RMS charge value at higher mass flow rate conditions owing to higher possibility of solids to hit with the walls of flexible joint.
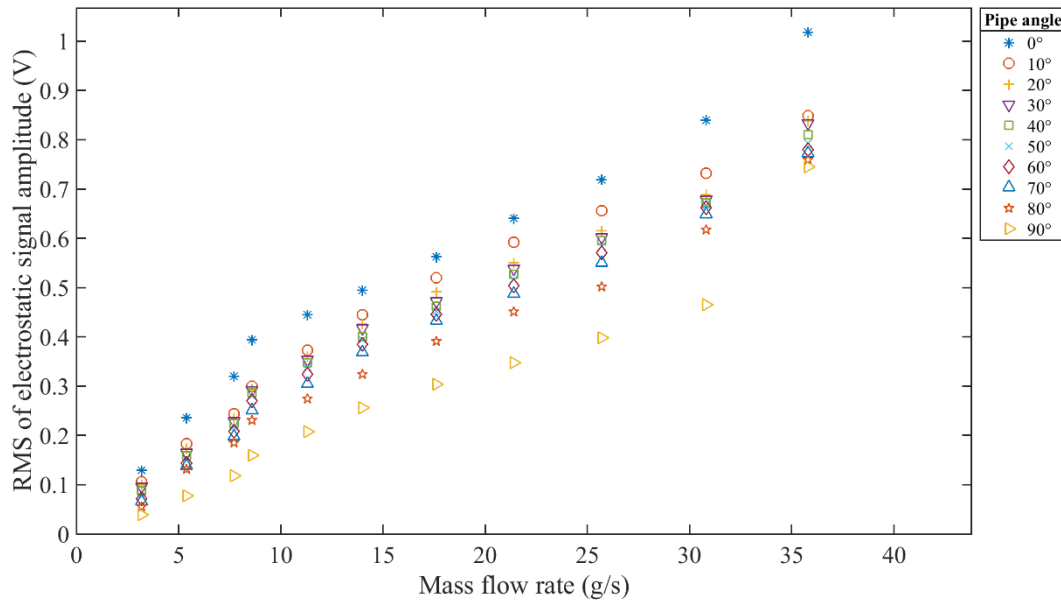
Figure 4-25: RMS of ring-shaped electrostatic sensor (E1) under different orientations of pipe

According to Gauss's law, the induced charge on the outer surfer of the electrode is directly proportional to the relative permittivity of the air and the permittivity of the vacuum [76]. Since the higher ambient temperature increases space polarization that gives a rise to relative permittivity of the air, therefore the magnitude of the charge also increases. Figure 4-26 shows the relationship

107

between the different mass flow rates of solids and the magnitude of the signal from electrostatic sensor under different ambient temperatures and constant RH 25%. From this relationship, it is quite evident that the magnitude of the electrostatic charge increases non-linearly with mass flow rate of solids. For a constant mass flow rate of solids, the increase in the ambient temperature has also resulted in a non-linear rise in the magnitude of the charge.
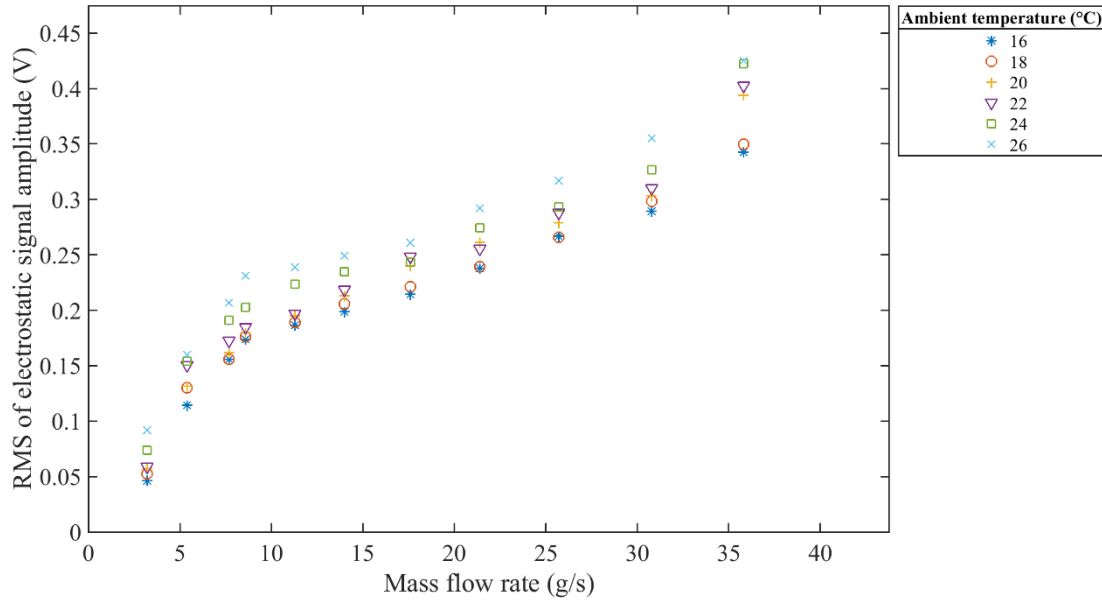
Figure 4-26: Relationship between the mass flow rate of solids and the RMS of electrostatic signal under different ambient temperatures and constant RH 25%.

Figure 4-27 shows the non-linear relationship between the mass flow rate of solids and the magnitude of the electrostatic signals under different ambient RH. The increasing RH has non-linearly decreased the magnitude of the signals owing to the fact that the higher RH in the lab causes the solids to absorb more moisture content as listed in Table 4-2. The solids with higher moisture content cover the surface area of the electrodes that results in obtaining weaker electrostatic signals. Moreover, a higher RH value results in reduced relative permittivity and hence the lower magnitude of the electrostatic charge.
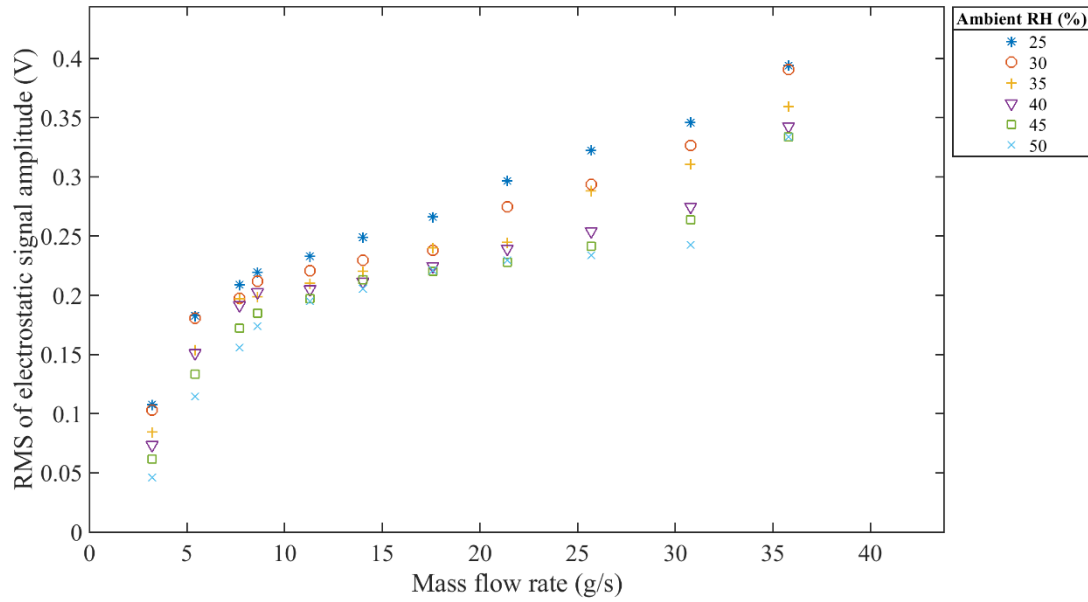
Figure 4-27: Relationship between mass flow rate of solids and RMS of electrostatic signal under different ambient RH and constant ambient temperature 16 ℃.

4.8.2 Performance of Ring and Arc-shaped Electrostatic Sensors and Their Combinations

As a first step towards evaluating the performance of the different categories of the electrostatic sensors, individual electrodes (ring and arc-shaped as presented in Chapter 3) are used for mass flow rate measurement with direct technique in conjunction with optimisation algorithm as shown in Figure 4-28 . The averaged NRMSE of mass flow rate measurement of solids using electrostatic sensors, lying in the same category of electrodes, remains similar to each other to a higher extent as shown in Figure 4-28. However, the performance of the arc-shaped electrostatic sensors is better than the ring-shaped ones owing to the reason that they provide more concise and stable output by calculating the average of the measurements taken through four arcs of the same ring.

Maximum and the minimum relative errors differ from electrode to electrode and this is due to irregular movement of solids in the whole cross-section of pipe. An improvement in NRMSE can be seen in Table 4-4 when a new electrode is added into the combination regardless of the fact that the NRMSE value of the individual electrode is higher than the others in the combination. This is due to the fact that some electrodes perform better in some cases while the others fail to provide the reasonable accuracy. Therefore, combination of their RMS values normalizes the cases where

109

a particular electrode does not perform well. The standard deviation of relative error in measured mass flow rates with different combinations of sensors is shown in Figure 4-29. Figure 4-30 shows the performance of FA in each iteration when applied with the fusion of all electrostatic sensors along with the respective values of a and b. Stability in the fitness value after 14th iteration shows that the FA has approached to its saturation state by giving the best suitable parameters a and b. Convergence of the program in such earlier stage of the iterations shows the effectiveness of the FA.
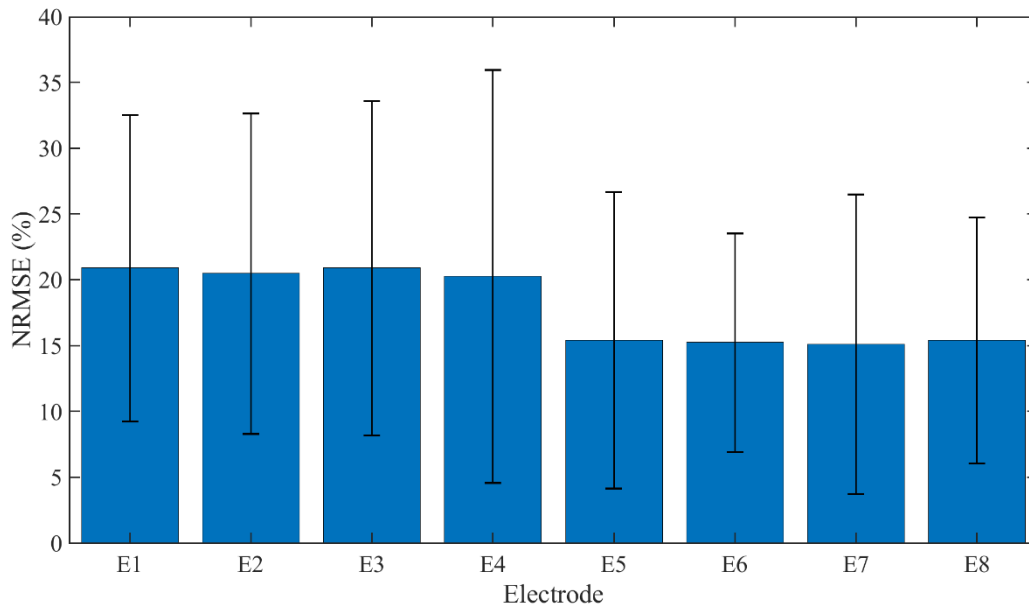
Figure 4-28: Performance of individual electrostatic sensors for mass flow rate measurement

Table 4-4: Combination of sensors for data fusion and their resulting NRMSE for the measured mass flow rate of solids

Combination name	Sensor combination	NRMSE (%)
i	E7	17.56
ii	E7, E6	17.49
iii	E7, E6, E5	16.21
iv	E7, E6, E5, E8	15.33
v	E7, E6, E5, E8, E2	15.05
vi	E7, E6, E5, E8, E2, E4	14.92
vii	E7, E6, E5, E8, E2, E4, E1	14.81

viii	E7, E6, E5, E8, E2, E4, E1, E2	14.73
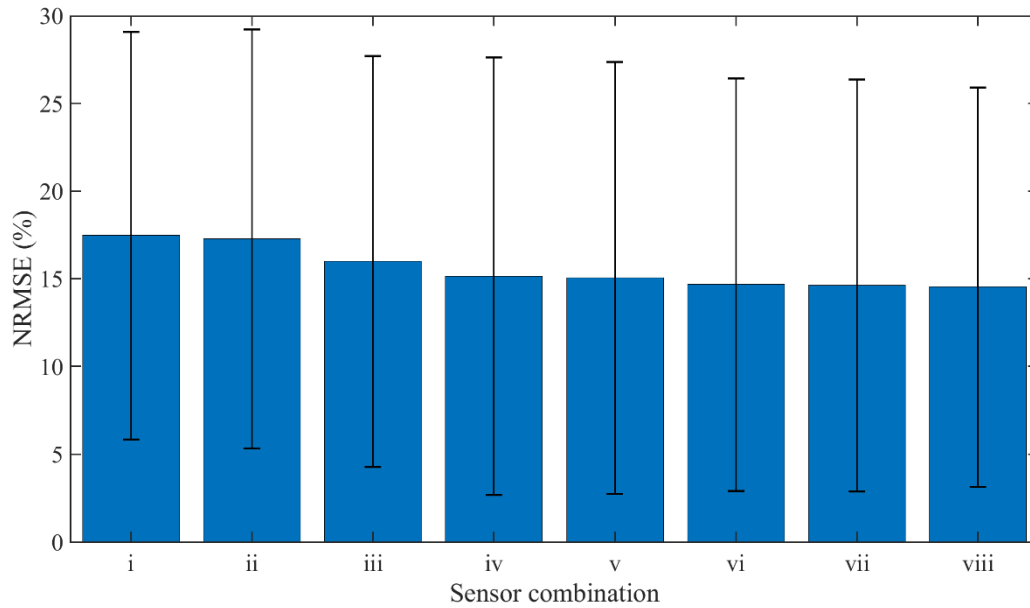
Figure 4-29: Performance of different combinations of electrostatic sensors for mass flow rate measurement
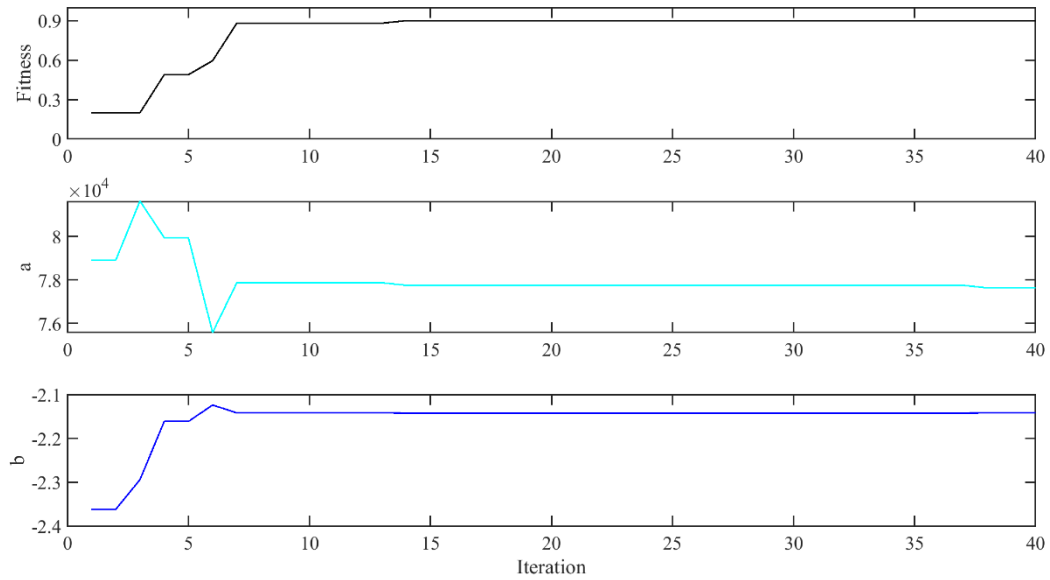
Figure 4-30: Performance of the FA algorithm

4.8.3 Measured Mass Flow Rate of Solids Under Different Air Velocities

Figure 4-31 shows the mass flow rate measurements by using the optimized values of a and b for direct technique when applied with combination of all the electrostatic sensors. Figure 4-32 shows the repeatability of the mass flow rate measurement in terms of NSTD under different air velocities. Mass flow rate measurements are more repeatable at lower air velocities, whereas the measurements become unstable at higher air velocities. All the measurements, taken with different air velocities under horizontal pipe orientation, fluctuate dramatically with high relative error, as shown in Figure 4-33. The averaged NRMSE of all the mass flow rate measurements is very high for different air velocities except at the velocity 22.1 m/s where the optimisation of a and b is carried out as shown in Figure 4-34. The optimized values of a and b are found to be 70000 and -2.28, respectively.
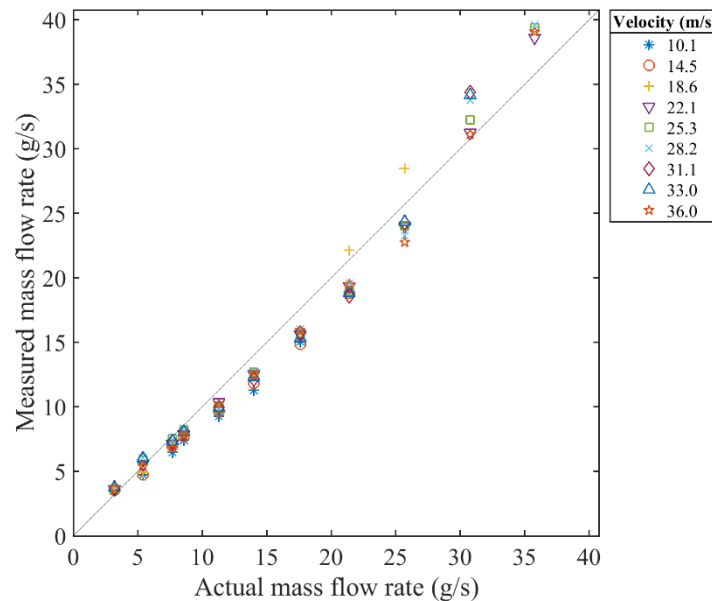
Figure 4-31: Mass flow rate measurement of solids under different air velocities
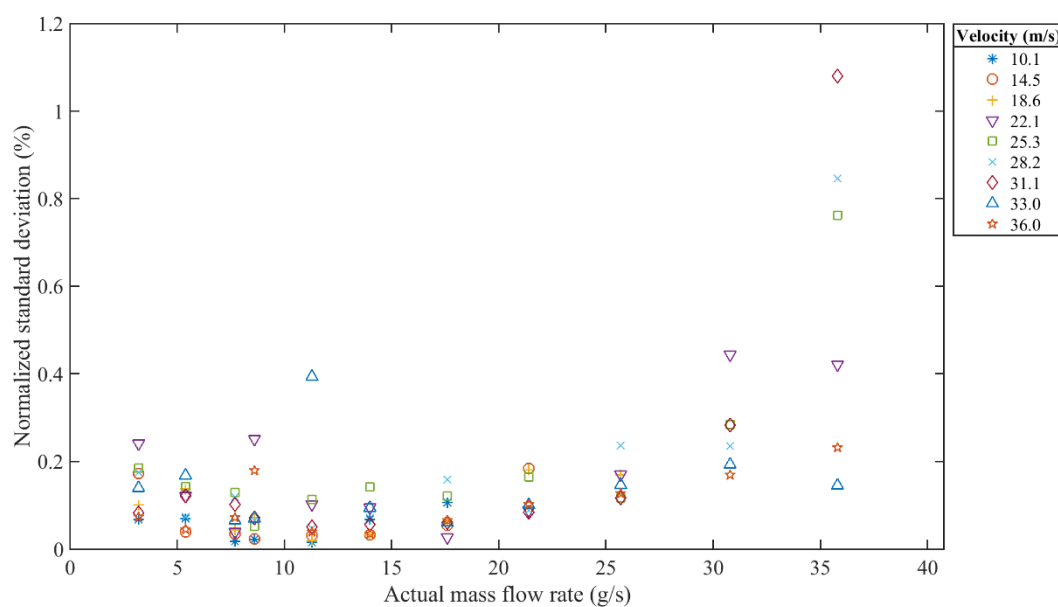
Figure 4-32: NSTD in mass flow rate measurement of solids rate under different air velocities
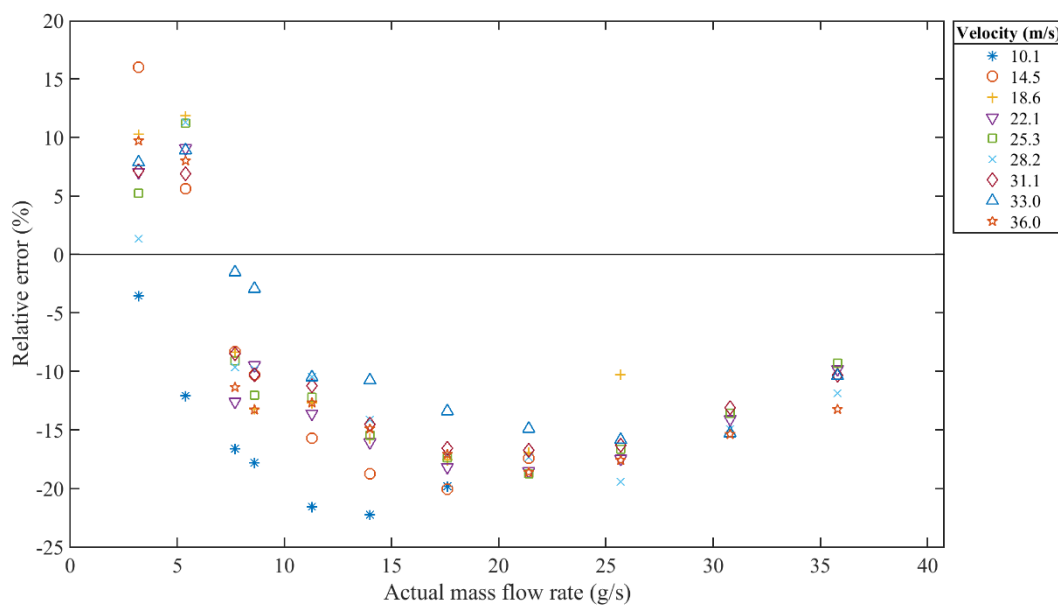
Figure 4-33: Relative error in mass flow rate measurement of solids under different air velocities
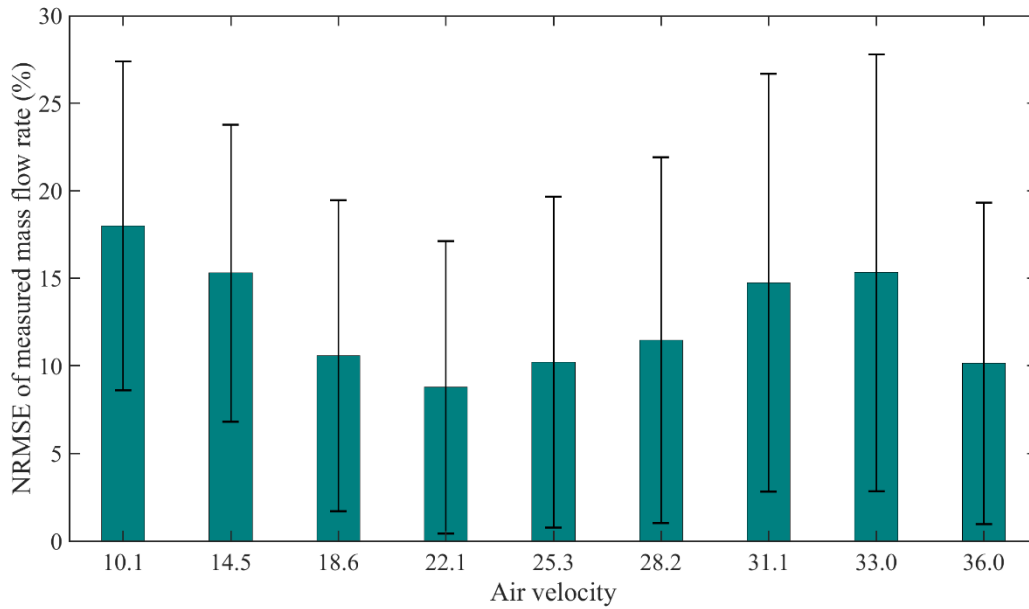
Figure 4-34: NRMSE of mass flow rate measured under different solids velocity

4.8.4 Measured Mass Flow Rate of Solids Under Different Pipe Orientations

Figure 4-35 shows the measurement of eleven mass flow rates under ten different pipe angles. Different shapes are used in the graph to show the measurements under different pipe angles. However, the measurements under same angle are ploted with different colours to differentiate the different mass flow rates especially the ones which are closely located on the graph. To investigate the only impact of different pipe orientations, solids velocity and other experimental conditions are kept constant. Repeatability of the measured results is assessed through NSTD as shown in Figure 4-36. As the pipe is tilted towards vertical orientation, particles hit with the wall of flexible joint and start moving downwards with high consistency. Therefore, higher repeatability can be seen in the measurement at tilted angles. Furthermore, the tilted pipe angles also influence the RMS charge level of particles. Therefore, higher relative errors are observed in the measurements taken at more tilted pipe angles as shown in Figure 4-37. As shown in Figure 4-38, the averaged NRMSE of all the mass flow rate conditions increases proportionally with pipe angle starting from $0°$ where the measurement system is calibrated with the values of a and b. The optimized values of a and b are found to be 71000 and -2.12, respectively.
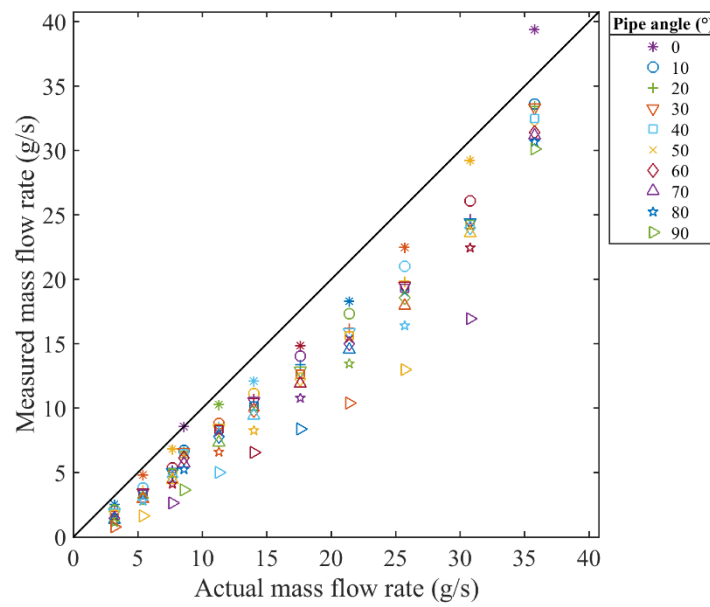
Figure 4-35: Mass flow rate measurement of solids under different pipe orientations
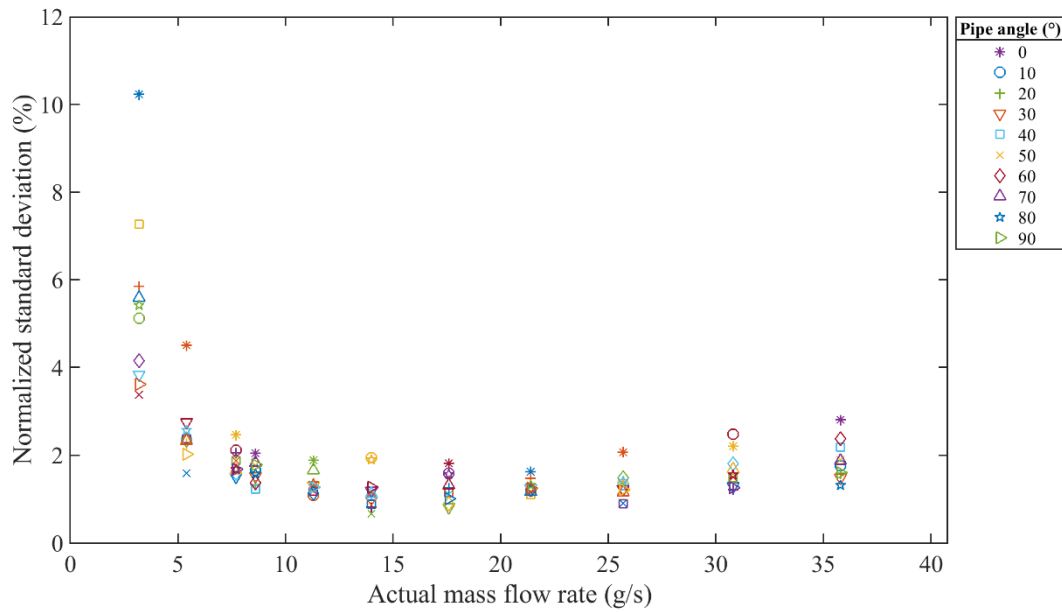
Figure 4-36: NSTD of mass flow rate measurement of solids under different pipe orientations
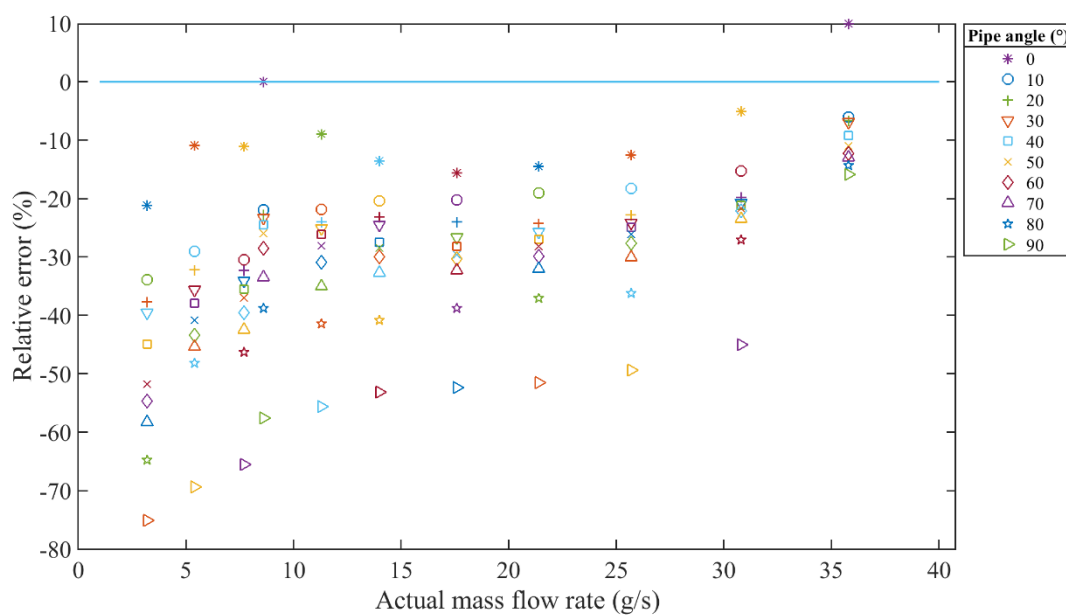
Figure 4-37: Relative error in mass flow rate measurement of solids under different pipe orientations
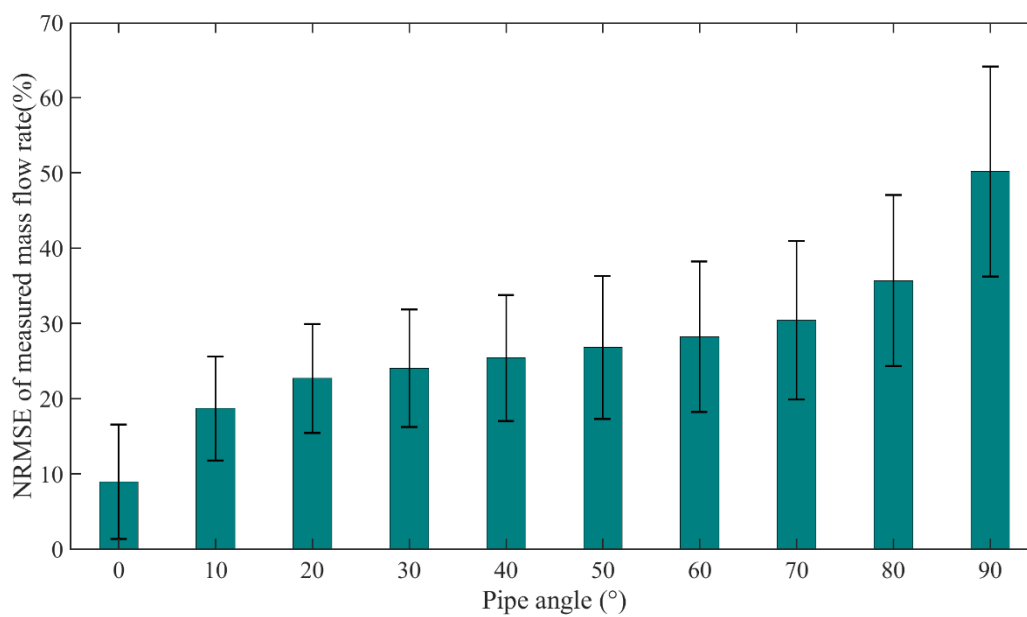
Figure 4-38: NRMSE of mass flow rate measurements under different pipe orientations

4.8.5 Measured Mass Flow Rate of Solids Under Different Ambient Conditions

Figure 4-39 and Figure 4-40 shows the measured mass flow rates of solids under six different ambient temperatures and RH. The characteristics of the electrostatic sensor data under different ambient conditions are non-linear. It is hard to keep all the points on or near the ideal straight line. However, the FA has used the fitness function to find and optimize the values of a and b in a way that can cause least amount of error in all the mass flow rate measurements under different ambient conditions. The characteristics of the electrostatic sensor changes with different ambient conditions as discussed in Section 4.8.1, therefore it is hard to map all the points, lying under different ambient conditions, near the reference line. In fact, the a and b parameters of the direct measurement technique are set in a way to keep the measurements under 16 °C and 25% RH close to the reference line. The measurements are less repeatable for the mass flow rates of solids under the ambient temperatures and RH higher than 16 °C and 25%, respectively, and this fact is more prominent for the higher mass flow rates of solids as shown in Figure 4-41 and Figure 4-42. The characteristics of the signals from electrostatic sensor for different mass flow rates are non-linear, therefore the relative error between the measured and the actual mass flow rate is higher at the lower and the higher mass flow rates of solids as shown in Figure 4-43 and Figure 4-44. The averaged NRMSE of mass flow rate measurement gives the better estimate of error under different ambient conditions. The system is calibrated at 16 °C and 25% RH, therefore the NRMSE is increasing from the points at which the system is calibrated as shown in Figure 4-45 and Figure 4-46. The optimized values of a and b are found to be 71500 and -2.42, respectively.
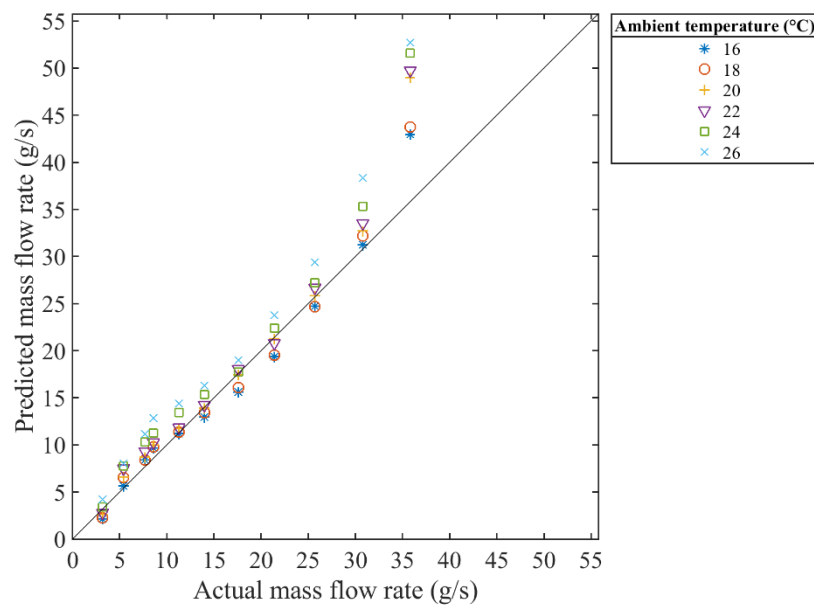
Figure 4-39: Mass flow rate measurement under different ambient temperature and constant ambient RH 25%.
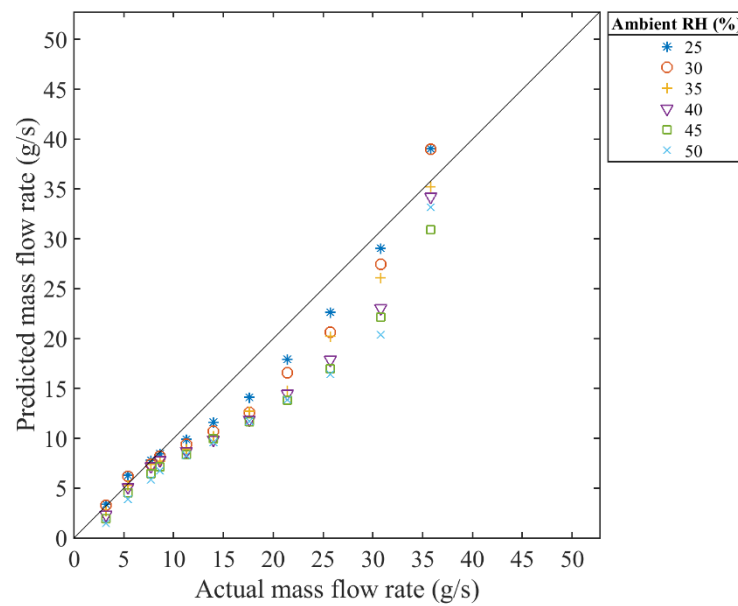
Figure 4-40: Mass flow rate measurement under different ambient RH and constant ambient temperature 16 °C.
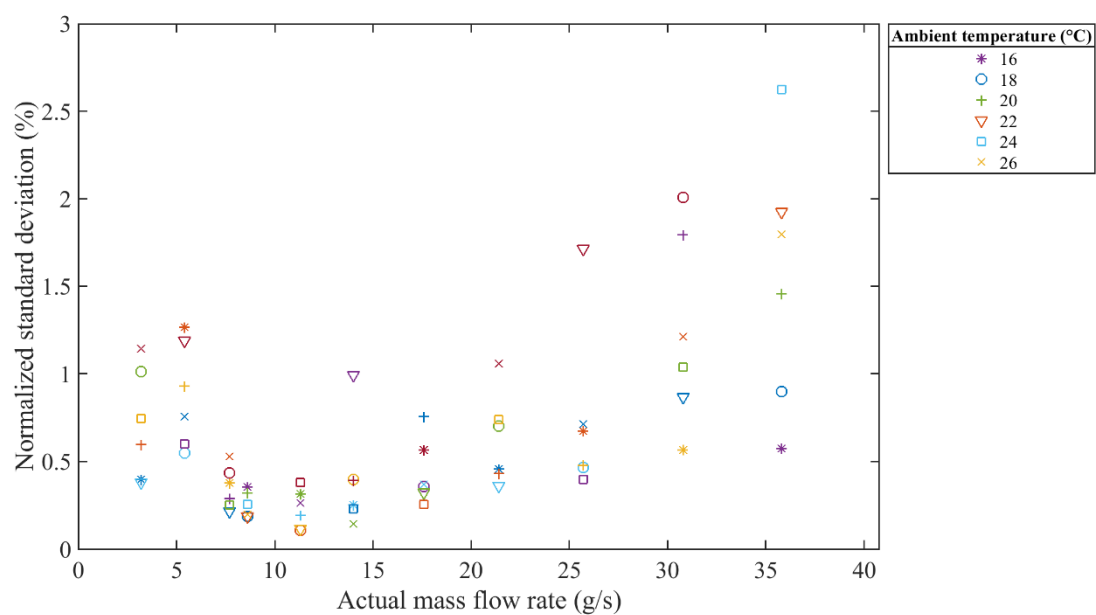
Figure 4-41: Normalized standard deviation of mass flow rate measurement under different ambient temperature and constant ambient RH 25%.
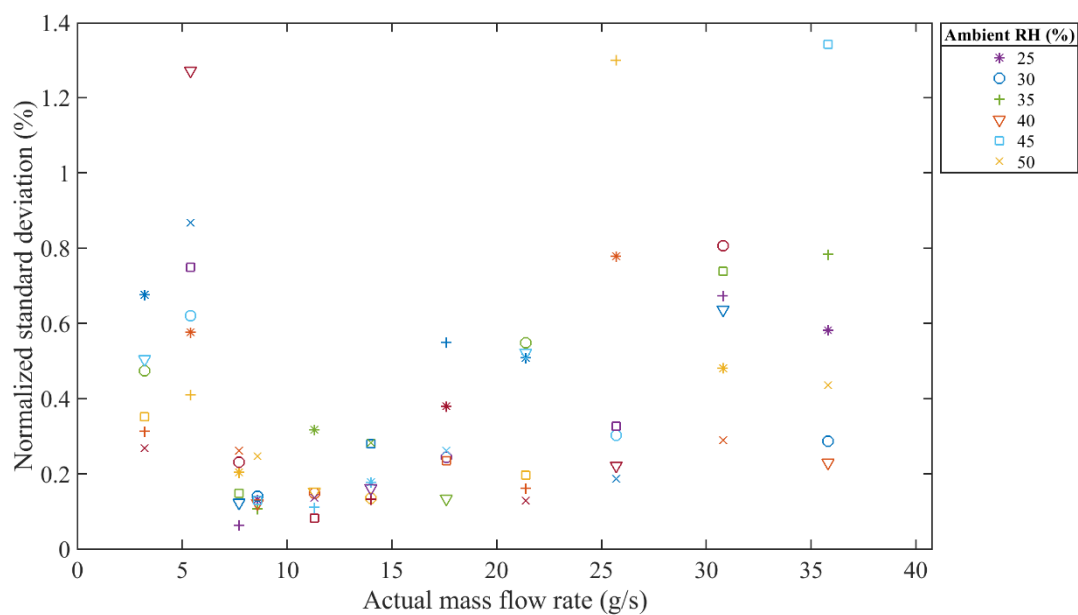
Figure 4-42: Normalized standard deviation of mass flow rate measurement under different ambient RH and constant ambient temperature 16 °C.
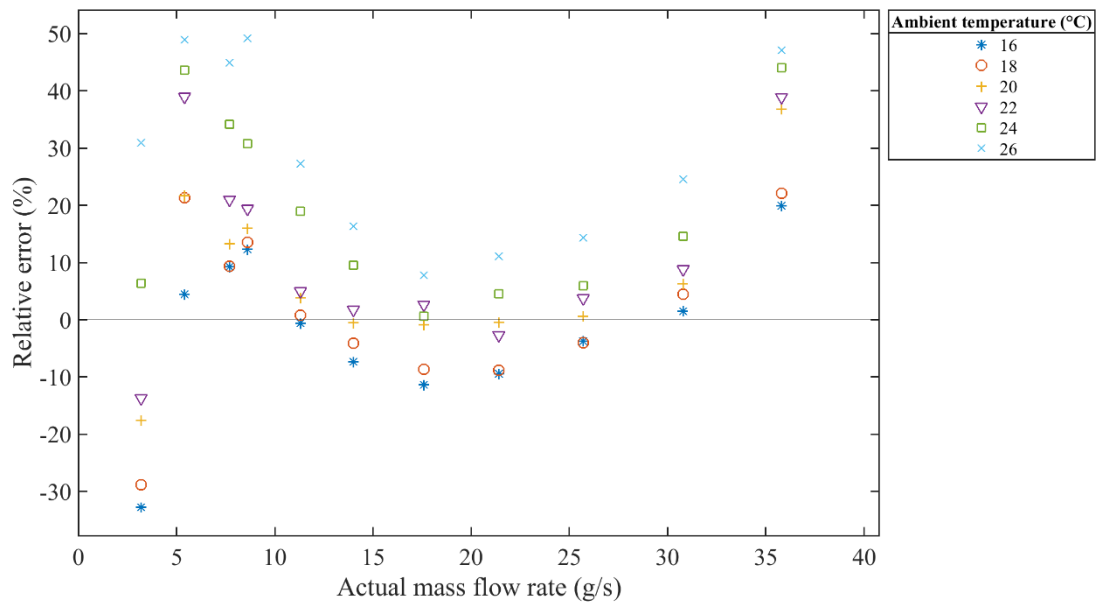
Figure 4-43: Relative error in mass flow rate measurement under different ambient temperature and constant ambient RH 25%.
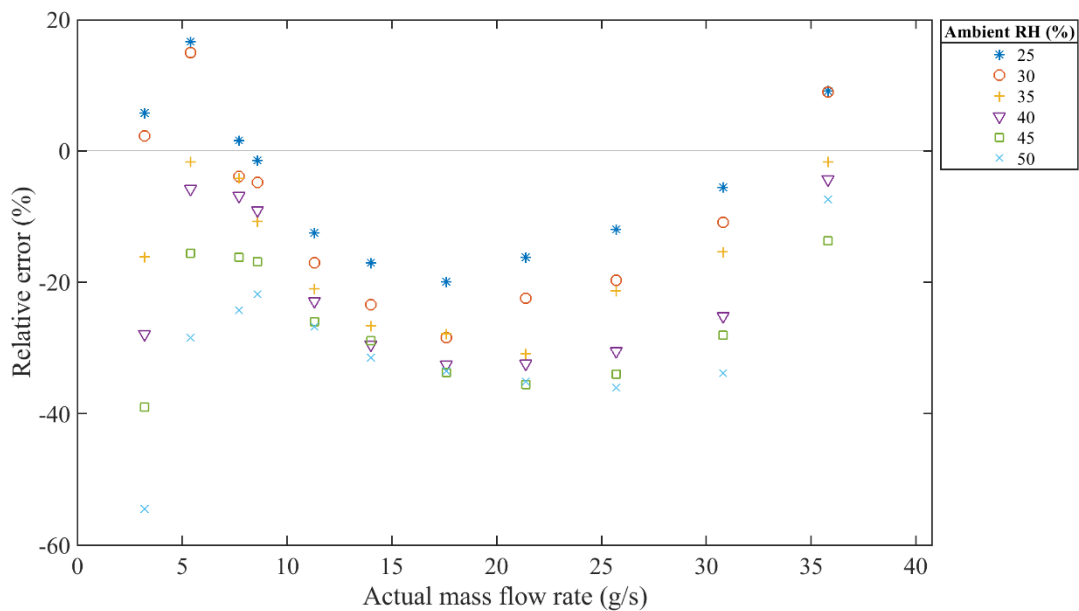
Figure 4-44: Relative error in mass flow rate measurement under different ambient RH and constant ambient temperature 16 ℃.
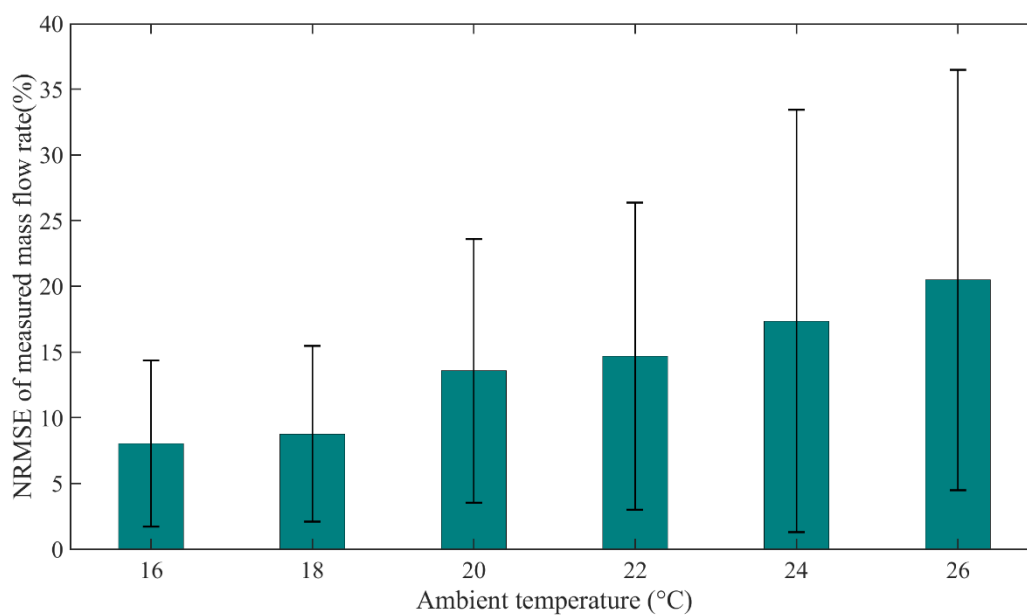
Figure 4-45: NRMSE of mass flow rate measurement under different ambient temperature and constant ambient RH 25%.
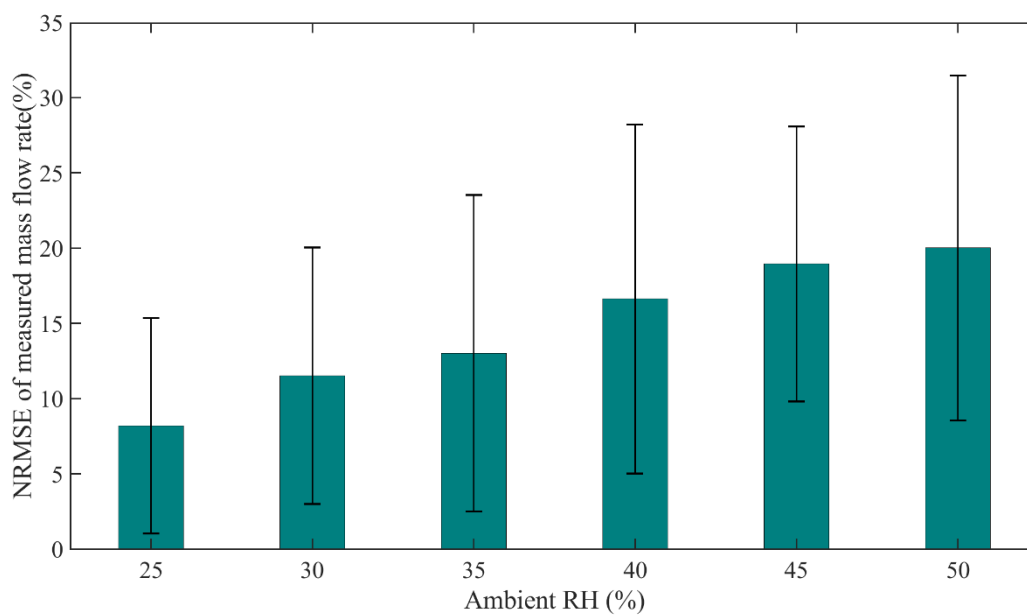
Figure 4-46: NRMSE of measured mass flow rate under different ambient temperature and constant ambient temperature 16 °C

4.9 Summary

This chapter presents the measurement of particle velocity and mass flow rate of solids using electrostatic sensors with direct measurement techniques under different air velocities, pipe orientations and ambient conditions. The relative error in particle velocity remains lower if measured with consecutively placed upstream/downstream electrodes. The higher concentration of solids moves from bottom to the left area of the pipe by changing the pipe orientation from horizontal to vertical. The fusion of data from different sensors can reduce the relative error in the measurement of flow parameter where a particular type of sensor does not perform well. Therefore, an improvement in the overall mass flow rate measurement has been seen by increasing the number of electrodes. This outcome is observed through ofline measurement of mass flow rate of solids using dataset with known labels of reference mass flow rates. The four arcs, covering the top, front, back and the bottom of the pipe, provide more concise and localized information about the flow parameters. Therefore, the average mass flow rate of solids, measured with four arc-shaped electrostatic sensors, has given the better performance as compared to the ring-shaped electrostatic sensors. The main advantage of direct method of mass flow rate measurement is that it takes less number of samples from the electrostatic sensor signals and produce more repeatable results provided the system is fully calibrated and the physical parameters remain constant. However, the method is inferential which is derived from particle velocity and the solids concentration. The RMS charge level of the electrostatic sensors, mainly used to determine the solids concentration, has been found affected by the physical factors, including air velocities, pipe orientations and the ambient conditions. The averaged NRMSE of the mass flow rate measurement has been increased from 8% to 18%, 50% and 35% under different air velocities, pipe orientations and ambient conditions, respectively. The measurement methodology requires recalibration of the system with new values of the parameters a and b under different physical conditions which is often time consuming. With the development of on-going analytical modelling of gas-solids two-phase flow, engineering judgements in the application of modern machine learning techniques may have a part to play to deal with the challenges of flow measurements under different physical conditions.

Chapter 5 Mass Flow Rate Measurement of Solids Through Data-Driven Modeling Under Different Conveying Air Velocities

5.1 Introduction

The inferential method of mass flow rate measurement of solids with particles velocity and solids concentration using electrostatic sensor is widely used. However, the main problem in applying such technique is that the RMS charge level of electrostatic sensor output depends on various physical factors, including particle velocity, ambient conditions and pipe orientations as seen in Chapter 4. This chapter presents data driven modelling as a potential method for the mass flow rate measurement of solids with a minimized impact of the physical factors. In this study, the experiments for the measurement of mass flow rate of solids through data-driven modeling are carried out under different air velocities. The technique for the mass flow rate measurement of solids under different conveying air velocities is based on the multi-modal sensing and data driven modelling. Data driven models, including artificial neural network (ANN), support vector machine (SVM), and convolutional neural network (CNN), are established through training with statistical features extracted from the post-processed data from the sensing system. The sensor selection is implemented by applying SVM model on the data of different combinations of sensors. Statistical features are selected based on partial mutual information algorithm and Akaike information criterion. Experimental data, presented in Chapter 4 on direct techniques of measurement of particle velocity and mass flow rate of solids, is used to train and test the data driven models. This chapter is only focused on the technique to compensate the effect of different air velocities on mass flow rate measurement. However, there is still a need to compensate other physical factors such as pipe orientations, ambient conditions and particle types that will be presented in Chapter 6.

5.2 Pre-processing of Sensor Data

5.2.1 Clustering of Data

The experimental work is related to different mass flow rates of solids that is controlled through vibratory feeder. Mass flow rate of solids is dependent on the magnitude of the vibrations produced by vibratory feeder. During this process of conveying solids through vibratory feeder, one of the significant phenomena, which has been observed after analyzing the raw data of sensor value is that the amount of solids provided by the feeder, does not remain consistent even if the magnitude of vibrations has been set constant. The non-consistent results are produced due to the reason that it is not possible to keep the volume of the solids uniform on the surface of vibratory feeder. Infact, in some cases, few of the solids absorb the moisture from air, which make them look like a cluster. After moving through all over the surface of vibratory feeder, when this cluster falls inside the pipe, it results in a rapid increase in the flow rate. Figure 5-1 shows how the raw values of electrostatic sensor data look like with different mass flow rates. From this figure, it is quite evident that one group of mass flow rate has many variations in it.
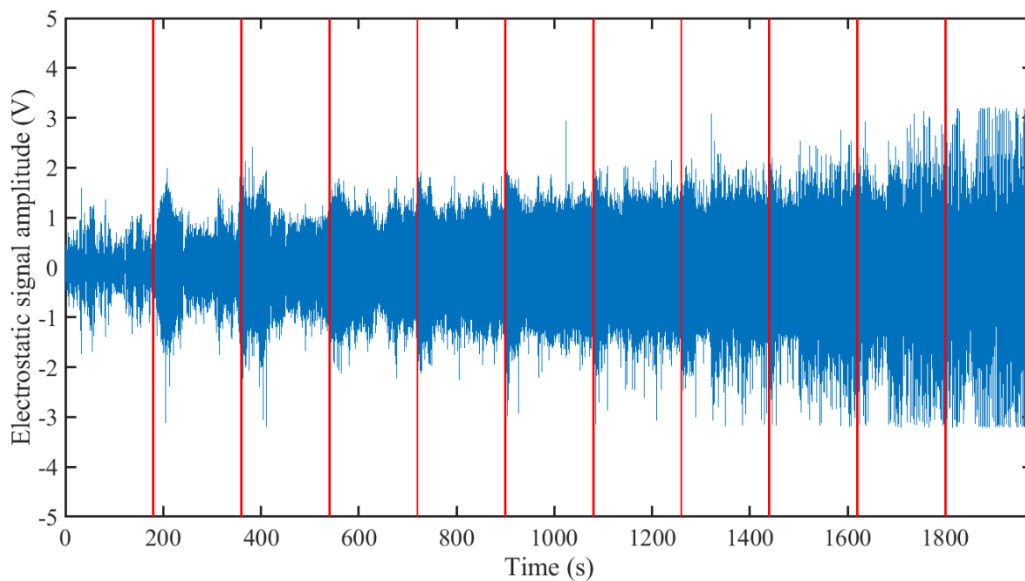
Figure 5-1: Raw signals from electrostatic sensor (E1) with different mass flow rates

Considering the example of first group in the plot, during the experimentation, the intended flow rate was set 3.2 g/s but there is a very high possibility that this group might contain some data samples, which belongs to the next group of mass flow rate that is 5.4 g/s or may be the group after

124

next group. One of the most imperative matter, which must be taken into attention, is that the process of prediction is highly based on careful tuning of weights and biasing parameters of data-driven modeling. Having these kind of multiple flow points in one group can mislead the training process. Therefore, the given scenario suggests to apply a clustering algorithm on raw sensor values to put them in a specific group of flow rate where they actually belong. For this purpose, K-means clustering algorithm is applied to put the data samples into their correct clusters. Given a number k, divide all the data into separate clusters in a given partition into k, each with a center doubling as a representative. These centers are updated in iterations and then the data samples are assigned to the clusters based on smallest distance between data sample and the center of the cluster. The whole process repeats until the centers are not moving [77] as explained in Figure 5-2.
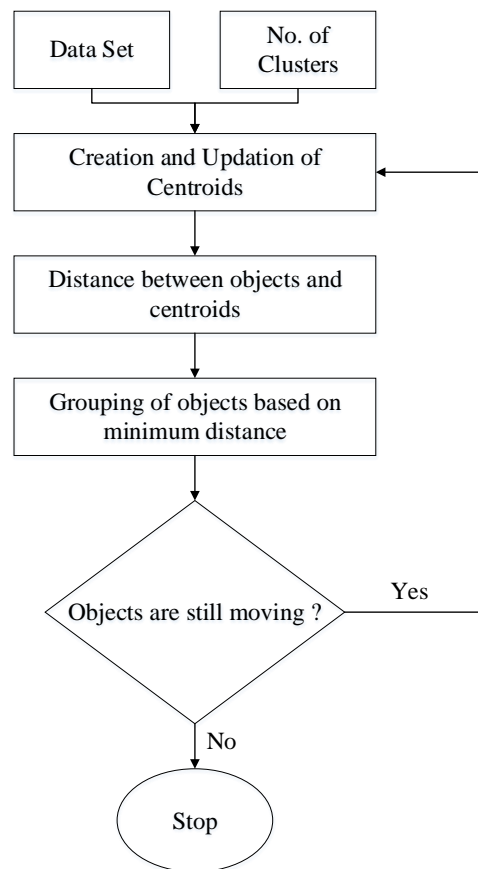
Figure 5-2: K-means clustering algorithm

The distance between the centers of the clusters and the data point is calculated and then based on the nearest neighbor method, it can be decided that which data point belongs to which cluster.

Similar set of steps are followed in multiple iterations after updating the center point of each cluster and due to this reason, it has been considered a simple yet the fast clustering algorithm in this study. The number of mass flow rate conditions are eleven, therefore the overall data points are gathered in eleven clusters. Figure 5-3 shows the electrostatic sensor signals after the clustering algorithm is applied on the raw signals. As it can be seen, the amplitude of the signals increases with the mass flow rate as well as particle velocity. The signal conditioning unit is powered from a ±3.3 V power supply, therefore the magnitude of all the signals remains within ±3.3 V without saturation.
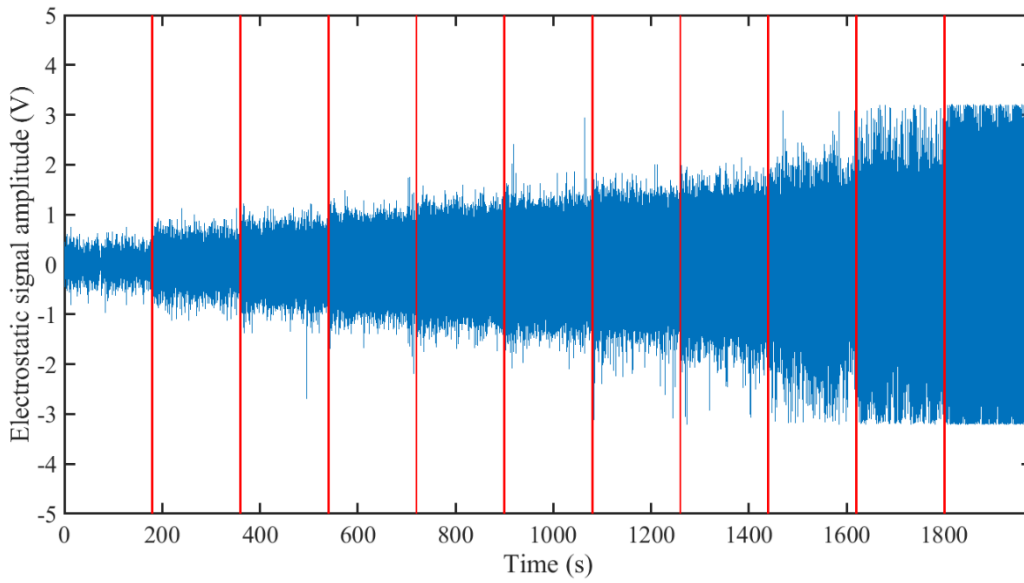
Figure 5-3: Signals from electrostatic sensor (E1) after clustering algorithm

5.2.2 Removal of Outliers

The performance of the data driven models is critically associated with the magnitude of the raw signals from the sensors. Therefore, the signals obtained from the sensing system need to be pre-processed prior to extraction of the features by removing the outliers. All the samples higher than $E(|x|) + \alpha\sigma$ are considered as outliers and removed from data, where $E(|x|)$ is the mean of the absolute data over one sampling period and it is calculated by

$$E(|x|) = \frac{1}{N}\sum_{i=1}^{N}|x_i|$$

(5-1)

126

Where x_i and x_i' contain the original and updated samples, respectively, σ is the standard deviation of the data over one sampling period, N is the total number of samples, and α is a constant that determines the confidence limits of the outlier.

Fig. 5-4 shows the impact of different α values on a typical time series electrostatic signal obtained at a constant velocity of 25.3 m/s and 11 mass flow rates of solids listed in Table 5-2 and Table 5-3. As it is discussed in Section II-D, most of the outliers exist outside the three standard deviations from the mean. A high number of samples is detected as outliers by having a smaller value of α as shown in Fig. 5-4 (a). The number of detected outliers has reasonably reduced by extending the limits of α as shown in Fig. 5-4 (b) and Fig. 5-4 (c). However, no data point lies after the six standard deviations from the mean and hence the algorithm is unable to detect any outlier at α=7 as shown in Fig. 5-4 (d). Based on the position of outliers in the sensor data, α is set to 4 and 6 for the signals of electrostatic sensors and DP transducer, respectively, that has removed the outliers without affecting the actual signals.

(a)

(b)
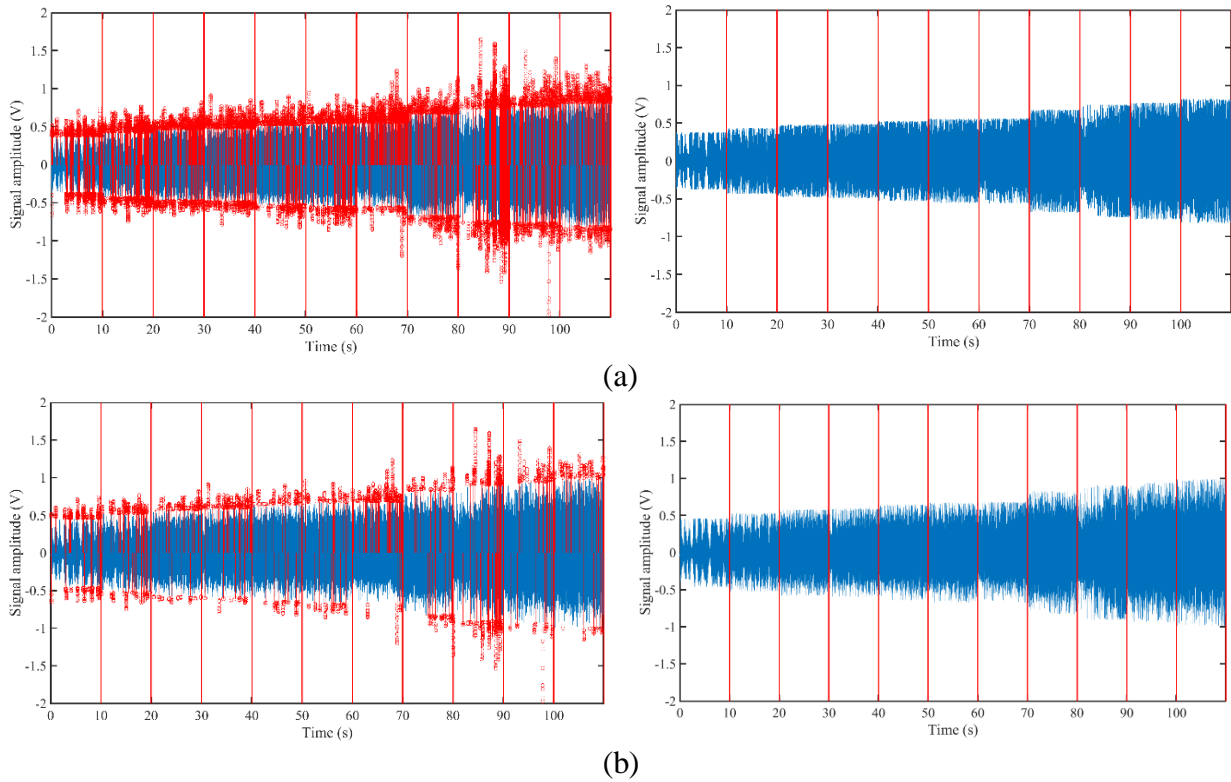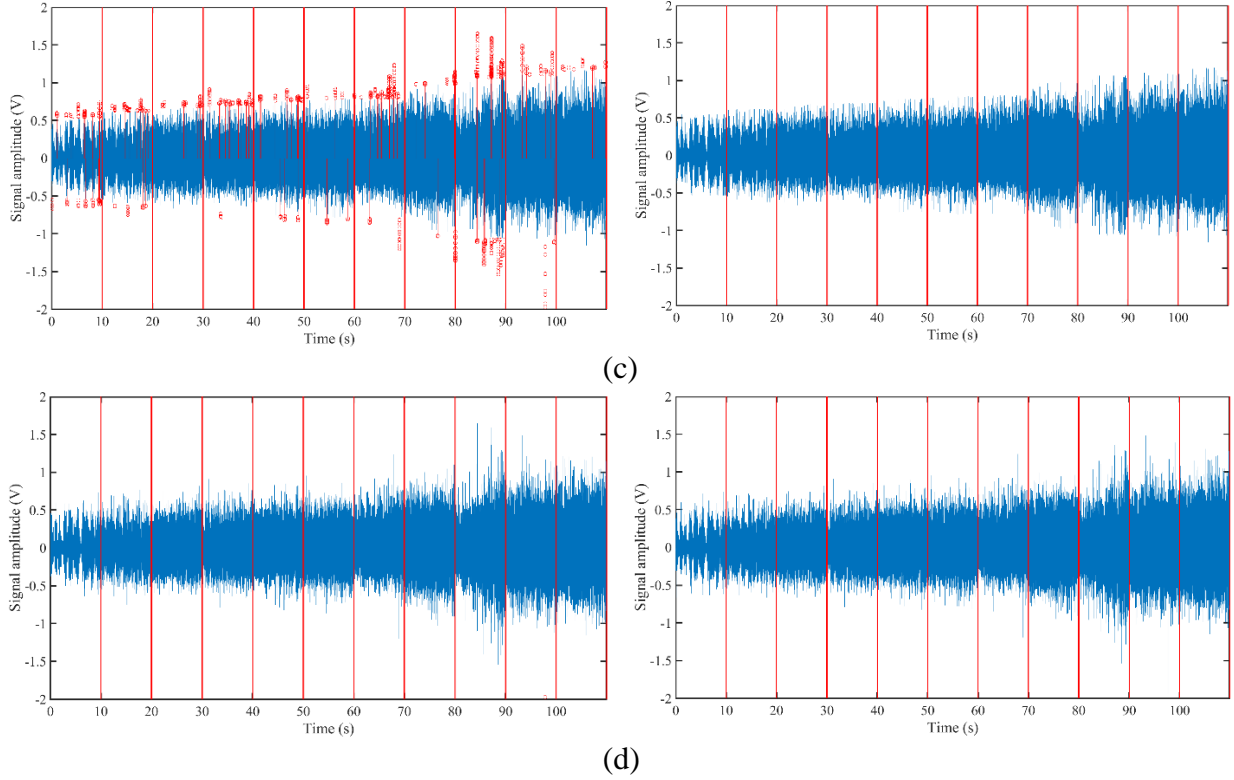
Fig. 5-4. Electrostatic signal with outliers highlighted on the left and removed on the right (a) $\alpha = 4$ (b) $\alpha = 5$ (c) $\alpha = 6$ (d) $\alpha = 7$

For all the sensor data in this study, an outlier is a data point which lies outside of the three standard deviations from the mean. In this study α is set to 4, as determined experimentally.

5.2.3 Extraction of Features

It is not straightforward to define the appropriate statistical features for the establishment and efficient training of data driven models and the ultimate measurement of the mass flow rate of solids. An extensive literature review was carried out in this study to compile and consider all the possible statistical features that may be used as inputs for the data driven models. The magnitude of all the sensor signals varies with the movement of solids. Therefore, the data driven models are trained with statistical features based on the magnitude of each sensor signal in three different domains, including time, frequency, and time-frequency domains. Table 5-1 summarizes the symbols and the equations to extract the features from the outputs of the electrostatic sensors and the DP transducer. A total of 13 statistical features, including root mean square, mean, standard deviation, skewness, kurtosis, number of zero crossings, variance, peak to peak, entropy, simple

128

sign integral, waveform length, Willison amplitude and slope sign integral, are extracted in the time domain. In the frequency domain five features, including peak amplitude, signal power, mean fluctuation frequency, spectral entropy and shape factor, are extracted. Likewise, mean, root mean square, variance, four quartile frequencies and mean crossing irregularity are also extracted in the time-frequency domain.

Table 5-1: Statistical features that are considered in this study

	Description	Symbol	Equation
Time Domain	Root Mean Square	RMS$_T$	$\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$
	Mean	$\bar{x}$	$\frac{1}{N}\sum_{i=1}^{N} x_i$
	Standard Deviation	σ	$\sqrt{\frac{\sum_{i=1}^{N}|x_i - \bar{x}|^2}{N-1}}$
	Skewness	SKW	$\frac{1}{N\sigma^3}\left(\sum_{i=1}^{N}|x_i - \bar{x}|^3\right)$
	Kurtosis	KRT	$\frac{1}{N\sigma^4}\left(\sum_{i=1}^{N}|x_i - \bar{x}|^4\right)$
	Number of Zero Crossings	ZC	$(x_i > 0\ \&\ x_{i+1} < 0)\ \|\|\ (x_i < 0\ \&\ x_{i+1} > 0)$
	Variance	VAR$_T$	$\frac{1}{N-1}\left(\sum_{i=1}^{N}|x_i - \bar{x}|^2\right)$
	Peak to Peak	PP	Max(x)-Min(x)
	Entropy	H$_T$	$\sum_{i=1}^{N} p_i \log(p_i)$
	Simple Sign Integral	SSI	$\sum_{i=1}^{N}|x_i|^2$
	Waveform Length	WL	$\sum_{i=2}^{N}|x_i - x_{i-1}|$
	Willison Amplitude	WAMP	$\sum_{i=2}^{N} f(|x_i - x_{i-1}|)$
	Slope Sign Changes	SSL	$(x_i > x_{i-1}\ \&\ x_i > x_{i+1})\ \|\|\ (x_i < x_{i-1}\ \&\ x_i < x_{i+1})$
Frequency Domain	Peak Amplitude	A$_p$	$max(|S|)$
	Signal Power	SP	$\sum_{i=1}^{N} S(i)$
	Mean Fluctuation Frequency	FF	$\frac{\sum_{i=1}^{N} S(i)X(i)}{\sum_{i=1}^{N} S(i)}$
	Spectral Entropy	SE	$-\sum_{i=1}^{N} d_i \log_{10}(d_i)$
	Shape Factor	SF	$\frac{1}{\bar{S}}\left(\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(S_i - \bar{S})^2}\right)$
Tim	Mean	$\bar{C}$	$\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} C_{i,j}$

Root Mean Square	RMS$_{TF}$	$\sqrt{\dfrac{1}{N^2}\displaystyle\sum_{i=1}^{N}\sum_{j=1}^{N} C_{i,j}^2}$	
Variance	VAR$_{TF}$	$\dfrac{1}{N-1}\left(\displaystyle\sum_{i=1}^{N}\sum_{j=1}^{N}|C_{i,j}-\bar{C}|\right)^2$	
Quartile Frequency 1	QFR1	f_{25}/f_{75}	
Quartile Frequency 2	QFR2	f_{25}/f_{90}	
Quartile Frequency 3	QFR3	f_{50}/f_{75}	
Quartile Frequency 4	QFR4	f_{50}/f_{90}	
Mean Crossing Irregularity	MCI	$\dfrac{1}{\bar{C}}\sqrt{VAR_{TF}}$	

5.2.4 Selection of Features

There are a total of 546 features extracted from 21 post-processed sensor signals. Some of the features are more essential than others, while some may be intrinsically linked or overlapped. The unnecessary and redundant features not only use extra memory storage but also raise over-fitting problems, over-killing the computing power, and all the time associated with it. With the use of the partial mutual information (PMI) algorithm [78], redundancy between the features can be identified. The PMI algorithm calculates only that information between input feature x and target output y which has not been accounted for the calculation of information between previously selected feature set w and output y. PMI for distinct data samples is represented as:

$$\text{PMI} = \frac{1}{N}\sum_{i=1}^{N}\log\frac{p(x',y')}{p(x')p(y')} \tag{5-2}$$

where x' = x - E[x/w] and y' = y - E[y/w]. Operator E[•] represents the expected value of a variable x or y, given variables w have already been selected. p(x'), p(y'), and p(x',y') are the respective univariate and joint probability densities estimated with a total of N samples. Therefore, if some features are already selected in w, then variables x' and y' are the residuals of a new feature x and the output y regarding the selected features in w, respectively.

Selecting a higher number of useful features can improve the performance of the data driven models. However, more features can also increase the complexity of the models. Therefore, the Akaike information criterion (AIC) is applied to identify the optimal number of features [79].

$$\text{AIC} = N\,Ln\left(\frac{1}{N}\sum_{i=1}^{N} u_i^2\right) + 2p \tag{5-3}$$

where N is the number of samples, p is the number of model parameters and u is the residual of the desired outputs.

The procedure of selecting useful features is summarised as follows:

1) Calculate the PMI value of all the features using Equation (5-2)
2) Sort all the features in a descending order based on their PMI values
3) Apply Equation (5-3) on all the features one by one, starting from the features with the highest PMI
4) Keep selecting the features if the AIC decreases. Otherwise, terminate the selecting procedure

It is worth noting that the feature selection is different from dimensionality reduction. Here the problem is not to reduce the dimension but to remove the irrelevant and unwanted features while still having a sufficient number of features to keep the model's generalization intact. It should be noted that the feature selection is only implemented before the development of data driven models to determine suitable inputs for data driven models. Once the most relevant features are determined, this step will no longer be required, and the models can be trained and tested with the short-listed features. The program for the signal processing is provided in Appendix 1.

5.3 Data-Driven Models

Suitable data driven models are required to map the complex relationship between the sensor data and the mass flow rate of solids to be measured. In view of their advantages in multiphase flow measurement [41], as outlined in the Introduction section, three data driven models, including ANN, SVM, and CNN, are selected and deployed in this study. The architectures of the BP-ANN and SVM models are shown in

Figure 5-5 and Figure 5-6, respectively. The input dimension of these models depends on the number of selected features. The number of neurons in the hidden layer is determined by [45]

$$L \leq 2M + 1 \tag{5-4}$$

$$L \leq \frac{N}{M + 1} \tag{5-5}$$

where *L* is the number of neurons and N and M are the numbers of samples and features, respectively.

The ANN model is developed by training a network of neurons to represent the inherent relationship between the input data and the intended measurand on output. The ANN model, used in this research, is a three-layer feedforward network that is trained based on the backpropagation learning method. Nodes present in the hidden layer receive values from the input layer and passes a quantitative value on through a pre-defined activation function. An activation function in the ANN model performs the complex computation in the hidden layer and then transfers the outcome to the output layer. Activation functions are mainly used to introduce the non-linearity in the model [46].

SVM models are one of the most popular and widely implemented data driven algorithms which perform linear regression in a high dimensional feature space and tend to reduce model complexity. The type of SVM model, implemented in this research, is regression for the measurement of mass flow rate of solids. As the output is a real and continuous number, therefore it becomes very challenging to predict the information at hand which has infinite possibilities. In the case of regression, a margin of tolerance is set in approximation to the SVM which would have already requested from the problem. SVM regression is considered a non-parametric technique because it relies on kernel functions. The kernel trick is useful to minimize the computational complexity of the input data which is comprised of several statistical features in the original space.

The theory of artificial neural networks has given rise to the idea of deep learning. Unlike the neural networks that were developed many years ago, modern methods of deep learning have cracked the code for scale on big data, generalization, and training stability. These are the models powered by the information that can reach the best statistical accuracy when exposed to a high volume of data. Despite all the benefits that modern deep learning models have on traditional models, it is not guaranteed that deep learning models will always give an output of the best accuracy. There are many types of deep learning models available. As shown in

Figure 5-7, the CNN model is the one implemented primarily in this research due to its simplicity and inexpensive cost of computing, compared to other deep learning models [80]. CNN is a deep learning class that is most widely used for image data processing and evaluation. The CNN model is able to interpret the information lying in the data from multiple sensors and suppress the

redundancy through deriving the layer structures. The CNN model, therefore, accepts the direct post-processed sensor signals as inputs. The complexity and size of the layers in the CNN system are the key factors that must be addressed in the design of a CNN architecture. Including a larger number of convolution layers may help improve design quality by improving the consistency of the derived elements. However, at the same time, it can also increase the cost of processing and generate over-fitting issues. Therefore, it is a common practice to begin this layer design process with a very shallow model and then gradually add additional layers of different window sizes and stride values until the test accuracy reaches saturation.
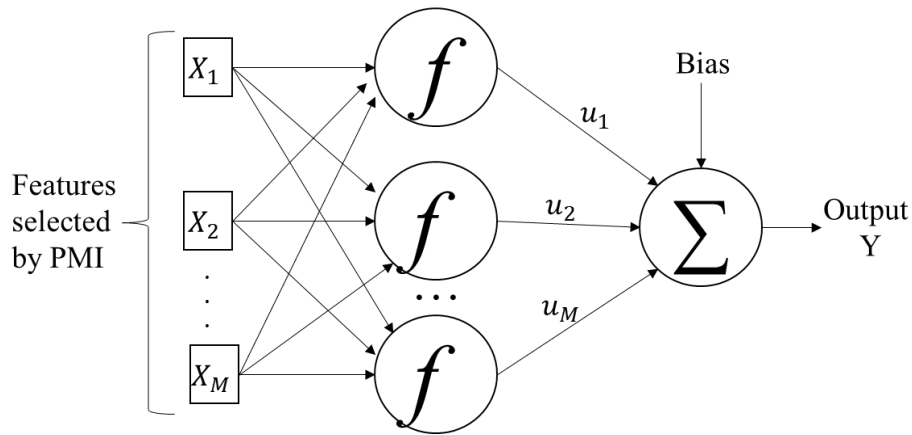
Figure 5-5: Structure of ANN.
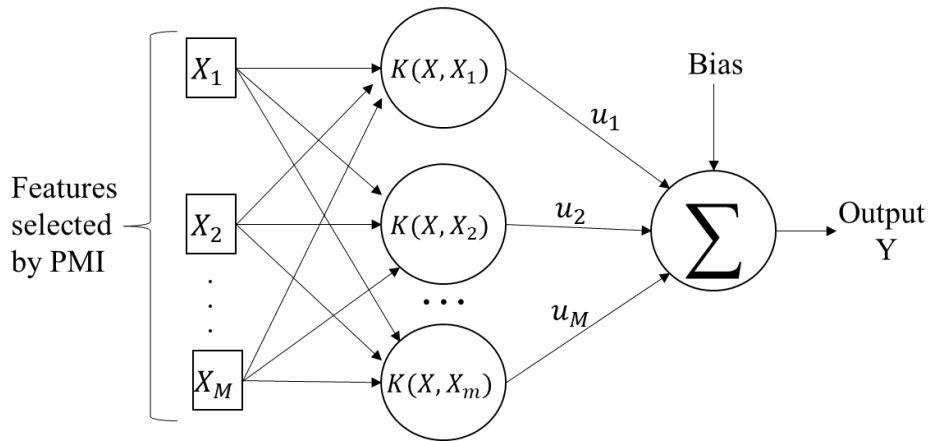
Figure 5-6. Structure of SVM.
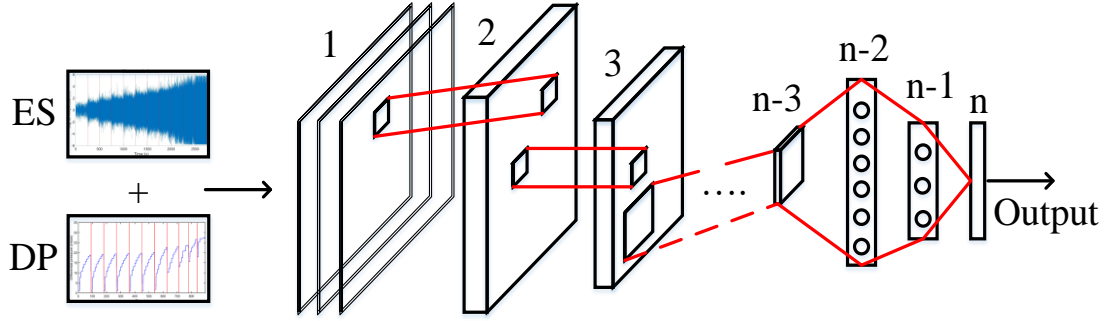
Figure 5-7: Structure of CNN.

5.3.1 Optimization of Number of Sensors Required

The usefulness of a sensor can be investigated through a model based selection method. The model is in a fixed structure with the same feature extraction mechanism. In this case, the sensor signal is the only factor affecting the performance of the model. As the SVM model has fixed three-layer structure and it takes a moderate volume of dataset and thus less time for training, therefore SVM model is applied to evaluate the importance of ring and arc-shaped electrostatic sensors and the DP transducer in this study. Three most commonly used features, including RMS, standard deviation and variance, are extracted from the post-processed sensor signal as inputs to the SVM model. The model output is mass flow rate of solids. Normalized root mean square error (NRMSE) is applied to assess the performance of the SVM models and it is determined from

$$NRMSE = \frac{1}{\bar{y}}\sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k - \widehat{y_k})^2} \qquad (5\text{-}6)$$

where y_k ($k = 1$ to N) is the reference mass flow rate of solids, $\bar{y}$ is the mean of the mass flow rate of y_k, $\widehat{y_k}$ is the predicted solids mass flow rate and N is the total number of samples used.

The sensor selection procedure is summarised as follows:

1) Initialise the selected sensor set S with null, potential sensor set P with E1, E2, E3, E4, E5, E6, E7, E8, DP
2) Develop the SVM model based on each sensor from P combined with all the sensors in S
3) Find out the sensor which makes the model developed in step (2) yielding lowest NRMSE

4) Move the sensor from P to S

5) Repeat step (2) until P is empty

A total of 9 combinations will be obtained from step (4) and the best combination is the determined by the lowest NRMSE or the NRMSE has reached a certain level.

5.4 Experiments with Different Air Velocities

5.4.1 Measurement Principle

The suggested solution to the measurement problem, as shown in

Figure 5-8, is based on multi-modal sensing and data driven models. The multi-modal sensing system includes an array of ring-shaped electrostatic sensors, four arrays of arc-shaped electrostatic sensors and a DP transducer. Electrostatic sensors are mainly chosen for the mass flow rate measurement of solids due to their non-intrusive character, simple structure, and inexpensive capital cost. The DP transducer can determine the drop in line pressure caused by the varying concentration of solids in the pneumatic pipeline. After the signals from all the sensors are denoised (Section 5.2.2), the statistical features are extracted from the data from all the electrostatic sensors and the DP transducer to train the data driven model. The required solids velocity is measured using all the ring and arc-shaped electrostatic sensors with data fusion and the cross-correlation signal processing technique [16]. Adding the particle velocity in the data driven models can potentially result in accurate and repeatable predictions of the mass flow rate of solids.

A data driven model is used to establish the complex relationship between the sensor data and the mass flow rate of solids to be measured. To determine the optimal number of inputs for the data driven model, extraction and selection of statistical features are undertaken prior to the development of the model. The optimal structure of the model is developed during the training and validation process. For the measurement of mass flow rate of solids, the data driven model is trained with the selected statistical features extracted from the post-processed sensor data in the time, frequency and the time-frequency domains. The raw sensor data is processed prior to the extraction of statistical features to avoid the false training with absurd data labels. The proposed measurement strategy is implemented in the MATLAB version 2020b. The program for the proposed measurement strategy is provided in Appendix 2.
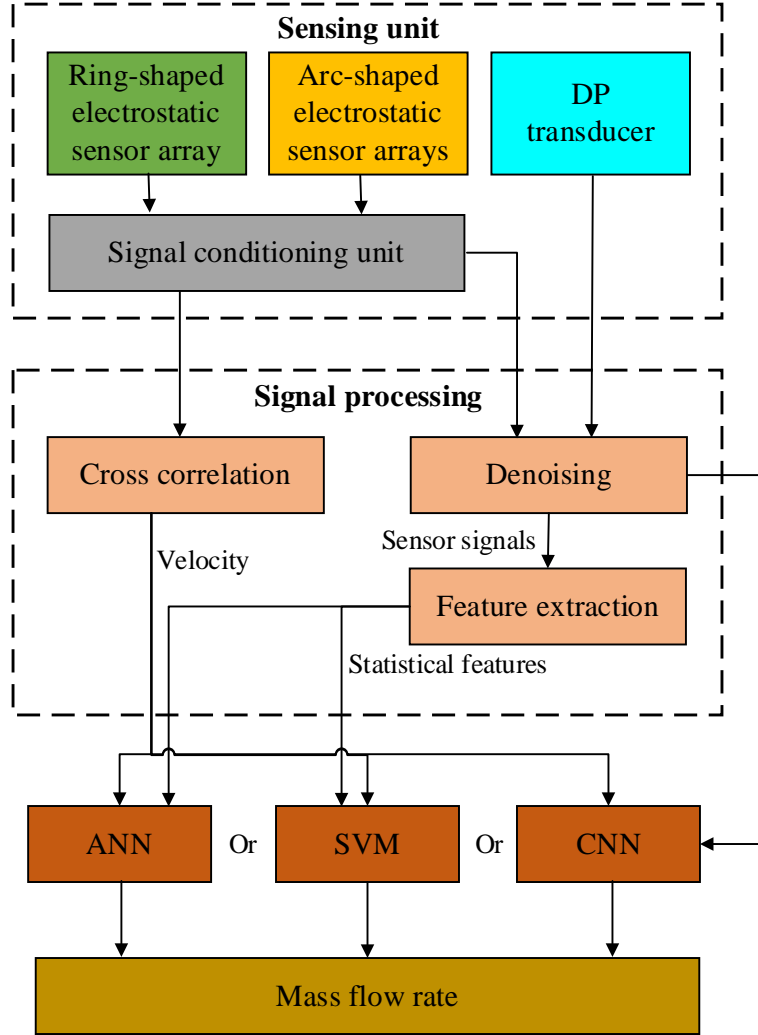
Figure 5-8: Principle of mass flow rate measurement through modal sensing and data driven modelling.

5.4.2 Experimental Conditions

The data to test the models are classified into seen and unseen conditions. The seen test conditions include mass flow rates of solids which are also used for the training of models as listed in Table 5-2. All the data in Table 5-2 is divided into 70% for training and 30% for testing. However, the unseen test conditions are kept entirely different from the mass flow rates of solids used for the training of models as listed in Table 5-3. All the unseen test conditions are chosen in a way to assess the prediction accuracy of a data driven model for the points lying between two adjacent training points. Two of the unseen mass flow rate conditions, 3.2 g/s and 35.8 g/s, are chosen from

outside the training and test data limits to evaluate the generalization ability of the models. The large gap of values between the higher mass flow rates of solids in the training data is due to nonlinear relationship between the VFD control and the mass flow rate of solids. The data-driven models are made generalized enough to predict any value lying between two heavily spaced training conditions.

As the velocity and mass flow rate of solids is controlled, respectively, with the suction pump and the vibratory feeder through the VFD, the set points for velocity and mass flow rate are chosen based on the distribution of frequency values of the VFD. For example, a suitable range of VFD frequencies to control the vibratory feeder lies between 30 and 50 Hz. Therefore, eleven set points are chosen for mass flow rate by varying the frequency between 30 and 50 Hz with a difference of 2 Hz in every set point. It is worth noting that the relationship between the VFD frequency and the mass flow rates is not linear and hence this is the reason of having larger gaps at higher mass flow rates. It should be noted that at lower air velocity conditions, higher mass flow rates are not workable because the particles will not be in a suspension in the pipeline, which may lead to blockage of the pipeline. The practically inappropriate combinations of a lower particle velocity and a higher mass flow rate of solids are marked out with a dash in Table 5-2 and Table 5-3.

Table 5-2: Combinations of air velocity and mass flow rate for training and testing of Models

		Mass flow rate (g/s)				
		5.4	8.6	14.0	21.4	30.8
Air velocity (m/s)	10.1	✓	✓	✓	–	–
	14.5	✓	✓	✓	✓	–
	18.6	✓	✓	✓	✓	–
	22.1	✓	✓	✓	✓	✓
	25.3	✓	✓	✓	✓	✓
	28.2	✓	✓	✓	✓	✓
	31.1	✓	✓	✓	✓	✓
	33.0	✓	✓	✓	✓	✓
	36.0	✓	✓	✓	✓	✓

Table 5-3: Combinations of air velocity and mass flow rate for testing of models with unseen conditions

		Mass flow rate (g/s)					
		3.2	7.7	11.3	17.6	25.7	35.8
Air velocity (m/s)	10.1	✓	✓	✓	✓	–	–
	14.5	✓	✓	✓	✓	–	–
	18.6	✓	✓	✓	✓	✓	–
	22.1	✓	✓	✓	✓	✓	✓
	25.3	✓	✓	✓	✓	✓	✓
	28.2	✓	✓	✓	✓	✓	✓
	31.1	✓	✓	✓	✓	✓	✓
	33.0	✓	✓	✓	✓	✓	✓
	36.0	✓	✓	✓	✓	✓	✓

5.4.3 Sensor Data Under Horizontal Pipe Orientation

Fig. 5-9 shows the post-processed signals under all the conditions from the ring-shaped electrostatic sensor (E1 as presented in Chapter 3). All the boundaries, highlighted with red vertical lines, represent the sensor data under seen and unseen mass flow rates of solids, as listed in Table 5-2 and Table 5-3, increasing from left to right. Sensor data plotted under the same mass flow rate condition also belong to the different air velocities, increasing from left to right. As can be seen, the amplitude of the signals increases with the mass flow rate as well as particle velocity. The signal conditioning unit is powered from a ±3.3 V power supply, therefore the magnitude of all the signals remains within ±3.3 V without saturation. Meanwhile, the signal conditioning unit is designed such that the range of the signals under all unseen test conditions is not out of the limits of the supply voltage ±3.3 V. The signals from other sensors are similar in terms signal patterns and ranges as all the signal conditioning units are powered from the same ±3.3 V power supply.
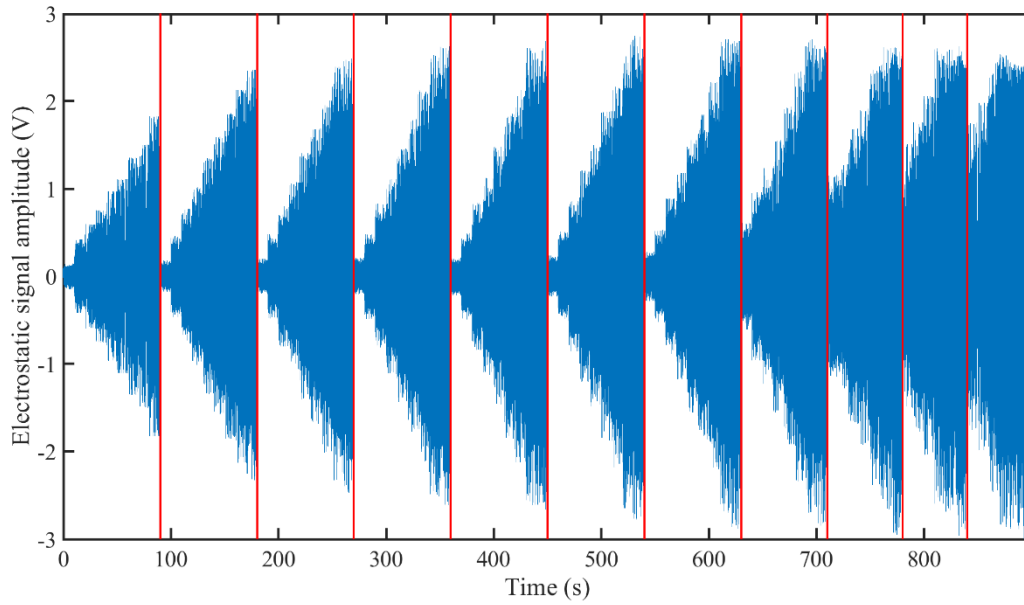
Fig. 5-9. Post-processed signals from the electrostatic sensor E1.

Figure 5-10 shows the relationship between the RMS of a post-processed signal from the E1 electrode of the ring-shaped electrostatic sensor and the mass flow rate of solids under different air velocity conditions. Each data point on the graph also indicates the uncertainty in both mass flow rate and RMS of electrostatic signal that is calculated from the reference data of mass flow rate of solids and the recorded electrostatic signal under different conditions. The magnitude of electrostatic signal increases nonlinearly by increasing the mass flow rate and the velocity of solids. Since a high volume of large solids (less surface charge) is concentrated at the bottom of the pipe due to gravitational effect, the RMS of the signal, obtained from the arc-shaped electrode at the bottom of the pipe, is lower than others, as shown in Figure 5-11. Nevertheless, the RMS values from the electrodes at the front and back are comparatively higher than that at the bottom owing to the higher number of small particles in the suspension. The RMS of the signal from the sensor at the top is lower than those at the front and the back due to the lower solids concentration along the top of the pipe. However, at much lower mass flow rates (<10 g/s), the outputs from the top, front and the back electrodes yield very similar RMS magnitudes due to the fully suspended flow pattern except some large particles moving relatively slow at the bottom giving rise to a lower RMS. Figure 5-12 shows the dependency of the differential pressure on the mass flow rate and velocity of solids in the pipeline. Drop in line pressure also increases nonlinearly with the mass

139

flow rate of solids and air velocity due to higher friction faced by the solids with the pipe wall. The DP signal at lowest velocity is much lower than the signals at other velocity because the impact of air velocity 10.1 m/s is relatively lower on the line pressure. Each data point on the plot also shows the uncertainty calculated for reference data of mass flow rate of solids and the measured line pressure with the DP transducer.
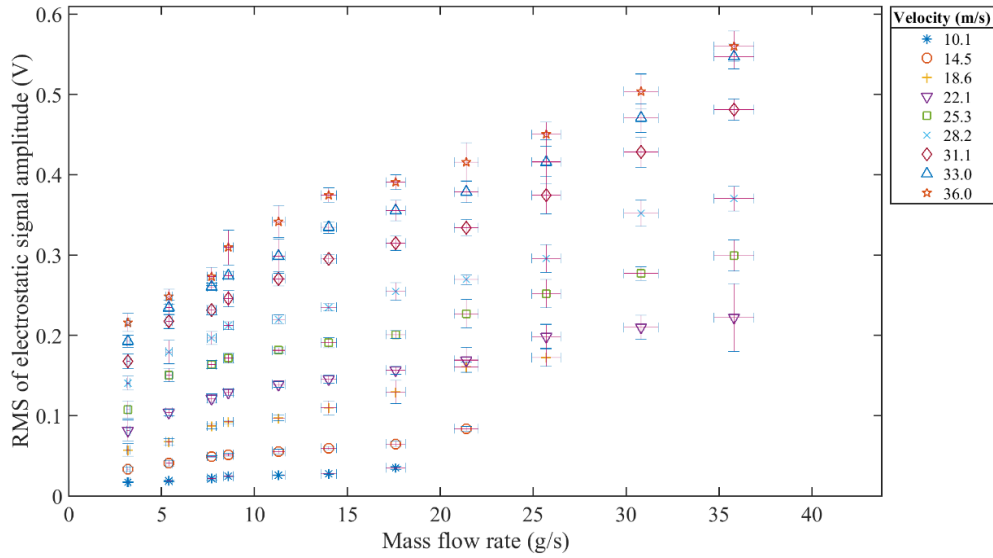
Figure 5-10: RMS of the electrostatic signal from ring-shaped electrostatic sensor E1 under horizontal pipe orientation.
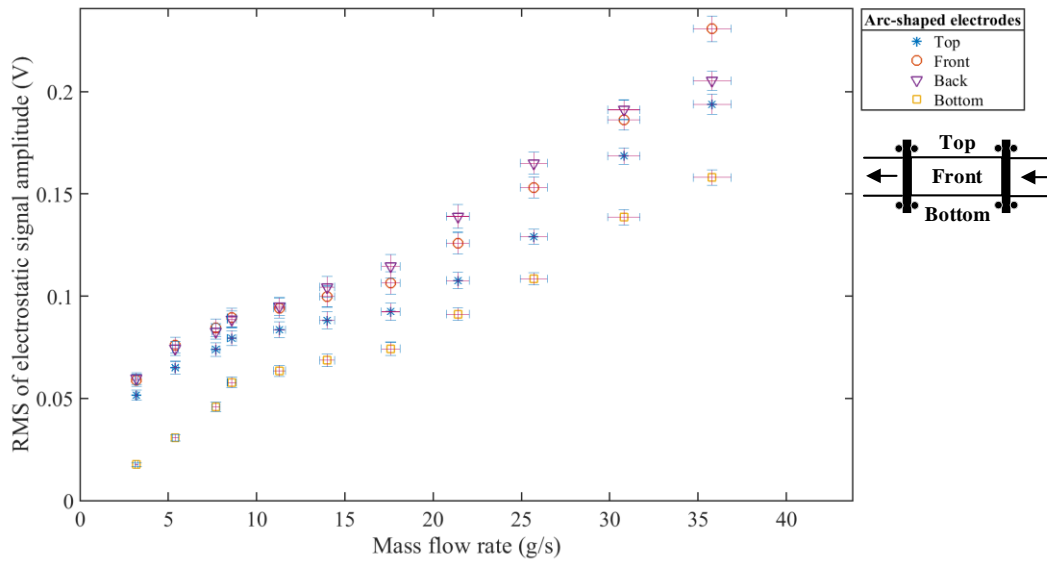
Figure 5-11: RMS of electrostatic signal from arc-shaped electrostatic sensors E5a, E5b, E5c, E5d with air velocity 22.1 m/s under horizontal pipe orientation.

140
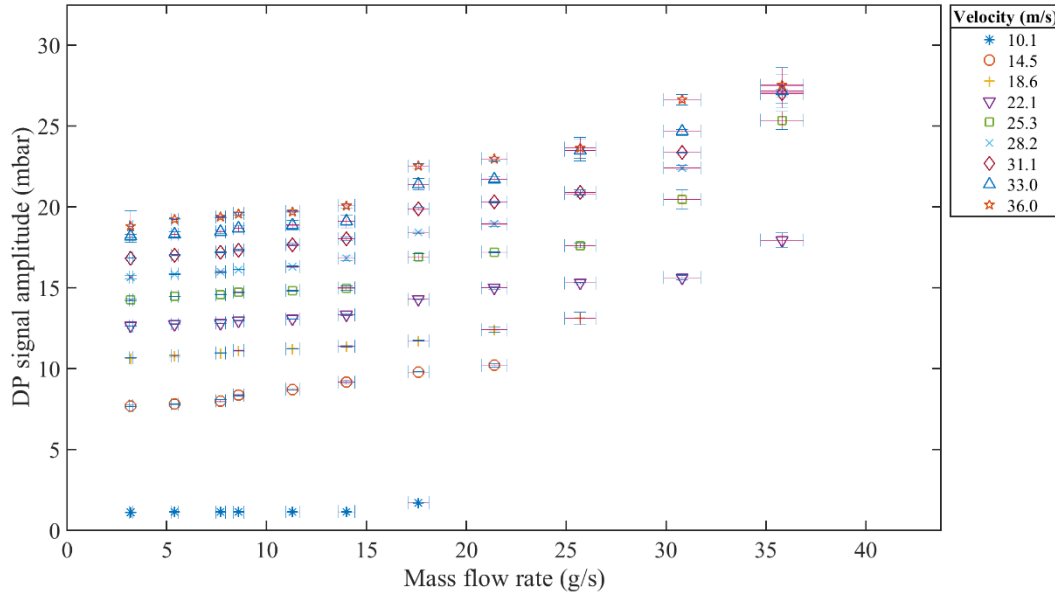
Figure 5-12: Output of the DP transducer under horizontal pipe orientation.

5.4.4 Sensor Data Under Vertical Pipe Orientation

For each pair of air velocity and mass flow rate of solids, the magnitude of the signals from ring-shaped sensors in vertical pipe orientation is slightly increased compared to the horizontal pipe orientation due to higher gravitational effect and solids-pipe wall friction as shown in Figure 5-13. However, a substantial variation is observed in the localized charge values due to change in flow regime under vertical pipe orientation, as shown in Figure 5-14. The larger solids with less surface charge, moving along the upper horizontal pipeline, hit with the wall of the pipe bend and start moving along the left side of the vertical pipeline, resulting in lower RMS charge. The smaller, but more solids at the front, back and the right regions of the pipe yield higher RMS values of the signals from the sensors at such locations. Meanwhile, after downwards turning at the bend, fine particles at the right of the pipe give the lowest RMS value. The variations in the DP signal amplitude with the mass flow rate of solids under vertical pipe orientation is demonstrated in Figure 5-15. The drop in the line pressure in this case is escalated a little compared to that in horizontal pipe orientation due to higher effect of gravity on the gas, solids, and the solids-wall friction at the pipe bend.
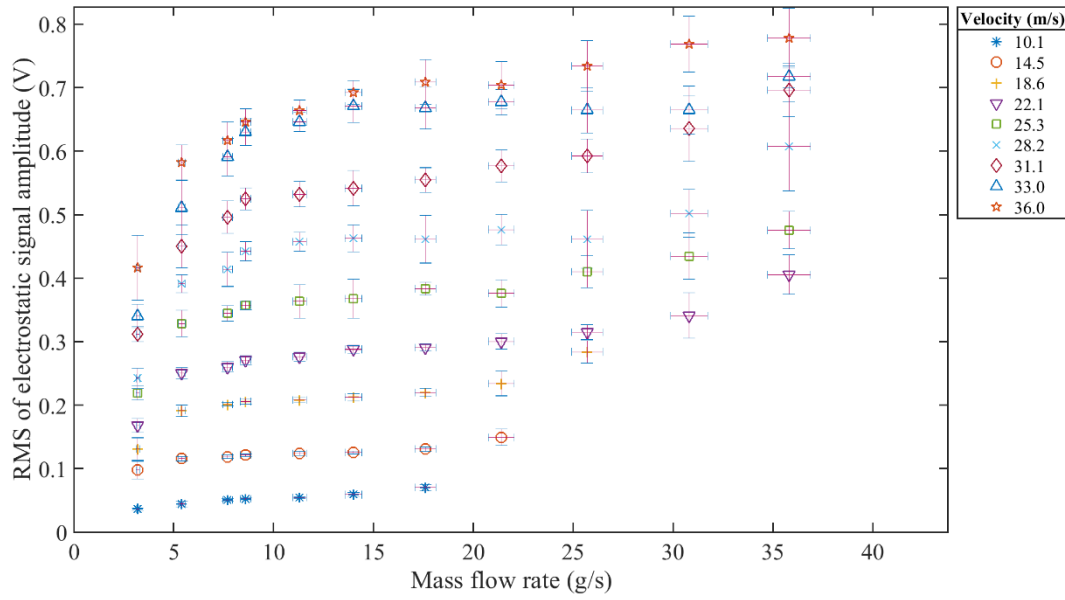
Figure 5-13: RMS of the electrostatic signal from ring-shaped electrostatic sensor E1 under vertical pipe orientation.
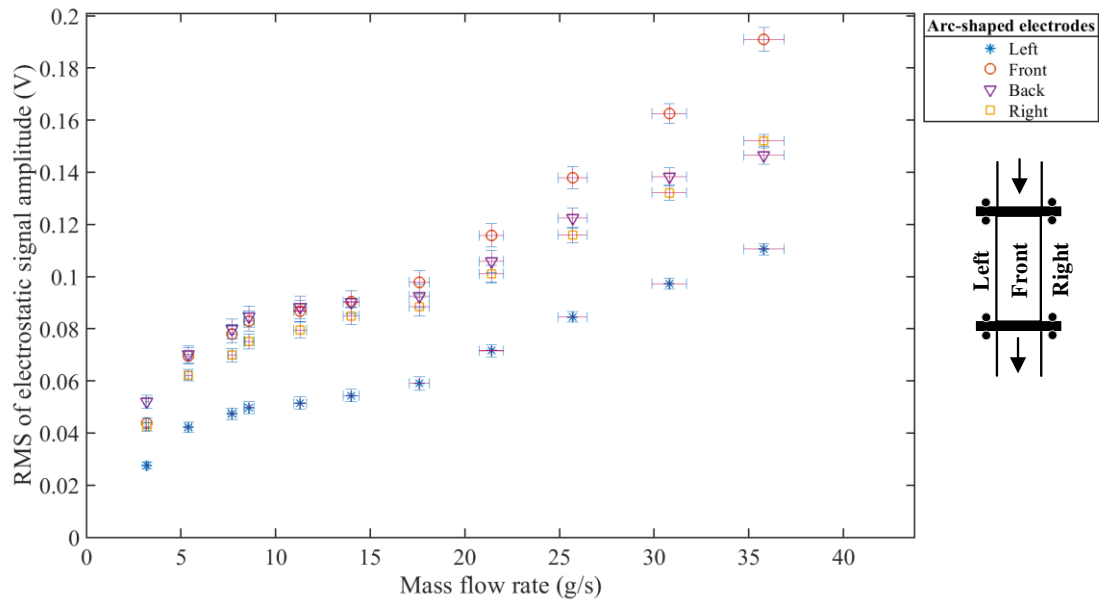
Figure 5-14: RMS of electrostatic signal from arc-shaped electrostatic sensors E5a, E5b, E5c, E5d with air velocity 22.1 m/s under vertical pipe orientation.
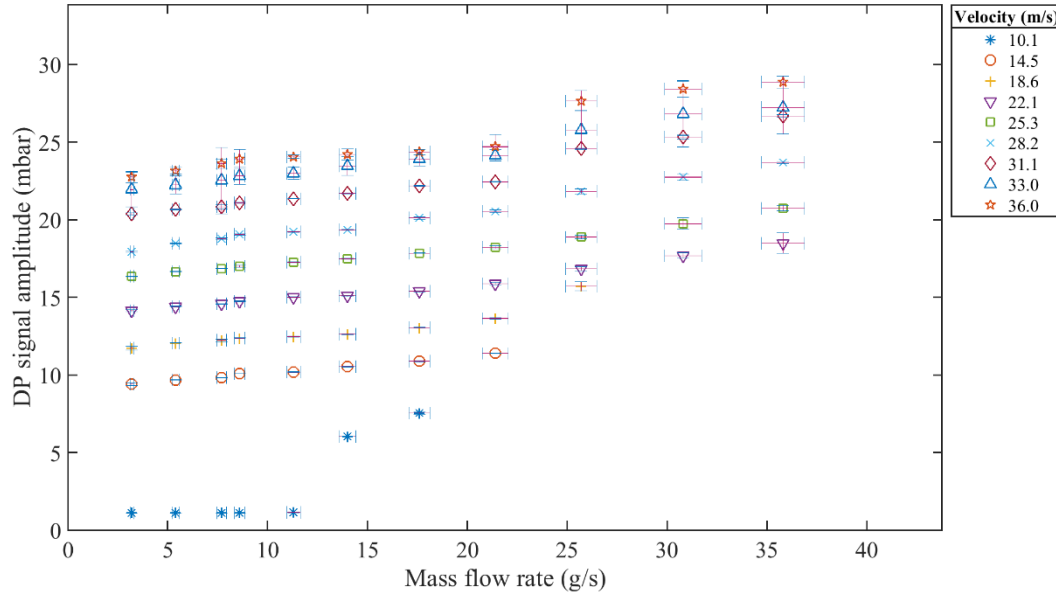
Figure 5-15: Output of the DP transducer under vertical pipe orientation.

5.4.5 Selection of Important Sensors

The training and test data for the purpose of sensor selection come from the experimental data collected under horizontal pipe orientation, with a fixed air velocity of 22.1 m/s and all the mass flow rate conditions listed in Table 5-2 and Table 5-3. Figure 5-16 shows the performance of the SVM models which were developed based on the signals from each sensor respectively. The error bars indicate the standard deviation of errors in predicted mass flow rates of solids [41]. Since each set of arc-shaped electrodes is made up of top, front, back and the bottom arcs, it provides localized and yet more precise information about the whole cross-section of pipe. Therefore, NRMSE calculated with arc-shaped electrodes (E5-E8) is lower compared to the others.
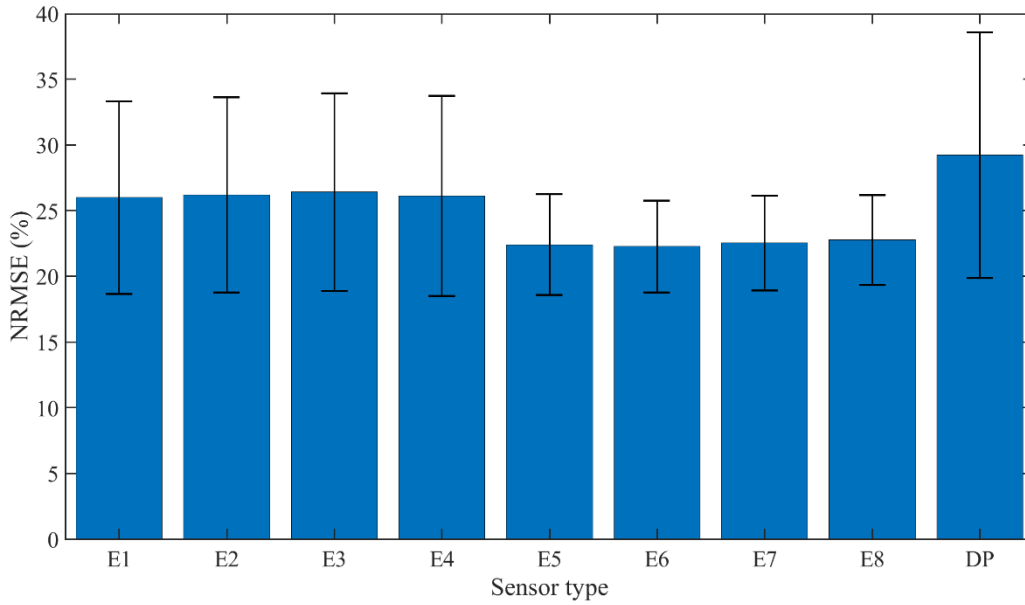
Figure 5-16: NRMSE of predicted mass flow rates based on the individual sensors.

Table 5-4 describes the name of the sensors used in each combination. The NRMSE values and standard deviations of each combination are shown in Figure 5-17.The NRMSE does not improve much after the combination (vi), therefore the combination (vi) is selected for the processing in subsequent sections.

Table 5-4: Combinations of Sensors Applied to SVM Model and Their NRMSE

Combination label	Sensor combination	NRMSE (%)
i	E5	22.29
ii	E5, E6	21.63
iii	E5, E6, E7	20.85
iv	E5, E6, E7, E1	18.56
v	E5, E6, E7, E1, E2	17.10
vi	E5, E6, E7, E1, E2, DP	16.07
vii	E5, E6, E7, E1, E2, DP, E3	16.05
viii	E5, E6, E7, E1, E2, DP, E3, E4	16.05
ix	E5, E6, E7, E1, E2, DP, E3, E4, E8	16.04
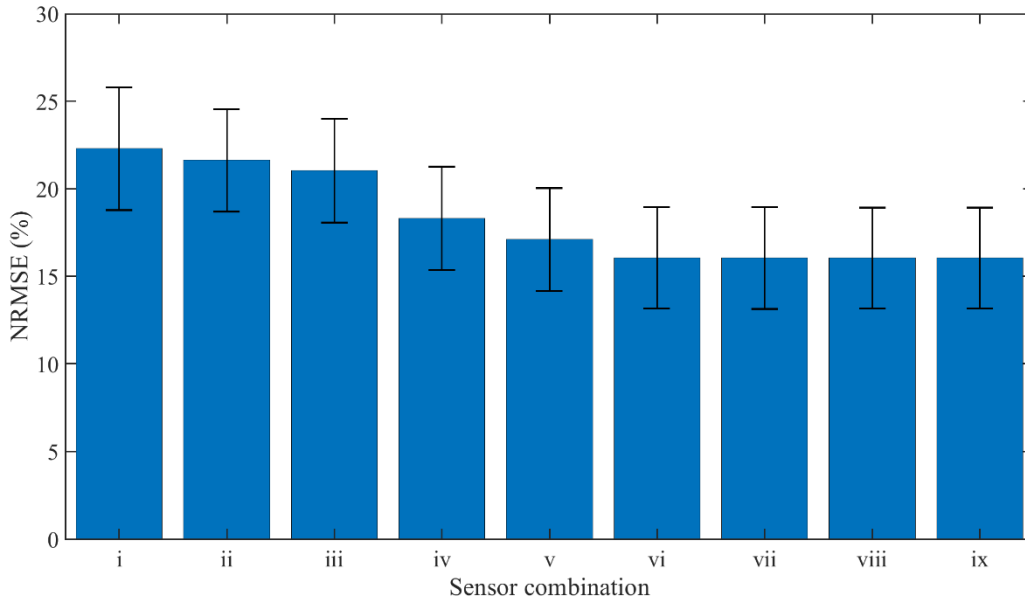
Figure 5-17: NRMSE of predicted mass flow rates based on the combination of sensors.

5.4.6 Selection of Window Size

Since the accuracy of the data driven models is highly dependent on the number of samples used to train the models, it is crucial to find the optimum value of the window size, which defines the length of the data that must be taken to extract the features to create a single sample. The ratio of the overall data length to the window size defines the number of samples that can be used for the training and the testing of the models. Therefore, with a fixed overall data length, a smaller window size will produce more number of samples. The overall data length is a product of data recording duration under one mass flow rate and velocity case (30 s), number of mass flow rates of solids (11) and the number of air velocities (9). As the flow of particles from the vibratory feeder via the test section of the pneumatic pipeline to the suction pump is often uneven, which can result in slight variations in the sensor signals, a small window size, e.g. 2 s, can result in large fluctuations in the extracted features. Depending on the robustness of the models, different window sizes can impact the accuracy of the models differently. An iterative analysis was carried out to evaluate the optimal size of the window for each of the data driven models, as listed in Table 5-5. A window size below the minimum value can negatively influence the performance of the models. A larger window size is good for the accuracy of the models, but more data are acquired and processed. The CNN model, in this case, is ideal in terms of the window size requirement. However, in this

145

study, the basis for making a comparison between the three models is on the prediction accuracy and not on the window size requirement. Therefore, to have a fair comparison between the three models, a larger window size of 10 s is used in this study. However, for practical applications a window size of 1-2 s is appropriate to track significant changes in flow conditions. In this case, the CNN model is advantageous as it requires 2 s of data to yield the satisfactory predictions.

Table 5-5: Data slicing of ANN, SVM and CNN

Model	Overall data length(s)	Minimum window size (s)	No. of samples
ANN	2970	10	297
SVM	2970	6	495
CNN	2970	2	1485

5.4.7 Shortlisted Features

For the purpose of training the data driven models, a total of 113 features are selected from the available 390 (Table 5-1) through the use of the PMI algorithm. Table 5-6 summarizes these features that contain all the following two characteristics determined through the PMI value of each feature:

1) Selected features have a high degree of relevance with the target output.
2) Selected features have a low value of redundancy with each other.

The features are selected mainly because they provide substantial information amongst others to determine the mass flow rate of solids. The number of selected features is different for each category of sensors and this is mainly due to the reason that for some of the statistical features, a sensor that appears first in the array of electrostatic sensors (E1) has provided enough information. Selection of the same statistical feature for the sensor appearing at the second place in the array (E2) will yield a redundant information, for example, WL in this case. A similar trend is also seen for the feature count in the arc-shaped sensors, for example, E5a, E6a and E7a. It has to be noted that the orientation of the pipe does not affect the magnitude of the signals from sensors and hence the patterns of the signals. Therefore, all the selected features will remain same regardless of the pipe orientation. It must be pointed out that, despite the fact that we have a certain knowledge of correlations between the sensor signals, the DP transducer output and the flow characteristics, it is not straightforward to interpret the exact physical meanings of all the features. This is a common challenge when data driven modelling techniques are applied to resolve difficult measurement

problems. With the development of on-going analytical modelling of gas-solids two-phase flow, engineering judgements in the application of modern machine learning techniques may have a part to play in future.

Table 5-6: Features selected by PMI

Sensor	Selected features	Feature count
E1	RMS_T, σ, VAR_T, PP, SSI, WL, A_p, $\bar{C}$, RMS_{TF}, VAR_{TF}	10
E2	RMS_T, σ, VAR_T, PP, SSI, A_p, $Mean_{TF}$, VAR_{TF}	8
E5a	RMS_T, σ, VAR_T, PP, SSI, WL, A_p, $\bar{C}$, RMS_{TF}, VAR_{TF}	10
E5b	RMS_T, σ, ZC, VAR_T, PP, SSI, SSL, A_p, $\bar{C}$, RMS_{TF}	10
E5c	RMS_T, σ, PP, SSL, $\bar{C}$, RMS_{TF}	6
E5d	RMS_T, σ, VAR_T, PP, SSI, WL, A_p, $\bar{C}$, RMS_{TF}, VAR_{TF}	10
E6a	RMS_T, σ, VAR_T, PP, SSI, A_p, $\bar{C}$, RMS_{TF}	8
E6b	RMS_T, σ, VAR_T, PP, SSI, SSL, A_p, RMS_{TF}	8
E6c	RMS_T, σ, A_p, $\bar{C}$, RMS_{TF}	5
E6d	RMS_T, σ, VAR_T, WL, SSL, A_p, $\bar{C}$, RMS_{TF}, VAR_{TF}	9
E7a	RMS_T, σ, VAR_T, SSI, WL, RMS_{TF}, VAR_{TF}	7
E7b	RMS_T, σ, VAR_T, SSI, A_p, RMS_{TF}, VAR_{TF}	7
E7c	RMS_T, σ	2
E7d	RMS_T, σ, VAR_T, A_p, $\bar{C}$, RMS_{TF}, VAR_{TF}	7
DP	RMS_T, $\bar{x}$, σ, VAR_T, SSI, A_p	6

5.4.8 Development of ANN, SVM and CNN Models

The data driven models are reinitialized and retrained several times with different parameters to develop the optimized structure. The optimized internal parameters of the ANN and the SVM model are listed in Table 5-7. For the development of the ANN model, sigmoid activation function provides a more generic solution for the non-linear input data to map it into mass flow rate of solids. The tuning of weight values of the ANN model is undertaken with backpropagation method owing to the reason that it is simplest in structure and takes less time to converge. However, in the case of the SVM model, radial basis function (RBF) (Section 2.3.2) is one of the most suitable kernel functions for the non-linear input data. RBF is preferred when there is no prior knowledge available about the input data. Based on the nature of the sensor data, RBF has appropriately transformed the input data into another feature space where the data is suitably mapped into the correct mass flow rates of solids. Sequential minimal optimization (SMO) is used as a solver to solve the quadratic programming problem for the SVM model. SMO works as an iterative

147

algorithm that breaks down the whole optimization problem into small sub-problems which can be solved analytically. A separate cost function works in conjunction with SMO to evaluate the solution. The choice for the rest of the parameters of the ANN and SVM models is based on the best trade off between the training time of the models and the level of prediction accuracy.

The layers and parameters of the CNN model are designed and selected through trial and error by keeping in mind the factors mentioned in Section II-F. Development of the CNN model started by keeping the lower number of convolutional layers with a smaller sized convolution window. The number of layers is then increased in each iteration until the accuracy reached saturation. The optimized CNN model, with the parameters listed in

Table 5-8, is comprised of multiple layers that include Convolutional Layer (CL), Batch Normalization Layer (BNL), Max Pooling Layer (MPL), Rectified Linear Unit (ReLu), SoftMax Layer, Dropout Layer, and Fully Connected Layer. In the first layer, 20,000 post-processed samples extracted from each of the 15 sensors in combination (combination vi in Table 5-4) are stored in 15 different rows constituting an image matrix of dimension 15×20000. In the subsequent layers, this image data pass through 12 convolutional layers with different sized convolution windows and the stride values to extract the information. The convolution layer is the core of the CNN model to extract the meaningful information. The dimension of the input data is substantially high, therefore a set of multiple convolution layers with a high range of window sizes is applied to gather most of the useful information from the input data. An optimized number of convolution layers is selected by trading off between prediction accuracy and the processing time by iteratively adding more and more layers. An MPL with a standard window size is only used in the 2nd layer to reduce the dimension of the data to a reasonable level. MPL is chosen over average pooling layer (AVL) due to its supremacy in prediction accuracy as observed during the development process. However, BNL and ReLu are the parts of each hidden layer purpose of which is to standardize the input for each mini batch and to keep the values in positive range, respectively.

Apart from process of designing the layer, there are some parameters which need to be selected prior to training of the model, as summarized in Table 5-9. Batch size in the training process defines the number of images that can be processed by all the layers of CNN in one time. As the dimension of input data is high, therefore a smaller batch size is chosen to process the data

148

appropriately. In order to bring the level of loss function below 5%, all the training data is repeated in 30 epochs. The model parameters are updated in each iteration based on the parameters of the previous iteration with a momentum. Setting the momentum to zero or one defines no contribution or maximal contribution from the previous iteration, respectively. A constructive difference is observed in the performance by keeping the momentum between 0.8 and 1. The value of the initial learning rate lies between 0 and 1. A very small or exceedingly large learning rate can slow down the process or cause the learning to stop prematurely. The CNN model narrows down the volume of information in each epoch and requires a smaller value of learning factor. Therefore, a learning factor is chosen initially to be 0.05 and then decreased in each epoch by a drop factor of 0.002. Lastly, the training process terminates if the number of times the loss function keeps giving the value greater than the previous smallest loss as similar as the validation patience.

Table 5-7: Parameters of the ANN and SVM models

Parameter	ANN	SVM
Type of model	Regression	Regression
Activation/Kernel Function	Sigmoid	RBF
No. of hidden layers	1	1
No. of hidden nodes	35	25
Input dimension	114	114
Iterations Utilized	157	445
Solver	Back propagation	SMO

Table 5-8: Optimized parameters of the CNN model

Layer no.	Name of layer and size
1	15 X 20000 Image Input Layer
2	4 X 4 CL with stride of 4, BNL, ReLu, 4 X 4 MPL
3	4 X 4 CL with stride of 2, BNL, ReLu
4	8 X 8 CL with stride of 4, BNL, ReLu
5	16 X 16 CL with stride of 8, BNL, ReLu
6	16 X 16 CL with stride of 4, BNL, ReLu
7	32 X 32 CL with stride of 8, BNL, ReLu
8	32 X 32 CL with stride of 4, BNL, ReLu
9	32 X 32 CL with stride of 4, BNL, ReLu
10	64 X 64 CL with stride of 8, BNL, ReLu
11	64 X 64 CL with stride of 4, BNL, ReLu
12	64 X 64 CL with stride of 2, BNL, ReLu
13	128 X 128 CL with stride of 2, BNL, ReLu

14	SoftMax Layer
15	Dropout Layer
16	Fully Connected Layer
17	Regression Layer

Table 5-9: Training parameters of the CNN model

Parameter	Value
Batch size	6
Max epochs	30
Momentum	0.95
Initial learning rate	0.05
Learning rate drop factor	0.002
Validation patience	20

5.4.9 Mass Flow Rate Measurement with ANN, SVM and CNN Models Under Horizontal Pipe Orientation

Figure 5-18 to Figure 5-20 show the predicted mass flow rates of solids with the three trained data driven models, respectively, under horizontal pipe orientation. Results for each model are compared with the ideal straight line that shows the proximity of the predicted results to the reference mass flow rate. The models are tested under eleven different mass flow rate conditions where each condition carries the data of nine different velocities. The outputs of the data driven models are based on the random values of the weights and the biases in the structure, therefore the predicted mass flow rates are randomly distributed around the reference lines.

The error in the predicted mass flow rates of solids under unseen test conditions is higher than that under the seen test conditions. Figure 5-21 shows the poor generalization ability of the ANN model with maximum relative error of 18%. The higher prediction error at lower mass flow rates of solids (< 7.7 g/s) is due to the extremely dilute volumetric concentration of solids that results in obtaining a weaker and fluctuating electrostatic signal. The higher air velocities (> 25.3 m/s) cause more dilute two-phase flow that yields higher error in the predictions under unseen mass flow rates. Compared to the ANN model, the SVM performs better due to its improved generalization abilities and keeps the relative error within $\pm13\%$, as shown in Figure 5-22. The SVM model avoids the over-fitting of data by maintaining a balance between the prediction error at validation and the unseen dataset. The validation dataset, in this case, is a portion of seen test conditions. Therefore,

the prediction error with the SVM model for unseen mass flow rates is not significantly higher than those seen mass flow rates. The CNN model performs well when there is a high volume of sensor data that can be used to formulate enough images to input the model. The relative error between the measured and the actual mass flow rates of solids remains within ±2% under all conditions except the mass flow rates 3.2 g/s and 35.8 g/s which lie out of the training data range, as shown in Figure 5-23. The cross-sectionally averaged and localized particle velocity are also input features for the model. For this reason, the error in the predicted mass flow rate of solids is also dependent on the ability of the data driven model to suitably fit all the data with such complexity.

From Figure 5-24 to Figure 5-26, it is evident that the maximum normalized standard deviation (NSTD) of the predicted mass flow rates of solids with the trained ANN, SVM and the CNN models remains within 14%, 10% and 5%, respectively. The comparison between the NSTD values for the three models also shows that the predictions made with the CNN are more repeatable than the ANN and SVM models.
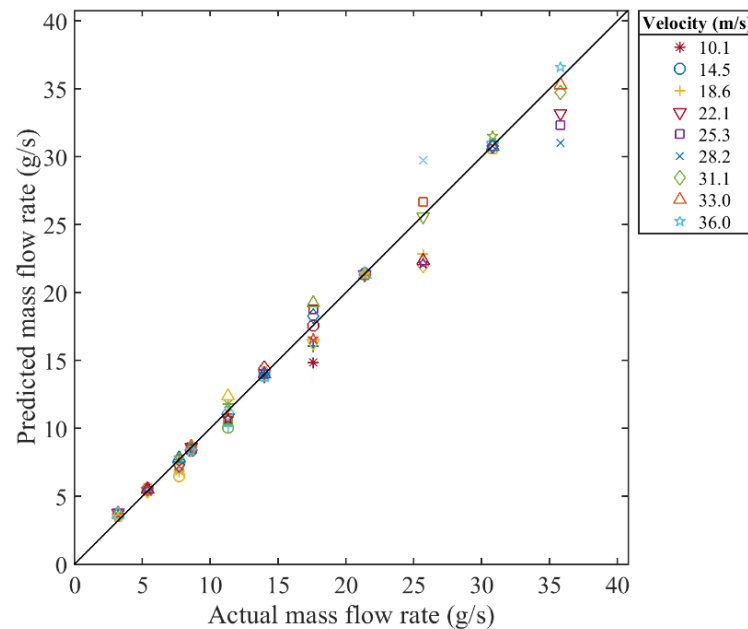
Figure 5-18: Mass flow rate measurement with ANN under horizontal pipe orientation.
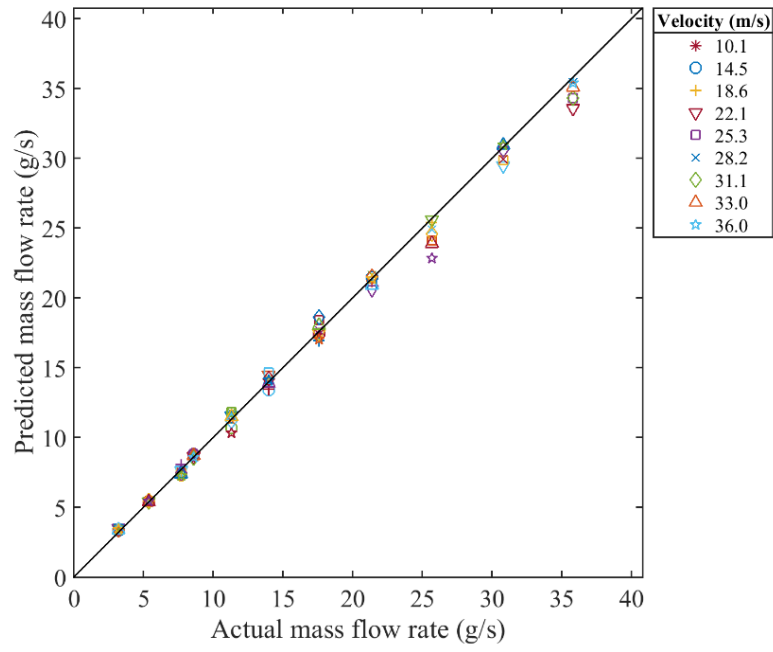
Figure 5-19: Mass flow rate measurement with SVM under horizontal pipe orientation.
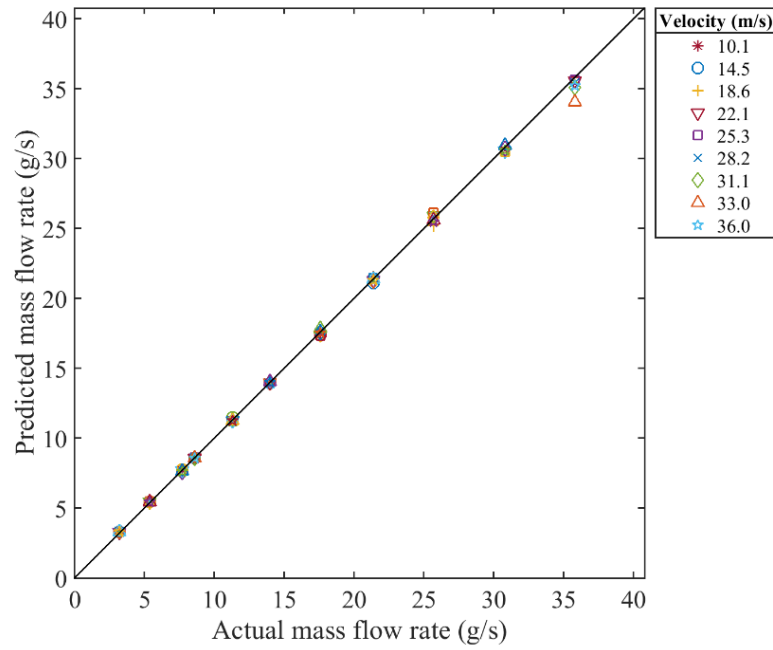
Figure 5-20: Mass flow rate measurement with CNN under horizontal pipe orientation.
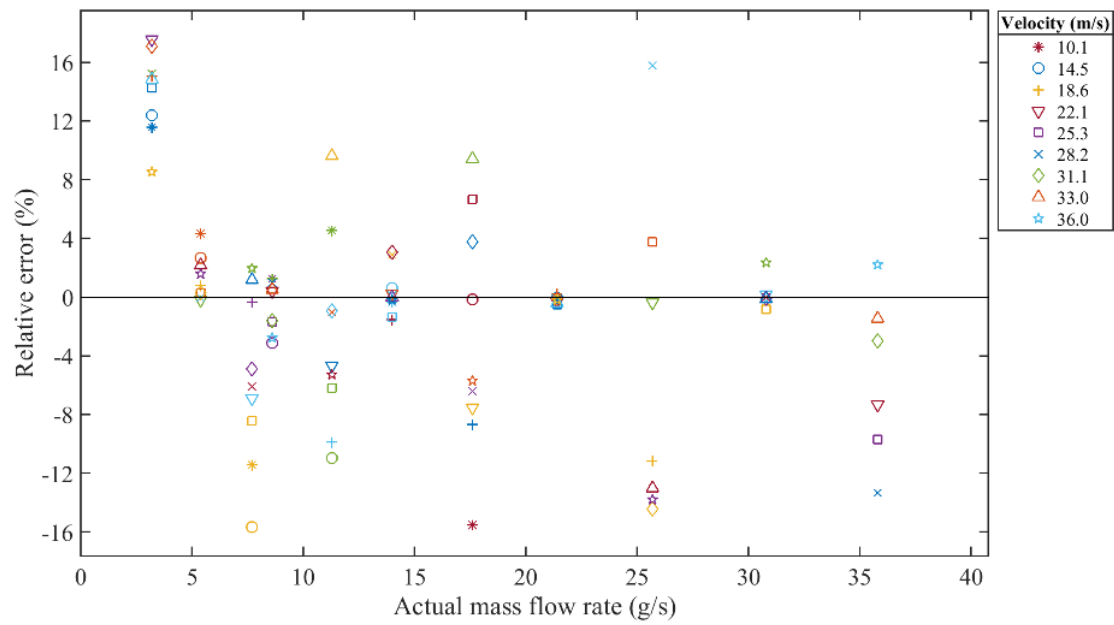
Figure 5-21: Relative error with ANN under horizontal pipe orientation.
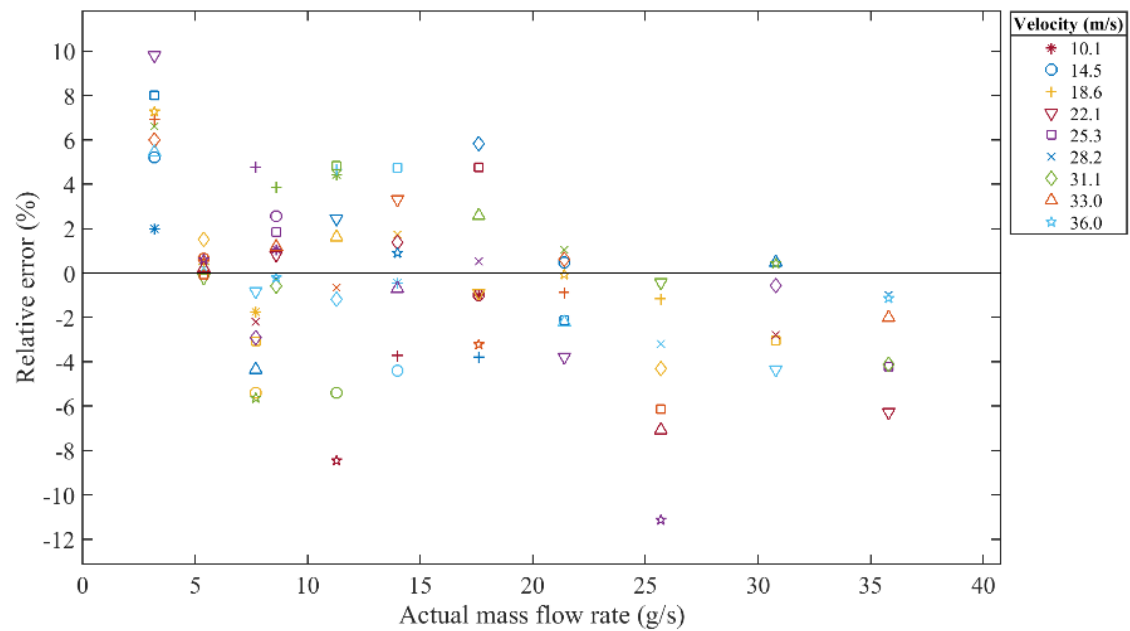
Figure 5-22: Relative error with SVM under horizontal pipe orientation.

153
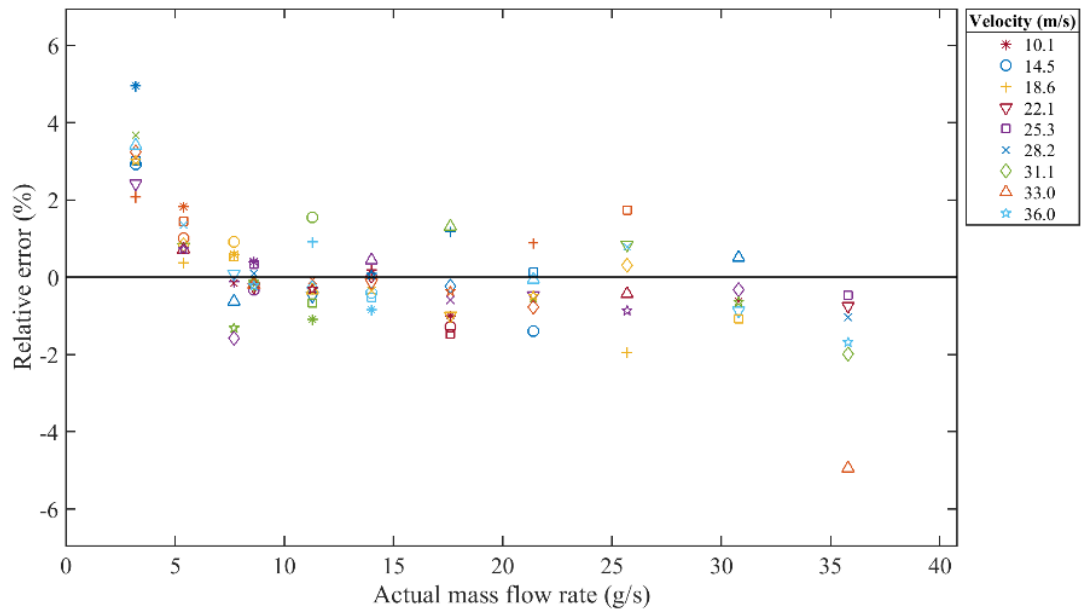
Figure 5-23: Relative error with CNN under horizontal pipe orientation.
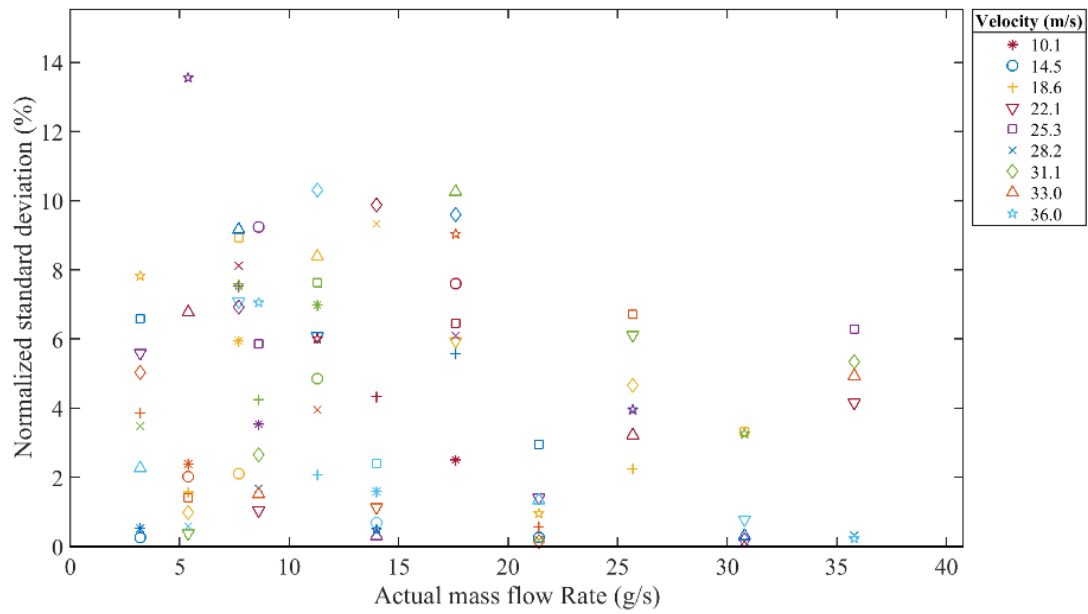
Figure 5-24: Normalized SD with ANN under horizontal pipe orientation.

154
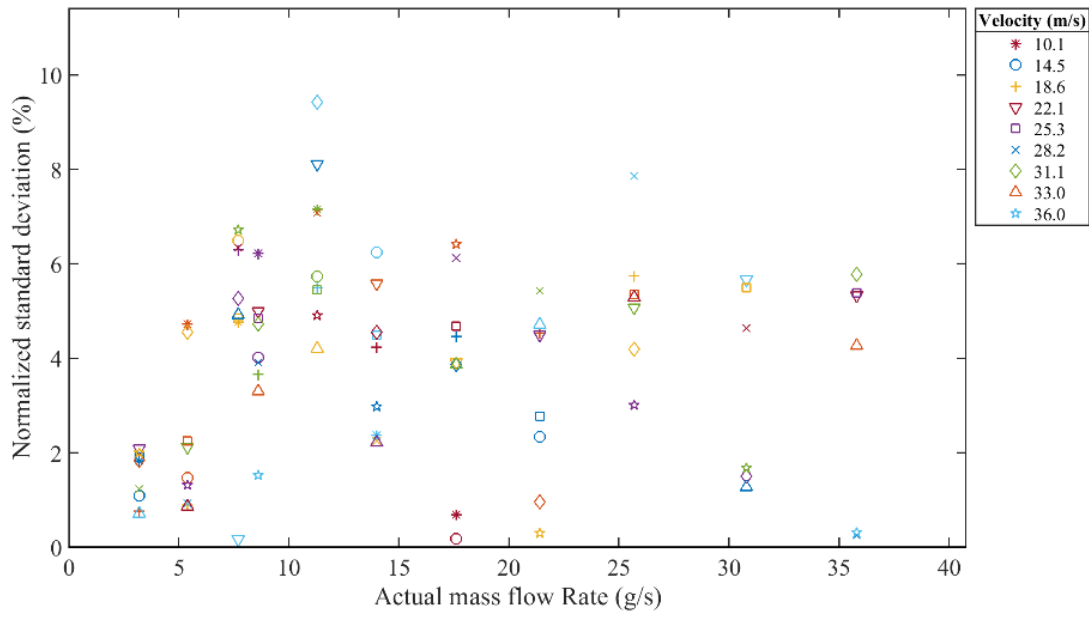
Figure 5-25: Normalized SD with SVM under horizontal pipe orientation.
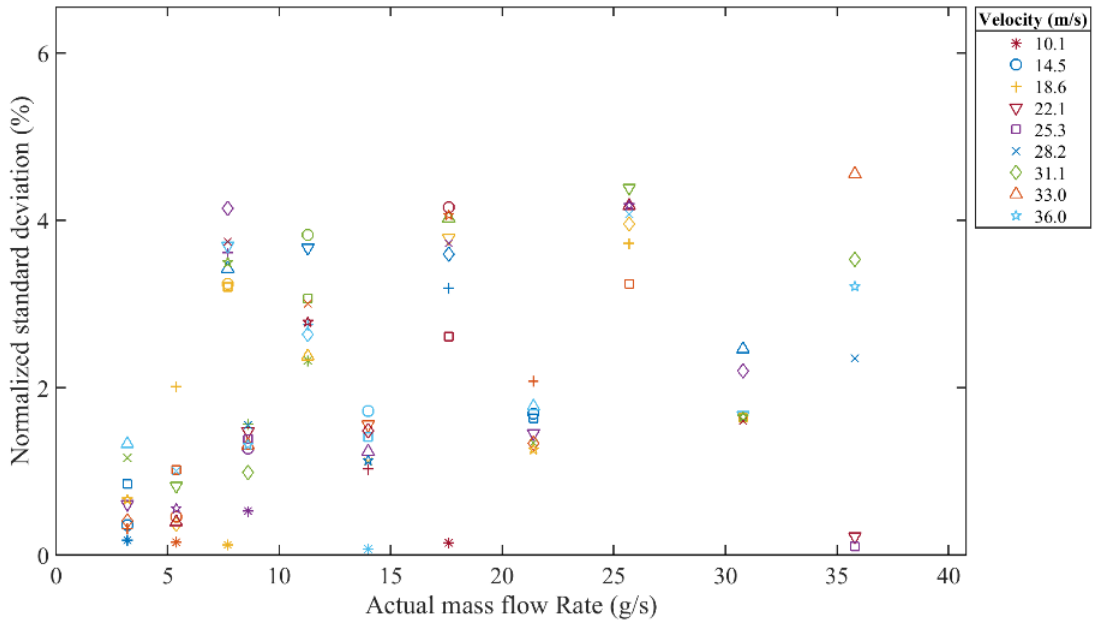
Figure 5-26: Normalized SD with CNN under horizontal pipe orientation.

5.4.10 Mass Flow Rate Measurement with ANN, SVM and CNN Models Under Vertical Pipe Orientation

Figure 5-27 to Figure 5-29 plot the predicted mass flow rates of solids with the three models under vertical pipe orientation. For all the models, the predicted mass flow rates under higher velocities yield higher relative errors due to loss of solids energy after hitting with the bend of the pipeline, as can be seen in Figure 5-30 to Figure 5-32. The data driven models are trained and tested with a different dataset under vertical pipe orientation. Therefore, the relative errors in predictions made with the ANN, SVM and the CNN models are similar to those under horizontal pipe orientation. In the vertical pipe orientation, the ANN, SVM and the CNN model has predicted the mass flow rate of solids with relative error within ±16%, ±14%, ±8%, respectively. However, compared to those under horizontal pipe orientation, mass flow rate measurements under vertical pipe orientation are more repeatable (Figure 5-33 to Figure 5-35) due to the fact that, after hitting the bend of the pipeline, solids start moving downwards more consistently in the vertical pipeline. The mass flow rates predicted with ANN, SVM and the CNN model are repeatable with maximum NSTD of 12%, 8% and 5%, respectively.
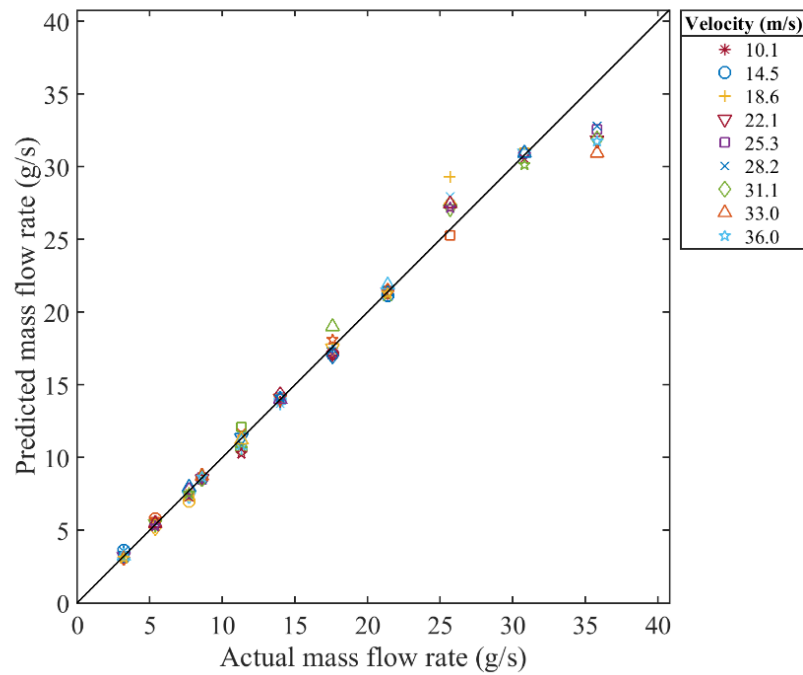
Figure 5-27: Mass flow rate measurement with ANN under vertical pipe orientation.
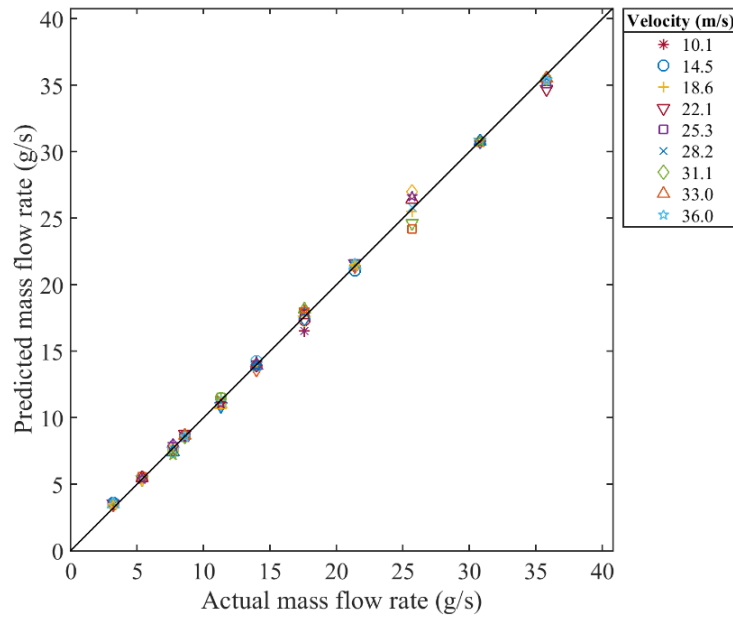
Figure 5-28: Mass flow rate measurement with SVM under vertical pipe orientation.
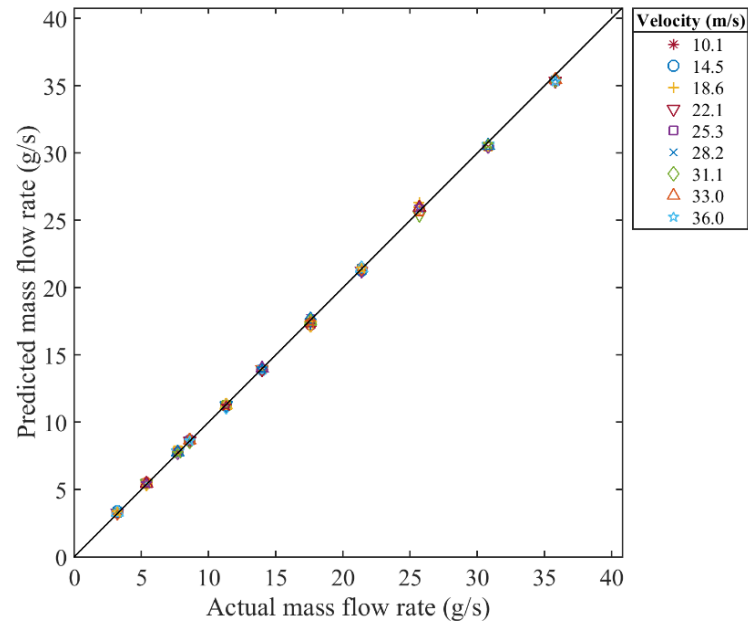
Figure 5-29: Mass flow rate measurement with CNN under vertical pipe orientation.
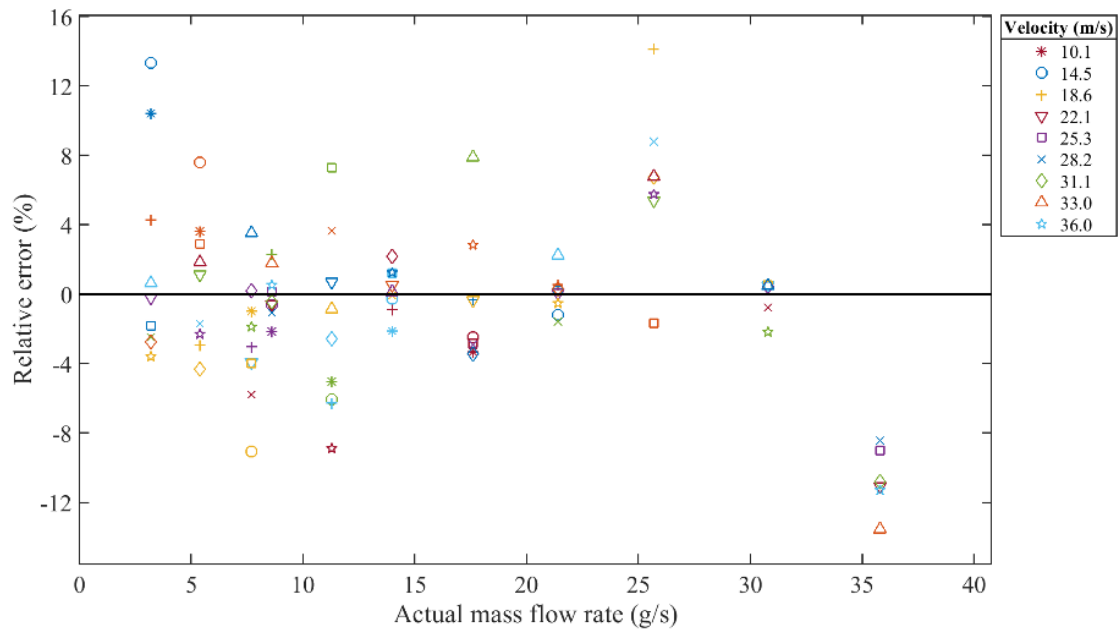
Figure 5-30: Relative error with ANN under vertical pipe orientation.
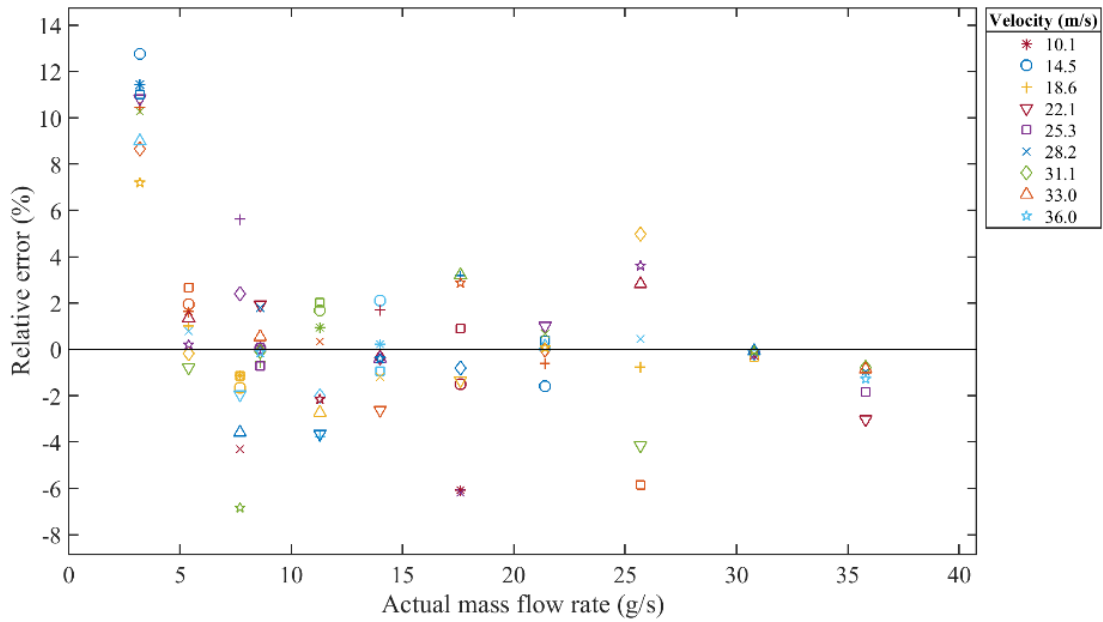
Figure 5-31: Relative error with SVM under vertical pipe orientation.

158
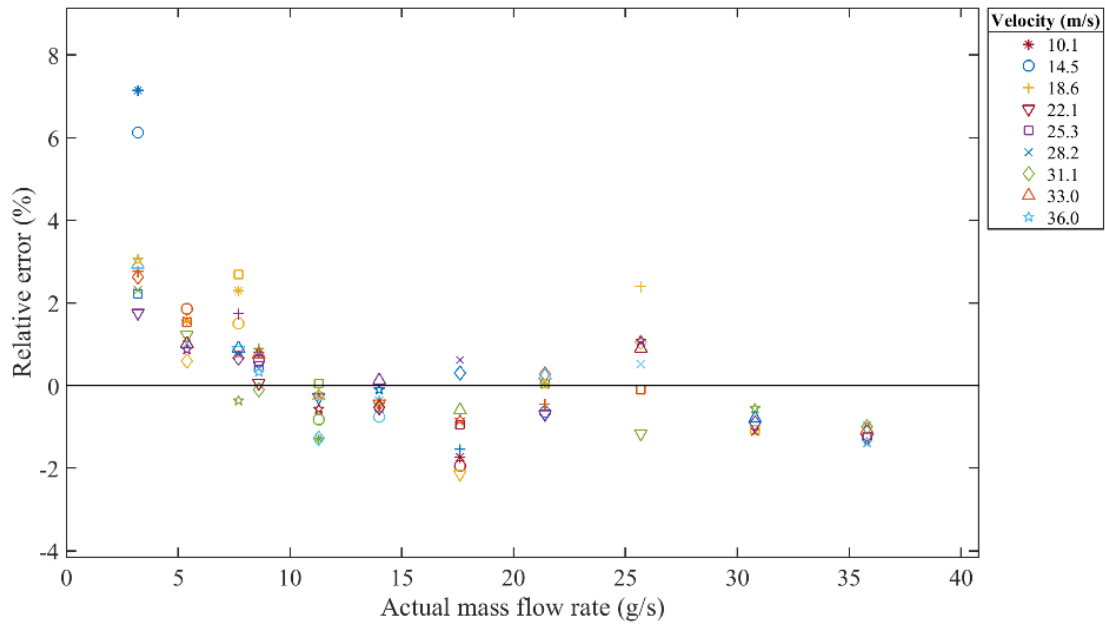
Figure 5-32: Relative error with CNN under vertical pipe orientation.
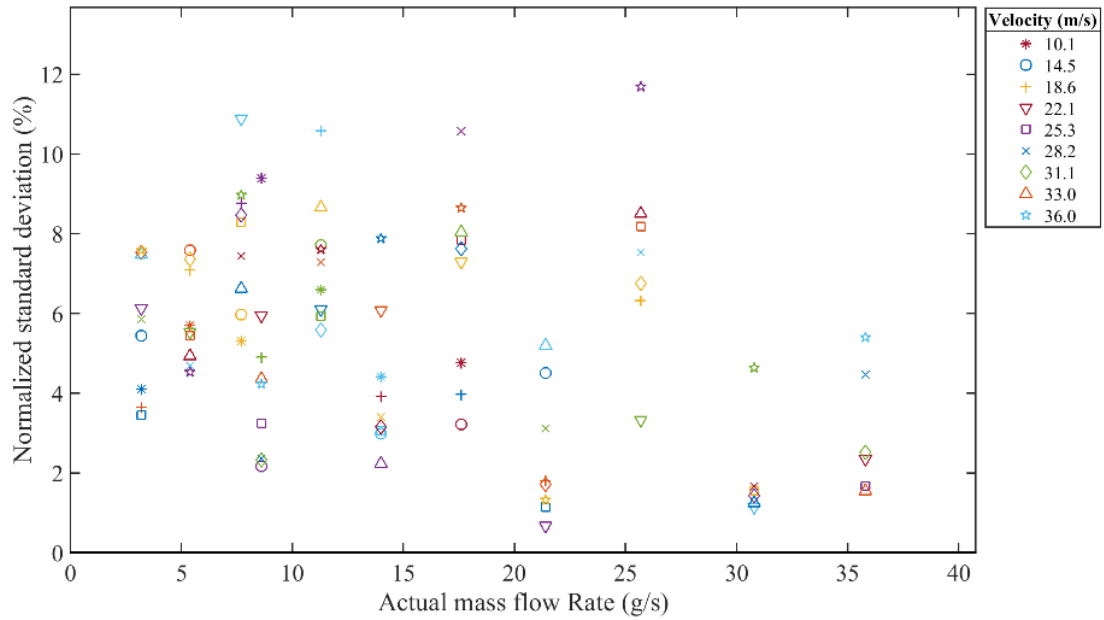
Figure 5-33: Normalized SD with ANN under vertical pipe orientation.
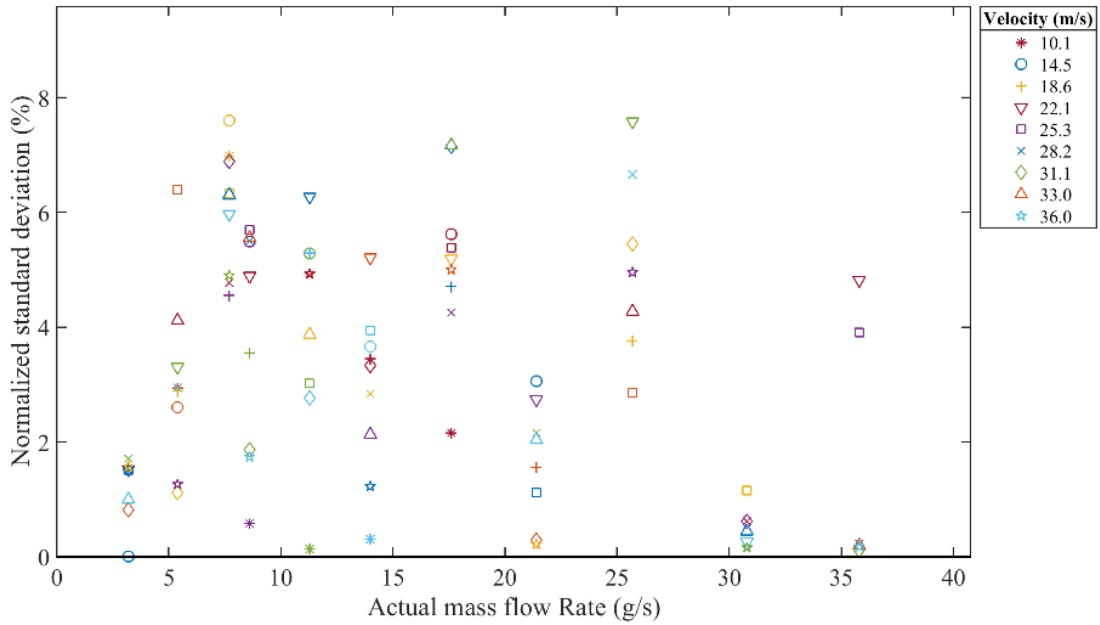
Figure 5-34: Normalized SD with SVM under vertical pipe orientation.
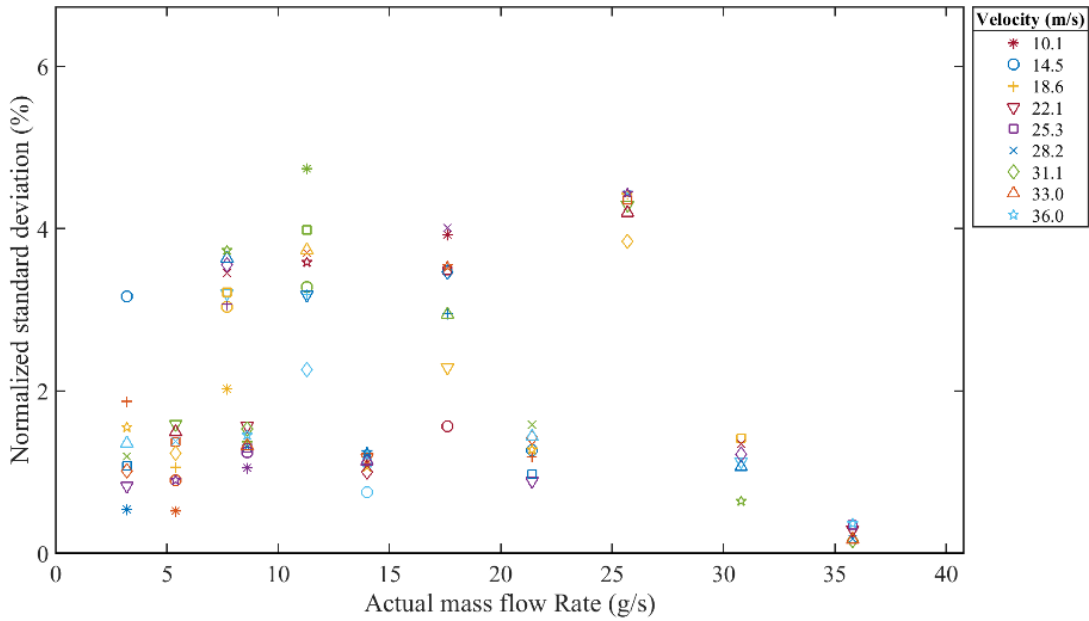
Figure 5-35: Normalized SD with CNN under vertical pipe orientation.

5.4.11 Performance Comparison Between ANN, SVM and CNN Models

A comparison is made between the ANN, SVM and the CNN models in terms of maximum relative error and NSTD under all the seen and unseen test conditions. Table 5-10 summarizes the

performance comparison between the three models. The structure of the ANN model is based on the very basic idea of a neural network where the weights and the biases are randomly initialized and tuned, which is not a suitable method to establish the relationship between the characteristics of sensor outputs and the intended measurand. For this very reason, the ANN model has not performed well for both seen and unseen conditions. However, the SVM model has generalized the data comparatively better than the ANN model because it works on the principle of transforming the data into a multi-dimensional feature space with an increasing degree of the polynomial using the kernel function to best fit the data. Furthermore, the mass flow rate measurement via the SVM model has lower NSTD values, demonstrating a better repeatability of the models. This outcome is in agreement with the earlier research on gas-liquid two phase flow measurement through SVM modelling [41] [81]. However, in comparison with both ANN and SVM models, the CNN model produces consistently more accurate and more repeatable results. This outcome is believed to be due to the fact that the CNN is capable of extracting the high-level information from the complex image data through multiple convolutional layers in an incremental method.

The maximum error and NSTD for all the models under seen test conditions are both lower than those under unseen test conditions because the models are trained with similar mass flow rates as the seen test conditions. The consistence in the solids movement in vertical pipe orientation increases after the solids hit with the bend of the pipeline. Therefore, compared to horizontal pipe orientation, the maximum error and NSTD for the seen test conditions are smaller than those in vertical pipe orientation. However, for unseen test conditions, the maximum error and NSTD remains in close agreement for both orientations.

Table 5-10: Comparison between the ANN, SVM and CNN models

		Seen test conditions			Unseen test conditions		
		ANN	SVM	CNN	ANN	SVM	CNN
Max. Error (%)	Horizontal	13.52	5.92	1.65	17.14	12.38	5.87
	Vertical	6.83	3.16	2.26	17.95	13.74	7.14
Max NSTD (%)	Horizontal	9.78	6.32	2.45	14.05	7.64	4.76
	Vertical	7.73	5.68	1.92	12.52	7.89	5.07

5.5 Summary

In this chapter, the results of mass flow rate measurement of solids through multi-modal sensing and data-driven modeling approach are presented. The results presented have suggested that, for all the seen and unseen test conditions, the CNN model has outperformed the ANN and the SVM models with a relative error within ±8% while the normalized standard deviation within 5% in both horizontal and vertical pipe orientations. The ANN and the SVM models have yielded relative errors of ±18% and ±14%, respectively, with normalized standard deviations of 14% and 8%, respectively. The direct relationship between the signal from DP transducer, particle velocity and mass flow rate of solids has constructively enhanced the performance of the models. The performance of the SVM model remains same for all the individually used electrostatic sensors for the same shaped electrodes. However, the SVM model does not perform well when trained with the combination of several electrostatic sensors from the same type of electrodes due to over-fitting problem of the model. It should be stressed that not all the sensors can provide the same number of statistical features that are useful for the data driven models. Some statistical features are useful while the others are regarded as redundant based on the partial mutual information algorithm. Compared to the ANN and SVM models, the CNN model requires a smaller window size to extract the complete information from the sensor data to produce one prediction.

Chapter 6 Mass Flow Rate Measurement of Solids Through Data-Driven Modeling Under Different Pipe Orientations, Ambient Conditions and Particle Types

6.1 Introduction

The Chapter 5 explains the issues related to inferential method of mass flow rate measurement under different physical factors and also presents the methodology based on data-driven modelling and the results of mass flow rate measurement under different conveying air velocities. This chapter presents the results of mass flow rate measurement through data-driven modeling under different pipe orientations and ambient conditions. The chapter also demonstrates the strategy to measure the ratio of two different particle types, such as coal and biomass in the pneumatic pipeline. To compensate the impact of different pipe orientations on mass flow rate measurement, a low-cost accelerometer is integrated with the sensing system. Data-driven models, based on support vector machine (SVM), are developed by taking the selected features from post-processed sensor data and infer the mass flow rate of solids in different pipe orientations. The partial mutual information algorithm is applied to quantify the importance of each feature. The firefly algorithm is used to optimize the selection of useful features and tune the learning parameters of SVM models. Experimental tests were conducted on a pneumatic conveying test rig circulating flour over the mass flow rate of solids from 3.2 g/s to 35.8 g/s in pipe orientations from 0° to 90°. Performance comparisons are made between the conventional SVM model and the optimised SVM models with the training data from horizontal orientation and different orientations, respectively. In order to compensate the impact of ambient conditions, the thermo-hygrometer probe. containing a temperature and a RH sensor, is non-intrusively placed inside the pipe and it is interfaced with the sensing system. Performance of ANN, SVM and CNN models are compared for the mass flow rate measurement of solids under ambient tempertures ranging from 16 °C to 26 °C and the ambient RH ranging from 25% to 50%. As the different ratios of coal and biomass may exhibit

different relative permitivities, a capacitive sensor along with high resolution and high precision LCR meter is also integrated with the sensing system. The method to measure the coal/biomass ratios through data-driven modeling is also trained and tested with variety of solids flow rates.

6.2 Mass Flow Rate Measurement Under Different Pipe Orientations

6.2.1 Experimental Conditions

Experimental tests were conducted under the conditions outlined in Table 6-1. The test section was arranged in 10 different orientations. For each pipe angle, the mass flow rate of solids was adjusted from 3.2 g/s to 35.8 g/s. The particle velocity, in this case, was kept constant at 18 m/s. During the experiments, the ambient conditions were controlled at a temperature of 23°C and humidity of 46%. For each test condition, the sensor signals were recorded for the duration of 30 seconds at the sampling rate of 20 kHz. These raw data will be analysed and processed further for training and testing of the SVM models.

Table 6-1: Test conditions

Test conditions	Value
Mass flow rate of particles (g/s)	3.2, 5.4, 7.7, 8.6, 11.3, 14.0, 17.6, 21.4, 25.7, 30.8, 35.8
Pipe angles (°)	0,10,20,30,40,50,60,70,80,90
Air velocity (m/s)	18
Ambient temperature (°C)	23
Relative humidity (%)	46

6.2.2 Proposed Measurement Strategy

The overall measurement strategy, as shown in Figure 6-1, starts with the collection of raw data from the sensing unit which consists of an array of ring-shaped and four arrays of arc-shaped electrostatic sensors, a DP transducer, and an accelerometer. These three types of sensors are selected in order to acquire sufficient and complementary information about the physical characteristics of the flow. During pneumatic transportation, solids are electrically charged due to triboelectric charging effect. Electrostatic sensors are used to sense the electrostatic charge and the amplitude of electrostatic signals is related to the velocity, concentration and hence mass flow rate of solids [1, 19, 20]. The differential pressure is the pressure difference between the inlet and outlet of a pipe section. It is directly related to the flow velocity and solids concentration. [82].

164

Therefore, the output of the DP transducer contains information about the mass flow rate of solids. The raw data needs to be segregated into the corresponding clusters of mass flow rate conditions. Unnecessary peaks and the fluctuations in the sensing data are removed at the stage of removal of outliers. All the possible statistical features, presented in the previous study [83], are extracted from all the smoothened and conditioned sensor data. The PMI is used to measure the information between each statistical feature and its corresponding output. Based on the values of PMI, each feature is prioritized by keeping the most useful features on the top. FA is used to optimize the selection of input features and the learning parameters of the SVM model including kernel scale, delta gradient tolerance, gap tolerance, and epsilon. All the arrows filled with a grey colour in Figure 1 show the permanent connection between blocks. However, the white arrow connections in Figure 6-1 show the temporary existence of the FA during the training process. Once the features are selected and the parameters of the model are optimized during the learning process, the developed SVM model will be directly used to infer the mass flow rate of solids. The program for the proposed measurement strategy is provided in Appendix 3.
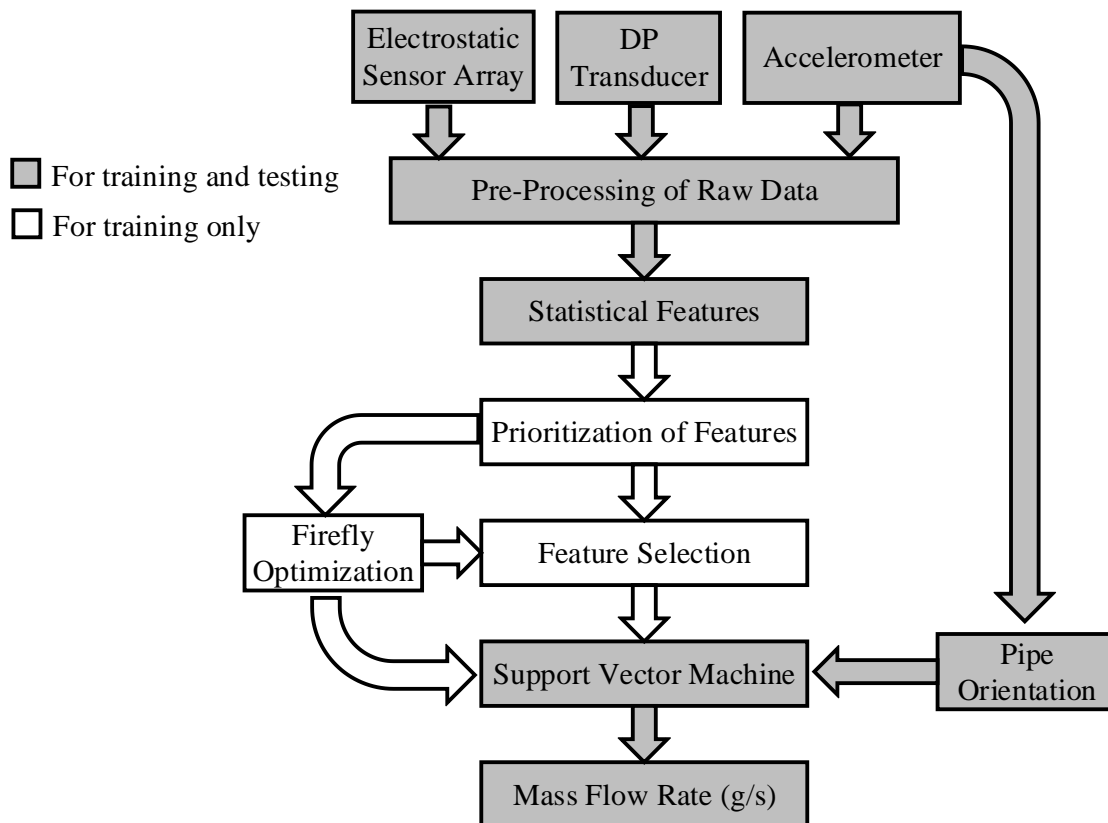
Figure 6-1: Measurement strategy

6.2.2.1 Optimization of Data-Driven Modelling

The nature of the sensor signals becomes further complex when recorded in different orientations of pipe. Therefore, a data-driven model is required to establish the relationship between the input features and the mass flow rate of solids. Owing to the good generalisation ability and prediction accuracy, SVM is a more suitable data-driven model for gas-solid flow measurement [83]. The typical structure of SVM model which is used for this purpose is presented in Chapter 5. Statistical features, optimally selected by PMI and FA (Section 6.2.3.4), are used as the inputs to the SVM model. The goal of the SVM model is to identify a hyperplane in an N-dimensional space with N number of features that classify the data points with the highest margins. Each node in the hidden layer converts the input data into another feature space using the kernel function. Compared with other kernel functions including linear, polynomial, and Gaussian, RBF is widely used in the domain of regression with better performance. The number of nodes in the hidden layer can be determined experimentally by the guidelines mentioned in [45]. In this experiment, the internal parameters of the model, including kernel scale, delta gradient tolerance, gap tolerance and epsilon, are tuned to their optimal values using the firefly optimization algorithm (Section 6.2.3.4). In the process of SVM training, the algorithm divides the element of the predictor matrix by the value of the kernel scale. Then the appropriate kernel norm is applied to compute the gram matrix. Delta gradient tolerance defines the tolerance for the gradient difference between the upper and lower violators in the SVM solver and is used to set the threshold for optimization convergence. Gap tolerance is the threshold defined to maximize the location of the decision boundary. The whole training process is terminated as soon as the threshold reaches the value of gap tolerance. The value of epsilon defines the half width of epsilon-insensitive zone which influences the number of support vectors used to create the regression function [84]. The larger the epsilon, the smaller the support vectors selected. On the other hand, a higher epsilon value results in more flat estimates. The value of the epsilon evaluates the accuracy of the approximated function. It relies entirely on the target values for the training set. If the epsilon is larger than the target range, good accuracy in the predicted result cannot be achieved. The relationship between epsilon, target output and the training data can be defined as

$$|y_n - (\beta x_n + b)| \leq \varepsilon \qquad \text{(6-1)}$$

Where y_n is the actual target label, x_n carries the training data and β is the minimal norm value.

The performance of the conventional SVM model is sensitive to its parameters such as kernel scale, delta gradient tolerance, gap tolerance, and epsilon. Careful tuning of these parameters can significantly improve the prediction accuracy. Furthermore, a little change in the dimension of the input feature vector can significantly affect the tuning of these parameters. Therefore, an optimization algorithm is required to determine the best combination of all the parameters. Metaheuristic algorithms, influenced by the nature of many dynamic entities, play a critical role in modern global optimization algorithms, soft computing, and cognitive intelligence. Among the new metaheuristics, the FA is found to be very effective with multi-modal and global/local optimization issues as presented in Chapter 4.

The whole optimization process starts with the random initialization of the population that contains SVM model parameters. The fitness value of each set of model parameters is evaluated using the fitness function:

$$I_{fitness} = 9.5 \times 10^{10} e^{-11 \sum_{i=1}^{N_c} a_i err_i} + 0.3 \tag{6-2}$$

where N_c is the total number of mass flow rate conditions, err_i and a_i are the averaged relative error and scaling factor for the ith mass flow rate condition, respectively. The fitness function produces the value based on the error between SVM predicted and actual results. All the other constants in the fitness function are adjusted in a way to keep the fitness value in between 0 and 1. The way each of the SVM model parameters moves towards optimal value in conjunction with fitness value in each iteration defines the relationship between noisy non-linear and the SVM parameters optimisation (Section 4.4).

Based on the intensity values of each firefly computed by the fitness function, each firefly makes a move towards the firefly of the highest intensity value. The position of ith firefly, that represents the value of kth parameter of the model, can be updated by

$$x_{ik} = x_{ik} + \beta_o e^{-\gamma \cdot r_{ij}} \left(x_{ik} - x_{jk} \right) + \alpha S_k (rand_{ik} - 0.5) \tag{6-3}$$

where r_{ij} can be computed from,

$$r_{ij} = \sqrt{\sum_{k=1}^{N_p}\left(x_{ik} - x_{jk}\right)^2}$$

<div align="right">(6-4)</div>

where x_{ik} and x_{jk} are the positions of ith and j^{th} fireflies which represent the values of kth parameter of the SVM model.

The whole algorithm keeps updating the parameters of the SVM model until either the iterations reach the maximum number of iterations or the intensity of the firefly reaches the threshold.

6.2.3 Results and Discussion

6.2.3.1 Sensor Signals From DP Transducer and Accelerometer

The variations in the DP signal amplitude with the mass flow rate of solids in different pipe orientations are plotted in Figure 6-2. With reference to Newton's second law of motion, the drop in line pressure is directly proportional to the solids-wall friction and solids gravity. The drop in line pressure increases with the mass flow rate of solids as a higher concentration of solids introduces more solids-wall friction and hence more energy is required to convey the solids from the upstream to the downstream of the spoolpiece. The velocity of conveying air was kept constant throughout the experiments, so the increasing mass flow rate of solids entails rising solids concentration and hence higher pressure drop. Furthermore, for a given mass flow rate of solids, a rising trend is observed in differential pressure with an increasing pipe angle because of the increasing gravitational effect on solids.
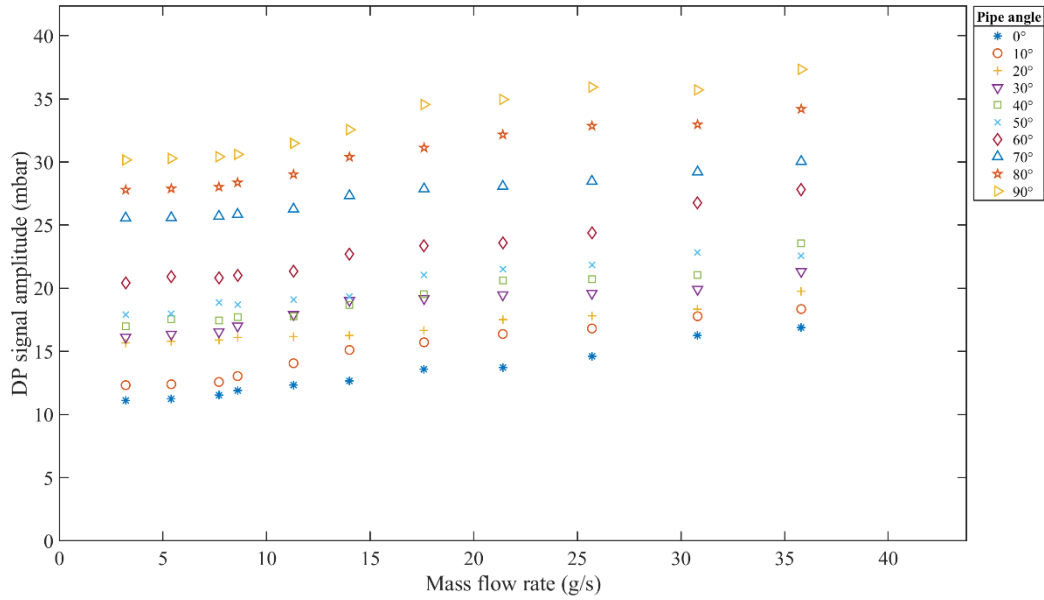
Figure 6-2: Relationship between the DP signal and mass flow rate of solids

RMS of electrostatic signals is a key feature in the time domain, indicating the amplitude of electrostatic signals, which is closely related to solids velocity and concentration as discussed in Chapter 4. PSD is a feature in the frequency domain, providing complementary information about the two-phase flow. DP signal amplitude is directly related to the velocity of the mixture flow and solids concentration. In view of their physical meanings, they are three important, independent and complementary features to infer the mass flow rate of solids.

The purpose of installing an accelerometer on the sensing unit is to collect the information about the pipe orientation. Therefore, the signal output from this sensor has nothing to do with the vibration in the pipe due to mass flow rate of solids but the orientation of the pipe. Figure 6-3 shows the averaged value of the post-processed signal from the accelerometer in different pipe orientations.
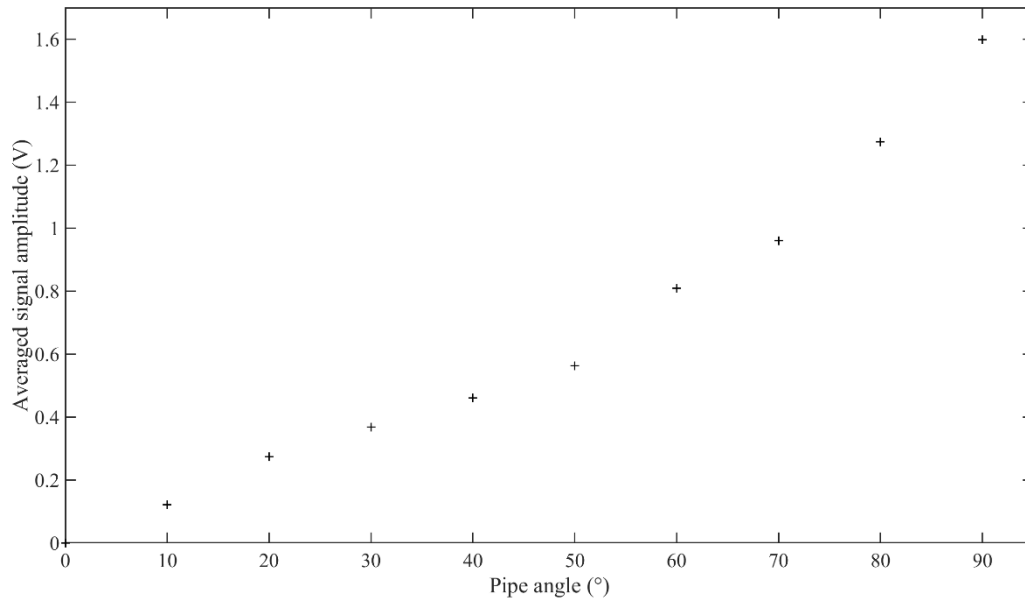
Figure 6-3: Relationship between the output of the accelerometer and the pipe orientation

6.2.3.2 Conventional SVM Model Trained Under Horizontal Pipe Orientations

The conventional SVM model was trained with the dataset and parameters presented in Chapter 4. Training data were collected when the sensing unit was installed at a fixed angle of $0°$. However, testing was done with the data collected in different angles ($10°$, $30°$, $50°$, $70°$ and $90°$). Figure 6-4 shows the predicted mass flow rate from the conventional SVM model at eleven different mass flow rates. The straight line shows the scenario where the predicted measurements are ideally assumed to be the same as the reference. A significant amount of variation from the ideal straight line can be seen in the predicted measurements, especially at the lower mass flow rate and larger pipe angles. This is due to the fact that the movement of sensing head towards vertical orientation introduces the change in flow regime and result in different characteristics in sensor signals. Because of the limitation of the generalisation ability of the data-driven models, the SVM model, developed based on the data from a single pipe orientation, cannot measure the mass flow rate at other pipe orientations.
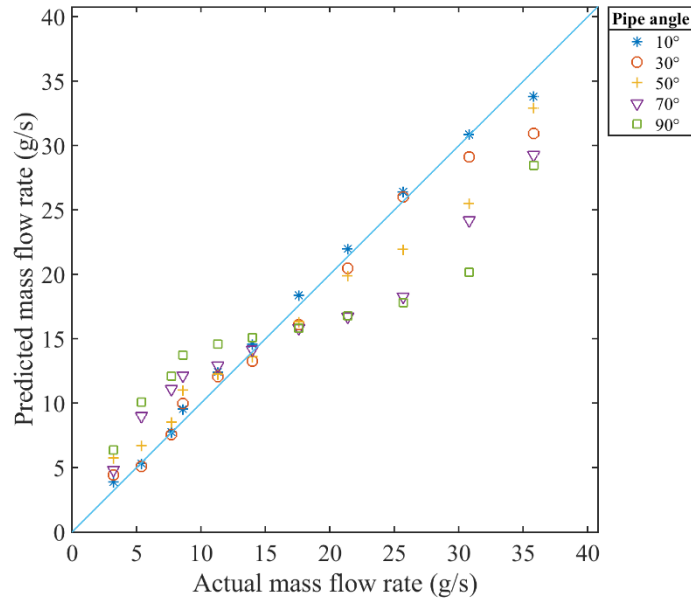
Figure 6-4: Results from the conventional SVM model trained under horizontal pipe conditions

Figure 6-5 shows that the relative error is moderately higher under the lower mass flow rate conditions and reduces significantly with the mass flow rate of solids. This is caused by the flow regime effects in the pneumatic conveying pipe. Flow regime stays homogenous at a lower mass flow rate due to the uniform distribution of particles. The higher ratio of particles in uniform distribution loses the energy after hitting with the curved wall of the flexible joint and hence causes the measurement error. As the mass flow rate increases, the flow regime becomes stratified due to the reason that at constant velocity and higher solids concentration, most of the particles travel in the lower part of the pipe. The particles which are moving along the lower part of the pipe can easily pass through the bend of the pipe and hence lower measurement error due to the reason that the particles are least effected by the bend. Repeatability of these measurements is also observed by calculating the normalized standard deviation, as shown in Figure 6-6. Loss of particles energy at lower mass flow rates produces inconsistent and less reliable sensor data compared to the higher mass flow rates which affect the repeatability of the measurements accordingly. Variations in the predicted mass flow rate measurement demonstrate the impact of the pipe orientation on the mass flow rate measurement.
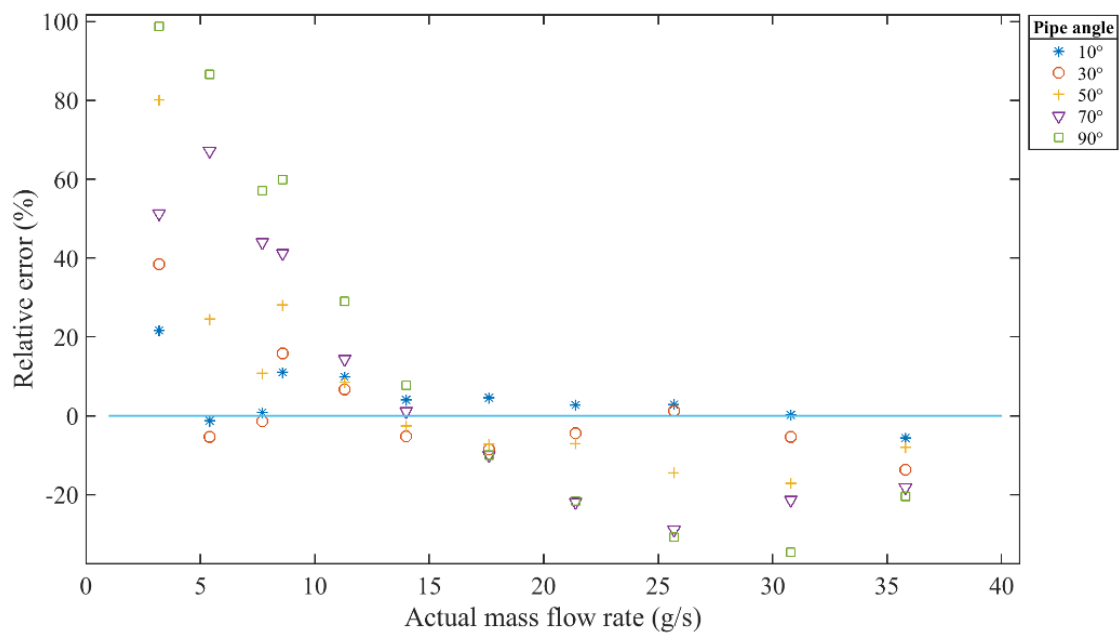
Figure 6-5: Relative error of the conventional SVM model trained under horizontal pipe conditions
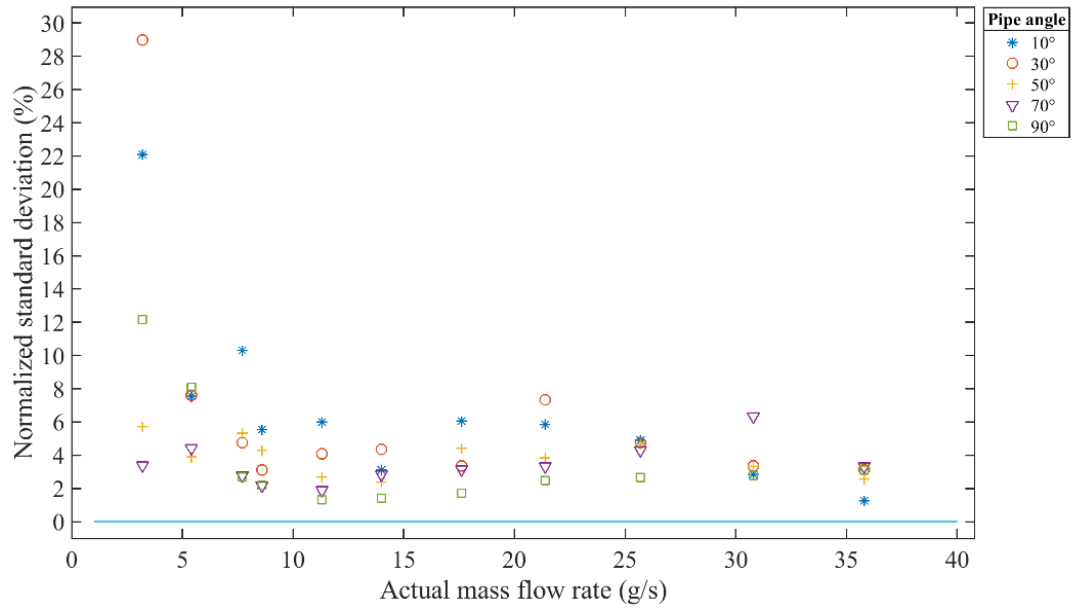
Figure 6-6: Normalized STD of the conventional SVM model trained under horizontal pipe conditions

6.2.3.3 Conventional SVM Model Trained Under Different Pipe Orientations

As the orientation of the sensing unit does have an impact on the characteristics of sensor signals and affect the performance of the data-driven model, sensor signals in different orientations are added to the training data to make the model adaptive to the orientation of the sensing unit. The experimental test conditions for training data and test data are summarized in

Table 6-2. A new model based on conventional SVM was trained in five different angles and eleven mass flow rates. However, the test data were collected completely different from the conditions of training data to analyse the generalization capability of the model. In the training process, the selection of useful features and SVM model parameters were manually determined by trial and error method. Table 6-3 shows the parameters used in the conventional SVM model.

Table 6-2: Dataset for training and testing

Mass Flow Rate (g/s)	Training Angle (°)	Test Angle (°)
3.2, 5.4, 7.7, 8.6, 11.3, 14.0, 17.6, 21.4, 25.7, 30.8, 35.8	0	10
	20	30
	40	50
	60	70
	80	90

Table 6-3: Parameters of the conventional SVM model

Parameter	Value
Kernel Scale	25
Delta Gradient Tolerance	0.005
Gap Tolerance	0.009
Epsilon	0.001
Number of Input Features	120

Figure 6-7 shows the predicted measurements under eleven mass flow rates in five different angles of the sensing unit. The proximity of measured mass flow rate values with the ideal straight line demonstrates the effectiveness of compensation of pipe orientation. The relative error of the

predicted measurements from conventional SVM trained in different pipe orientations is shown in Figure 6-8. In comparison with the results in Figure 6-8, the relative error is significantly reduced to within ±40%. Likewise, the repeatability of the measurements has also been improved and the maximum normalized standard deviation is reduced down to 9% as shown in Figure 6-9.
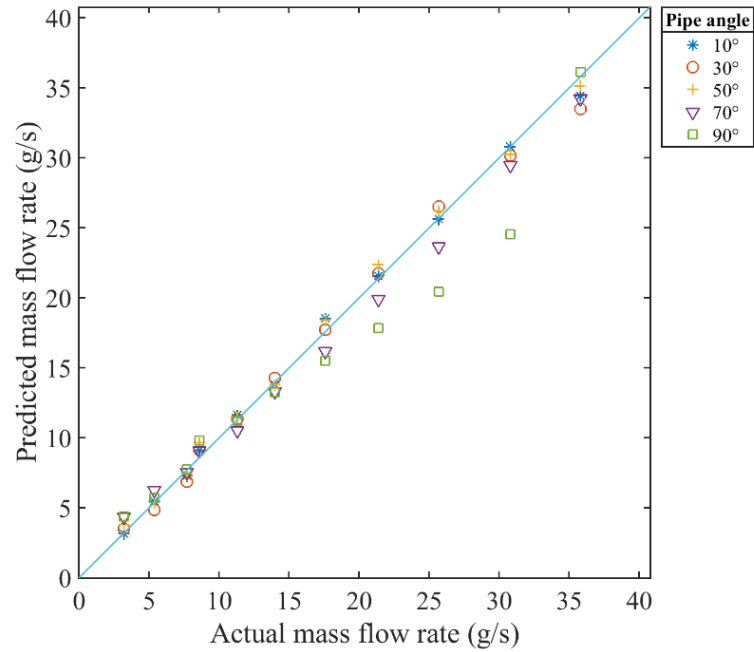
Figure 6-7: Results from the conventional SVM model trained in different pipe orientations
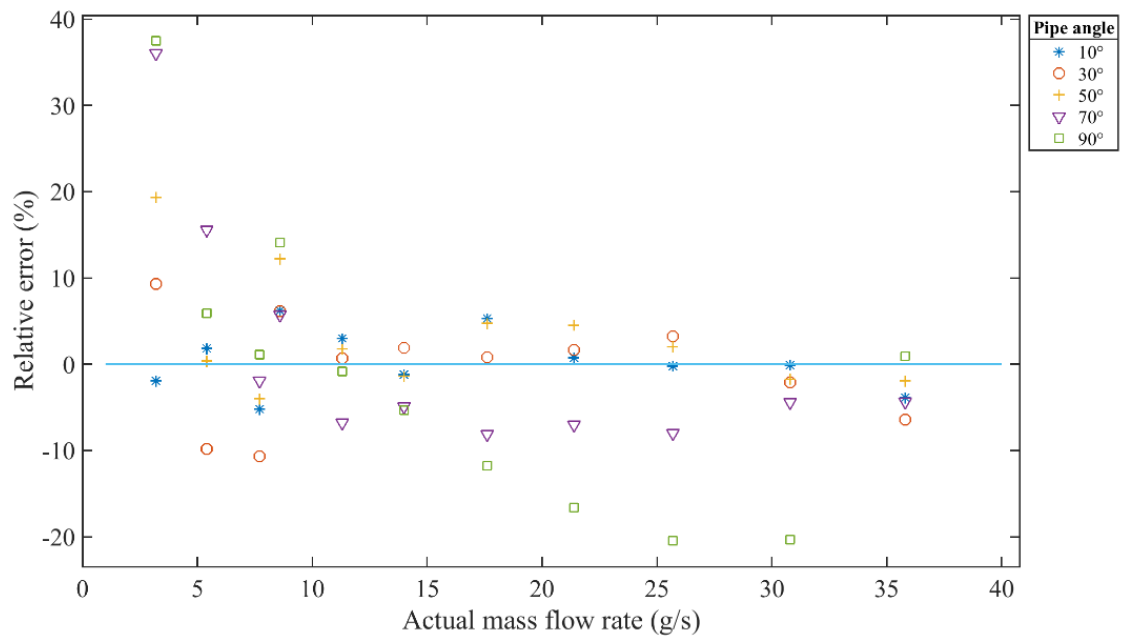
Figure 6-8: Relative error of the conventional SVM model trained in different pipe orientations
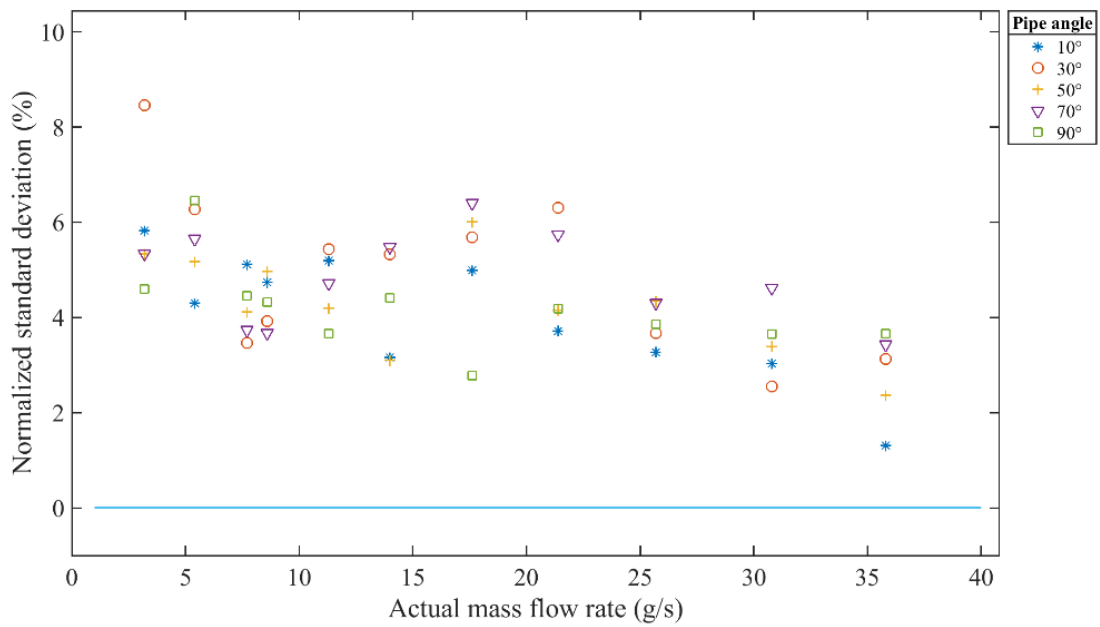
Figure 6-9: Normalized STD of the conventional SVM model trained in different pipe orientations

6.2.3.4 Optimized SVM Model Trained Under Different Pipe Orientations

In this section the SVM model is trained and tested using the data outlined in

Table 6-2. The performance of the SVM model is also influenced by the learning parameters of the model, therefore FA is applied to find the optimal parameters in the SVM model. The FA has created 300 elements in the entire population where each element contains all the tuning parameters of the SVM model, as mentioned in Section 2.3. The FA has calculated the fitness value of each element in the population using Equation (6-2) to analyze the performance. The higher fitness value of an element indicates that the element is closer to the optimal value of the model parameters. Based on the selection of the best element, FA then keeps creating a new population around that best element of the previous population. FA has generated the optimized SVM parameters based on the best element of the final population when the algorithm is terminated. The whole algorithm terminates when either the accuracy of the SVM model reaches the desired level or if there are no significant changes observed in the accuracy of the SVM model for a specific number of iterations. In this case, FA has converged to the optimized solution in 76 iterations by using the parameters, as listed in Table 6-4. The parameters for FA are chosen through trial and error by limiting the maximum number of iterations (N_{max}) to 500.

Table 6-4: Parameters of FA

Parameter	Value
α	0.2
β	4
ϒ	0.01
S	[65, 0.3, 0.4, 1, 70]
N_p	300
N_{max}	500

The performance of the FA in each iteration including the value of fitness function and the parameters of the SVM model, including kernel scale, delta gradient tolerance, gap tolerance, epsilon, and the number of input features can be seen in Figure 6-10. Stability in the fitness value after 76th iteration shows that the FA has approached to its saturation state by giving the best suitable parameters mentioned in Table 6-5.
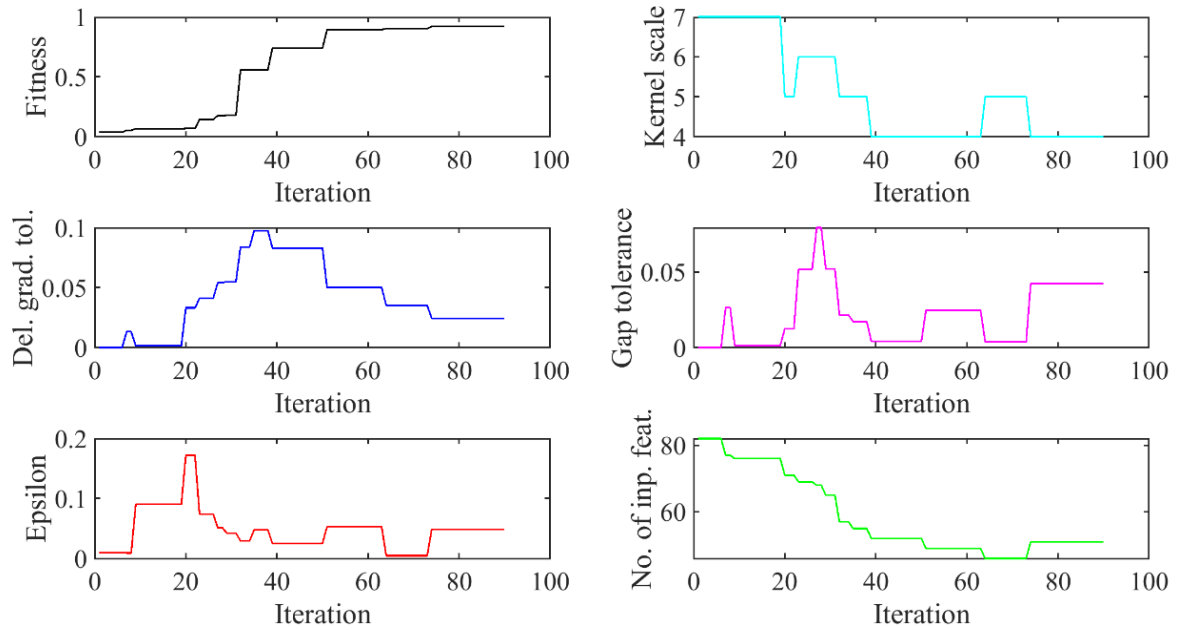
Figure 6-10: Performance of the FA

Table 6-5: Optimized parameters of the SVM model

Parameter	Value
Kernel Scale	4
Delta Gradient Tolerance	0.0481
Gap Tolerance	0.0472
Epsilon	0.0514
Number of Input Features	52

Figure 6-11 shows the mass flow rate measurements from the optimized SVM model. The very close proximity of measured points to the ideal straight line shows the effectiveness of the proposed model. A deep-down analysis is undertaken based on the relative error between measured and the reference mass flow rate conditions shown in Figure 6-12. Measurements taken at the angle of 10 and 30 degrees have a relative error within the range of ±5% for lower mass flow rate conditions and it reduces down to ±1% for higher mass flow rates. At the angle of 50 degrees,

relative error fluctuates between ±5% at lower mass flow rate conditions but it stays reasonably low at higher mass flow rate conditions. Measurements, taken at 70 and 90 degrees, exhibit the relative error that is heavily dispersed between ±12% because the orientation of the sensing unit is approaching towards vertical orientation and the particles are partially disturbed by the walls of the flexible joint. As shown in Figure 6-13, the repeatability of the measurements is improved and the normalized standard deviation is within 7%.
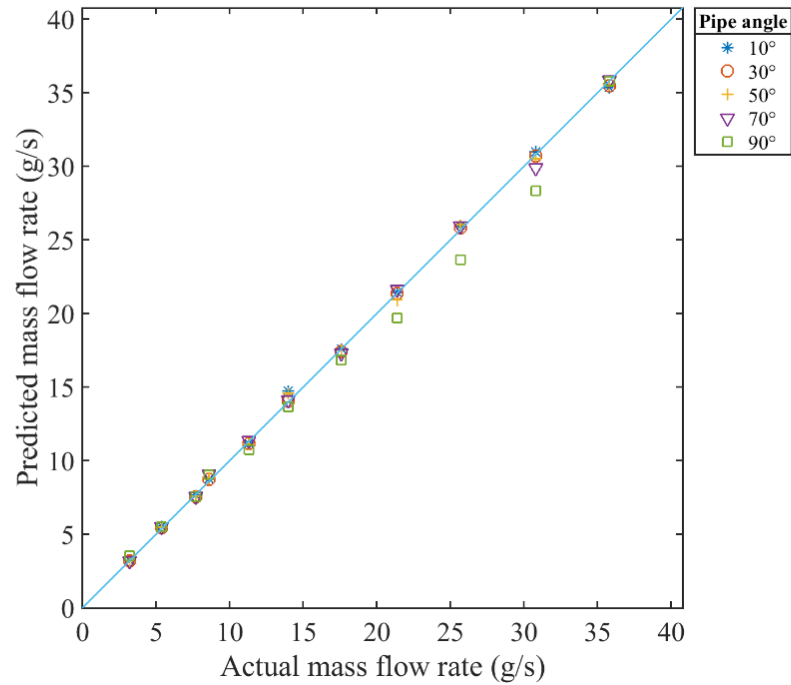
Figure 6-11: Results from the optimized SVM model trained in different pipe orientations
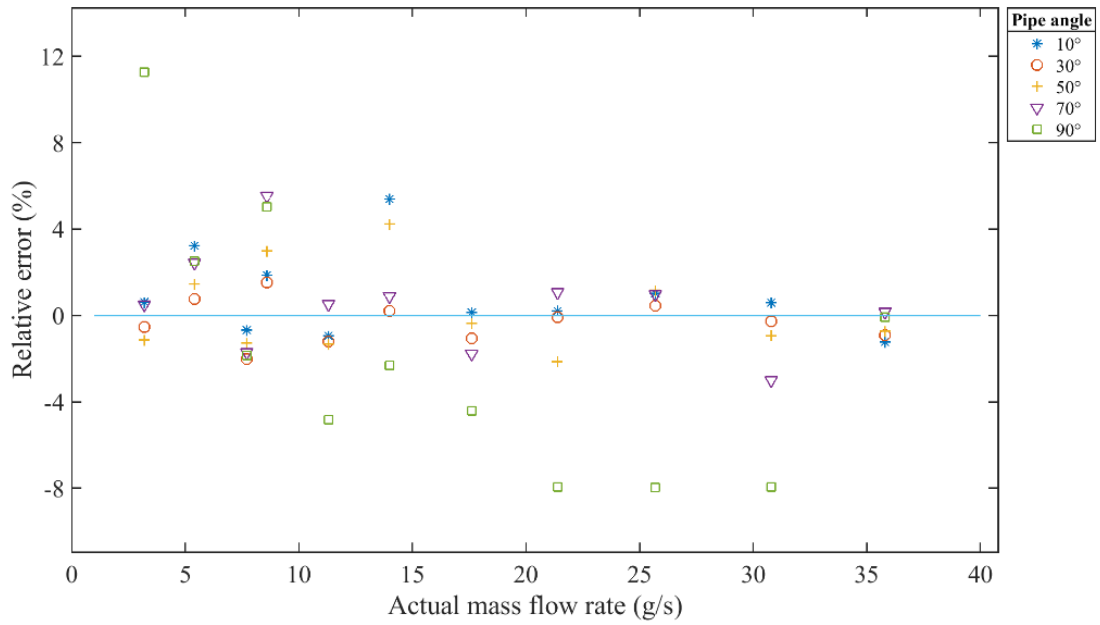
Figure 6-12: Relative error of the optimized SVM model trained in different pipe orientations
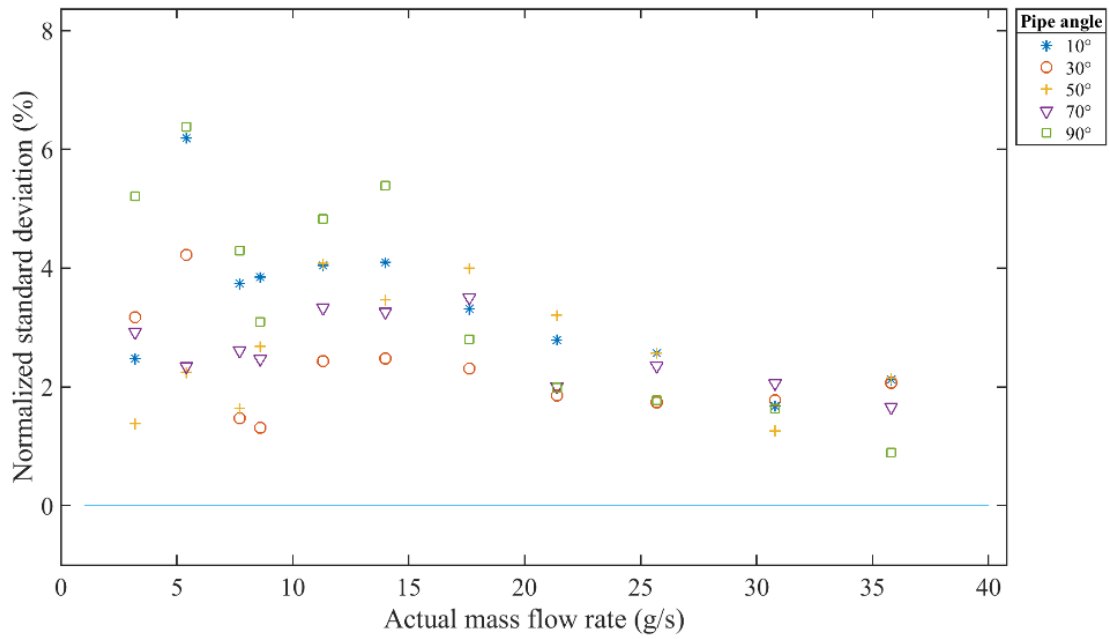
Figure 6-13: Normalized STD of the optimized SVM model trained in different pipe orientations

179

6.3 Mass Flow Rate Measurement Under Different Ambient Conditions

6.3.1 Experimental Conditions

Table 6-6 describes the mass flow rates of solids which are run at six different ambient temperatures. The rise in temperature can cause a drop in moisture level. Therefore, the air inside the lab was circulated with the air outside the lab to maintain the RH around a constant set point inside the lab. Every time a new ambient condition is set in the lab, the test rig was run idly for 10 minutes to maintain the same ambient conditions inside the pneumatic pipeline. Table 6-7 shows the combinations of mass flow rates of solids with six different ambient RH and constant ambient temperature 20 °C. Experiments were run in several days with different weather conditions to gather the data under different RH. A constant ambient temperature was maintained using two commercial level air conditioning units.

Table 6-6: Test matrix with different mass flow rates and room temperatures at constant RH 35%

		Mass flow rate (g/s)										
		Training and test conditions					Unseen test conditions					
		5.4	8.6	14.0	21.4	30.8	3.2	7.7	11.3	17.6	25.7	35.8
Ambient temperature (°C)	16	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	18	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	22	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	24	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	26	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 6-7: Test matrix with different mass flow rates and RH at constant room temperature 20 °C

		Mass flow rate (g/s)										
		Training and test conditions					Unseen test conditions					
		5.4	8.6	14.0	21.4	30.8	3.2	7.7	11.3	17.6	25.7	35.8
RH (%)	25	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	35	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	40	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	45	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

6.3.2 Measurement Principle

The suggested solution for the measurement of mass flow rate of solids, as shown in Figure 6-14, is based on the multi-modal sensing and data driven models. The multi-modal sensing system includes an array of ring-shaped electrostatic sensors, four arrays of arc-shaped electrostatic sensors, a DP transducer and a thermo-hygrometer containing temperature and RH sensors. The thermo-hygrometer provides the supplementary information required to compensate the impact of ambient conditions. A data driven model is used to establish the complex relationship between the sensor data and the mass flow rate of solids to be measured under different ambient conditions. The statistical features are extracted and shortlisted prior to the development of the model. The optimal structure of the model is developed during the training and validation process. For the measurement of mass flow rate of solids, the data driven model is trained with the selected statistical features extracted from the post-processed sensor data in the time, frequency and the time-frequency domains. The raw sensor data is processed prior to the extraction of statistical features to avoid the false training with absurd data labels. The proposed measurement strategy is implemented in the MATLAB version 2020b. The program for the proposed measurement strategy is provided in Appendix 4.
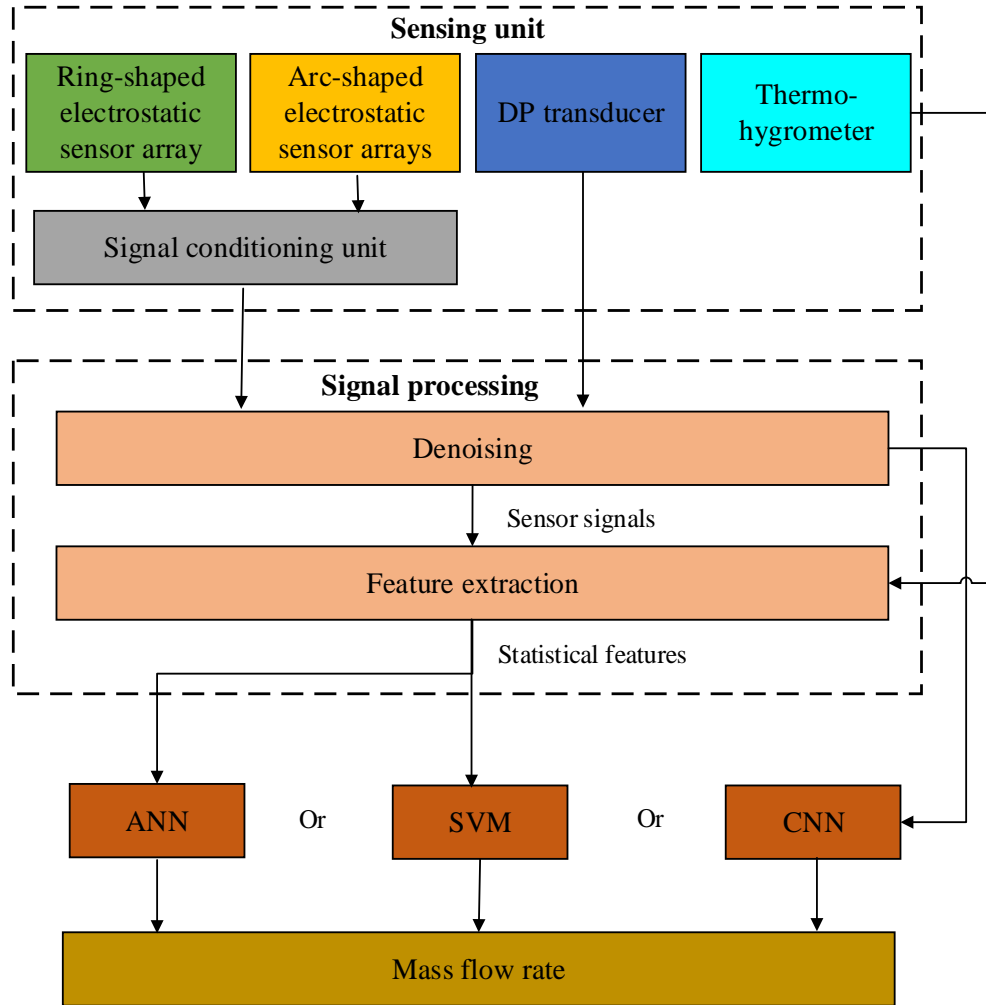
Figure 6-14: Principle of mass flow rate measurement through data-driven modelling under different ambient conditions

6.3.3 Data from Thermo-Hygrometer

The pneumatic system takes the air from the lab, therefore different temperatures are observed inside the pipe, as shown in Figure 6-15, under different ambient temperatures with uncertainty of 0.5 °C measured with thermo-hygrometer. RH inside the pipe that stays around 35% with a standard deviation of 1% measured with thermo-hygrometer. For the purpose of running the experiments under different ambient RH, the ambient temperature was kept constant using two commercial level air conditioning units. The measured temperature inside the pipe stays around 20 °C with standard deviation of 0.3 °C. Figure 6-16 shows the measured RH inside pipe under

six different RH conditions. The RH inside the pipe remains near the RH that is set in the lab with standard deviation of 1% measured with thermo-hygrometer.
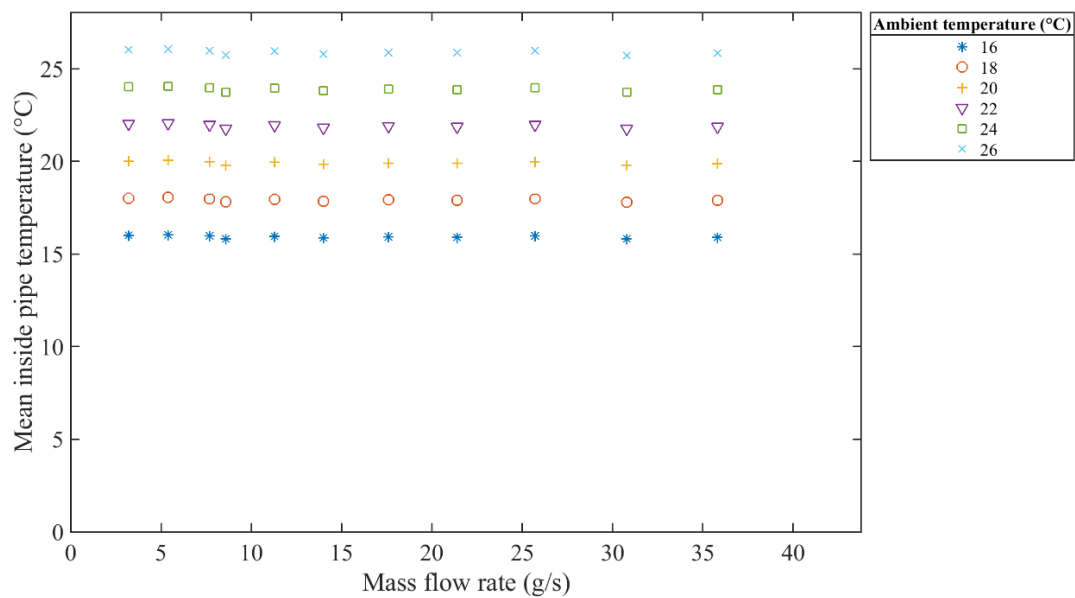
Figure 6-15: Temperature inside the pipe under different mass flow rates of solids and ambient temperatures with RH 35%.
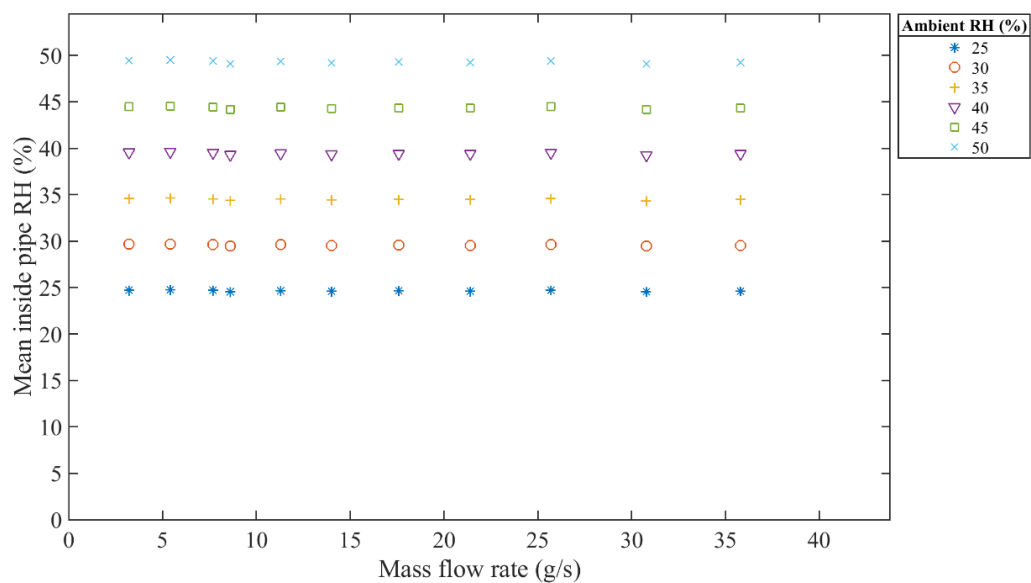
Figure 6-16: RH inside the pipe under different mass flow rates of solids, ambient RH and constant ambient temperature 20 °C.

6.3.4 Measurement with ANN, SVM and CNN

Figure 6-17 to Figure 6-19 shows the measurement of mass flow rate of solids under different ambient temperatures and RH, as outlined in Table 6-6 and Table 6-7, respectively, with ANN, SVM and CNN models. The relative error in mass flow rate of solids measured with SVM is lower compared to relative errors in the measurements generated by ANN and CNN, as can be seen in Figure 6-20 to Figure 6-22. Based on the nature and complexity of the sensor data, the SVM model has effectively developed a relationship between the input data and the output. The relative error in mass flow rate measurement under different ambient RH is also lower than the relative error with different ambient temperatures. The results predicted with SVM model are also highly repeatable compared to those predicted with ANN and the CNN model with maximum NSTD of 7%.
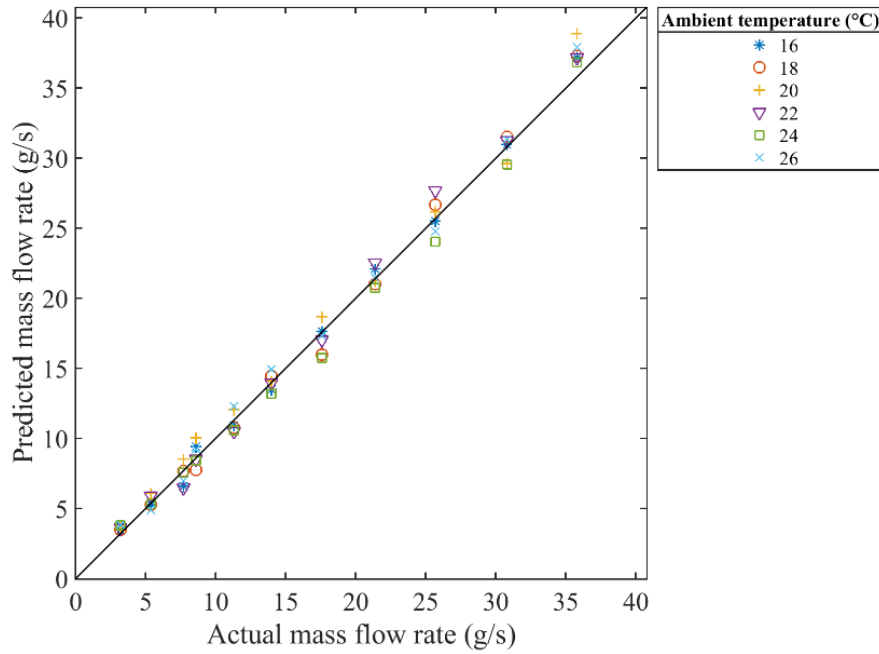
(a)
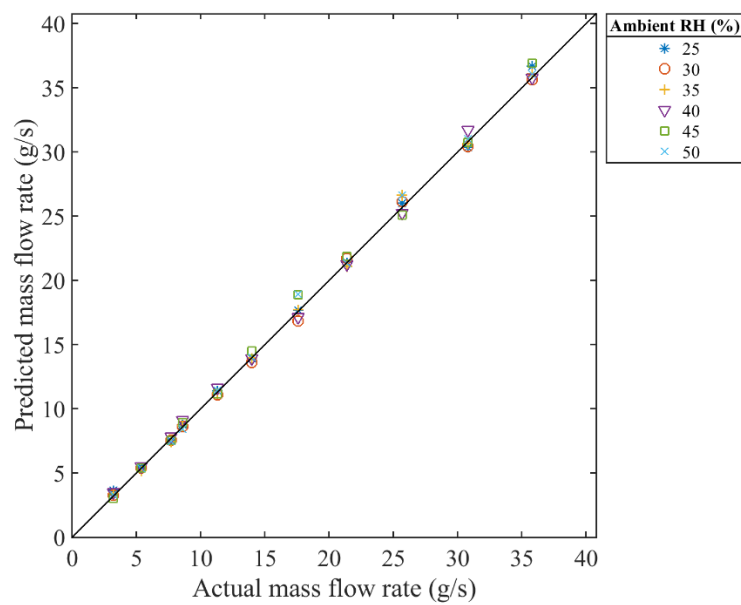
(b)

Figure 6-17: Mass flow rate measurement with ANN (a) different ambient temperatures (b) different ambient RH
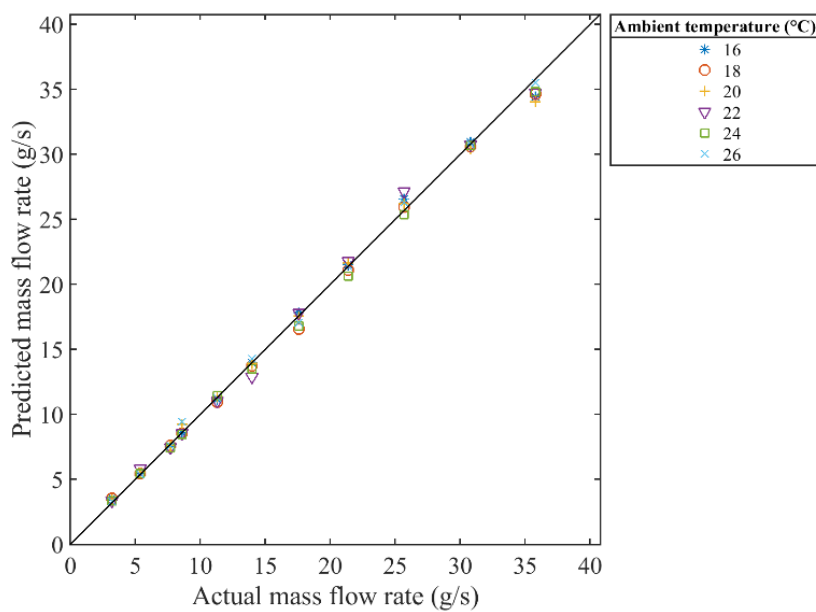
(a)
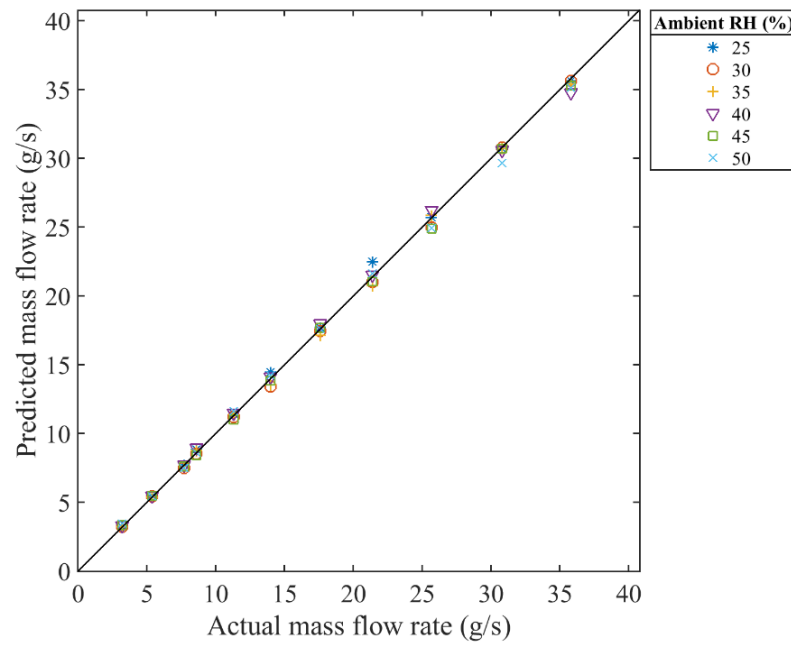
(b)

Figure 6-18: Mass flow rate measurement with SVM (a) different ambient temperatures (b) different ambient RH

(a)

186
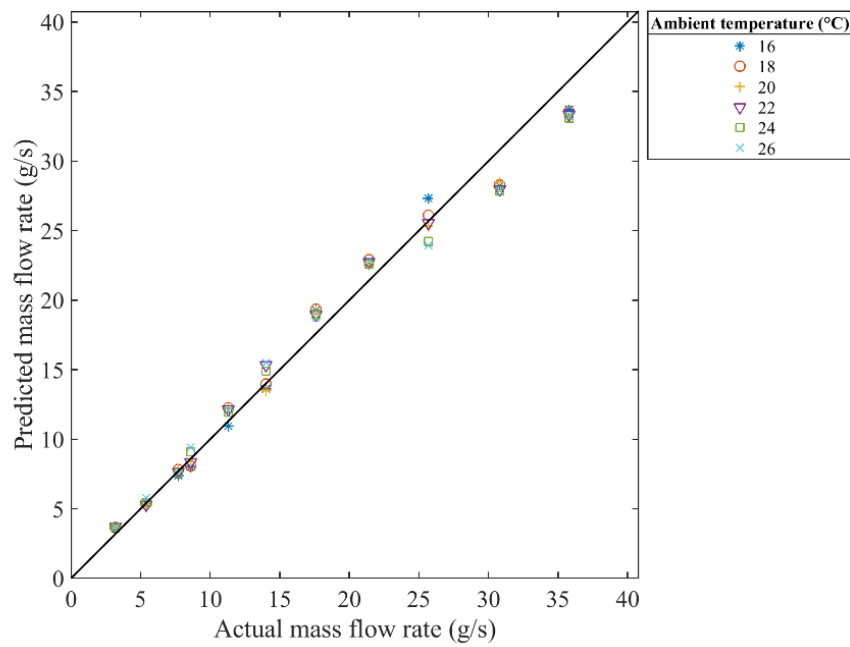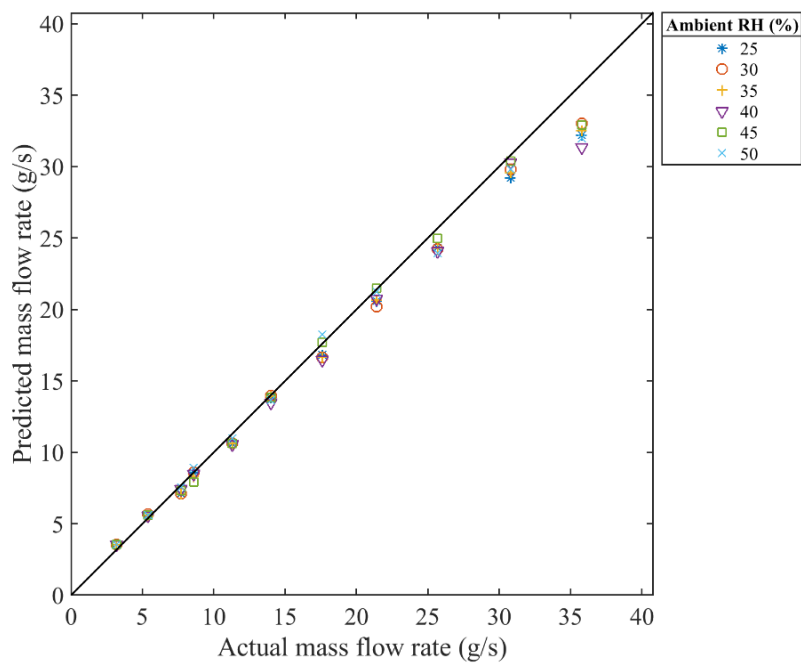
(b)

Figure 6-19: Mass flow rate measurement with CNN (a) different ambient temperatures (b) different ambient RH
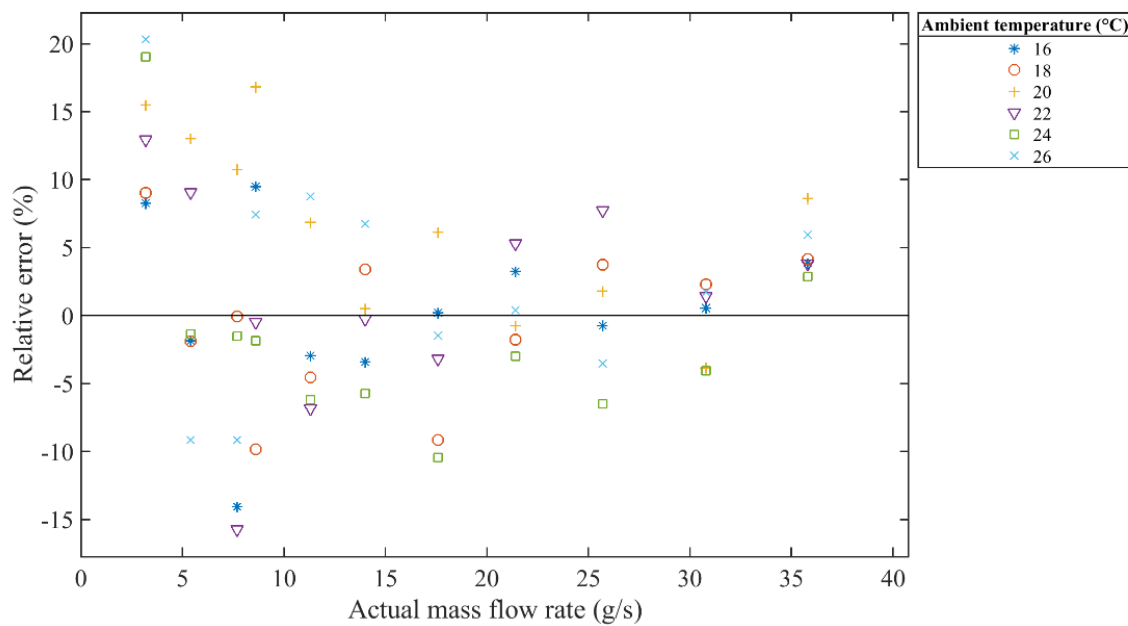
(a)
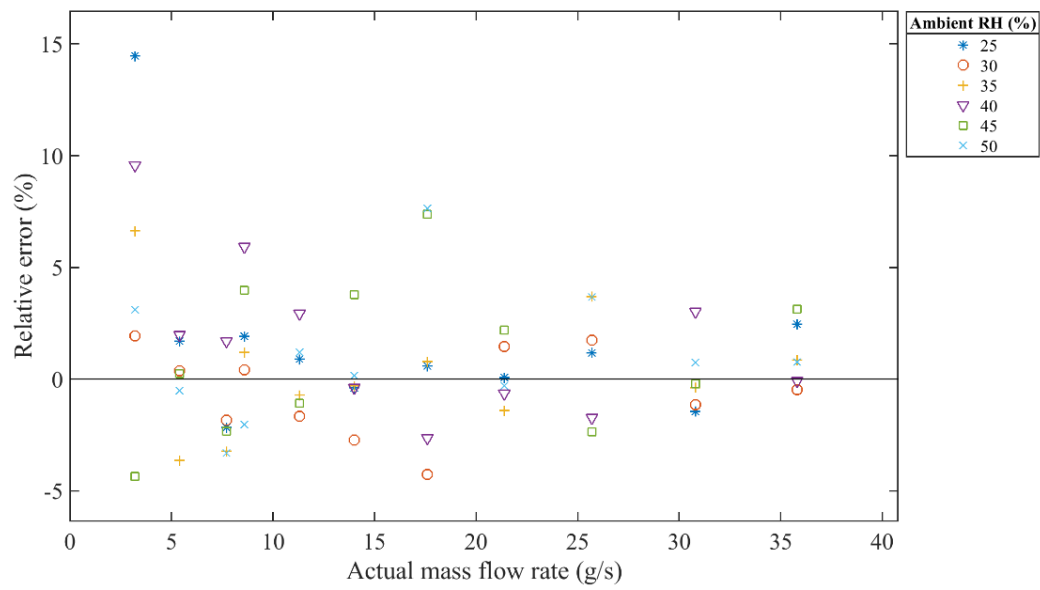
(b)

Figure 6-20: Relative error of mass flow rate measurement with ANN (a) different ambient temperatures (b) different ambient RH

(a)

188
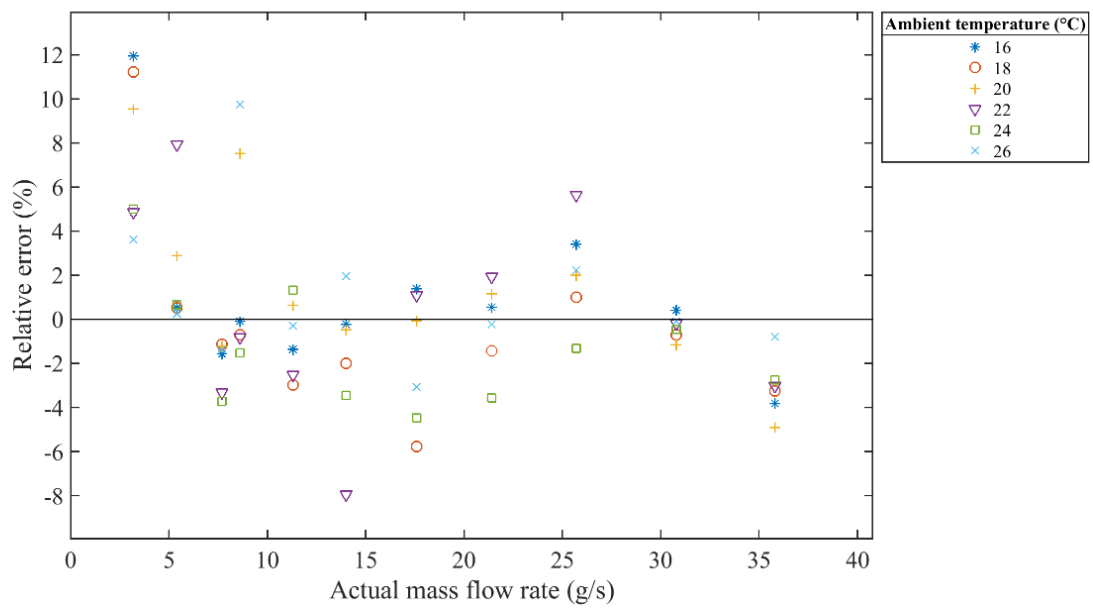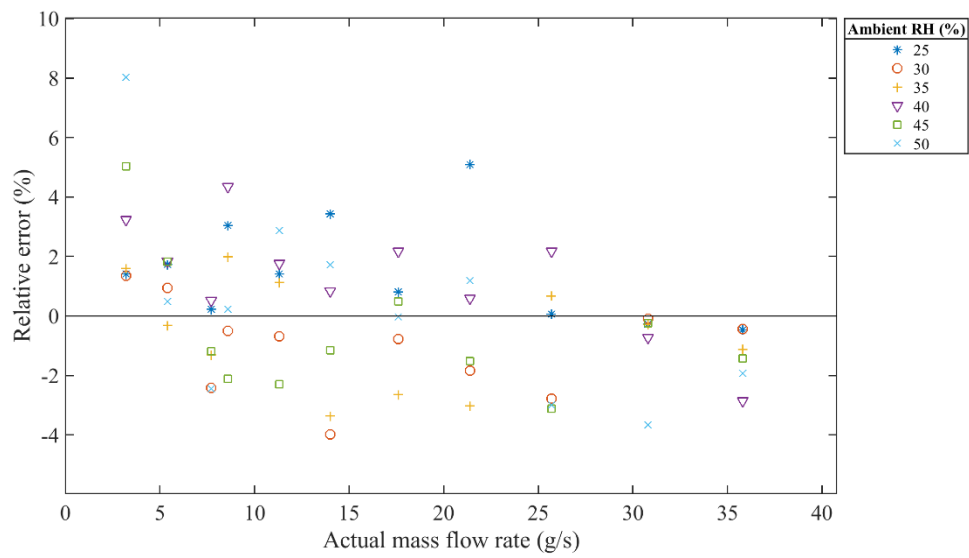
(b)

Figure 6-21: Relative error of mass flow rate measurement with SVM (a) different ambient temperatures (b) different ambient RH

(a)

189
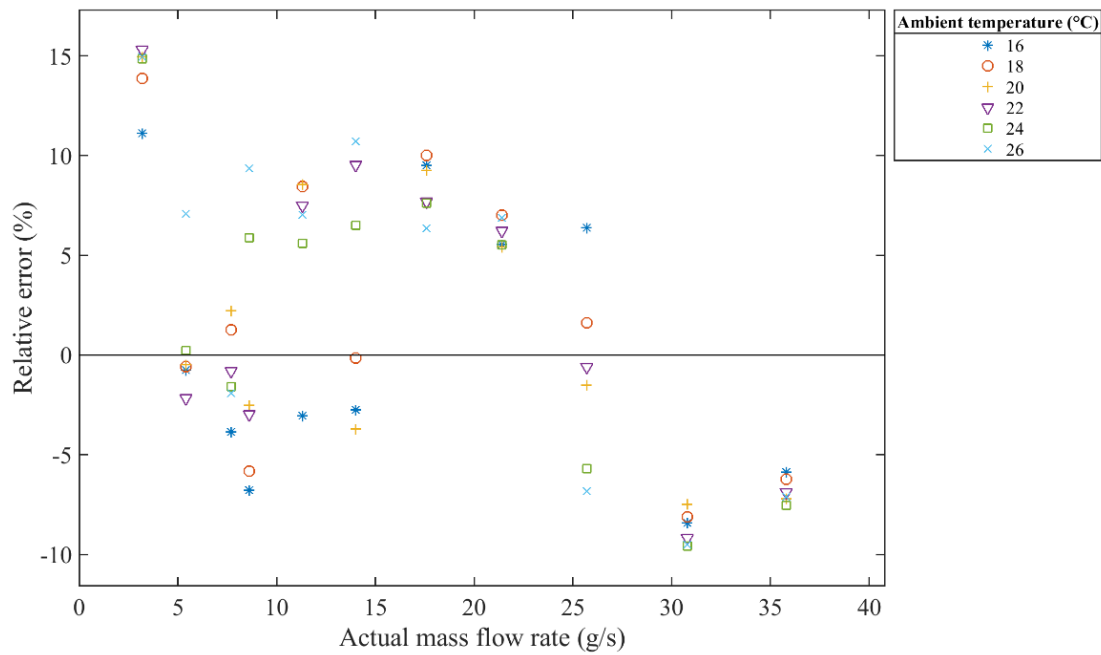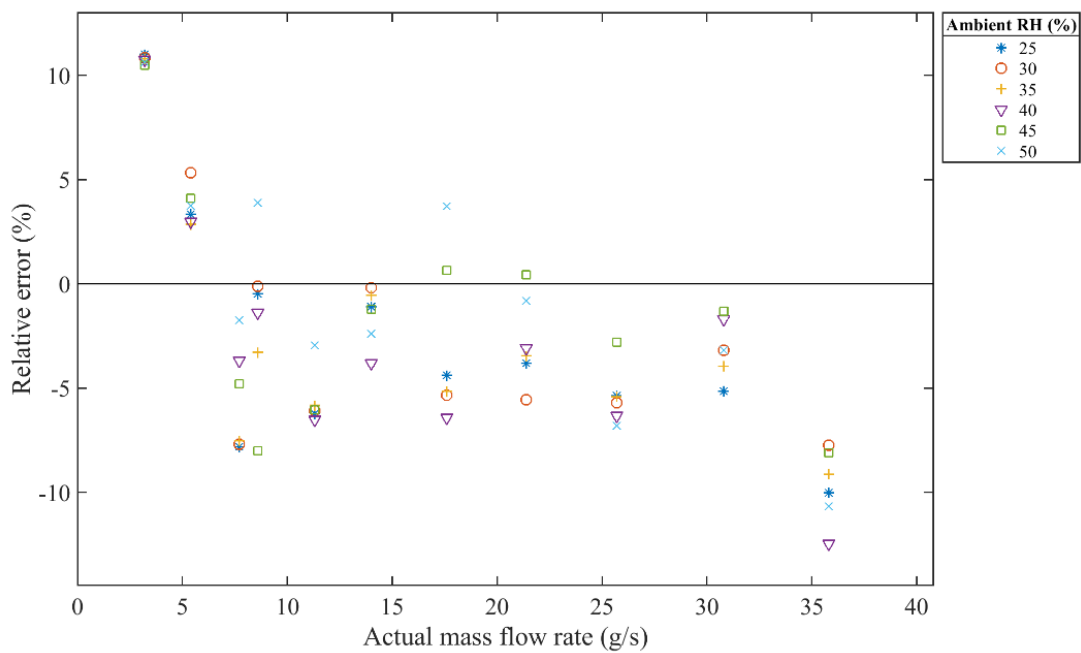
(b)

Figure 6-22: Relative error of mass flow rate measurement with CNN (a) different ambient temperatures (b) different ambient RH
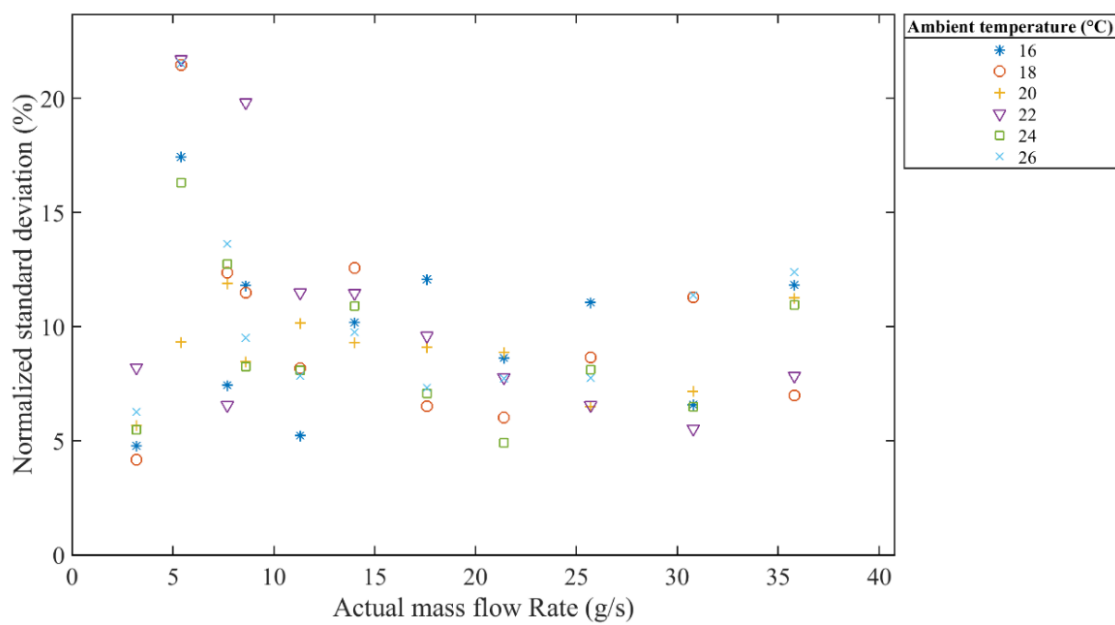
(a)
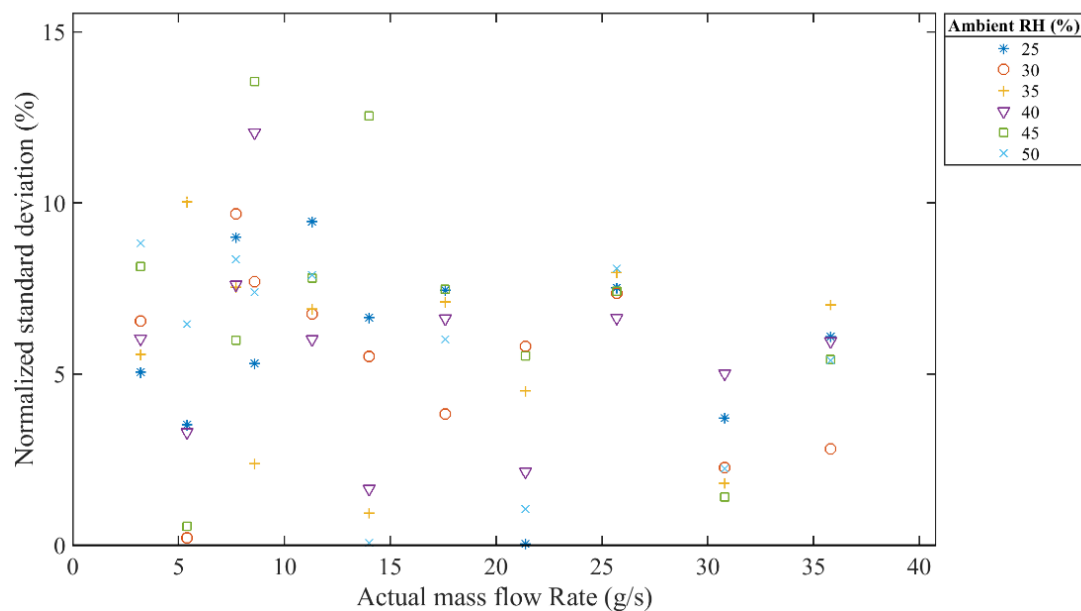
(b)

Figure 6-23: NSTD of mass flow rate measurement with ANN (a) different ambient temperatures (b) different ambient RH
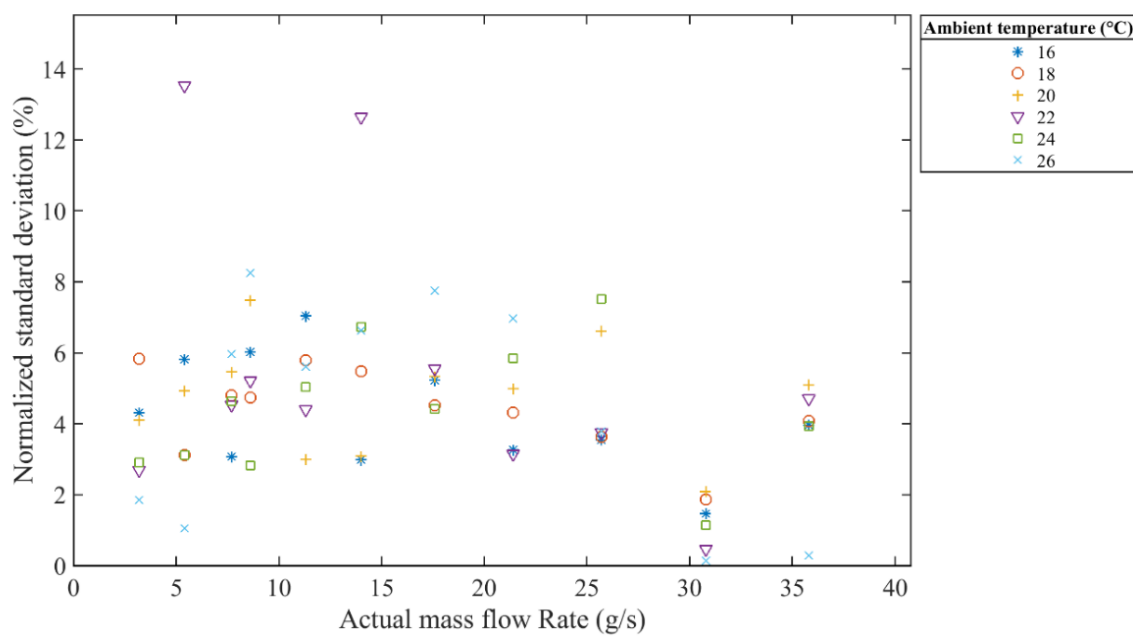
(a)
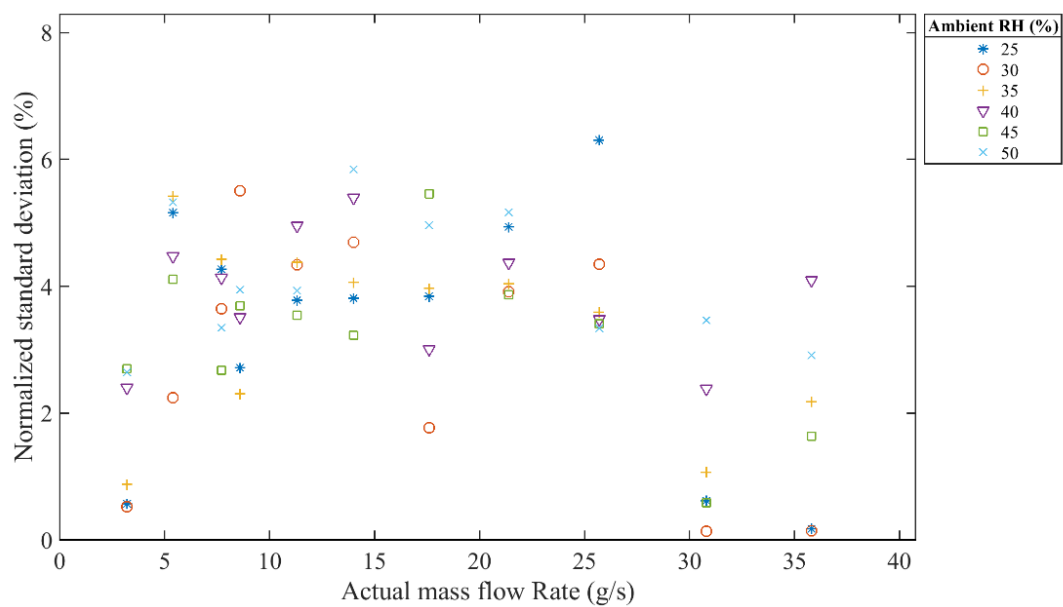
(b)

Figure 6-24: NSTD of mass flow rate measurement with SVM (a) different ambient
temperatures (b) different ambient RH
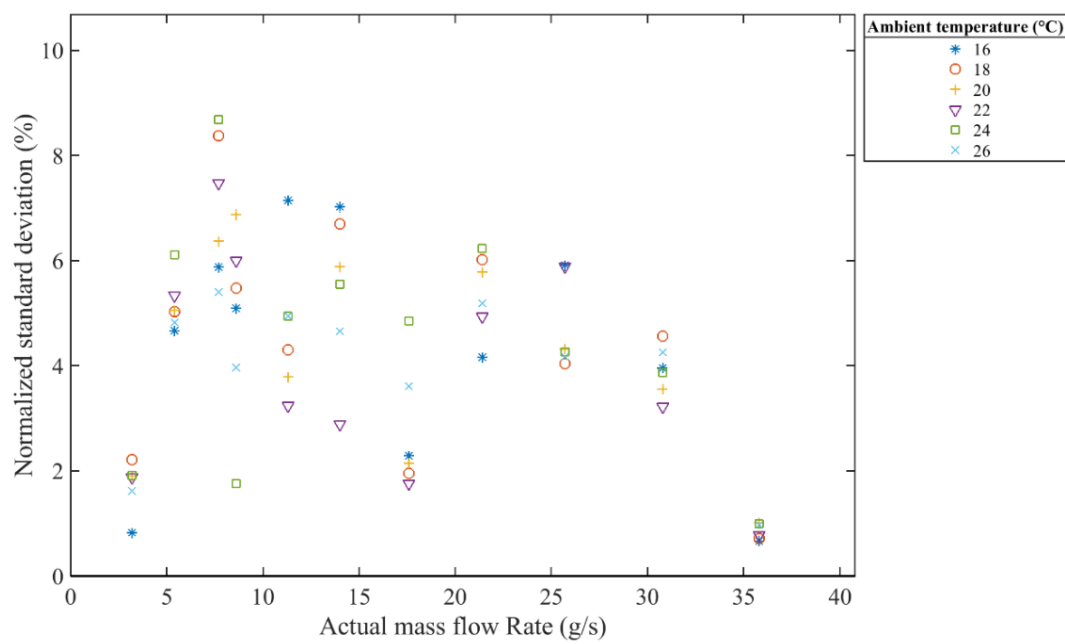
(a)
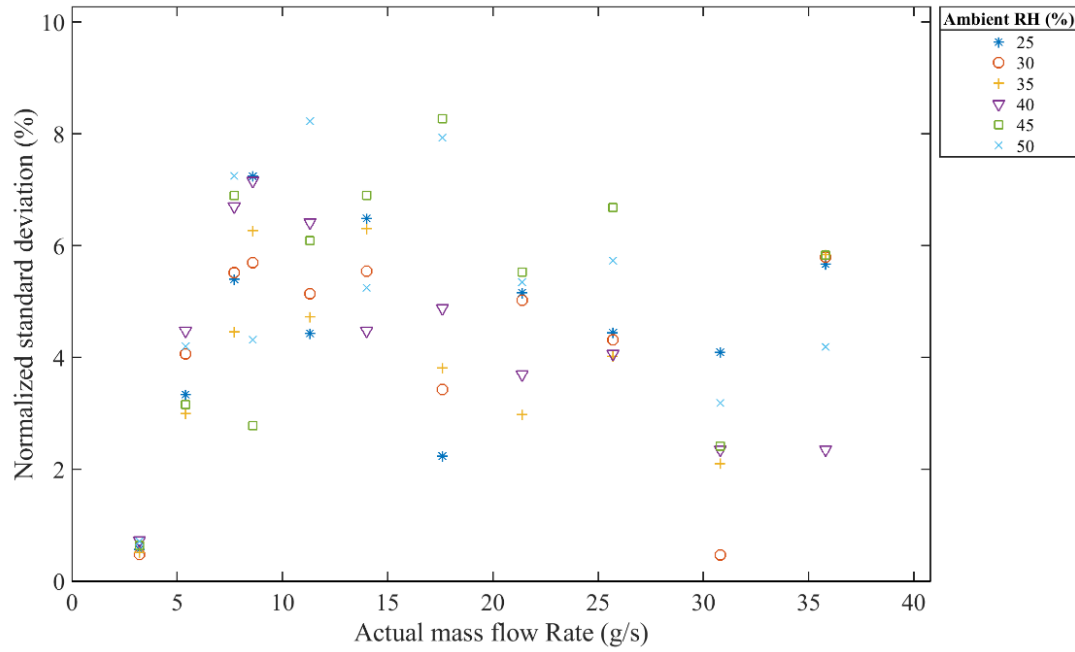
(b)

Figure 6-25: NSTD of mass flow rate measurement with CNN (a) different ambient temperatures (b) different ambient RH

6.4 Measurement of Individual Ratios of Amino Plastic and Biomass in the Mixture of Amino/Biomass/Air

6.4.1 Experimental Conditions

The experiments were carried out using amino plastic as substitute of coal due to health and safety concerns. The physical properties of these particle types are explained in Chapter 3. The data to test the models are classified into seen and unseen conditions. The seen test conditions include volumetric flow rates of solids which are also used for the training of models as listed in Table 6-8. All the data in Table 6-8 is divided into 70% for training and 30% for testing. However, the unseen test conditions are kept entirely different from the mass flow rates of solids used for the training of the models. All the unseen test conditions are chosen in a way to assess the prediction accuracy of a data driven model for the points lying between two adjacent training points. Two of the unseen mass flow rate conditions, 0/100 and 100/0, are chosen from outside the training and test data limits to evaluate the generalization ability of the models. It has to be noted that the solids

193

flow rate is controlled by volume instead of mass and this is due to the fact that coal and biomass has different densities. A change in coal/biomass ratio will bring a huge difference in the overall mass of the mixture. Furthermore, the vibratory feeder can only supply a consistent amount of mixture by volume and not by mass.

Table 6-8: Training and test conditions of coal/biomass ratios for data-driven models

		Coal/Biomass ratio										
		Training and test conditions					Unseen test conditions					
		10/90	30/70	50/50	70/30	90/10	0/100	20/80	40/60	60/40	80/20	100/0
Solids flow rate (ml/s)	1.66	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2.50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	4.50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	6.25	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	9.50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	12.50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	17.50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	22.67	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	27.00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	32.33	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	36.50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

6.4.2 Measurement Principle

Figure 6-26 shows the overall measurement strategy that starts with the collection of data from the sensing unit which consists of an array of ring-shaped electrostatic sensors, four arrays of arc-shaped electrostatic sensors, a capacitive sensor and a DP transducer (as presented in Chapter 3). Different ratios of coal/biomass may exhibit different charging properties, dielectric constant and drop in line pressure, therefore three types of sensors are selected in order to acquire adequate information about the physical characteristics of the flow. Unnecessary peaks and the glitches in the sensing data are removed at the stage of removal of outliers. The PMI is used to measure the information between each statistical feature and its corresponding output. The program for the proposed measurement strategy is provided in Appendix 5.
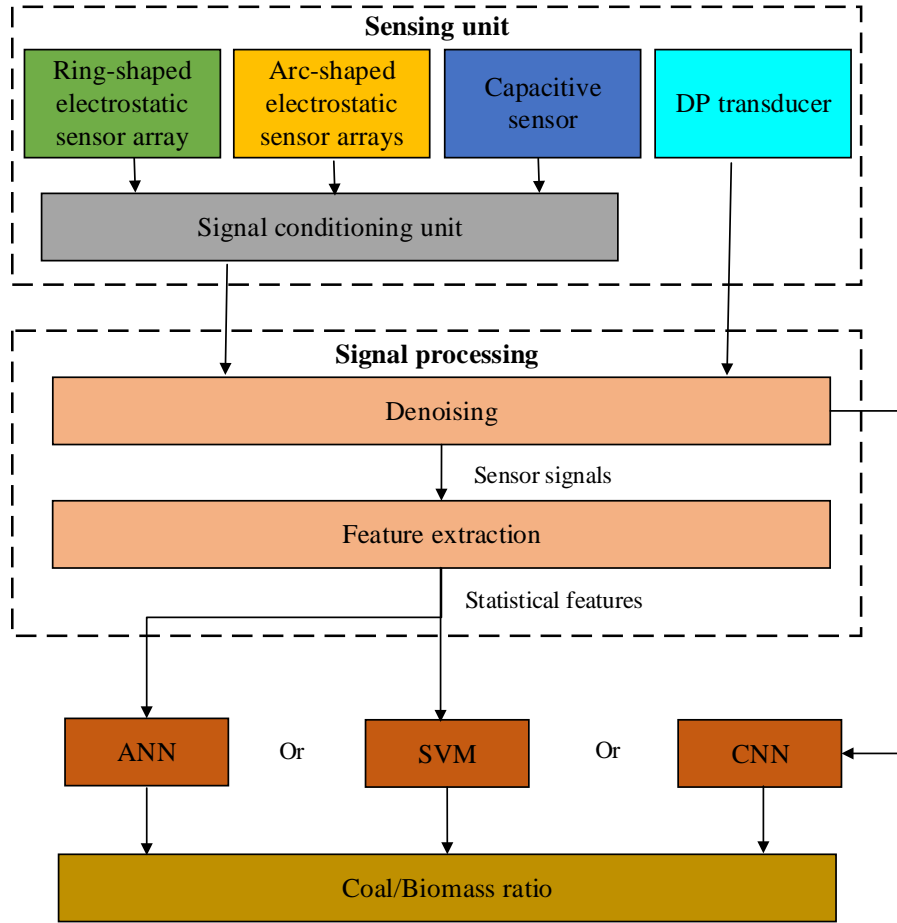
Figure 6-26: Measurement principle of coal/biomass ratio through data-driven modelling

6.4.3 Sensor Data

Figure 6-27 shows the relationship between the RMS of a post-processed signal from the E1 electrode of the ring-shaped electrostatic sensor and the coal/biomass ratios under different volumetric flow rates of solids. The reference of the volumetric flow rate of solids was based on a container with labeled volume levels and a timer. The coal particles are smaller in size and carry more surface charge, therefore the magnitude of electrostatic signal increases nonlinearly by increasing the concentration of coal particles in the coal/biomass mixture and the volumetric flow rate of solids. Since the coal particles have higher value of dielectric constant, the increasing concentration of coal particles in the coal/biomass mixture causes higher permittivity in the pipe and hence the increasing magnitude of the signal from capacitive sensor as shown in Figure 6-28. It is worth noting that the solids flow rate has very low impact on the signal of capacitive sensor. Figure 6-29 shows the dependency of the differential pressure on the coal/biomass ratios and the

195

solids flow rate in the pipeline. Drop in line pressure also increases nonlinearly with the increasing ratio of coal particles in coal/biomass mixture and the solids flow rate due to higher friction faced by the coal particles (with higher density) with the pipe wall.
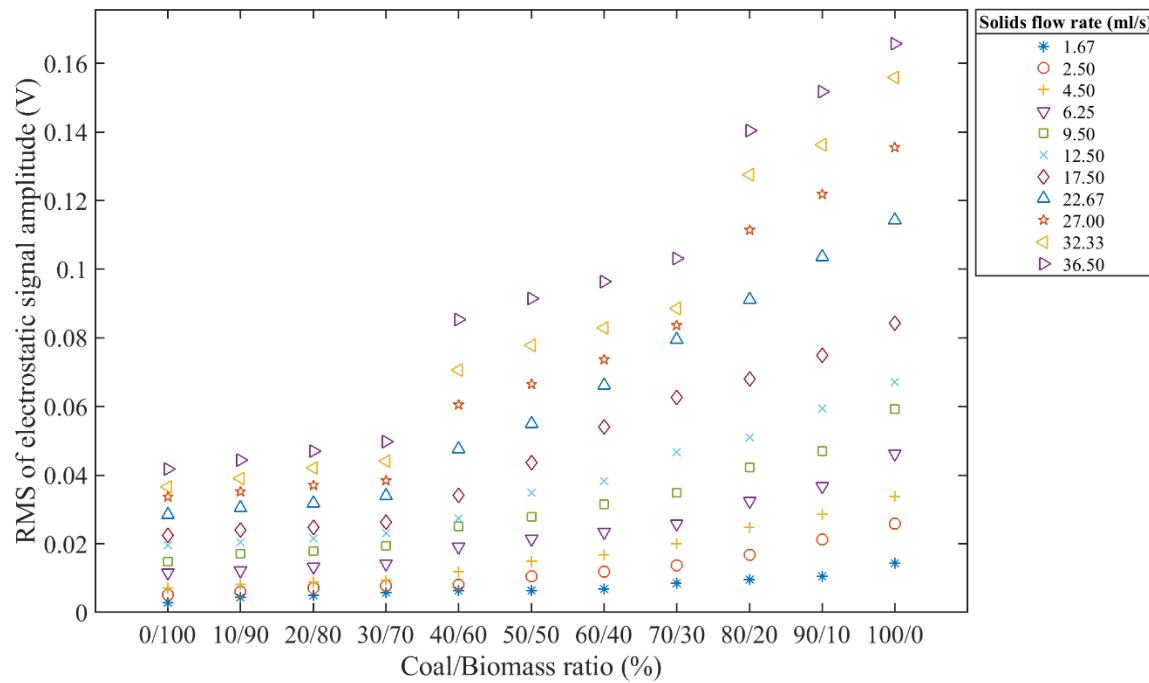
Figure 6-27: RMS of signal from ring-shaped electrostatic sensor under different coal/biomass ratios and volumetric flow rates of solids
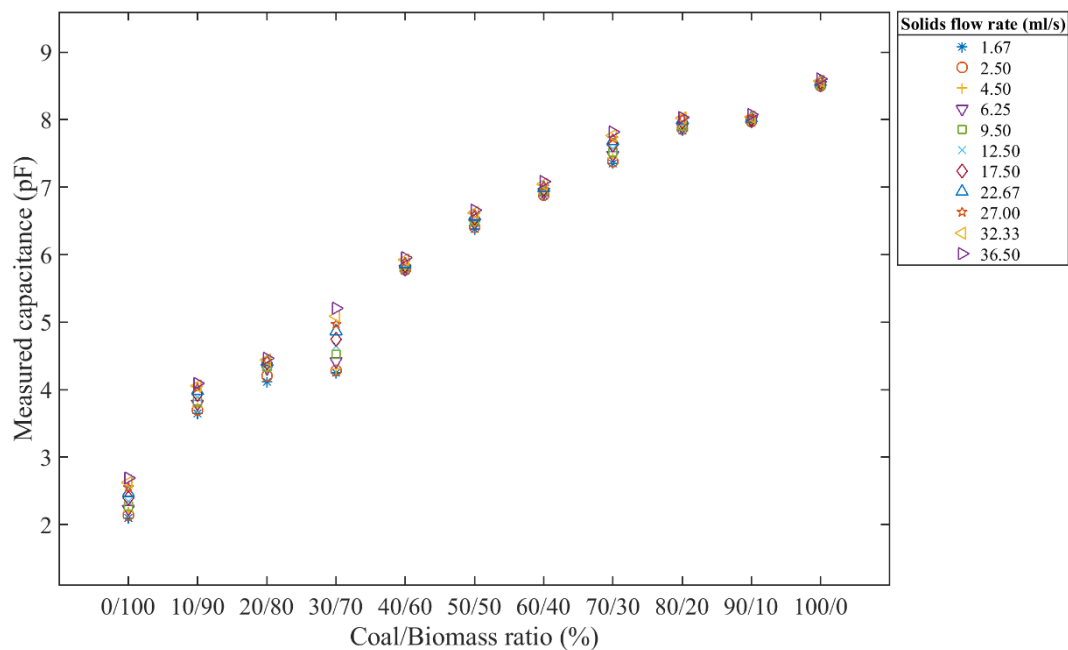
Figure 6-28: Measured capacitance values under different coal/biomass ratios and volumetric flow rates of solids
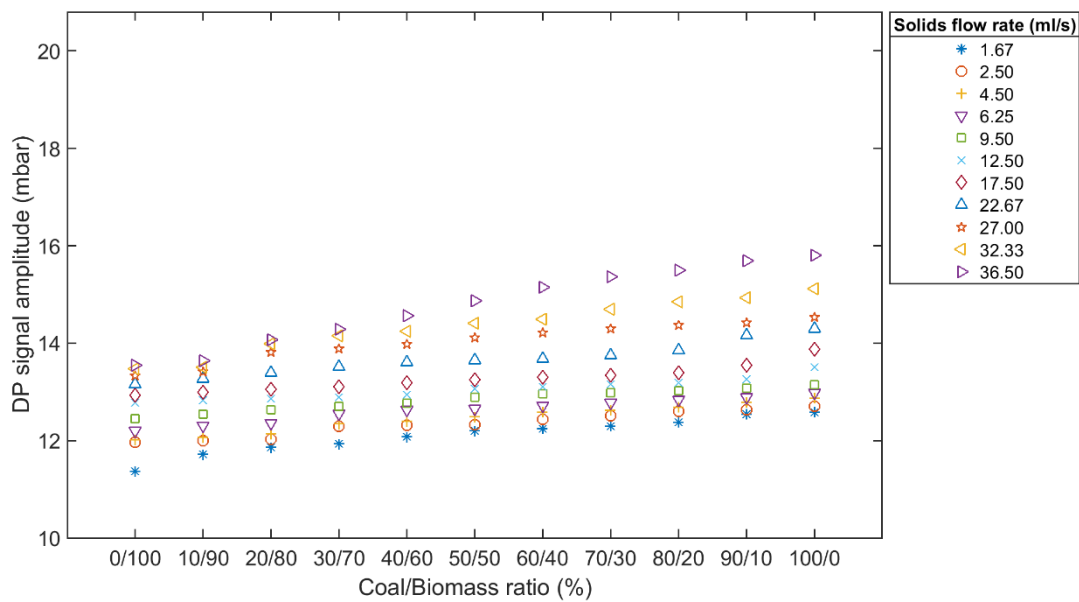
Figure 6-29: Averaged signal from DP transducer under different coal/biomass ratios and volumetric flow rates of solids

6.4.4 Measurement with ANN, SVM and CNN

Figure 6-30 to Figure 6-32 show the predicted coal/biomass ratios with the three trained data driven models, including ANN, SVM and CNN models. Results of each model are compared with the ideal straight line that shows the proximity of the predicted results to the reference coal/biomass ratio. The models are tested under eleven different coal/biomass ratios where each ratio also carries the data of eleven different solids flow rates. The outputs of the data driven models are based on the random values of the weights and the biases in the structure, therefore the predicted mass flow rates are randomly distributed around the reference lines.

The error in the predicted coal/biomass ratio under unseen test conditions is higher than that under the seen test conditions. Figure 6-33 shows the poor generalization ability of the ANN model with maximum relative error of 22%. Compared to the ANN model, the SVM performs better due to its improved generalization abilities and keeps the relative error within ±14%, as shown in Figure 6-34. The SVM model avoids the over-fitting of data by maintaining a balance between the prediction error at validation and the unseen dataset. The validation dataset, in this case, is a portion of seen test conditions. Therefore, the prediction error with the SVM model for unseen coal/biomass ratios is not significantly higher than those seen coal/biomass ratios. The CNN model performs well when there is a high volume of sensor data that can be used to formulate enough images to input the model. The relative error between the measured and the actual coal/biomass ratios remains within ±9% under all the seen and unseen test conditions, as shown in Figure 6-35.

The maximum normalized standard deviation (NSTD) of the predicted coal/biomass ratios with the trained ANN, SVM and the CNN models remains within 12%, 26% and 6%, respectively, as shown in Figure 6-36 to Figure 6-38. The comparison between the NSTD values for the three models also shows that the predictions made with the CNN are more repeatable than the ANN and SVM models.
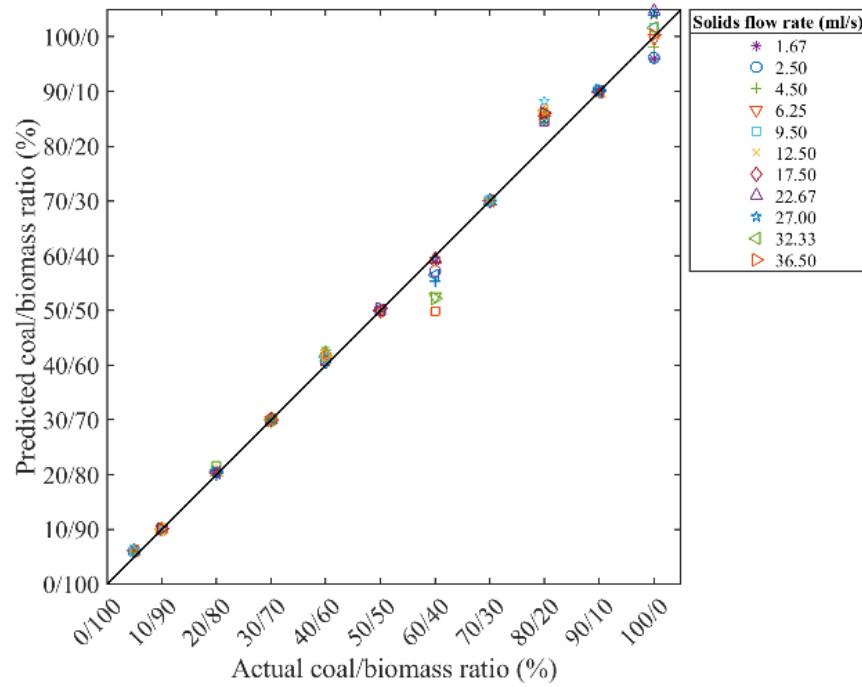
Figure 6-30: Measurement of coal/biomass ratio with ANN
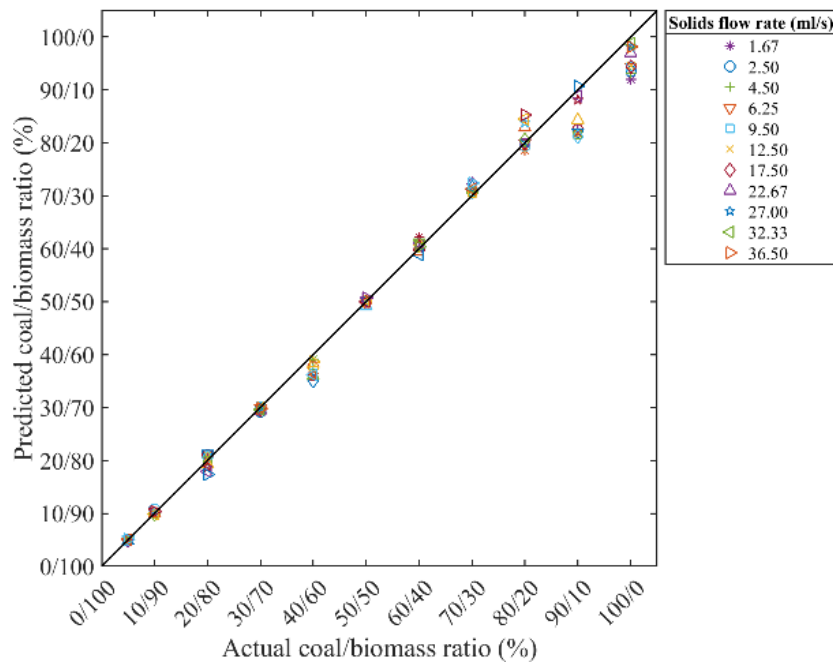
Figure 6-31: Measurement of coal/biomass ratio with SVM
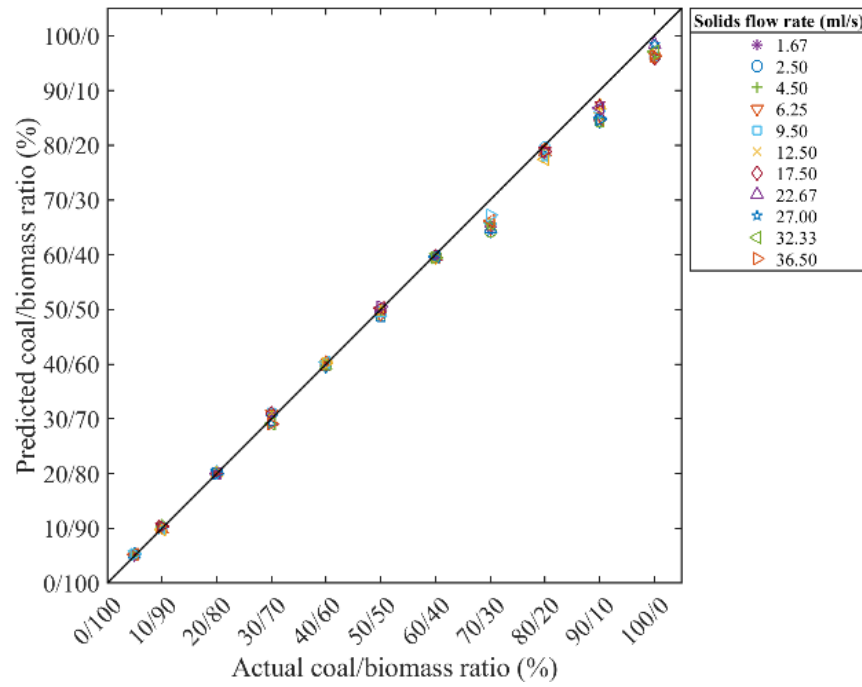
Figure 6-32: Measurement of coal/biomass ratio with CNN
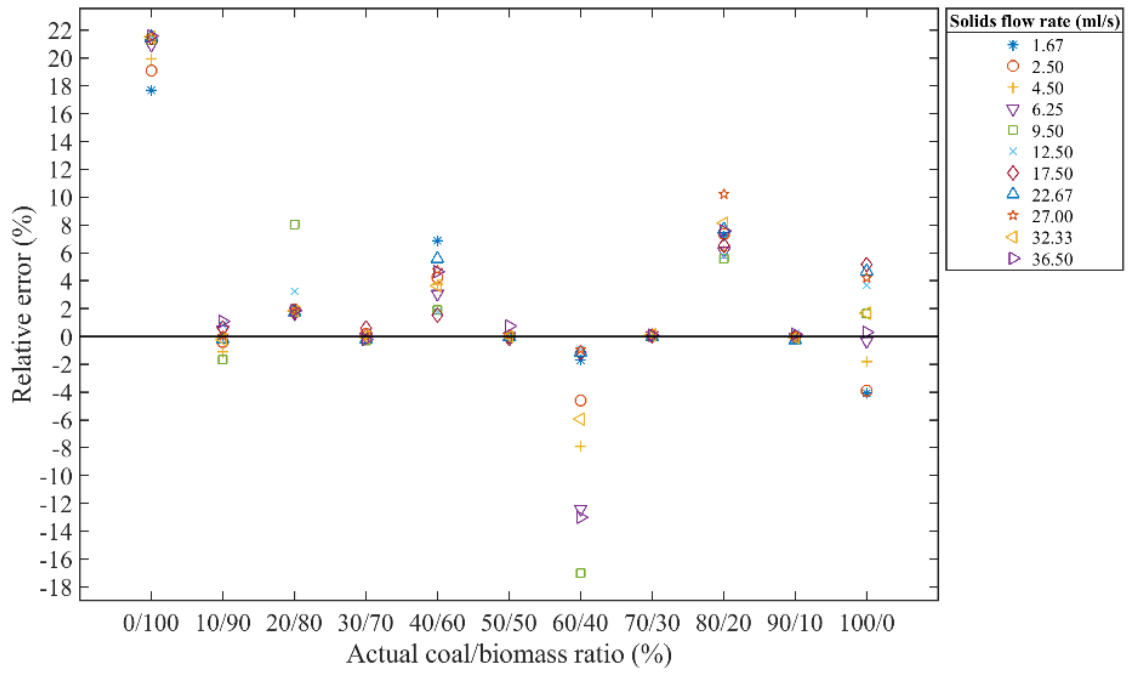
Figure 6-33: Relative error of measurement of coal/biomass ratio with ANN
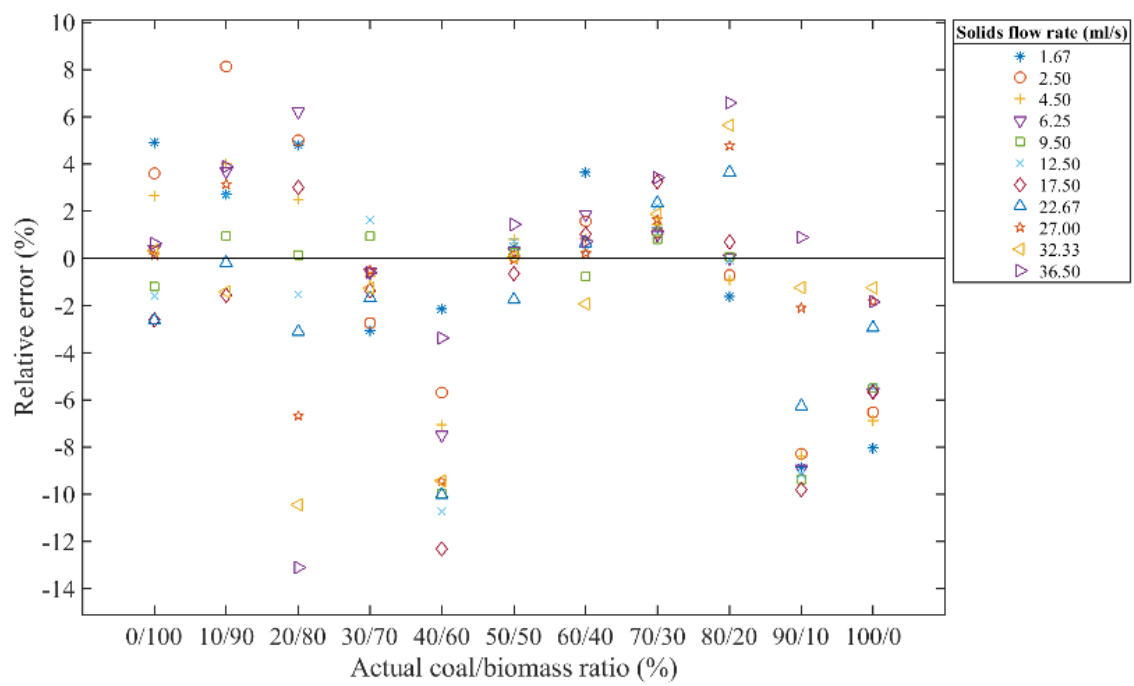
Figure 6-34: Relative error of measurement of coal/biomass ratio with SVM
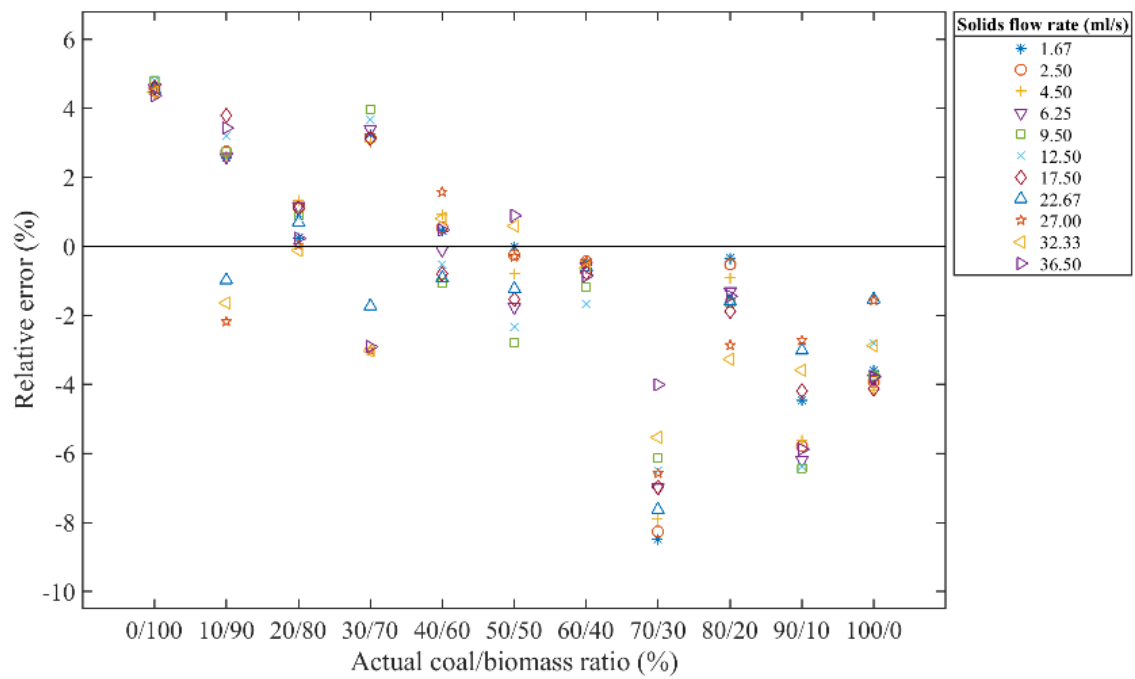
Figure 6-35: Relative error of measurement of coal/biomass ratio with CNN
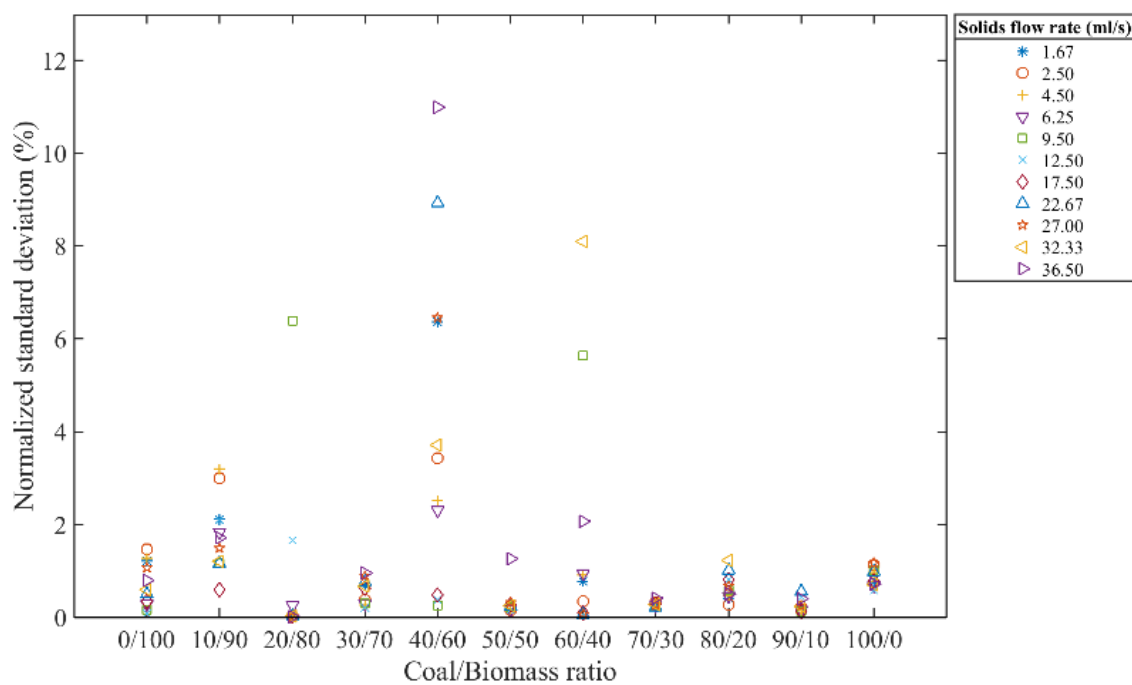
Figure 6-36: NSTD of measurement of coal/biomass ratio with ANN
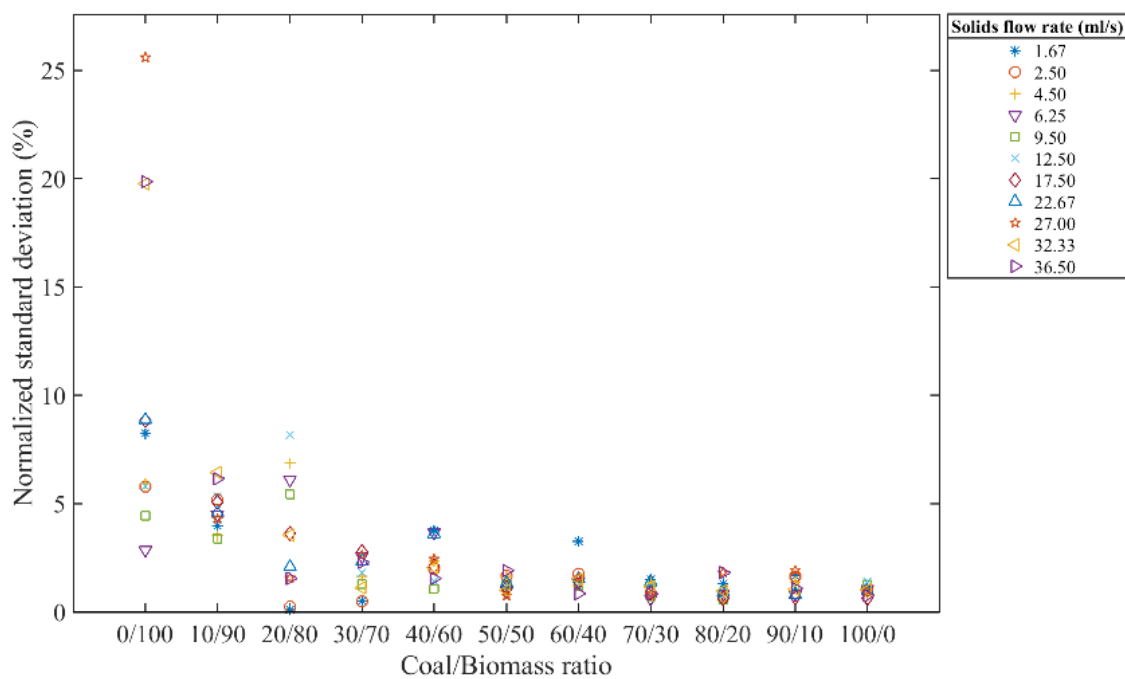
Figure 6-37: NSTD of measurement of coal/biomass ratio with SVM
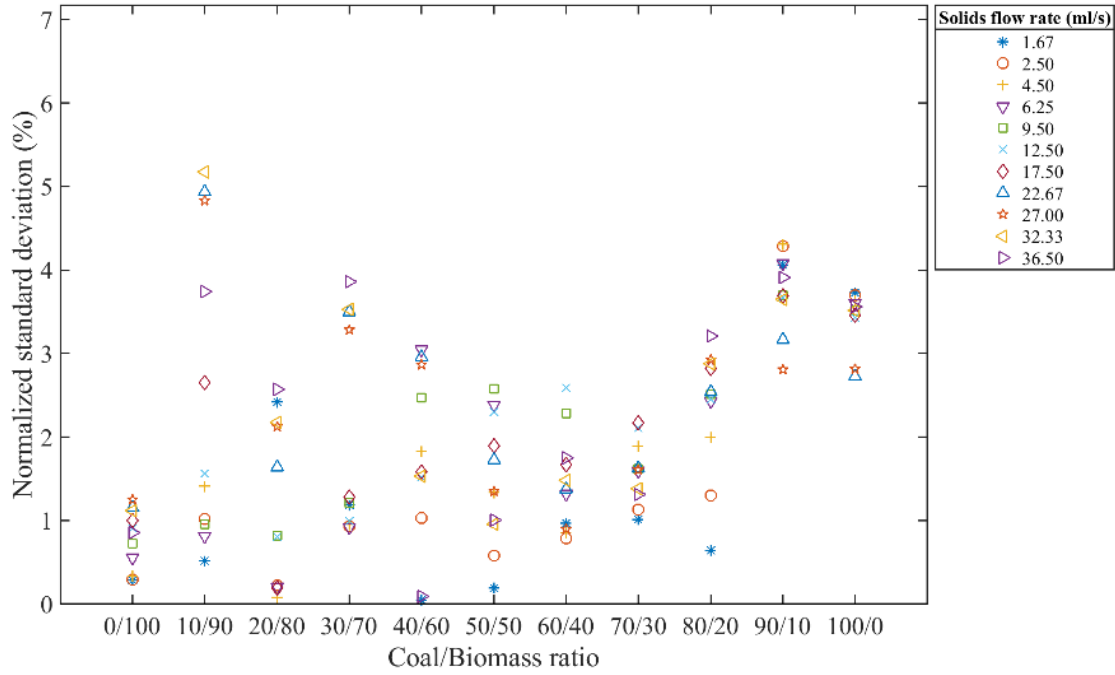
Figure 6-38: NSTD of measurement of coal/biomass ratio with CNN

6.5 Summary

This chapter presents the results of mass flow rate measurement under different pipe orientations and ambient conditions and the measurement of coal/biomass ratios through data-driven modelling approach. For the measurement of mass flow rate of solids under different pipe orientations, a multi-modal sensing unit, including ring and arc-shaped electrostatic sensors, a DP transducer and an accelerometer, in conjunction with optimized SVM model, has been proposed to measure the mass flow rate of solids in different pipe orientations. It has been observed that the RMS magnitude of the signal and corresponding PSD reduce down as the pipe moves from the horizontal to vertical orientation. The performance of the conventional SVM model trained under horizontal pipe conditions is unsatisfactory due to the limited generalization ability and lack of flow information in different pipe orientations. By training the SVM model in different pipe orientations, the relative error has been reduced from ±100% to ±40%. As the firefly optimization algorithm has been used to determine the parameters of the SVM model and the number of input features, the relative error

203

has been reduced to ±12% with the maximum normalized standard deviation of 7%. The results have demonstrated the effectiveness of the proposed methodology to measure the mass flow rate of solids in an air suspension in different pipe orientations.

A probe containing a temperature and a RH sensor has been installed non-intrusively with the sensing head to measure the temperature and RH values inside the pipe for the trials with different ambient conditions. From the analysis of sensor data, it has been seen that the temperature and RH inside the pipe remains close to the ambient conditions which were set at the beginning of each experiment. The inclusion of temperature and RH sensor data in the model has compensated and reduced the relative error in the mass flow rate measurement under different ambient conditions. Based on the nature of the data and the complexity, the SVM model has outperformed the ANN and the CNN models by giving the relative error within ±8%. Furthermore, a capacitive sensor along with high resolution

LCR meter has been interfaced with the sensing unit for the measurement of coal/biomass ratios. Analysis of sensor data shows that the different coal/biomass ratios cause the significant change in the magnitude of the electrostatic charge and the drop in line pressure. However, the different solids flow rates also influence the characteristics of the signals from electrostatic sensors and the DP transducer that make it difficult to identify whether the change is occurring because of coal/biomass ratio or the solids flow rate. On the contrary, the characteristics of the signals from capacitive sensor have not been observed to be significantly influenced by solids flow rate. Experimental results suggest that the CNN model, in this case, perform better than ANN and the SVM models with relative error within ±9%.

Chapter 7 Conclusions and Recommendations for Future Work

7.1 Introduction

This chapter presents the conclusions that have been drawn from the research programme conducted and makes the recommedations for the future work in the field of gas-solid flow.

The research work presented in this thesis concerned with a technique for the mass flow rate measurement of pneumatically conveyed solids based on multi-modal sensing and data driven modelling. The multi-modal sensing system is comprised of an array of ring-shaped electrostatic sensors, four arrays of arc-shaped electrostatic sensors, a differential-pressure transducer, accelerometer, temperature, humidity and capacitive sensor. Data driven models, including artificial neural network (ANN), support vector machine (SVM), and convolutional neural network (CNN), have been established through training with statistical features extracted from the post-processed data from the sensing system. Performance of the inferential method of mass flow rate measurement of solids using electrostatic sensors has also been investigated under variety of physical conditions.

The sensing unit has been modified by integrating a accelerometer to determine the orientation of pipe to produce the angle compensated mass flow rate of solids. A probe containing a temperature and a RH sensor is installed non-intrusively inside the pipe to compensate the effects of ambient conditions on mass flow rate measurement. Two electrodes covering the whole cross-section of the pipe have been flush-mounted inside the pipe to formulate a capacitive sensor for the measurement of ratios of different particle types in the complex mixture of air and two different type of particles (coal and biomass). The test rig has been modified to make the placement of sensing unit fully flexible for the trials under different pipe orientations. A control system, comprised of two VFDs and a DAQ unit, has been desiged to control the velocity of air and the solids concentration through computer during the process of dataset collection. Multiple datasets

have been collected under variety of flow conditions, including different mass flow rates, air velocities, pipe orientations, ambient conditions and the particle types. The soft computing techniques, such as SVM and PMI, have been applied, respectively, to evaluate the importance of all the sensors and each of the statistical feature extracted from the shorlisted sensors. The data-driven models, including ANN, SVM and CNN models, have been applied to establish the relationship between sensor data and the mass flow rate of solids. Performance of these models has also been compared in terms of accuracy, window size requirement and repeatability of the measurements.

7.2 Conclusions

7.2.1 Overall Measurement Platform

A sensing unit, comprised of an array of ring-shaped and four arrays of arc-shaped electrostatic sensors, a DP transducer, a temperature, a RH, an accelerometer and a capacitive sensor is developed for the measurement of mass flow rate of solids under different physical conditions. The inclusion of multiple types of sensors in the sensing unit has made the measurement system capable of compensating different physical conditions. Each array of multiple electrostatic sensors has the advantage of providing more relaibale information about the flow parameters. The conversion of 4 to 20 mA signal, from DP transducer, to 0 to 10 V has also been elaborated in the view of specifications from the manufacturer. The test rig, that is used to carry out all the experiments, is capable of controlling air velocity and solids concentration through air suction pump and the vibratory feeder. Two high speed commercial level VFDs have been interfaced with the DAQ to control the suction power and the magnitude of vibrations through computer. It has become more convenient to control the suction power and the magnitude of vibration during the experiments through computer using two VFDs. The integration of VFDs with suction pump and the vibratory feeder has helped reduce the uncertaininity in the reference particle velocity and the solids concentration which was quantified through normalized standard deviation of the refrence data.

The holding structure of the sensing unit has been modified in such a way that enables the sensing unit to be placed in any orientation. The dynamic movement of the sensing head along with two flexible pieces of pipe has facilitated to conduct the experiments in different pipe angles. The data acquisition and the control system has the advantage of obtaining 50,000 samples from all the

sensors in one single matrix. The DAQ unit has been used for the dual purpose of acquiring the sensor data and controlling the VFDs simultaneously. The programme of data acquisition has the ability of collecting all the sensor data under given parameters without human intervention in operating suction pump and the vibratory feeder. The developed user interface to produce the measurement results in the real-time continuous manner only requires a user to start the programme. Upon starting the measurement programme, the results are produced using pre-trained data-driven models and the real-time test data from all the sensors.

7.2.2 Inferential Method for Mass Flow Rate Measurement of Solids

There is a low degree of error for the measurement of particle velocity through cross-correlation based signal processing technique when measured with sequentially positioned upstream and the downstream electrodes. By shifting the pipe orientation from horizontal to vertical, the larger concentration of solids travels from the bottom to the left side of the pipe. In case a specific type of sensor does not function well, it has been seen that the relative error in the measurement of a flow parameter can be reduced by combining the data from various sensors. As a result, increasing the number of electrodes has resulted in an improvement in the overall accuracy of the mass flow rate measurement. The four arcs, which span the top, front, back, and bottom of the pipe, give more concise and localised flow parameter information. Consequently, when comparing the averaged mass flow rate of solids measured with four arc-shaped electrostatic sensors to the ring-shaped electrostatic sensors, the averaged mass flow rate of solids recorded with four arc-shaped electrostatic sensors has offered the superior performance.

The key benefit of the direct technique of mass flow rate measurement is that it requires fewer samples from electrostatic sensor signals and produces more repeatable results, as long as the system is properly calibrated and the physical parameters stay constant. The approach, however, is inferential and is based on particle velocity and solids concentration. Physical factors such as air velocities, pipe orientations, and ambient conditions have been found that can impact the RMS charge level of electrostatic sensors, which is mainly used to measure the solids concentration. It is not possible to select the values of parameters a and b in a way to keep the relative error of mass flow rate measurement low for all the values of physical factors.

With the help of a optimization algorithm, the values of the parameters a and b are selected to optimize the overall degree of relative error in the measurements under different physical factors.

Under varied air velocities, pipe orientations, and ambient circumstances, the mean NRMSE of the mass flow rate measurement has been raised from 8% to 18%, 50%, and 35%, respectively. The measurement principle necessitates the recalibration of the system with new values of the parameters a and b under various physical circumstances, which can be time intensive.

7.2.3 Mass Flow Rate Measurement Through Data-Driven Modeling

With the development of on-going analytical modelling of gas-solids two-phase flow, engineering judgements in the application of modern machine learning techniques have played a important role to deal with the challenges of flow measurements under different physical conditions. The data-driven models, including ANN, SVM and the CNN models have been established through shortlisted sensor data for the the measurement of mass flow rate of solids. The generalization ability of the models has been tested over a range of seen and unseen test conditions. The CNN model has outperformed the ANN and the SVM models with a relative error within ±8% while the normalized standard deviation within 5% in both horizontal and vertical pipe orientations for the mass flow rate measurement with different air velocities. The ANN and the SVM models have yielded relative errors of ±18% and ±14%, respectively, with normalized standard deviations of 14% and 8%, respectively.

The characteristics of the DP transducer has constructively enhanced the performance of the models. Similar type of electrostatic sensors has given same performance. However, the SVM model does not perform well when trained with the combination of several electrostatic sensors from the same type of electrodes due to over-fitting problem of the model. It has also been observed that all the sensors does not provide the same number of statistical features that are useful for the development of data driven models. Based on the partial mutual information algorithm, it has been determined that only certain statistical characteristics are valuable while others are considered redundant. The CNN model, in comparison to the ANN and SVM models, requires a smaller window size to extract all of the information from the sensor data in order to make a single prediction.

The inclusion of an accelerometer in the multi-modal sensing unit in conjunction with an optimised SVM model has been proposed to measure the mass flow rate of solids in different pipe orientations, . As the pipe goes from a horizontal to a vertical position, the RMS magnitude of the signal and the accompanying PSD both decrease. Due to its limited generalisation capacity and

lack of flow information in multiple pipe orientations, the performance of the traditional SVM model developed under horizontal pipe circumstances is poor. By training the SVM model in different pipe orientations, the relative error has been reduced from ±100% to ±40%. As the firefly optimization algorithm has been used to determine the parameters of the SVM model and the number of input features, the relative error has been reduced to ±12% with the maximum normalized standard deviation of 7%.

For the experiments with varied environmental conditions, a probe comprising a temperature and RH sensor has been placed non-intrusively in the sensing head to monitor the temperature and RH values within the pipe. The temperature and RH inside the pipe stay near the ambient parameters that were set at the start of each trial, according to sensor data analysis. The addition of temperature and relative humidity sensor data to the model reduced the relative error in mass flow rate measurement under various ambient conditions. The SVM model surpassed the ANN and CNN models by delivering a relative error within 8%, based on the nature of the data and its complexity.

The sensing unit has also been interfaced with a capacitive sensor and a high-resolution LCR meter for the measurement of coal/biomass ratios. The various coal/biomass ratios produce a substantial shift in the amplitude of the electrostatic charge and a reduction in line pressure, according to the sensor data presented. The characteristics of the signals from electrostatic sensors and the DP transducer have also been influenced by the varied solids flow rates, making it difficult to determine whether the change is due to the coal/biomass ratio or the solids flow rate. The characteristics of the capacitive sensor signals, on the other hand, do not seem to be significantly impacted by the solids flow rate. In this case, experimental findings show that the CNN model outperforms the ANN and SVM models with a relative error within 9%.

7.3 Recommendations for Future Work

7.3.1 Combination of Inferential and Data-Driven Modelling Techniques

This research programme has presented the results of mass flow rate measurement with inferential and data-driven modelling techniques. Both of these techniques has some advantages and disadvantages. For example, the inferential technique provides more accurate yet the repeatable measurements provided the physical conditions are constant throughout experiments. Data-driven modelling techniques, on the other side, perform well under different physical conditions,

however, requires a high volume of data for training. In this case, it might be a good idea to combine both of these techniques to balance the flaws of eachother. The inferential method can be implemented for the mass flow rate measurement of solids while using the data-driven modeling techniques for the tuning of parameters 'a' and 'b' of inferential method. Since the measurement strategy is not entirely dependent on the data-driven models, therefore that effort of collecting large volume of data can be avoided.

7.3.2 Trials on Different Test Rigs

All the experiments, that were carried out in the different studies, are based on negative pressure test rig. The negative pressure test rig uses negative pressure to convey the solids at high velocity along both horizontal and vertical pipes. A suction pump is used as source of air. In this case, all the solids move in a counter-clockwise direction and fall downwards in the same direction as gravitational force. However, a positive test rig uses positive pressure to convey the solids at high velocity along both horizontal and vertical pipes. The vacuum is created with a blower fan while making the solids move in a clockwise direction and against the direction of gravitational force in the vertical pipe section. The solids will be moving against the gravitational force and hence it may cause a higher drop in line pressure and reduction in the magnitude of electrostatic sensor signals. Consequently, the different test rig setup may influence the characteristics of the sensor signals and result in retrtaining of the data-driven models.

7.3.3 Trials on Different Pipe Diameters

The data-driven models are trained and tested with the data extracted from the sensors installed on the spoolpiece of constant diameter of 50 mm. The distribution of the solids across whole cross-section of pipe in different orientations has been presented through analysis of sensor data. However, the flow pattern of solids may go from deposition to suspension if the trials are run with the pipe of higher diameters. In this case, solids may have less chances to collide with eachother and with the walls of pipeline and hence a lower magnitude of electrsostatic charge is expected. There could also be a less drop in line pressure due to less friction between solids and the pipe wall. As a results of that, the data-driven models may need to be retrained with the sensor data under different pipe diameters.

7.3.4 Trials Under Full Plant Conditions

The actual conditions in engineering can be very different from the laboratory conditions, so the models trained in a laboratory may not perform well in practical applications. In order to make the models work well in the real world, it is imperative to collect sufficient and representative datasets under actual operational conditions on the plant where the models to be deployed. The models that have been previously developed under laboratory conditions will need to be re-trained for the intended industrial application, though the overall methodology (model structure, selected features etc.) and the whole measurement system (sensors, data acquisition unit, computing unit etc.) remain the same. Moreover, an in-situ training functionality should be incorporated so that the online training of the developed models can be updated regularly to enhance the performance of the models in the field. The main advantage of the technique based on data-driven models is that it can be applied on an industrial process plant that requires the measurement of mass flow rate of solids, provided that the datasets are collected under the actual plant conditions and are representative of the range of the conditions of the process plant. The trials under full plant conditions may require little changes to the hardware of the measurement system.

References

[1]   Y. Zheng and Q. Liu, "Review of techniques for the mass flow rate measurement of pneumatically conveyed solids," *Measurement*, vol. 44, no. 4, pp. 589–604, May 2011.

[2]   J. R. Coombes, "Development of electrostatic and piezoelectric sensor arrays for determining the velocity and concentration profiles and size distribution of pneumatically conveyed bulk solids," Ph.D. dissertation, School of Engineering, University of Kent, Aug. 2016.

[3]   Y. Yan, "Mass flow measurement of bulk solids in pneumatic pipelines," *Measurement Science and Technology*, vol. 7, no. 12, pp. 1687–1706, Dec. 1996.

[4]   D. Mills, *Pneumatic Conveying Design Guide*, 2nd ed., B.-H. Publications, Ed. Oxford, UK, 2004.

[5]   Y. Yan, "Guide to the flow measurement of particulate solids in pipelines. part 1: Fundamentals and principles." *Powder Handling and Processing*, vol. 13, no. 4, pp. 343–352, Oct. 2001.

[6]   H. Cui and J. R. Grace, "Pneumatic conveying of biomass particles: a review," *China Particuology*, vol. 4, no. 3, pp. 183–188, Jul. 2006.

[7]   X. Qian, Y. Yan, L. Wang, and J. Shao, "An integrated multi-channel electrostatic sensing and digital imaging system for the on-line measurement of biomass-coal particles in fuel injection pipelines," *Fuel*, vol. 151, pp. 2–10, Jul. 2015.

[8]   Y. Yan, Y. Hu, L. Wang, X. Qian, W. Zhang, K. Reda, J. Wu, and G. Zheng, "Electrostatic sensors - their principles and applications," *Measurement*, vol. 169, p. 108506, Feb. 2021.

[9]   R. Abdul Rahim, L. Leong, K. Chan, M. Rahiman, and J. Pang, "Real time mass flow rate measurement using multiple fan beam optical tomography," *ISA Transactions*, vol. 47, no. 1, pp. 3–14, 2008.

[10]   N. O. Mahony, T. Murphy, K. Panduru, D. Riordan, and J. Walsh, "Acoustic and optical sensing configurations for bulk solids mass flow measurements," in *2016 10th International Conference on Sensing Technology (ICST)*, Nanjing, China, Nov. 2016, pp. 1–6.

[11]   N. A. Zulkiflli, S. Ibrahim, M. H. F. Rahiman, J. Pusppanathan, R. A. Rahim, K. S. Tee, F. A. Phang, N. D. Nawi, and N. M. N. Ayob, "Ultrasound tomography hardware system for multiphase flow imaging," in *2019 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, Kuala Lumpur, Malaysia, Sep. 2019, pp. 264–268.

[12]   X. Wang, Y. Hu, H. Hu, and L. Li, "Evaluation of the performance of capacitance sensor for concentration measurement of gas/solid particles flow by coupled fields," *IEEE Sensors Journal*, vol. 17, no. 12, pp. 3754–3764, June 2017.

[13]   A. Bilgic, J. Kunze, V. Stegemann, J. Hogendoorn, and M. Zoeteweij, "Multiphase flow metering with nuclear magnetic resonance spectroscopy," in *AMA Conferences*, Nurnberg, Germany, May 2015, pp. 292–297.

[14]   J. Li, F. Fu, S. Li, C. Xu, and S. Wang, "Velocity characterization of dense phase pneumatically conveyed solid particles in horizontal pipeline through an integrated electrostatic sensor," *International Journal of Multiphase Flow*, vol. 76, pp. 198–211, Nov. 2015.

[15]   Y. Hu, Y. Yan, X. Qian, and W. Zhang, "A comparative study of induced and transferred charges for mass flow rate measurement of pneumatically conveyed particles," *Powder Technology*, vol. 356, pp. 715–725, Nov. 2019.

[16]   X. Qian, Y. Yan, X. Huang, and Y. Hu, "Measurement of the mass flow and velocity distributions of pulverized fuel in primary air pipes using electrostatic sensing techniques," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 5, pp. 944–952, May 2017.

[17]   X. Qian, X. Huang, H. Yonghui, and Y. Yan, "Pulverized coal flow metering on a full-scale power plant using electrostatic sensor arrays," *Flow Measurement and Instrumentation*, vol. 40, pp. 185 – 191, 2014.

[18]   J. B. Gajewski, "Electrostatic nonintrusive method for measuring the electric charge, mass flow rate, and velocity of particulates in the two-phase gas-solid pipe flows - its only or as many as 50 years of historical evolution," *IEEE Transactions on Industry Applications*, vol. 44, no. 5, pp. 1418–1430, Sep. 2008.

[19]   C. Xu, S. Wang, and Y. Yan, "Spatial selectivity of linear electrostatic sensor arrays for particle velocity measurement," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 1, pp. 167–176, Sep. 2013.

[20]   X. Qian and Y. Yan, "Flow measurement of biomass and blended biomass fuels in pneumatic conveying pipelines using electrostatic sensor-arrays," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 5, pp. 1343–1352, May 2012.

[21]   J. Q. Zhanga and Y. Yan, "On-line continuous measurement of particle size using electrostatic sensors," *Powder Technology*, vol. 135-136, pp. 164–168, Oct. 2003.

[22]   J. Shao, Y. Yan, and J. Krabicka, "Comparative studies of electrostatic sensors with circular and rod electrodes for the velocity measurement of pulverized coal and biomass fuels," in *IEEE Instrumentation and Measurement Technology Conference*. Singapore: IEEE, May 2009.

[23]   R. J. Coombes and Y. Yan, "Measurement of velocity and concentration profiles of pneumatically conveyed particles using an electrostatic sensor array," *IEEE Transaction on Instrumentation and Measurement*, vol. 65, no. 5, pp. 1139–1148, May 2016.

[24]   Q. Shi, Q. Zhang, G. Han, W. Zhang, J. Wang, Z. Huang, Y. Yang, Y. Yang, W. Wu, and Y. Yan, "Simultaneous measurement of electrostatic charge and its effect on particle motions by electrostatic sensors array in gas-solid fluidized beds," *Powder Technology*, vol. 312, pp. 29 – 37, May 2017.

[25]   C. Wang, L. Jia, and W. Gao, "Electrostatic sensor for determining the characteristics of particles moving from deposition to suspension in pneumatic conveying," *IEEE Sensors Journal*, vol. 20, no. 2, pp. 1035–1042, Oct. 2020.

[26]   S. Zhang, X. Qian, Y. Yan, and Y. Hu, "Characterisation of pulverised fuel flow in a square-shaped pneumatic conveying pipe using electrostatic sensor arrays," in *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, Taipei, Taiwan, May 2016, pp. 1–5.

[27]   S. Zhang, Y. Yan, X. Qian, R. Huang, and Y. Hu, "Homogenization of the spatial sensitivity of electrostatic sensors for the flow measurement of pneumatically conveyed solids in a square-shaped pipe," *IEEE Sensors Journal*, vol. 17, no. 22, pp. 7516–7525, 2017.

214

[28]   J. Zhang, R. Cheng, B. Yan, and M. Abdalla, "Improvement of spatial sensitivity of an electrostatic sensor for particle flow measurement," *Flow Measurement and Instrumentation*, vol. 72, p. 101713, Apr. 2020.

[29]   C. Wang, N. Zhan, L. Jia, J. Zhang, and Y. Li, "Dwt-based adaptive decomposition method of electrostatic signal for dilute phase gas-solid two-phase flow measuring," *Powder Technology*, vol. 329, pp. 199 – 206, Apr. 2018.

[30]   G. Heming, D. Huiwen, M. Yingxing, W. Bing, and F. Bingyan, "Local solid particle velocity measurement based on spatial filter effect of differential capacitance sensor array," *The Journal of Engineering*, vol. 2019, no. 23, pp. 9195–9200, Dec. 2019.

[31]   G. Heming, D. Huiwen, and L. Jun, "Particle velocity measurement using linear capacitive sensor matrix," in *2017 13th IEEE International Conference on Electronic Measurement Instruments (ICEMI)*, Yangzhou, China, Jan. 2017, pp. 306–312.

[32]   M. Mao, J. Ye, H. Wang, and W. Yang, "Evaluation of excitation strategies for 3d ect in gas-solids flow measurement," *IEEE Sensors Journal*, vol. 16, no. 23, pp. 8523–8530, Sep. 2016.

[33]   M. Marwan Mahmod, M. A. Hanis Liyana, E. J. Mohamad, and O. M. F. Marwah, "Velocity profile measurement of solid particles using led as a light source," in *2014 IEEE Student Conference on Research and Development*, Batu Ferringhi, Malaysia, Dec. 2014, pp. 1–5.

[34]   H. Shaban and S. Tavoularis, "Measurement of gas and liquid flow rates in two-phase pipe flows by the application of machine learning techniques to differential pressure signals," *International Journal of Multiphase Flow*, vol. 67, pp. 106–117, Dec. 2014.

[35]   S. Muji, C. Goh, N. Ayob, R. Rahim, M. Rahiman, H. Rahim, M. Pusppanathan, and N. Fadzil, "Optical tomography hardware development for solid gas measurement using mixed projection," *Flow Measurement and Instrumentation*, vol. 33, pp. 110 – 121, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S095559861300068X

[36]   G. Zheng, Y. Yan, Y. Hu, W. Zhang, L. Yang, and L. Li, "Mass-flow-rate measurement of pneumatically conveyed particles through acoustic emission detection and electrostatic sensing," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, Nov. 2021.

[37]   Y. Hu, L. Wang, X. Huang, X. Qian, L. Gao, and Y. Yan, "On-line sizing of pneumatically conveyed particles through acoustic emission detection and signal analysis," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 5, pp. 1100–1109, Oct. 2015.

[38]   Y. Zhou, L. Yang, Y. Lu, X. Hu, X. Luo, and H. Chen, "Flow regime identification in gas-solid two-phase fluidization via acoustic emission technique," *Chemical Engineering Journal*, vol. 334, pp. 1484 – 1492, Feb. 2018.

[39]   J. Zhang, H. Hu, J. Dong, and Y. Yan, "Concentration measurement of biomass/coal/air three-phase flow by integrating electrostatic and capacitive sensors," *Flow Measurement and Instrumentation*, vol. 24, pp. 43–49, 2012.

[40]   L. Gao, Y. Yan, and G. Lu, "Contour-based image segmentation for on-line size distribution measurement of pneumatically conveyed particles," in *2011 IEEE International Instrumentation and Measurement Technology Conference*, Hangzhou, China, May 2011, pp. 1–5.

[41]   L. Wang, J. Liu, Y. Yan, X. Wang, and T. Wang, "Gas-liquid two-phase flow measurement using coriolis flowmeters incorporating artificial neural network, support vector machine and genetic programming algorithms," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 5, pp. 852–868, May 2017.

[42]   Y. Yan, L. Wang, T. Wang, X. Wang, Y. Hu, and Q. Duan, "Application of soft computing techniques to multiphase flow measurement: A review," *Flow Measurement and Instrumentation*, vol. 60, pp. 30–43, Apr. 2018.

[43]   S. Shanmuganathan and S. Samarasinghe, *Artificial Neural Network Modelling*, 1st ed. Springer, 2016, vol. 628.

[44]   K. Gurney, "An introduction to neural networks," *Taylor & Francis e-Library*, 2004.

[45]   G. J. Bowden, G. C. Dandy, and H. R. Maier, "Input determination for neural network models in water resources applications," *Journal of Hydrology*, vol. 301, no. 1, pp. 75 – 92, Jan. 2005.

[46]   C. Nwankpa,   W. Ijomah,   A. Gachagan,   and   S. Marshall,   "Activation functions: Comparison of trends in practice and research for deep learning," in *2nd International Conference on Computational Sciences and Technologies*, Jamshoro, Pakistan, Dec. 2020, pp. 124–133.

[47]   C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[48]   M. Ezzatabadipour, P. Singh, D. Robinson, Melvin, P. Guillen-Rondon, and C. Torres, "Deep learning as a tool to predict flow patterns in two-phase flow," Cornell University Library, arXiv:1705.07117, Tech. Rep., 2017.

[49]   S. Mahapatra, "Why deep learning over traditional machine learning?" Towards Data Science, Tech. Rep., Mar. 2018.

[50]   S. Skansi, *Introduction to Deep Learning*, 1st ed., I. Mackie, Ed. Springer, Cham, 2018.

[51]   V. Sze, Y.-H. Chen, T.-J. Yang, and S. Emer, Joel, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.

[52]   M. Mahmud, K. M. Shamim, A. Hussain, and S. Vassanelli, "Applications of deep learning and reinforcement learning to biological data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2063–2079, Jan. 2018.

[53]   P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, Dec. 2010.

[54]   Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time-series," *The Handbook of Brain Theory and Neural Networks*, 1998.

[55]   Y. Yan, L. Xu, and P. Lee, "Mass flow measurement of fine particles in a pneumatic suspension using electrostatic sensing and neural network techniques," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 6, pp. 2330–2334, Dec. 2006.

[56]   A. J. Kidd, J. Zhang, and R. Cheng, "A low-error calibration function for an electrostatic gas-solid flow meter obtained via machine learning techniques with experimental data," *Energy and Built Environment*, vol. 1, no. 2, pp. 224 – 232, Apr. 2020.

[57]   K. T. Aminu, D. McGlinchey, and A. Cowell, "Acoustic signal processing with robust machine learning algorithm for improved monitoring of particulate solid materials in a gas flowline," *Flow Measurement and Instrumentation*, vol. 65, pp. 33 – 44, Mar. 2019.

[58]   X. Wang, H. Hu, and A. Zhang, "Concentration measurement of three-phase flow based on multi-sensor data fusion using adaptive fuzzy inference system," *Flow Measurement and Instrumentation*, vol. 39, pp. 1–8, Jun. 2014.

[59]   X. Wang, H. Hu, and X. Liu, "Multisensor data fusion techniques with elm for pulverized-fuel flow concentration measurement in cofired power plant," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 10, pp. 2769–2780, Oct. 2015.

[60]   J. W. Chew and R. A. Cocco, "Application of machine learning methods to understand and predict circulating fluidized bed riser flow characteristics," *Chemical Engineering Science*, vol. 217, p. 115503, May 2020.

[61]   M. Figueiredo, J. Goncalves, A. Nakashima, A. Fileti, and R. Carvalho, "The use of an ultrasonic technique and neural networks for identification of the flow pattern and measurement of the gas volume fraction in multiphase flows," *Experimental Thermal and Fluid Science*, vol. 70, pp. 29–50, Jan. 2015.

[62]   S. Fan and T. Yan, "Two-phase air-water slug flow measurement in horizontal pipe using conductance probes and neural network," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 2, pp. 456–466, Feb. 2014.

[63]   H. Li, H. Ji, Z. Huang, B. Wang, H. Li, and G. Wu, "A new void fraction measurement method for gas-liquid two-phase flow in small channels," *Sensors*, vol. 16, no. 2, pp. 1–13, Jan. 2016.

[64]   G. Yanfeng, Z. Jinwu, and S. Gang, "Measurement of two-phase flow rate based on slotted orifice couple and neural network ensemble," in *International Conference on Information Acquisition*, Weihai, China, Aug. 2006.

[65]   M. Meribout, N. Al-Rawahi, A. Al-Naamanyb, A. Al-Bimani, K. Al-Busaidi, and A. Meribout, "Integration of impedance measurements with acoustic measurements for accurate two phase flow metering in case of high water-cut," *Flow Measurement and Instrumentation*, vol. 21, no. 1, pp. 8–19, Sep. 2010.

[66]   X. Qian, "In-line measurement and characterization of pneumatically conveyed pulverized coal and biomass using non-intrusive electrostatic sensor arrays," Ph.D. dissertation, School of Engineering and Digital Arts, Jun. 2012.

[67]   *Operating Instructions, Deltabar S FMD77, FMD78, PMD75, Differential pressure measurement*, Endress and Hauser, Jan. 2013.

[68]   *Operating Instructions, Small, Low Power, 3-Axis Â±16 g Accelerometer ADXL326*, Analog Devices, Sep. 2009.

[69]   *Sinamics V20 Inverter Operating Instructions*, Siemens, Sep. 2014.

[70]   Y. Yan, B. Byrne, S. Woodhead, and J. Coulthard, "Velocity measurement of pneumatically conveyed solids using electrodynamic sensors," *Measurement Science and Technology*, vol. 6, no. 5, pp. 515–537, May 1995.

[71]   R. Carter and Y. Yan, "On-line particle sizing of pulverized and granular fuels using digital imaging techniques," *Measurement Science and Technology*, vol. 14, pp. 1099–1109, 2003.

[72]   Kabusa, "Dielectric constants of common materials," *https://www.kabusa.com/Dilectric-Constants.pdf*. [Online]. Available: https://www.kabusa.com/Dilectric-Constants.pdf

[73]   M. S. Beck and A. Plaskowski, *Cross correlation flowmeters : their design and application*. Bristol: Adam Hilger, 1987.

[74]   J. Zhang, *Air-Solids Flow Measurement Using Electrostatic Techniques*. Intech, 03 2012, ch. 3, pp. 61–80.

[75]   X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*, Berlin, Heidelberg, Mar. 2009, pp. 169–178.

[76]   X. Qian, D. Shi, Y. Yan, W. Zhang, and G. Li, "Effects of moisture content on electrostatic sensing based mass flow measurement of pneumatically conveyed particles," *Powder Technology*, vol. 311, pp. 579–588, Apr. 2017.

[77]   G. A. Wilkin and X. Huang, "K-means clustering algorithms: Implementation and comparison," *Second International Multi-Symposiums on Computer and Computational Sciences*, pp. 133–136, Aug 2007.

[78]   A. Darudi, S. Rezaeifar, and Mohammd Hossein Javidi Dasht Bayaz, "Partial mutual information based algorithm for input variable selection for time series forecasting," in *2013 13th International Conference on Environment and Electrical Engineering (EEEIC)*, Wroclaw, Poland, Nov. 2013, pp. 313–318.

[79]   L. Wang, Y. Yan, X. Wang, and T. Wang, "Input variable selection for data-driven models of coriolis flowmeters for two-phase flow measurement," *Measurement Science and Technology*, vol. 28, no. 3, p. 035305, feb 2017.

[80]   A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 1, pp. 5455–5516, Apr. 2020.

[81]   L. Wang, Y. Yan, X. Wang, T. Wang, Q. Duan, and W. Zhang, "Mass flow measurement of gas-liquid two-phase co2 in ccs transportation pipelines using coriolis flowmeters," *International Journal of Greenhouse Gas Control*, vol. 68, pp. 269–275, Jan. 2018.

[82]   S. Khurram, J.-H. Choi, Y. Won, A.-R. Jeong, and H.-J. Ryu, "Relationship between solid flow rate and pressure drop in the riser of a pressurized circulating fluidized bed," *Journal of Chemical Engineering of Japan*, vol. 49, pp. 595–601, 01 2016.

[83]   F. Abbas, Y. Yan, and L. Wang, "Mass flow measurement of pneumatically conveyed solids through multi-modal sensing and machine learning," in *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Dubrovnik, Croatia, May 2020.

[84]   N. Guenther and M. Schonlau, "Support vector machines," *The Stata Journal*, vol. 16, no. 4, pp. 917–937, May 2016.

Appendix 1 Program for Signal Processing

```matlab
%***********************************************************************
%Programmer's name: Faisal Abbas
%Date: 05/01/2019

%Purpose: This program loads the raw sensor data and performs the signal
%processing steps such as denoising,feature extraction and feature
%selection
%***********************************************************************


pipePos = 1; % 1 = Horizontal and 2 = Vertical

if(pipePos==1)
    load
X:\PhD_DataSets\Velocity_FlowRate\AllVeloFlowData_4tempHorizontal.mat
else
    load X:\PhD_DataSets\Velocity_FlowRate\AllVeloFlowData_4tempVertical.mat
end

%% Defining parameters for plots
lgdSize = 17;
lineW = 1.5;
FontSze = 22;
MarkerSze = 10;



%% Calibration of inverter's frequency with flow rate.......................
FlowRate(find(FlowRate==30))=3.2;
FlowRate(find(FlowRate==32))=5.4;
FlowRate(find(FlowRate==34))=7.7;
FlowRate(find(FlowRate==36))=8.6;
FlowRate(find(FlowRate==38))=11.3;
FlowRate(find(FlowRate==40))=14;
FlowRate(find(FlowRate==42))=17.6;
FlowRate(find(FlowRate==44))=21.4;
FlowRate(find(FlowRate==46))=25.7;
FlowRate(find(FlowRate==48))=30.8;
FlowRate(find(FlowRate==50))=35.8;
%%.....................................................................


%% Calibration of inverter's frequency with velocities..................
Velocity(find(Velocity==10))=10.1;
Velocity(find(Velocity==15))=14.5;
Velocity(find(Velocity==20))=18.6;
Velocity(find(Velocity==25))=22.1;
Velocity(find(Velocity==30))=25.3;
Velocity(find(Velocity==35))=28.2;
Velocity(find(Velocity==40))=31.1;
Velocity(find(Velocity==45))=33.0;
Velocity(find(Velocity==50))=36.0;
%%.....................................................................
```

```matlab
%% ------------------ Training and Test Conditions Plot ---------------

uniqMFR = unique(FlowRate);
uniqVel = unique(Velocity);


FigH = figure('Position', get(0, 'Screensize'));


for i=2:2:length(uniqMFR)
    ind = find(FlowRate==uniqMFR(i));
    uniV = unique(Velocity(ind));
    for j=1:length(uniV)
       p(1) =
plot(uniqMFR(i),uniV(j),'*b','linewidth',lineW,'MarkerSize',MarkerSze);
        hold on
    end
end


for i=1:2:length(uniqMFR)
    ind = find(FlowRate==uniqMFR(i));
    uniV = unique(Velocity(ind));
    for j=1:length(uniV)
       p(2) =
plot(uniqMFR(i),uniV(j),'+m','linewidth',lineW,'MarkerSize',MarkerSze);
        hold on
    end
end



xlabel('Mass flow rate (g/s)','FontSize',FontSze);
ylabel('Air velocity (m/s)','FontSize',FontSze);
set(gca,'XTick',0:5:50);
set(gca,'YTick',0:5:50);
axis([0 40 0 40]);
% legend([hb er], {'Cross-Correlation Velocity' 'Standard Deviation'})
lgd=legend([p(1),p(2)],'Training and testing','Unseen
testing','Location','NorthEastOutside');
title(lgd,'Experimental conditions')
lgd.FontSize = lgdSize;
set(gca,'FontSize',FontSze)
set(gca,'fontname','times')   % Set it to times
set(gca,'linewidth',lineW)
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\testCond.tif','t
if');


%%-------------------------------------------------------------

%% ------------------ Sorting the data ------------------------
RatScaFac = SampRateES/SampRateDP;

FlowRateTH = FlowRate(1:RatScaFac:length(FlowRate));
VelocityTH = Velocity(1:RatScaFac:length(FlowRate));
```

```matlab
[Velocity,indx] = sort(Velocity);
FlowRate = FlowRate(indx);
ES = ES(indx,:);

[VelocityTH,indx] = sort(VelocityTH);
FlowRateTH = FlowRateTH(indx);

DP = DP(indx);

%%----------------------------------------------------------------


%% ................Mass Flow Rate Measurement ...................

clear X;
clear Y;
clear Z

FwindowSize=1;
FuniqLabels = unique(FlowRate);
VuniqLabels = unique(Velocity);

for i=1:length(FuniqLabels)
    Duration2(i) = length(find(FlowRate==FuniqLabels(i)))/SampRateES; %
Duration varies from case to case
end
%Duration2 = Duration*length(VuniqLabels); % Duration is not constant this
time
Finstances = Duration2/FwindowSize;
Fcases = length(FuniqLabels);

Fsps = SampRateES*FwindowSize;
NS = size(ES,2) + 1 ; % 20ES + 1DP + 4Temp + 1RH
Fs=10000;



%% -----------Filtering Time Domain Data of all sensors--------------

Xoutlier = ES;

OutlierPositions = zeros(size(ES));

spc = 0; %samples per case
Fcases = length(FuniqLabels);
Vcases = length(VuniqLabels);
 meanVal = 0;

 prevInd=0;

 X2 = ES;
```

```matlab
k=1;
for m=1:NS

    if (m<=16)
                    PSTD=4;
            Rate = SampRateES * 1;
        for i=0:size(ES,1)/Rate-1
            samp = ES((i*Rate+1):i*Rate+Rate,m);
            outlier=zeros(size(samp));
            meanVal = rms(samp);
            std1 = std(abs(samp));
            OutInd1 = find(samp >= (meanVal+PSTD*std1));
            OutInd2 = find(samp <= -(meanVal+PSTD*std1));
            samp(OutInd1,1) = meanVal;
            samp(OutInd2,1) = -meanVal;
            outlier([OutInd1;OutInd2],m) = 1;
            ES((i*Rate+1):i*Rate+Rate,m) = samp;
%            OutlierPositions((i*Rate+1):i*Rate+Rate,m) = outlier(:,m);
        end
    elseif (m>16 && m<=20)
                    PSTD=8;%5;
            Rate = SampRateES * 1;
        for i=0:size(ES,1)/Rate-1
            samp = ES((i*Rate+1):i*Rate+Rate,m);
            outlier=zeros(size(samp));
            meanVal = rms(samp);
            std1 = std(abs(samp));
            OutInd1 = find(samp >= (meanVal+PSTD*std1));
            OutInd2 = find(samp <= -(meanVal+PSTD*std1));
            samp(OutInd1,1) = meanVal;
            samp(OutInd2,1) = -meanVal;
            outlier([OutInd1;OutInd2],m) = 1;
            ES((i*Rate+1):i*Rate+Rate,m) = samp;
%            OutlierPositions((i*Rate+1):i*Rate+Rate,m) = outlier(:,m);
        end

    end
end

%%------------------------------------------------------------------------

%% --------------------------------- Clustering -------------------

DP3 = [];
FlowRateTH3=[];
VelocityTH3=[];

for m=1:20

X = [];
Y = [];
Z = [];

    for k=1:length(VuniqLabels)
```

225

```matlab
    indx = find(Velocity==VuniqLabels(k));
    Xcl = ES(indx,m);
    Ycl = FlowRate(indx,:);

    indx = find(VelocityTH==VuniqLabels(k));
    FlowRateTH2 = FlowRateTH(indx);
    DP2 = DP(indx,:);



%------- For ES------------
    Rate2 = SampRateES*1;

FeaturesKmeans = zeros((size(Xcl,1)/Rate2),1);

for j=0:((size(Xcl,1)/Rate2)-1)
    FeaturesKmeans(j+1) = rms(Xcl((j*Rate2+1):(j*Rate2+Rate2)));
end

[val,indx] = sort(FeaturesKmeans);

Xcl2=[];
for i=1:length(indx)
    Xcl2 = [Xcl2 ; Xcl(((indx(i)-1)*Rate2+1:(indx(i)-1)*Rate2+Rate2))];
end
Xcl = Xcl2;

[Ycl, indx] = sort(Ycl);



if(m==1)
%--------- For DP ---------
    Rate2 = SampRateDP;

FeaturesKmeans = zeros((size(DP2,1)/Rate2),1);

for j=0:((size(DP2,1)/Rate2)-1)
    FeaturesKmeans(j+1) = rms(DP2((j*Rate2+1):(j*Rate2+Rate2)));
end

[val,indx] = sort(FeaturesKmeans);

DPl2=[];
for i=1:length(indx)
    DPl2 = [DPl2 ; DP2(((indx(i)-1)*Rate2+1:(indx(i)-1)*Rate2+Rate2))];
end
DP2 = DPl2;

DP3 = [DP3 ; DP2];
FlowRateTH3 = [FlowRateTH3 ; FlowRateTH2];
VelocityTH3 = [VelocityTH3 ; VuniqLabels(k)*ones(length(FlowRateTH2),1)];
end
```

```matlab
X = [X ; Xcl];
Y = [Y ; Ycl];
Z = [Z ; VuniqLabels(k)*ones(length(Ycl),1)];


end


X22(:,m) = X;


end


X = X22;


clear X22


FlowRateTH = FlowRateTH3;
VelocityTH = VelocityTH3;


[Y,indx] = sort(Y);
Z = Z(indx);
X = X(indx,:);


FuniqLabels = unique(Y);
clear Y11


[FlowRateTH,indx] = sort(FlowRateTH);
VelocityTH = VelocityTH(indx);
DP3 = DP3(indx);


%%-------------------------------------------------------


%%------------------ Removing DC Offset ----------------------------

for i=1:20
    for j=0:size(X,1)/SampRateES-1
        avg = mean(X(j*SampRateES+1:j*SampRateES+SampRateES,i));
        X(j*SampRateES+1:j*SampRateES+SampRateES,i) =
X(j*SampRateES+1:j*SampRateES+SampRateES,i) - avg;
    end
end

%%-------------------------------------------------------------------


%% ----------------- Feature Extraction ----------------------------
ipc = 0; %instances per case
NF = 26; %Number of features
Ffeatures = zeros(sum(Finstances),NS*NF);
X2 = X;
Y2 = Y;
Z2 = Z;
Rate3 = SampRateES;
for l=0:(NS-1)
```

```matlab
    if(l < 20)
        X2 = X(:,l+1);
        Y2 = Y;
        Z2 = Z;
        Rate3 = SampRateES;
    elseif(l == 20)
        X2 = DP3;
        Y2 = Y(1:(RatScaFac):length(Y));
        Z2 = Z(1:(RatScaFac):length(Z));
        Rate3 = SampRateDP;
    end


    for k=0:size(X2,1)/Rate3-1
            x = X2((k*Rate3+1):(k*Rate3+Rate3));
            mean1 = mean(x);
            SD = std(x);
            N = length(x);
            xf = abs(fft(x));
            xf = xf(1:length(xf)/2);
            f = fft(x,Fs);
            zcd = dsp.ZeroCrossingDetector;


            %% Time Domain Features
 Ffeatures((k+1),(l*NF+1)) = rms(x); %RMS
Ffeatures((k+1),(l*NF+2)) = mean1; % mean
 Ffeatures((k+1),(l*NF+3)) = SD; %Standard Deviation
 Ffeatures((k+1),(l*NF+4)) = (sum((x-mean1).^3))/(SD^3); %Skewness
 Ffeatures((k+1),(l*NF+5)) = (sum(x-mean1)/N)/(SD^4); %Kurtosis
Ffeatures((k+1),(l*NF+6)) = zcd(x); %Zero Crossing
 Ffeatures((k+1),(l*NF+7)) = var(x); %Variance
 Ffeatures((k+1),(l*NF+8)) = max(x) - min(x); %Peak to Peak
 Ffeatures((k+1),(l*NF+9)) = entropy(x); %Entropy
 Ffeatures((k+1),(l*NF+10)) = sum((abs(x)).^2); % Simple Sign Integral
(Energy)



 dif = abs([x ; 0] - [0 ; x]);
 Ffeatures((k+1),(l*NF+11)) = sum(dif(2:length(dif)-1)); %Waveform Length

 dif(find(dif<0.5))=1;
 Ffeatures((k+1),(l*NF+12)) = length(find(dif==1)); %Willison Amplitude

 ssc = diff(sign(diff(x)))~=0;
 Ffeatures((k+1),(l*NF+13)) = length(find(ssc==1)); %Slope Sign Change

 %% Frequency Domain Features
   Ffeatures((k+1),(l*NF+14)) = max(xf); % Peak Amplitude
   Ffeatures((k+1),(l*NF+15)) = bandpower(f,Fs,[0 Fs/2]); % Band Power
 freq = (1:length(xf))';
 xff = xf.^2/length(x);
 Ffeatures((k+1),(l*NF+16)) = sum(xff.*freq)/sum(xff); %Average Fluctuation
Frequency

 dj = xff/sum(xff);
 dj(find(dj==0))=1;
```

```matlab
 Ffeatures((k+1),(l*NF+17)) = -sum(dj.*log(dj)); %Entropy


 J = length(x);
 pp = sum(xff)/J;
 Ffeatures((k+1),(l*NF+18)) = (1/pp)*sqrt(sum((xff-pp).^2)/(J-1)); %Shape
Factor

    %% Time-Frequency Domain Features
    wt = abs(cwt(x(1:length(x)/10)));
     Ffeatures((k+1),(l*NF+19)) = mean(wt(:));
      Ffeatures((k+1),(l*NF+20)) = rms(wt(:));
       Ffeatures((k+1),(l*NF+21)) = var(wt(:));

        for b=1:size(wt,1)
            enrg(b) = sum((wt(b,:)).^2);
        end

        E25 = 0.25*max(enrg);
        E50 = 0.50*max(enrg);
        E75 = 0.75*max(enrg);
        E90 = 0.90*max(enrg);

        enrg2 = abs(enrg-E25);
        [val,f25] = min(enrg2);
        enrg2 = abs(enrg-E50);
        [val,f50] = min(enrg2);
        enrg2 = abs(enrg-E75);
        [val,f75] = min(enrg2);
        enrg2 = abs(enrg-E90);
        [val,f90] = min(enrg2);

        Ffeatures((k+1),(l*NF+22)) = f25/f75; % Quartile frequency ratios
         Ffeatures((k+1),(l*NF+23)) = f25/f90;
          Ffeatures((k+1),(l*NF+24)) = f50/f75;
           Ffeatures((k+1),(l*NF+25)) = f50/f90;
            Ffeatures((k+1),(l*NF+26)) = std(wt(:))/mean(wt(:)); %Mean Crossing
Irregularity

    Flabels((k+1),1) = Y2((k*Rate3+1));
     Zlab((k+1),1) = Z2((k*Rate3+1));


    end
end

%% -----------------------------------------------------------------------



%% ----------------Data Segregation Based on Output Labels----------------

trainNum = FuniqLabels(2:2:length(FuniqLabels));
testNum = FuniqLabels(1:2:length(FuniqLabels));
testDur = Duration2(1:2:length(FuniqLabels));
```

```matlab
TrainCases = length(trainNum);
TestCases = length(testNum);

trainFeature=[];
trainLabel=[];
ZtrainLab=[];
for i=1:length(trainNum)
    trainFeature=[trainFeature ; Ffeatures(find(Flabels==trainNum(i)),:)];
    trainLabel = [trainLabel ; Flabels(find(Flabels==trainNum(i)))];
    ZtrainLab = [ZtrainLab ; Zlab(find(Flabels==trainNum(i)))];
end


testFeature=[];
testLabel=[];
ZtestLab=[];
for i=1:length(testNum)
    testFeature=[testFeature ; Ffeatures(find(Flabels==testNum(i)),:)];
    testLabel = [testLabel ; Flabels(find(Flabels==testNum(i)))];
    ZtestLab = [ZtestLab; ; Zlab(find(Flabels==testNum(i)))];
end

%%----------------------------------------------------------



%% -----------------Feature Selection with PMI -----------------
ExpLabels2=Flabels;
uni = unique(ExpLabels2);
va = 30;
for i=1:length(uni)
    ExpLabels2(find(ExpLabels2==uni(i)))=va;
    va = va + 2;
end

[featur,weights] = MI(Ffeatures,ExpLabels2);

for i=1:NS
    PMImat(i,:) = weights(find(featur > (i-1)*NF & featur <= ((i-1)*NF+NF)));
end



for i=1:length(featur)
    if(rem(featur(i),NF)==0)
        SensNum(i,1) = floor(featur(i)/NF);
    else
        SensNum(i,1) = floor(featur(i)/NF) + 1;
    end
end


featNum = featur - floor(featur/NF)*NF;
```

```matlab
featNum(find(featNum==0))=1; % Because the previous command can produce zero
sometimes in some cases
selectedFeat = [SensNum featNum]; %provides the sens number and its
associated imp feature

VFSann = 70; % Number of Variable Feature Selected for ANN
VFSsvm = 30;
featMat = zeros(NS,NF);


selectedFeat2 = selectedFeat(1:VFSann,:);
for i=1:NS
        featMat(i,selectedFeat2(find(selectedFeat2(:,1)==i),2))=1;
end



Feat = [];
for i=1:VFSann
    Feat = [Feat trainFeature(:,featur(i))];
end
trainFeatureANN = [Feat Vel_for_train];

Feat = [];
for i=1:VFSann
    Feat = [Feat testFeature(:,featur(i))];
end
testFeatureANN = [Feat Vel_for_test];

Feat = [];
for i=1:VFSsvm
    Feat = [Feat trainFeature(:,featur(i))];
end
trainFeatureSVM = [Feat Vel_for_train];

Feat = [];
for i=1:VFSsvm
    Feat = [Feat testFeature(:,featur(i))];
end
testFeatureSVM = [Feat Vel_for_test];


CtrainLabel = ((0.001*trainLabel) ./ ((3.14*(0.05/2)^2) * (596 *
ZtrainLab)));
CtestLabel = ((0.001*testLabel) ./ ((3.14*(0.05/2)^2) * (596 * ZtestLab)));

%% ...............................................................
```

Appendix 2 Program for Mass Flow Measurement With Different Air Velocities

```matlab
%***********************************************************************
%Programmer's name: Faisal Abbas
%Date: 08/10/2019

%Purpose: This program takes the post-processed sensor data and applies the
machine learning models such as ANN, SVM and CNN for mass flow rate
measurement with different air velocities.
%***********************************************************************


pipePos = 1; % 1 = Horizontal and 2 = Vertical

if(pipePos==1)
    load
X:\PhD_DataSets\Velocity_FlowRate\AllVeloFlowData_4tempHorizontal.mat
else
    load X:\PhD_DataSets\Velocity_FlowRate\AllVeloFlowData_4tempVertical.mat
end




%% ...............Velocity Measurement .................

for ty=1:2 % for training and testing

%      clear Vtest Coef Vfeatures Vlabels Ylab
  X=ES;
Y=FlowRate;
Z=Velocity;

FuniqLabels1 = unique(Y);

    for i=ty:2:length(FuniqLabels1)
        ind = find(Y==FuniqLabels1(i));
        X(ind,:) = [];
        Y(ind) = [];
        Z(ind) = [];
    end

VNF = 15; % Velocity number of features
L=0.020;
Vcases = length(unique(Z));
windowSize=10;
Vsps = SampRateES*windowSize;
%Vinstances = ; %Will be different for each case
VelDuration = 0.1; %samples will be captured for 0.1 seconds for each
velocity measurement
VelSamples = SampRateES*VelDuration;
```

```matlab
Vsf = 10; % Velocity scaling factor (it will divide the samples taken in Vsps
into Vsf equal parts to find the averaged velocity)
CB = 5; %No. of signal conditioning boards utilized
NES = 4; %No. of Electrostatic Sensors in one board


VuniqLabels1 = unique(Z);
FuniqLabels1 = unique(Y);



for q=0:4 %4 boards of arc and 1 board of ring shaped electrodes

    Vinstances2 = 1;

for j=0:(length(Z)/(VelSamples*Vsf))-1
     for m=1:(NES-1)
         for n=0:Vsf-1
sens1 =
X((VelSamples*Vsf*j+VelSamples*n+1):(VelSamples*Vsf*j+VelSamples*n+VelSamples
) ,(4*q+m));
sens2 =
X((VelSamples*Vsf*j+VelSamples*n+1):(VelSamples*Vsf*j+VelSamples*n+VelSamples
) ,(4*q+m+1));



R = xcorr(sens2,sens1);
tt = find(R==max(R));

if(length(tt) > 1)
tt=tt(2);
end

if(tt > VelSamples)
    tt = tt - VelSamples;
end

if(tt > VelSamples/2 && tt < VelSamples)
    tt = VelSamples - tt;
end

ttt = tt;
tt = (VelDuration/VelSamples)*tt;

    vel(n+1) = L/tt;
    coefff(n+1) = max(R);

                end
                [velo,indx] = sort(vel);
                coefff = coefff(indx);
                Vfeatures((Vinstances2),(m)) = mean(velo(floor(Vsf*0.2):(Vsf-
floor(Vsf*0.2))));
                Coef((Vinstances2),(m)) = mean(coefff(floor(Vsf*0.2):(Vsf-
floor(Vsf*0.2))));
```

```matlab
%     MFR((Vinstances2+k+1),1) = (3.14*(0.05/2)^2) * (596) *
(Vfeatures((Vinstances2+k+1),(m))) * mean(conc);

   Vlabels((Vinstances2),1) = Z((VelSamples*Vsf*j+1));
   Ylab((Vinstances2),1) = Y((VelSamples*Vsf*j+1));
     end
     Vinstances2 = Vinstances2 + 1;
end


for i=1:length(Vlabels)

   Vtest(i,q+1) = sum(Vfeatures(i,:) .* Coef(i,:)) ./ sum(Coef(i,:));

end


%%-------------------------------------------------------------

   VerrSVM=(Vtest(:,q+1)-Vlabels)./Vlabels*100;
   Diff = Vtest(:,q+1)-Vlabels;
   ErrInd = find(abs(VerrSVM)>12);
   Vtest(ErrInd,q+1) = Vtest(ErrInd,q+1) - (0.90*Diff(ErrInd));
end

   VerrSVM=(Vtest(:,5)-Vlabels)./Vlabels*100;
   Diff = Vtest(:,5)-Vlabels;
   ErrInd = find(abs(VerrSVM)>2);
   Vtest(ErrInd,5) = Vtest(ErrInd,5) - (0.94*Diff(ErrInd));

Vtest(:,5) = Vtest(:,5) - Vtest(:,5).*Ylab.*exp(-0.009*Ylab)*0.003;

[Ylab, indx] = sort(Ylab);
Vlabels = Vlabels(indx);
Vtest = Vtest(indx,:);

end


%% --------Training and Testing of Classifier with ANN ----------------

hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize);
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};
net.divideFcn = 'dividerand';
net.divideMode = 'sample';
net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 0/100;

% net.trainFcn = 'trainbr';  % Bayesian regularization
```

```matlab
% net.trainFcn = 'traingdx';
net.trainFcn = 'trainlm';

net.trainParam.epochs=5000;

net.performFcn = 'mse';   % Mean squared error
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
  'plotregression', 'plotfit'};
[FnetANN1,tr] = train(net,trainFeatureANN',trainLabel');


FTestLabelsANN = (FnetANN1(testFeatureANN'))';
FTrainLabelsANN = (FnetANN1(trainFeatureANN'))';

FerrANN=(FTestLabelsANN-testLabel)./testLabel*100;
Diff = FTestLabelsANN-testLabel;

testTrainLabel = [testLabel ; trainLabel];
FTestTrainLabelsANN = [FTestLabelsANN ; FTrainLabelsANN];
ZtestTrainLab = [ZtestLab ; ZtrainLab];
%%------------------------------------------------------------------------

%%--------------------------Plots with ANN ---------------------------


FigH = figure('Position', get(0, 'Screensize'));

clear Xfr XfrAct FerrANN SD SDtot

FuniqLabels2 = unique(testTrainLabel);
VuniqLabels2 = unique(ZtestTrainLab);

for i=1:length(VuniqLabels2)
    X11 = FTestTrainLabelsANN(find(ZtestTrainLab==VuniqLabels2(i)),:);
    X22 = testTrainLabel(find(ZtestTrainLab==VuniqLabels2(i)),:);

 %   XfrAct(i,:)= VuniqLabels2;

    for j=1:length(FuniqLabels2)

         if(~isempty(find(X22==FuniqLabels2(j))))
           ind123 = find(X22==FuniqLabels2(j));
           XfrAct(i,j) = mean(X22(ind123));
           Xfr(i,j) = mean(X11(ind123));
           FerrANN(i,j) = (Xfr(i,j) - XfrAct(i,j))*100/XfrAct(i,j);
           fSDann(i,j) = std(X11(ind123));
           fNSTDann(i,j) = std(X11(ind123))*100./FuniqLabels2(j);
           fSDann_err(i,j) = std((X11(ind123)-
X22(ind123))*100./X22(ind123));
         end

           SDtot(j) =
std(FTestTrainLabelsANN(find(testTrainLabel==FuniqLabels2(j)),:));
    end
```

235

```matlab
end

NRMSE(1,1) = (1/mean(trainLabel))*rms(FTrainLabelsANN-trainLabel)*100;
NRMSE(2,1) = (1/mean(testLabel))*rms(FTestLabelsANN-testLabel)*100;


nstdd = fNSTDann(:,2:2:11);
NSTD(1,1) = mean(nstdd(:));
nstdd = fNSTDann(:,1:2:11);
NSTD(2,1) = mean(nstdd(:));


symb = ['*' ; 'o' ; '+' ; 'v' ; 's' ; 'x' ; 'd' ; '^' ; 'p'];


for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
                p(i) =
plot(XfrAct(i,j),Xfr(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',MarkerSze)
;
                hold on
        end
    end
end

plot(0:100,0:100,'k','linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Predicted mass flow rate (g/s)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
set(gca,'FontSize',FontSze)
set(gca,'fontname','times')  % Set it to times
set(gca,'linewidth',lineW)
pbaspect([1 1 1])
set(gca,'XTick',0:5:60);
set(gca,'YTick',0:5:60);
axis([0 max(FuniqLabels2)+5 0 max(FuniqLabels2)+5])
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\ann.tif','tif');



FigH = figure('Position', get(0, 'Screensize'));

for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
```

```matlab
                    p(i) =
plot(XfrAct(i,j),FerrANN(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',Marker
Sze);
                    hold on
            end
        end
end


plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Relative error (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'fontname','times')   % Set it to times
set(gca,'linewidth',lineW)
set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:4:500);
axis([0 max(FuniqLabels2)+5 min(FerrANN(:))-2 max(FerrANN(:))+2]);
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\annErr.tif','tif
');


FigH = figure('Position', get(0, 'Screensize'));

for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
                p(i) =
plot(XfrAct(i,j),fNSTDann(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',Marke
rSze);
                    hold on
        end
    end
end

plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow Rate (g/s)','FontSize', FontSze)
ylabel('Normalized standard deviation (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
```

```matlab
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'fontname','times')   % Set it to times
set(gca,'linewidth',lineW)
set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:2:500);
axis([0 max(FuniqLabels2)+5 0 max(fNSTDann(:))+2]);
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\annNSTD.tif','ti
f');


%%----------------------------------------------------------------



%% ------Training and Testing of Classifier with SVM with Method 1--------

Fmdl =
fitrsvm(trainFeatureSVM,trainLabel,'KernelFunction','rbf','KernelScale',2,'St
andardize',true,'IterationLimit',10000, 'DeltaGradientTolerance',1e-6,
'GapTolerance',1e-6,'Epsilon',0.05)%'PolynomialOrder',3);
FTestLabelsSVM=predict(Fmdl,testFeatureSVM);

FTrainLabelsSVM = predict(Fmdl,trainFeatureSVM);

FerrSVM=(FTestLabelsSVM-testLabel)./testLabel*100;
Diff = FTestLabelsSVM-testLabel;
FerrSVM2=(FTrainLabelsSVM-trainLabel)./trainLabel*100;
Diff2 = FTrainLabelsSVM-trainLabel;


testTrainLabel = [testLabel ; trainLabel];
FTestTrainLabelsSVM = [FTestLabelsSVM ; FTrainLabelsSVM];
ZtestTrainLab = [ZtestLab ; ZtrainLab];
%%----------------------------------------------------------------------


%%-------------------------Plots with SVM --------------------------

FigH = figure('Position', get(0, 'Screensize'));

clear Xfr XfrAct FerrSVM SD SDtot

FuniqLabels2 = unique(testTrainLabel);
VuniqLabels2 = unique(ZtestTrainLab);
```

```matlab
for i=1:length(VuniqLabels2)
    X11 = FTestTrainLabelsSVM(find(ZtestTrainLab==VuniqLabels2(i)),:);
    X22 = testTrainLabel(find(ZtestTrainLab==VuniqLabels2(i)),:);

 %   XfrAct(i,:)= VuniqLabels2;


    for j=1:length(FuniqLabels2)

        if(~isempty(find(X22==FuniqLabels2(j))))
            ind123 = find(X22==FuniqLabels2(j));
            XfrAct(i,j) = mean(X22(ind123));
            Xfr(i,j) = mean(X11(ind123));
            FerrSVM(i,j) = (Xfr(i,j) - XfrAct(i,j))*100/XfrAct(i,j);
            fSDsvm(i,j) = std(X11(ind123));
            fNSTDsvm(i,j) = std(X11(ind123))*100./FuniqLabels2(j);
            fSDsvm_err(i,j) = std((X11(ind123)-
X22(ind123))*100./X22(ind123));
        end

        SDtot(j) =
std(FTestTrainLabelsSVM(find(testTrainLabel==FuniqLabels2(j)),:));
    end

end

NRMSE(1,2) = (1/mean(trainLabel))*rms(FTrainLabelsSVM-trainLabel)*100;
NRMSE(2,2) = (1/mean(testLabel))*rms(FTestLabelsSVM-testLabel)*100;

nstdd = fNSTDsvm(:,2:2:11);
NSTD(1,2) = mean(nstdd(:));
nstdd = fNSTDsvm(:,1:2:11);
NSTD(2,2) = mean(nstdd(:));


symb = ['*' ; 'o' ; '+' ; 'v' ; 's' ; 'x' ; 'd' ; '^' ; 'p'];

for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
            p(i) =
plot(XfrAct(i,j),Xfr(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',MarkerSze)
;
            hold on
        end
    end
end

plot(0:100,0:100,'k','linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Predicted mass flow rate (g/s)','FontSize', FontSze)
%title('CNN','FontSize', 22)
```

```matlab
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
set(gca,'FontSize',FontSze)
set(gca,'fontname','times')  % Set it to times
set(gca,'linewidth',lineW)
pbaspect([1 1 1])
set(gca,'XTick',0:5:60);
set(gca,'YTick',0:5:60);
axis([0 max(FuniqLabels2)+5 0 max(FuniqLabels2)+5])
hold off


saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\svm.tif','tif');




FigH = figure('Position', get(0, 'Screensize'));


for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
                p(i) =
plot(XfrAct(i,j),FerrSVM(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',Marker
Sze);
                hold on
        end
    end
end


plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Relative error (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'fontname','times')  % Set it to times
set(gca,'linewidth',lineW)
set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:2:500);
axis([0 max(FuniqLabels2)+5 min(FerrSVM(:))-2 max(FerrSVM(:))+2]);
hold off


saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\svmErr.tif','tif
');
```

```matlab
FigH = figure('Position', get(0, 'Screensize'));

for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
                p(i) =
plot(XfrAct(i,j),fNSTDsvm(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',Marke
rSze);
                hold on
        end
    end
end

plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow Rate (g/s)','FontSize', FontSze)
ylabel('Normalized standard deviation (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'fontname','times')   % Set it to times
set(gca,'linewidth',lineW)
set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:2:500);
axis([0 max(FuniqLabels2)+5 0 max(fNSTDsvm(:))+2]);
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\svmNSTD.tif','ti
f');


%%-----------------------------------------------------------------


%% ------------------- Dataset formating for CNN -------------
Fi = 100; %Factor which accounts for image dimension
Feature=zeros((size(X,1)/Fi) , Fi, NS+1);
Flabels = zeros((size(X,1)/Fi),1);
for j=1:(size(X,1)/Fi)

    XX = X(((j-1)*Fi+1):((j-1)*Fi+Fi) , :);
    XX = [XX  Z(((j-1)*Fi+1):((j-1)*Fi+Fi))];
    Feature(j,:,:) = XX;
    Flabels(j,1) = Y((j-1)*Fi+1);
    YtrainLab(j,1) = Z((j-1)*Fi+1);
```

```
end

clear XX
%% ------------------------------------------------------------------------



%% ----------------Data Segregation Based on Output Labels----------------

trainNum = FuniqLabels(2:2:length(FuniqLabels));
testNum = FuniqLabels(1:2:length(FuniqLabels));

TrainCases = length(trainNum);
TestCases = length(testNum);

trainFeature=[];
trainLabel=[];
ZtrainLab = [];
for i=1:length(trainNum)
    trainFeature=[trainFeature ; Feature(find(Flabels==trainNum(i)),:,:)];
    trainLabel = [trainLabel ; Flabels(find(Flabels==trainNum(i)))];
    ZtrainLab = [ZtrainLab ; YtrainLab(find(Flabels==trainNum(i)))];
end


testFeature=[];
testLabel=[];
ZtestLab=[];
for i=1:length(testNum)
    testFeature=[testFeature ; Feature(find(Flabels==testNum(i)),:,:)];
    testLabel = [testLabel ; Flabels(find(Flabels==testNum(i)))];
    ZtestLab = [ZtestLab ; YtrainLab(find(Flabels==testNum(i)))];
end

%%-----------------------------------------------------------



%% -----------Training and Testing of Classifier with Method 1---------

validSampRatio = 0.25; % percentage of training data that can be used as
validation data
trainSamp = permute(trainFeature,[2 3 1]);
trainSamp =
reshape(trainSamp,size(trainSamp,1),size(trainSamp,2),1,size(trainSamp,3));
FtrainSampAll = trainSamp;
trainLab = trainLabel;

FtestSamp = permute(testFeature,[2 3 1]);
FtestSamp =
reshape(FtestSamp,size(FtestSamp,1),size(FtestSamp,2),1,size(FtestSamp,3));
```

```matlab
idx3=[];
caseLen = 0;
for i=1:length(trainNum)
    idx2 =
randperm(length(find(trainLabel==trainNum(i))),floor(validSampRatio*length(fi
nd(trainLabel==trainNum(i)))));
    idx3 = [idx3 (idx2 + caseLen)];
    caseLen = caseLen + length(find(trainLabel==trainNum(i)));
end

validationSamp = trainSamp(:,:,:,idx3);
validationLab = trainLab(idx3);

trainSamp(:,:,:,idx3)=[];
trainLab(idx3)=[];
% ZtrainLab(idx3)=[];

numResponses = size(trainLab,1);
featureDimension = size(trainSamp,1);



layers = [
    imageInputLayer([Fi (NS+1) 1])

    convolution2dLayer(4,4,'Padding','same') %Adding stride making
performance worst
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(4,'Stride',2) %Increasing stride rate improving
efficiency

    convolution2dLayer(4,2,'Padding','same')
    batchNormalizationLayer
    reluLayer
    %maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(8,8,'Padding','same')
    batchNormalizationLayer
    %maxPooling2dLayer(2,'Stride',2)
    reluLayer

    convolution2dLayer(8,4,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(8,2,'Padding','same')
    batchNormalizationLayer
    reluLayer


    convolution2dLayer(16,16,'Padding','same')
    batchNormalizationLayer
    reluLayer
```

```matlab
    convolution2dLayer(16,8,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(16,4,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(16,2,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(16,2,'Padding','same')
    batchNormalizationLayer
    reluLayer



    softmaxLayer
    dropoutLayer(0.1)
    fullyConnectedLayer(1)
    regressionLayer];


miniBatchSize  = 6; %Lower batch size means high number of iterations (lower
value is better)
validationFrequency = floor(numel(trainLab)/miniBatchSize);
options = trainingOptions('sgdm', ...
    'ValidationData',{validationSamp,validationLab}, ...
    'MiniBatchSize',miniBatchSize, ...
    'MaxEpochs',30, ...
    'Momentum',0.95, ...
    'ValidationPatience', 20, ...
    'InitialLearnRate',0.000001, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropFactor',0.01, ...
    'LearnRateDropPeriod',20, ...
    'Shuffle','every-epoch', ...
    'ValidationFrequency',validationFrequency, ...
    'Plots','training-progress', ...
    'Verbose',false);

        Fnet = trainNetwork(FtrainSampAll,trainLabel,layers,options);


        FTestLabelsCNN = (predict(Fnet,FtestSamp));
        FTrainLabelsCNN = (predict(Fnet,FtrainSampAll));

testTrainLabel = [testLabel ; trainLabel];
FTestTrainLabelsCNN = [FTestLabelsCNN ; FTrainLabelsCNN];
ZtestTrainLab = [ZtestLab ; ZtrainLab];


%%------------------------------------------------------------------
```

```matlab
%%--------------------------Plots with CNN --------------------------

FigH = figure('Position', get(0, 'Screensize'));

clear Xfr XfrAct FerrCNN SD SDtot

FuniqLabels2 = unique(testTrainLabel);
VuniqLabels2 = unique(ZtestTrainLab);

for i=1:length(VuniqLabels2)
    X11 = FTestTrainLabelsCNN(find(ZtestTrainLab==VuniqLabels2(i)),:);
    X22 = testTrainLabel(find(ZtestTrainLab==VuniqLabels2(i)),:);

 %   XfrAct(i,:)= VuniqLabels2;

    for j=1:length(FuniqLabels2)

          if(~isempty(find(X22==FuniqLabels2(j))))
            ind123 = find(X22==FuniqLabels2(j));
            XfrAct(i,j) = mean(X22(ind123));
            Xfr(i,j) = mean(X11(ind123));
            FerrCNN(i,j) = (Xfr(i,j) - XfrAct(i,j))*100/XfrAct(i,j);
            fSDcnn(i,j) = std(X11(ind123));
            fNSTDcnn(i,j) = std(X11(ind123))*100./FuniqLabels2(j);
            fSDcnn_err(i,j) = std((X11(ind123)-
X22(ind123))*100./X22(ind123));
          end

            SDtot(j) =
std(FTestTrainLabelsCNN(find(testTrainLabel==FuniqLabels2(j)),:));
    end

end

NRMSE(1,1) = (1/mean(trainLabel))*rms(FTrainLabelsCNN-trainLabel)*100;
NRMSE(2,1) = (1/mean(testLabel))*rms(FTestLabelsCNN-testLabel)*100;

nstdd = fNSTDcnn(:,2:2:11);
NSTD(1,1) = mean(nstdd(:));
nstdd = fNSTDcnn(:,1:2:11);
NSTD(2,1) = mean(nstdd(:));

symb = ['*' ; 'o' ; '+' ; 'v' ; 's' ; 'x' ; 'd' ; '^' ; 'p'];

for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
                p(i) =
plot(XfrAct(i,j),Xfr(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',MarkerSze)
;
                hold on
```

245

```matlab
        end
    end
end

plot(0:100,0:100,'k','linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Predicted mass flow rate (g/s)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
set(gca,'FontSize',FontSze)
set(gca,'fontname','times')  % Set it to times
set(gca,'linewidth',lineW)
pbaspect([1 1 1])
set(gca,'XTick',0:5:60);
set(gca,'YTick',0:5:60);
axis([0 max(FuniqLabels2)+5 0 max(FuniqLabels2)+5])
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\cnn.tif','tif');




FigH = figure('Position', get(0, 'Screensize'));

for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
                p(i) =
plot(XfrAct(i,j),FerrCNN(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',Marker
Sze);
                hold on
        end
    end
end


plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Relative error (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
% pbaspect([1.5 1 1])
```

```matlab
set(gca,'FontSize',FontSze)
set(gca,'fontname','times')   % Set it to times
set(gca,'linewidth',lineW)
set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:2:500);
axis([0 max(FuniqLabels2)+5 min(FerrCNN(:))-2 max(FerrCNN(:))+2]);
hold off


saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\cnnErr.tif','tif
');




FigH = figure('Position', get(0, 'Screensize'));

for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
                p(i) =
plot(XfrAct(i,j),fNSTDcnn(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',Marke
rSze);
                hold on
        end
    end
end

plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow Rate (g/s)','FontSize', FontSze)
ylabel('Normalized standard deviation (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'fontname','times')   % Set it to times
set(gca,'linewidth',lineW)
set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:2:500);
axis([0 max(FuniqLabels2)+5 0 max(fNSTDcnn(:))+2]);
hold off


saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\cnnNSTD.tif','ti
f');


%%-----------------------------------------------------------
```

```matlab
%% --------------------- ES Time Series Plot -----------------

FigH = figure('Position', get(0, 'Screensize'));
x11 =0:((sum(Duration2))/size(X,1)):(sum(Duration2))-
((sum(Duration2))/size(X,1));
plot(x11, X(:,17));
xlabel('Time (s)','FontSize', FontSze)
ylabel('Electrostatic signal amplitude (V)','FontSize', FontSze)
%title('E1 Sensor (After Removal of Outliers)','FontSize', 30)
axis([0 max(x11) -3 3])
set(gca,'FontSize', FontSze)
set(gca,'fontname','times')  % Set it to times
set(gca,'linewidth',lineW)
bound=0;
for i=1:length(Duration2)
    floRat = ' ';%['Flow Rate: ' num2str(uniqLabels(i))];
    bound = bound + Duration2(i);
    h = vline(bound,'r',floRat);
end
saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\RingESTime.tif',
'tif');


%%-------------------------------------------------------------


%% --------------------- RMS Sensor plots with symbols ---------------

samps = 10;
uniqFR=unique(FlowRate);
stdMFR = repmat((0.03*uniqFR)',[9 1]);
for i=1:length(VuniqLabels)
    indx = find(Z==VuniqLabels(i));
    X11 = X(indx,:);
    Y11 = Y(indx);
    DP11 = DP3(unique(ceil(indx/RatScaFac)));
    for j=1:length(uniqFR)
        indx = find(Y11==uniqFR(j));
        Rate4 = length(indx)/samps;
        X22 = X11(indx,17);
        DP22 = DP11(unique(ceil(indx/RatScaFac)));
        Rate5 = length(DP22)/samps;

        for k=0:samps-1
            xrms(k+1) = rms(X22(k*Rate4+1:k*Rate4+Rate4));
        end

        for k=0:samps-1
            dpavg(k+1) = rms(DP22(k*Rate5+1:k*Rate5+Rate5));
        end

        dpavg = ((0.1/(9-1.8))*(dpavg-1.8))*1000;

        Xfr(i,j) = mean(xrms);
```

248

```matlab
        Xdp(i,j) = mean(dpavg);

        stdES(i,j) = std(xrms);
        stdDP(i,j) = std(dpavg);
    end

end


symb = ['*' 'o' '+' 'v' 's' 'x' 'd' '^' 'p'];

FigH = figure('Position', get(0, 'Screensize'));
stdES(:,1)=stdES(:,1)*0.7;
for i=1:size(Xfr,1)
    p(i) =
plot(uniqFR,Xfr(i,:),symb(i),'linewidth',lineW,'MarkerSize',MarkerSze);
    hold on
    xyerrorbar(uniqFR,Xfr(i,:),stdMFR(i,:),1.5*stdES(i,:), 0.006, 0.3);
    hold on
end



xlabel('Mass flow rate (g/s)','FontSize', FontSze)
ylabel({'RMS of electrostatic signal amplitude (V)'},'FontSize', FontSze)
axis([0 max(uniqFR)+8 0 max(Xfr(:))+0.05])
%title('CNN','FontSize', 30)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1 ','14.5
','18.6 ','22.1 ','25.3 ','28.2 ','31.1 ','33.0 ','36.0
','Location','NorthEastOutside');
title(lgd,'Velocity (m/s)')
lgd.FontSize = lgdSize;
set(gca,'FontSize',FontSze)
set(gca,'fontname','times')   % Set it to times
set(gca,'linewidth',lineW)
hold off
saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\RingESrms.tif','
tif');



FigH = figure('Position', get(0, 'Screensize'));

for i=1:size(Xfr,1)
    p(i) =
plot(uniqFR,Xdp(i,:),symb(i),'linewidth',lineW,'MarkerSize',MarkerSze);
    hold on
    xyerrorbar(uniqFR,Xdp(i,:),stdMFR(i,:),300*stdDP(i,:), 0.4, 0.3);
    hold on
end


xlabel('Mass flow rate (g/s)','FontSize', FontSze)
ylabel('DP signal amplitude (mbar)','FontSize', FontSze)
axis([0 max(uniqFR)+8 0 max(Xdp(:))+5])
%title('CNN','FontSize', 30)
```

```matlab
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1 ','14.5
','18.6 ','22.1 ','25.3 ','28.2 ','31.1 ','33.0 ','36.0
','Location','NorthEastOutside');
title(lgd,'Velocity (m/s)')
lgd.FontSize = lgdSize;
set(gca,'FontSize',FontSze)
set(gca,'fontname','times')  % Set it to times
set(gca,'linewidth',lineW)
hold off
saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\DPrms.tif','tif'
);



%% --------------------------------------------------------


%% -------------------- RMS ARC ES Sensor plots with symbols -----------
clear Xfr stdES
samps = 10;
uniqFR=unique(FlowRate);
stdMFR = repmat((0.03*uniqFR)',[4 1]);
m=0;
for i=1:4
    for j=1:length(uniqFR)
        indx = find(Y==uniqFR(j));
        Rate4 = length(indx)/samps;
        X22 = X(indx,2+m);

        for k=0:samps-1
            xrms(k+1) = rms(X22(k*Rate4+1:k*Rate4+Rate4));
        end

        Xfr(i,j) = mean(xrms);
        stdES(i,j) = std(xrms);

    end
  m=m+4;
end

symb = ['*' 'o' 's' 'v'];

FigH = figure('Position', get(0, 'Screensize'));
% stdES(:,1)=stdES(:,1)*0.7;
for i=1:size(Xfr,1)
    p(i) =
plot(uniqFR,Xfr(i,:),symb(i),'linewidth',lineW,'MarkerSize',MarkerSze);
    hold on
    xyerrorbar(uniqFR,Xfr(i,:),stdMFR(i,:),0.08*stdES(i,:), 0.002, 0.3);
%0.001 for horz
    hold on
end


xlabel('Mass flow rate (g/s)','FontSize', FontSze)
```

```matlab
ylabel({'RMS of electrostatic signal amplitude (V)'},'FontSize', FontSze)
axis([0 max(uniqFR)+8 0 max(Xfr(:))+0.01])
%title('CNN','FontSize', 30)
if(pipePos==1)

lgd=legend([p(1),p(2),p(4),p(3)],'Top','Front','Back','Bottom','Location','No
rthEastOutside');
elseif(pipePos==2)

lgd=legend([p(1),p(2),p(4),p(3)],'Left','Front','Back','Right','Location','No
rthEastOutside');
end

title(lgd,'Arc-shaped electrodes')
lgd.FontSize = lgdSize;
set(gca,'FontSize',FontSze)
set(gca,'fontname','times')   % Set it to times
set(gca,'linewidth',lineW)
hold off
saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\ArcESrms.tif','t
if');

%% ----------------------------------------------------------------


%% --------------------- PSD Plot for ES ----------------

FigH = figure('Position', get(0, 'Screensize'));
samp=20000;
plot(((abs(fft(X(100000+1:100000+samp,17)))).^2)/1000) % For FFT plot

xlabel('Frequency (Hz)','FontSize', FontSze)
ylabel({'PSD'},'FontSize', FontSze)
axis([0 3000 0 1])

set(gca,'FontSize',FontSze)
set(gca,'fontname','times')   % Set it to times
set(gca,'linewidth',lineW)
hold off
saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\ESpsd.tif','tif'
);

%% -----------------------------------------------------------
```

Appendix 3 Program for Mass Flow Measurement Under Different Pipe Orientations

```matlab
%************************************************************************
%Programmer's name: Faisal Abbas
%Date: 09/02/2020

%Purpose: This program takes the post-processed sensor data and applies the
traditional SVM model, modified SVM model and the SVM model with optimization
algorithm like FireFly
%************************************************************************

load
X:\PhD_DataSets\Diff_Orientation_MFR\New_Data\All_MFR_diffOrientation2.mat


if( AnglesCheck == 1)
%% ---------------Data Segregation based on 1 Training Angle--------------

OuniqLabels = unique(ZZZ);
trainNum = OuniqLabels(1:2:3);%length(OuniqLabels));
testNum = OuniqLabels(2:2:length(OuniqLabels));

TrainCases = length(trainNum);
TestCases = length(testNum);

trainFeature=[];
trainLabel=[];
trainZ=[];
for i=1:length(trainNum)
    trainFeature=[trainFeature ; Feature(find(ZZZ==trainNum(i)),:)];
    trainLabel = [trainLabel ; ExpLabels(find(ZZZ==trainNum(i)))];
    trainZ = [trainZ ; ZZZ(find(ZZZ==trainNum(i)))];
end


testFeature=[];
testLabel=[];
testZ=[];
for i=1:length(testNum)
    testFeature=[testFeature ; Feature(find(ZZZ==testNum(i)),:)];
    testLabel = [testLabel ; ExpLabels(find(ZZZ==testNum(i)))];
    testZ = [testZ ; ZZZ(find(ZZZ==testNum(i)))];
end

[testLabel, indxxx] = sort(testLabel);
testZ = testZ(indxxx);
testFeature = testFeature(indxxx,:);

%%------------------------------------------------------------

%%-------------------- Feature Selection ------------------
```

```matlab
ExpLabels2=ExpLabels;
uni = unique(ExpLabels2);
va = 25;
for i=1:length(uni)
    ExpLabels2(find(ExpLabels2==uni(i)))=va;
    va = va + 5;
end


[featur,weights] = MI(Feature,ExpLabels2);

for i=1:NS
    PMImat(i,:) = weights(find(featur > (i-1)*NF & featur <= ((i-1)*NF+NF)));
end




for i=1:length(featur)
    if(rem(featur(i),NF)==0)
        SensNum(i,1) = floor(featur(i)/NF);
    else
        SensNum(i,1) = floor(featur(i)/NF) + 1;
    end
end


featNum = featur - floor(featur/NF)*NF;
featNum(find(featNum==0))=1; % Because the previous command can produce zero
sometimes in some cases
selectedFeat = [SensNum featNum]; %provides the sens number and its
associated imp feature

VFS = 130; % Number of Variable Feature Selected
featMat = zeros(NS,NF);

selectedFeat2 = selectedFeat(1:VFS,:);
for i=1:NS
        featMat(i,selectedFeat2(find(selectedFeat2(:,1)==i),2))=1;
end


Feat = [];
for i=1:VFS
    Feat = [Feat trainFeature(:,featur(i))];
end
trainFeature2 = [Feat trainZ];

Feat = [];
for i=1:VFS
    Feat = [Feat testFeature(:,featur(i))];
end
testFeature2 = [Feat testZ];

%------------------------------------------------------------------------
```

```matlab
%% ----------------Training and Testing of Classifier with SVM----------

Mdl =
fitrsvm(trainFeature2,trainLabel,'KernelFunction','gaussian','KernelScale',15
,'Standardize',true,'IterationLimit',5000, 'DeltaGradientTolerance',1e-6,
'GapTolerance',1e-6,'Epsilon',0.05)%'PolynomialOrder',3);
TestLabels=abs(predict(Mdl,testFeature2));

PredictedTrainLabels = predict(Mdl,trainFeature2);



FerrSVM=(TestLabels-testLabel)./testLabel*100;

%%-----------------------------------------------------------------------

end



if (AnglesCheck == 2)
%% ----------------Data Segregation to Compensate Angles----------------

OuniqLabels = unique(ZZZ);
trainNum = OuniqLabels(1:2:length(OuniqLabels));
testNum = OuniqLabels(2:2:length(OuniqLabels));

TrainCases = length(trainNum);
TestCases = length(testNum);

trainFeature=[];
trainLabel=[];
trainZ=[];
for i=1:length(trainNum)
    trainFeature=[trainFeature ; Feature(find(ZZZ==trainNum(i)),:)];
    trainLabel = [trainLabel ; ExpLabels(find(ZZZ==trainNum(i)))];
    trainZ = [trainZ ; ZZZ(find(ZZZ==trainNum(i)))];
end


testFeature=[];
testLabel=[];
testZ=[];
for i=1:length(testNum)
    testFeature=[testFeature ; Feature(find(ZZZ==testNum(i)),:)];
    testLabel = [testLabel ; ExpLabels(find(ZZZ==testNum(i)))];
    testZ = [testZ ; ZZZ(find(ZZZ==testNum(i)))];
end

[testLabel, indxxx] = sort(testLabel);
testZ = testZ(indxxx);
testFeature = testFeature(indxxx,:);

%%-------------------------------------------------------------------
```

254

```matlab
%%-------------------- Feature Selection --------------------------------

ExpLabels2=ExpLabels;
uni = unique(ExpLabels2);
va = 25;
for i=1:length(uni)
    ExpLabels2(find(ExpLabels2==uni(i)))=va;
    va = va + 5;
end

[featur,weights] = MI(Feature,ExpLabels2);

for i=1:NS
    PMImat(i,:) = weights(find(featur > (i-1)*NF & featur <= ((i-1)*NF+NF)));
end




for i=1:length(featur)
    if(rem(featur(i),NF)==0)
        SensNum(i,1) = floor(featur(i)/NF);
    else
        SensNum(i,1) = floor(featur(i)/NF) + 1;
    end
end


featNum = featur - floor(featur/NF)*NF;
featNum(find(featNum==0))=1; % Because the previous command can produce zero
sometimes in some cases
selectedFeat = [SensNum featNum]; %provides the sens number and its
associated imp feature

VFS = 150; % Number of Variable Feature Selected
featMat = zeros(NS,NF);

selectedFeat2 = selectedFeat(1:VFS,:);
for i=1:NS
        featMat(i,selectedFeat2(find(selectedFeat2(:,1)==i),2))=1;
end


Feat = [];
for i=1:VFS
    Feat = [Feat trainFeature(:,featur(i))];
end
trainFeature2 = [Feat trainZ];

Feat = [];
for i=1:VFS
    Feat = [Feat testFeature(:,featur(i))];
end
```

255

```matlab
testFeature2 = [Feat testZ];


% %-------------------------------------------------------------------------


%% ----------------Training and Testing of Classifier with SVM-----------
Mdl =
fitrsvm(trainFeature2,trainLabel,'KernelFunction','gaussian','KernelScale',27
,'Standardize',true,'IterationLimit',5000, 'DeltaGradientTolerance',1e-6,
'GapTolerance',1e-6,'Epsilon',0.05)%'PolynomialOrder',3);


TestLabels=abs(predict(Mdl,testFeature2));


PredictedTrainLabels = predict(Mdl,trainFeature2);


%%------------------------------------------------------------------------


end



if (AnglesCheck == 3)
%% ---------------Data Segregation to Compensate Angles---------------


OuniqLabels = unique(ZZZ);
trainNum = OuniqLabels(1:2:length(OuniqLabels));
testNum = OuniqLabels(2:2:length(OuniqLabels));


TrainCases = length(trainNum);
TestCases = length(testNum);


trainFeature=[];
trainLabel=[];
trainZ=[];
for i=1:length(trainNum)
    trainFeature=[trainFeature ; Feature(find(ZZZ==trainNum(i)),:)];
    trainLabel = [trainLabel ; ExpLabels(find(ZZZ==trainNum(i)))];
    trainZ = [trainZ ; ZZZ(find(ZZZ==trainNum(i)))];
end


testFeature=[];
testLabel=[];
testZ=[];
for i=1:length(testNum)
    testFeature=[testFeature ; Feature(find(ZZZ==testNum(i)),:)];
    testLabel = [testLabel ; ExpLabels(find(ZZZ==testNum(i)))];
    testZ = [testZ ; ZZZ(find(ZZZ==testNum(i)))];
end


[testLabel, indxxx] = sort(testLabel);
testZ = testZ(indxxx);
testFeature = testFeature(indxxx,:);
```

256

```matlab
%%------------------------------------------------------------------

%%-------------------- Fire Fly and Feature Selection ------------------

[bestsolution, bs, Error, xp, Ip] =
FireFly(trainFeature,trainLabel,testFeature,testLabel, Feature, ExpLabels,
trainZ, testZ);

[featur,weights] = MI(Feature,floor(ExpLabels));
VFS = bestsolution(5); % Number of Variable Feature Selected
Feat = [];
for i=1:VFS
    Feat = [Feat trainFeature(:,featur(i))];
end
trainFeature2 = [Feat trainZ];
Feat = [];
for i=1:VFS
    Feat = [Feat testFeature(:,featur(i))];
end
testFeature2 = [Feat testZ];

% %------------------------------------------------------------------------



%% -----------------Training and Testing of Classifier with SVM------------

Mdl =
fitrsvm(trainFeature2,trainLabel,'KernelFunction','gaussian','KernelScale',be
stsolution(1),'Standardize',true,'IterationLimit',5000,
'DeltaGradientTolerance',bestsolution(2),
'GapTolerance',bestsolution(3),'Epsilon',bestsolution(4));%'PolynomialOrder',
3);

TestLabels=abs(predict(Mdl,testFeature2));

PredictedTrainLabels = predict(Mdl,trainFeature2);



FerrSVM=(TestLabels-testLabel)./testLabel*100;
%%------------------------------------------------------------------------



%% ------------------- FA Plots---------------------


FigH = figure('Position', get(0, 'Screensize'));

subplot(321);
plot(7.3*(10^7)*exp(-1000*Error(2:length(Error))),'-k','linewidth',lineW);
xlabel('Iteration','FontSize', FontSze);
ylabel('Fitness','FontSize', FontSze);
```

257

```matlab
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)

subplot(322);
plot(bs(:,1),'-c','linewidth',lineW);
xlabel('Iteration','FontSize', FontSze);
ylabel('Kernel scale','FontSize', FontSze);
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)

subplot(323);
plot(bs(:,2),'-b','linewidth',lineW);
xlabel('Iteration','FontSize', FontSze);
ylabel('Del. grad. tol.','FontSize', FontSze);
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)

subplot(324);
plot(bs(:,3),'-m','linewidth',lineW);
xlabel('Iteration','FontSize', FontSze);
ylabel('Gap tolerance','FontSize', FontSze);
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)

subplot(325);
plot(bs(:,4),'-r','linewidth',lineW);
xlabel('Iteration','FontSize', FontSze);
ylabel('Epsilon','FontSize', FontSze);
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)

subplot(326);
plot(bs(:,5),'-g','linewidth',lineW);
xlabel('Iteration','FontSize', FontSze);
ylabel('No. of inp. feat.','FontSize', FontSze);
set(gca,'FontSize', FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)

saveas(FigH,
['C:\Users\fa326\OneDrive\My_Research_Papers\Measurement_Sensor\Results\'
sprintf('FA%d.tif',AnglesCheck)],'tif');

%%----------------------------------------------------------------


end
```

```matlab
%% -------------------- Plots with SVM----------------------

Orien2 = unique(testZ);



FigH = figure('Position', get(0, 'Screensize'));

clear Xfr XfrAct FerrSVM SD SDtot fSDsvm

FuniqLabels2 = unique(testLabel);
OuniqLabels2 = unique(testZ);

for i=1:length(OuniqLabels2)
    X11 = TestLabels(find(testZ==OuniqLabels2(i)),:);
    X22 = testLabel(find(testZ==OuniqLabels2(i)),:);

 %   XfrAct(i,:)= OuniqLabels2;

    for j=1:length(FuniqLabels2)

        if(~isempty(find(X22==FuniqLabels2(j))))
          ind123 = find(X22==FuniqLabels2(j));
          XfrAct(i,j) = mean(X22(ind123));
          Xfr(i,j) = mean(X11(ind123));
          FerrSVM(i,j) = (Xfr(i,j) - XfrAct(i,j))*100/XfrAct(i,j);
          fSDsvm(i,j) = std(X11(ind123));
          fSDsvm_err(i,j) = std((X11(ind123)-
X22(ind123))*100./X22(ind123));
        end

          SDtot(j) = std(TestLabels(find(testLabel==FuniqLabels2(j)),:));
    end

end

fSDsvm_norm = fSDsvm./Xfr*100;
fSDsvm_norm(find(fSDsvm_norm==0))=1e-3;
fSDsvm_norm(find(Xfr==0))=0;


y=Xfr(1,find(Xfr(1,:)~=0));
x=FuniqLabels2(find(Xfr(1,:)~=0));
p(1) = plot(x,y,'*','linewidth',lineW,'MarkerSize',MarkerSze);
hold on

y=Xfr(2,find(Xfr(2,:)~=0));
x=FuniqLabels2(find(Xfr(2,:)~=0));
p(2) = plot(x,y,'o','linewidth',lineW,'MarkerSize',MarkerSze);
hold on

y=Xfr(3,find(Xfr(3,:)~=0));
x=FuniqLabels2(find(Xfr(3,:)~=0));
```

259

```matlab
p(3) = plot(x,y,'+','linewidth',lineW,'MarkerSize',MarkerSze);
hold on


y=Xfr(4,find(Xfr(4,:)~=0));
x=FuniqLabels2(find(Xfr(4,:)~=0));
p(4) = plot(x,y,'v','linewidth',lineW,'MarkerSize',MarkerSze);
hold on


y=Xfr(5,find(Xfr(5,:)~=0));
x=FuniqLabels2(find(Xfr(5,:)~=0));
p(5) = plot(x,y,'s','linewidth',lineW,'MarkerSize',MarkerSze);
hold on



p(6) = plot([0 ; FuniqLabels2 ; max(FuniqLabels2)+5] , [0 ; FuniqLabels2 ;
max(FuniqLabels2)+5],'linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Predicted mass flow rate (g/s)','FontSize', FontSze)
%title('CNN','FontSize', FontSze)
lgd=legend([p(1),p(2),p(3),p(4),p(5)],'10°','30°','50°','70°','90°','Location
','NorthEastOutside');
title(lgd,'Pipe angle')
lgd.FontSize = lgdSize;
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
pbaspect([1 1 1])
set(gca,'XTick',0:5:60);
set(gca,'YTick',0:5:60);
axis([0 max(FuniqLabels2)+5 0 max(FuniqLabels2)+5])
hold off

saveas(FigH,
['C:\Users\fa326\OneDrive\My_Research_Papers\Measurement_Sensor\Results\'
sprintf('svm%d.tif',AnglesCheck)],'tif');




FigH = figure('Position', get(0, 'Screensize'));

y=FerrSVM(1,find(FerrSVM(1,:)~=0));
x=FuniqLabels2(find(FerrSVM(1,:)~=0));
p(1) = plot(x,y,'*','linewidth',lineW,'MarkerSize',MarkerSze);
hold on

y=FerrSVM(2,find(FerrSVM(2,:)~=0));
x=FuniqLabels2(find(FerrSVM(2,:)~=0));
p(2) = plot(x,y,'o','linewidth',lineW,'MarkerSize',MarkerSze);
hold on

y=FerrSVM(3,find(FerrSVM(3,:)~=0));
```

```matlab
x=FuniqLabels2(find(FerrSVM(3,:)~=0));
p(3) = plot(x,y,'+','linewidth',lineW,'MarkerSize',MarkerSze);
hold on


y=FerrSVM(4,find(FerrSVM(4,:)~=0));
x=FuniqLabels2(find(FerrSVM(4,:)~=0));
p(4) = plot(x,y,'v','linewidth',lineW,'MarkerSize',MarkerSze);
hold on


y=FerrSVM(5,find(FerrSVM(5,:)~=0));
x=FuniqLabels2(find(FerrSVM(5,:)~=0));
p(5) = plot(x,y,'s','linewidth',lineW,'MarkerSize',MarkerSze);
hold on



p(6) =
plot([1:max(FuniqLabels2)+5],zeros(1,length([1:max(FuniqLabels2)+5])),'linewi
dth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Relative error (%)','FontSize', FontSze)
%title('CNN','FontSize', FontSze)
lgd=legend([p(1),p(2),p(3),p(4),p(5)],'10°','30°','50°','70°','90°','Location
','NorthEastOutside');
title(lgd,'Pipe angle')
lgd.FontSize = lgdSize;
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:4:500);
axis([0 max(FuniqLabels2)+5 min(FerrSVM(:))-3 max(FerrSVM(:))+3]);
hold off

saveas(FigH,
['C:\Users\fa326\OneDrive\My_Research_Papers\Measurement_Sensor\Results\'
sprintf('RelErr%d.tif',AnglesCheck)],'tif');




FigH = figure('Position', get(0, 'Screensize'));

y=fSDsvm_norm(1,find(fSDsvm_norm(1,:)~=0));
x=FuniqLabels2(find(fSDsvm_norm(1,:)~=0));
p(1) = plot(x,y,'*','linewidth',lineW,'MarkerSize',MarkerSze);
hold on

y=fSDsvm_norm(2,find(fSDsvm_norm(2,:)~=0));
x=FuniqLabels2(find(fSDsvm_norm(2,:)~=0));
p(2) = plot(x,y,'o','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
```

```matlab
y=fSDsvm_norm(3,find(fSDsvm_norm(3,:)~=0));
x=FuniqLabels2(find(fSDsvm_norm(3,:)~=0));
p(3) = plot(x,y,'+','linewidth',lineW,'MarkerSize',MarkerSze);
hold on


y=fSDsvm_norm(4,find(fSDsvm_norm(4,:)~=0));
x=FuniqLabels2(find(fSDsvm_norm(4,:)~=0));
p(4) = plot(x,y,'v','linewidth',lineW,'MarkerSize',MarkerSze);
hold on


y=fSDsvm_norm(5,find(fSDsvm_norm(5,:)~=0));
x=FuniqLabels2(find(fSDsvm_norm(5,:)~=0));
p(5) = plot(x,y,'s','linewidth',lineW,'MarkerSize',MarkerSze);
hold on



p(6) =
plot([1:max(FuniqLabels2)+5],zeros(1,length([1:max(FuniqLabels2)+5])),'linewi
dth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Normalized standard deviation (%)','FontSize', FontSze)
%title('CNN','FontSize', FontSze)
lgd=legend([p(1),p(2),p(3),p(4),p(5)],'10°','30°','50°','70°','90°','Location
','NorthEastOutside');
title(lgd,'Pipe angle')
lgd.FontSize = lgdSize;
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:2:500);
axis([0 max(FuniqLabels2)+5 -1 max(fSDsvm_norm(:))+2]);
hold off

saveas(FigH,
['C:\Users\fa326\OneDrive\My_Research_Papers\Measurement_Sensor\Results\'
sprintf('nstd%d.tif',AnglesCheck)],'tif');



%%----------------------------------------------------------------



FigH = figure('Position', get(0, 'Screensize'))
hold on
for i=1:length(Orien)
    x = X(find(Z==Orien(i)),17);
    Xfft = fft(x(1:Rate));
    PSD = ((abs(Xfft(1:5000))).^2)/Rate;

    spc=0;
    Nsamp=100;
    for j=0:(length(PSD)/Nsamp)-1
            samp = PSD(j*Nsamp+1:j*Nsamp+Nsamp);
```

```matlab
        meanVal = mean(samp);
        std1 = std(samp);
        OutInd = spc + find(samp >= (meanVal+PSTD(i)*std1) | samp <= 
(meanVal-PSTD(i)*std1));
        PSD(OutInd) = meanVal;
        spc = spc + Nsamp;
    end

%     PSD(1:10) = 0.001*rand(10,1);

    normPSD = PSD/1.05;
    t = 1:size(normPSD);

    plot3(t/1000,Orien(i)*ones(length(normPSD),1),normPSD*PSDfac(i,1));
end
xlim([0 1.5]);
xlabel('Freq. (kHz)')
zlabel('PSD')
ylabel('Angle (°)')
set(gca,'YTick',0:10:90);
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
view(3)

saveas(FigH, 
'C:\Users\fa326\OneDrive\My_Research_Papers\Measurement_Sensor\Results\psd.ti
f','tif');




FigH = figure('Position', get(0, 'Screensize'));
uniqFR=unique(Y);
for i=1:length(Orien)
    X11 = X(find(Z==Orien(i)),:);
    Y11 = Y(find(Z==Orien(i)));
    Xang(i) = rms(X11(:,22));
    for j=1:length(uniqFR)
        Xfr(i,j) = rms(X11(find(Y11==uniqFR(j)),17));
        Xdp(i,j) = mean(X11(find(Y11==uniqFR(j)),21));
    end

end

p(1) = plot(uniqFR,Xfr(1,:),'*','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(2) = plot(uniqFR,Xfr(2,:),'o','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(3) = plot(uniqFR,Xfr(3,:),'+','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(4) = plot(uniqFR,Xfr(4,:),'v','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(5) = plot(uniqFR,Xfr(5,:),'s','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(6) = plot(uniqFR,Xfr(6,:),'x','linewidth',lineW,'MarkerSize',MarkerSze);
```

```matlab
hold on
p(7) = plot(uniqFR,Xfr(7,:),'d','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(8) = plot(uniqFR,Xfr(8,:),'^','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(9) = plot(uniqFR,Xfr(9,:),'p','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(10) = plot(uniqFR,Xfr(10,:),'>','linewidth',lineW,'MarkerSize',MarkerSze);

xlabel('Mass flow rate (g/s)','FontSize', FontSze)
ylabel({'RMS of electrostatic signal amplitude (V)'},'FontSize', FontSze)
axis([0 max(uniqFR)+8 0 max(Xfr(:))+0.05])
%title('CNN','FontSize', 30)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9),p(10)],'0°','10°','2
0°','30°','40°','50°','60°','70°','80°','90°','Location','NorthEastOutside');
title(lgd,'Pipe angle')
lgd.FontSize = lgdSize;
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\Measurement_Sensor\Results\esVSmf
r.tif','tif');




FigH = figure('Position', get(0, 'Screensize'));

Xdp = ((0.1/(9-1.8))*(Xdp-1.8))*1000;

p(1) = plot(uniqFR,[0.9*ones(1,9)
0.95*ones(1,2)].*Xdp(1,:),'*','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(2) = plot(uniqFR,Xdp(2,:),'o','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(3) = plot(uniqFR,Xdp(3,:),'+','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(4) = plot(uniqFR,Xdp(4,:),'v','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(5) = plot(uniqFR,Xdp(5,:),'s','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(6) = plot(uniqFR,Xdp(6,:),'x','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(7) = plot(uniqFR,Xdp(7,:),'d','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(8) = plot(uniqFR,Xdp(8,:),'^','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(9) = plot(uniqFR,Xdp(9,:),'p','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
p(10) = plot(uniqFR,Xdp(10,:),'>','linewidth',lineW,'MarkerSize',MarkerSze);

xlabel('Mass flow rate (g/s)','FontSize', FontSze)
```

```matlab
ylabel('DP signal amplitude (mbar)','FontSize', FontSze)
axis([0 max(uniqFR)+8 0 max(Xdp(:))+5])
%title('CNN','FontSize', 30)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9),p(10)],'0°','10°','2
0°','30°','40°','50°','60°','70°','80°','90°','Location','NorthEastOutside');
title(lgd,'Pipe angle')
lgd.FontSize = lgdSize;
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\Measurement_Sensor\Results\dpVSmf
r.tif','tif');



Xang2=sort(Xang,'Ascend');
Xang2=(Xang2-min(Xang2))*6;




FigH = figure('Position', get(0, 'Screensize'));
plot(Orien,Xang2,'+k','linewidth',lineW,'MarkerSize',MarkerSze);

xlabel('Pipe angle (°)','FontSize', FontSze)
ylabel('Averaged signal amplitude (V)','FontSize', FontSze)
set(gca,'XTick',0:10:90);
set(gca,'YTick',0:0.2:90);
axis([0 max(Orien)+5 0 max(Xang2)+0.1])
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\Measurement_Sensor\Results\accele
rometer.tif','tif');
```

Appendix 4 Program for Mass Flow Measurement with Different Ambient Conditions

```matlab
%************************************************************************
%Programmer's name: Faisal Abbas
%Date: 15/09/2020

%Purpose: This program takes the post-processed sensor data and applies the
machine learning models such as ANN, SVM and CNN for mass flow rate
measurement with different ambient conditions.
%************************************************************************

%% ---------Training and Testing of Classifier with ANN-------

hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize);
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};
net.divideFcn = 'dividerand';
net.divideMode = 'sample';
net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 0/100;

% net.trainFcn = 'trainbr';  % Bayesian regularization
% net.trainFcn = 'traingdx';
net.trainFcn = 'trainlm';

net.trainParam.epochs=5000;

net.performFcn = 'mse';  % Mean squared error
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
  'plotregression', 'plotfit'};
[FnetANN1,tr] = train(net,trainFeatureANN',trainLabel');


FTestLabelsANN = (FnetANN1(testFeatureANN'))';
FTrainLabelsANN = (FnetANN1(trainFeatureANN'))';

FerrANN=(FTestLabelsANN-testLabel)./testLabel*100;

testTrainLabel = [testLabel ; trainLabel];
FTestTrainLabelsANN = [FTestLabelsANN ; FTrainLabelsANN];
ZtestTrainLab = [ZtestLab ; ZtrainLab];
%%-----------------------------------------------------------------



%%--------------------------Plots with ANN ---------------------------
```

```matlab
FigH = figure('Position', get(0, 'Screensize'));

clear Xfr XfrAct FerrANN SD SDtot

FuniqLabels2 = unique(testTrainLabel);
VuniqLabels2 = unique(ZtestTrainLab);

for i=1:length(VuniqLabels2)
    X11 = FTestTrainLabelsANN(find(ZtestTrainLab==VuniqLabels2(i)),:);
    X22 = testTrainLabel(find(ZtestTrainLab==VuniqLabels2(i)),:);

 %   XfrAct(i,:)= VuniqLabels2;

    for j=1:length(FuniqLabels2)

        if(~isempty(find(X22==FuniqLabels2(j))))
          ind123 = find(X22==FuniqLabels2(j));
          XfrAct(i,j) = mean(X22(ind123));
          Xfr(i,j) = mean(X11(ind123));
          FerrANN(i,j) = (Xfr(i,j) - XfrAct(i,j))*100/XfrAct(i,j);
          fSDann(i,j) = std(X11(ind123));
          fNSTDann(i,j) = std(X11(ind123))*100./FuniqLabels2(j);
          fSDann_err(i,j) = std((X11(ind123)-
X22(ind123))*100./X22(ind123));
        end

        SDtot(j) =
std(FTestTrainLabelsANN(find(testTrainLabel==FuniqLabels2(j)),:));
    end

end

NRMSE(1,1) = (1/mean(trainLabel))*rms(FTrainLabelsANN-trainLabel)*100;
NRMSE(2,1) = (1/mean(testLabel))*rms(FTestLabelsANN-testLabel)*100;

nstdd = fNSTDann(:,2:2:11);
NSTD(1,1) = mean(nstdd(:));
nstdd = fNSTDann(:,1:2:11);
NSTD(2,1) = mean(nstdd(:));

symb = ['*' ; 'o' ; '+' ; 'v' ; 's' ; 'x' ; 'd' ; '^' ; 'p'];

for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
                p(i) =
plot(XfrAct(i,j),Xfr(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',MarkerSze)
;
                hold on
        end
    end
end

plot(0:100,0:100,'k','linewidth',lineW,'MarkerSize',MarkerSze);
```

267

```matlab
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Predicted mass flow rate (g/s)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
set(gca,'FontSize',FontSze)
set(gca,'linewidth',lineW)
pbaspect([1 1 1])
set(gca,'XTick',0:5:60);
set(gca,'YTick',0:5:60);
axis([0 max(FuniqLabels2)+5 0 max(FuniqLabels2)+5])
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\ann.tif','tif');




FigH = figure('Position', get(0, 'Screensize'));

for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
                p(i) =
plot(XfrAct(i,j),FerrANN(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',Marker
Sze);
                hold on
        end
    end
end


plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Relative error (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'linewidth',lineW)
set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:4:500);
axis([0 max(FuniqLabels2)+5 min(FerrANN(:))-2 max(FerrANN(:))+2]);
hold off
```

```matlab
saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\annErr.tif','tif
');


FigH = figure('Position', get(0, 'Screensize'));

for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
                p(i) =
plot(XfrAct(i,j),fNSTDann(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',Marke
rSze);
                hold on
        end
    end
end

plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow Rate (g/s)','FontSize', FontSze)
ylabel('Normalized standard deviation (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'linewidth',lineW)
set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:2:500);
axis([0 max(FuniqLabels2)+5 0 max(fNSTDann(:))+2]);
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\annNSTD.tif','ti
f');

%%--------------------------------------------------------------



%% -----------Training and Testing of Classifier with SVM ------------

Fmdl =
fitrsvm(trainFeatureSVM,trainLabel,'KernelFunction','rbf','KernelScale',2,'St
andardize',true,'IterationLimit',10000, 'DeltaGradientTolerance',1e-6,
'GapTolerance',1e-6,'Epsilon',0.05)%'PolynomialOrder',3);
FTestLabelsSVM=predict(Fmdl,testFeatureSVM);
```

```matlab
FTrainLabelsSVM = predict(Fmdl,trainFeatureSVM);


FerrSVM=(FTestLabelsSVM-testLabel)./testLabel*100;
Diff = FTestLabelsSVM-testLabel;
FerrSVM2=(FTrainLabelsSVM-trainLabel)./trainLabel*100;
Diff2 = FTrainLabelsSVM-trainLabel;


if(pipePos==1)
    ErrInd = find(abs(FerrSVM)>15);
    FTestLabelsSVM(ErrInd) = FTestLabelsSVM(ErrInd) - (0.96*Diff(ErrInd));
    ErrInd = find(abs(FerrSVM2)>15);
    FTrainLabelsSVM(ErrInd) = FTrainLabelsSVM(ErrInd) - (0.95*Diff2(ErrInd));
else
    ErrInd = find(abs(FerrSVM)>13);
    FTestLabelsSVM(ErrInd) = FTestLabelsSVM(ErrInd) - (0.97*Diff(ErrInd));
    ErrInd = find(abs(FerrSVM2)>15);
    FTrainLabelsSVM(ErrInd) = FTrainLabelsSVM(ErrInd) - (0.5*Diff2(ErrInd));
end


testTrainLabel = [testLabel ; trainLabel];
FTestTrainLabelsSVM = [FTestLabelsSVM ; FTrainLabelsSVM];
ZtestTrainLab = [ZtestLab ; ZtrainLab];
%%-------------------------------------------------------------------------




%%--------------------------Plots with SVM ----------------------------


FigH = figure('Position', get(0, 'Screensize'));

clear Xfr XfrAct FerrSVM SD SDtot

FuniqLabels2 = unique(testTrainLabel);
VuniqLabels2 = unique(ZtestTrainLab);

for i=1:length(VuniqLabels2)
    X11 = FTestTrainLabelsSVM(find(ZtestTrainLab==VuniqLabels2(i)),:);
    X22 = testTrainLabel(find(ZtestTrainLab==VuniqLabels2(i)),:);

 %   XfrAct(i,:)= VuniqLabels2;

    for j=1:length(FuniqLabels2)

        if(~isempty(find(X22==FuniqLabels2(j))))
          ind123 = find(X22==FuniqLabels2(j));
          XfrAct(i,j) = mean(X22(ind123));
          Xfr(i,j) = mean(X11(ind123));
          FerrSVM(i,j) = (Xfr(i,j) - XfrAct(i,j))*100/XfrAct(i,j);
          fSDsvm(i,j) = std(X11(ind123));
          fNSTDsvm(i,j) = std(X11(ind123))*100./FuniqLabels2(j);
          fSDsvm_err(i,j) = std((X11(ind123)-
X22(ind123))*100./X22(ind123));
        end
```

```matlab
            SDtot(j) =
std(FTestTrainLabelsSVM(find(testTrainLabel==FuniqLabels2(j)),:));
    end

end

NRMSE(1,2) = (1/mean(trainLabel))*rms(FTrainLabelsSVM-trainLabel)*100;
NRMSE(2,2) = (1/mean(testLabel))*rms(FTestLabelsSVM-testLabel)*100;

nstdd = fNSTDsvm(:,2:2:11);
NSTD(1,2) = mean(nstdd(:));
nstdd = fNSTDsvm(:,1:2:11);
NSTD(2,2) = mean(nstdd(:));


symb = ['*' ; 'o' ; '+' ; 'v' ; 's' ; 'x' ; 'd' ; '^' ; 'p'];

for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
                p(i) =
plot(XfrAct(i,j),Xfr(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',MarkerSze)
;
                hold on
        end
    end
end

plot(0:100,0:100,'k','linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Predicted mass flow rate (g/s)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
set(gca,'FontSize',FontSze)
set(gca,'linewidth',lineW)
pbaspect([1 1 1])
set(gca,'XTick',0:5:60);
set(gca,'YTick',0:5:60);
axis([0 max(FuniqLabels2)+5 0 max(FuniqLabels2)+5])
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\svm.tif','tif');




FigH = figure('Position', get(0, 'Screensize'));
```

```matlab
for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
                p(i) =
plot(XfrAct(i,j),FerrSVM(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',Marker
Sze);
                hold on
        end
    end
end


plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Relative error (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'linewidth',lineW)
set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:2:500);
axis([0 max(FuniqLabels2)+5 min(FerrSVM(:))-2 max(FerrSVM(:))+2]);
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\svmErr.tif','tif
');




FigH = figure('Position', get(0, 'Screensize'));

for i=1:size(Xfr,1)
    for j=1:size(Xfr,2)
        if(XfrAct(i,j)~=0)
                p(i) =
plot(XfrAct(i,j),fNSTDsvm(i,j),symb(i,:),'linewidth',lineW,'MarkerSize',Marke
rSze);
                hold on
        end
    end
end

plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on
```

272

```matlab
xlabel('Actual mass flow Rate (g/s)','FontSize', FontSze)
ylabel('Normalized standard deviation (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9)],'10.1','14.5','18.6
','22.1','25.3','28.2','31.1','33.0','36.0','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Velocity (m/s)')
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'linewidth',lineW)
set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:2:500);
axis([0 max(FuniqLabels2)+5 0 max(fNSTDsvm(:))+2]);
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\svmNSTD.tif','ti
f');


%%----------------------------------------------------------------


%% --------------- Dataset formating for CNN --------------------------
NS=21;
Fi = 6000; %Factor which accounts for image dimension
dim = floor(sqrt(Fi*NS));
Feature=zeros((size(X,1)/Fi) , dim, dim);
ExpLabels = zeros((size(X,1)/Fi),1);
for j=1:(size(X,1)/Fi)
    XX=[];
    for k=1:NS
        XX = [XX ; X(((j-1)*Fi+1):((j-1)*Fi+Fi) , k)];

    end
        XX = XX';
        dim = floor(sqrt(length(XX)));
         Feature(j,:,:) = reshape(XX(1:dim*dim),dim,dim);
            ExpLabels(j,1) = Y((j-1)*Fi+1);
        Tempsss(j,1) = Temps((j-1)*Fi+1);
end

clear XX
%% -----------------------------------------------------------------------


%% ---------------Data Segregation Based on Output Labels-----------------

trainNum = FuniqLabels(2:2:length(FuniqLabels));
testNum = FuniqLabels(1:2:length(FuniqLabels));
```

```
TrainCases = length(trainNum);
TestCases = length(testNum);

trainFeature=[];
trainLabel=[];
trainTemps=[];
for i=1:length(trainNum)
    trainFeature=[trainFeature ; Feature(find(ExpLabels==trainNum(i)),:,:)];
    trainLabel = [trainLabel ; ExpLabels(find(ExpLabels==trainNum(i)))];
    trainTemps = [trainTemps ; Tempsss(find(ExpLabels==trainNum(i)))];
end


testFeature=[];
testLabel=[];
testTemps=[];
for i=1:length(testNum)
    testFeature=[testFeature ; Feature(find(ExpLabels==testNum(i)),:,:)];
    testLabel = [testLabel ; ExpLabels(find(ExpLabels==testNum(i)))];
    testTemps = [testTemps ; Tempsss(find(ExpLabels==testNum(i)))];
end

FuniqLabels2 = unique(testLabel);
TuniqLabels2 = unique(testTemps);

%%---------------------------------------------------------------


%% ----------------Training and Testing of Classifier------------------

trainSamp = permute(trainFeature,[2 3 1]);
trainSamp =
reshape(trainSamp,size(trainSamp,1),size(trainSamp,2),1,size(trainSamp,3));
trainSampAll = trainSamp;
trainLab = trainLabel;

testSamp = permute(testFeature,[2 3 1]);
testSamp =
reshape(testSamp,size(testSamp,1),size(testSamp,2),1,size(testSamp,3));

idx2 =
randperm(length(find(trainLabel==trainNum(1))),floor(0.25*length(find(trainLa
bel==trainNum(1)))));

idx3=[];
for i=0:length(trainNum)-1
    idx3 = [idx3 (idx2 + i*(length(trainLabel)/length(trainNum)))];
end

validationSamp = trainSamp(:,:,:,idx3);
validationLab = trainLab(idx3);

trainSamp(:,:,:,idx3)=[];
```

```matlab
trainLab(idx3)=[];

numResponses = size(trainLab,1);
featureDimension = size(trainSamp,1);




layers = [
    imageInputLayer([dim dim 1])

    convolution2dLayer(2,2,'Padding','same') %Adding stride making
performance worst
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2) %Increasing stride rate improving
efficiency

    convolution2dLayer(2,2,'Padding','same')
    batchNormalizationLayer
    reluLayer
    %maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(4,2,'Padding','same')
    batchNormalizationLayer
    %maxPooling2dLayer(2,'Stride',2)
    reluLayer

    convolution2dLayer(4,4,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(8,2,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(8,4,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(8,5,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(16,4,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(16,8,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(16,16,'Padding','same')
    batchNormalizationLayer
    reluLayer
```

```matlab
    convolution2dLayer(32,16,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(32,32,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(64,32,'Padding','same')
    batchNormalizationLayer
    reluLayer

    softmaxLayer
    dropoutLayer(0.1)
    fullyConnectedLayer(1)
    regressionLayer
    ];


miniBatchSize  = 6; %Lower batch size means high number of iterations (lower
value is better)
validationFrequency = floor(numel(trainLab)/miniBatchSize);
options = trainingOptions('sgdm', ...
    'ValidationData',{validationSamp,validationLab}, ...
    'MiniBatchSize',miniBatchSize, ...
    'MaxEpochs',30, ...
    'Momentum',0.95, ...
    'ValidationPatience', 50, ...
    'InitialLearnRate',0.000001, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropFactor',0.1, ...
    'LearnRateDropPeriod',20, ...
    'Shuffle','every-epoch', ...
    'ValidationFrequency',validationFrequency, ...
    'Plots','training-progress', ...
    'Verbose',false);

        net = trainNetwork(trainSamp,trainLab,layers,options);


        FTestLabelsCNN = (predict(net,testSamp));
        PredictedTrainLabels = (predict(net,trainSampAll));

FerrCNN=(FTestLabelsCNN-testLabel)./testLabel*100;
%%------------------------------------------------------------------


%%--------------------------Plots with CNN ----------------------------

RelErrCNN = (FTestLabelsCNN-testLabel)./testLabel*100;

FerrCNN=(FTestLabelsCNN-testLabel)./testLabel*100;
FerrCNN = FerrCNN(1:Ds:length(FerrCNN));
```

```matlab
FigH = figure('Position', get(0, 'Screensize'));

clear Xfr XfrAct FerrCNN SD SDtot


for i=1:length(TuniqLabels2)
    X11 = FTestLabelsCNN(find(testTemps==TuniqLabels2(i)),:);
    X22 = testLabel(find(testTemps==TuniqLabels2(i)),:);




    for j=1:length(FuniqLabels2)

        if(~isempty(find(X22==FuniqLabels2(j))))
          ind123 = find(X22==FuniqLabels2(j));
          XfrAct(i,j) = mean(X22(ind123));
          Xfr(i,j) = mean(X11(ind123));
          FerrCNN(i,j) = (Xfr(i,j) - XfrAct(i,j))*100/XfrAct(i,j);
          fSDcnn(i,j) = std(X11(ind123));
          fNSTDcnn(i,j) = std(X11(ind123))*100./FuniqLabels2(j);
          fSDcnn_err(i,j) = std((X11(ind123)-
X22(ind123))*100./X22(ind123));
        end

        SDtot(j) =
std(FTestLabelsCNN(find(testLabel==FuniqLabels2(j)),:));
    end

end


symb = ['*' 'o' '+' 'v' 's' 'x' 'd' '^' 'p'];

for i=1:size(Xfr,1)
    p(i) =
plot(XfrAct(i,:),Xfr(i,:),symb(i),'linewidth',lineW,'MarkerSize',MarkerSze);
    hold on
%    xyerrorbar(XfrAct(i,:),Xfr(i,:), 0.3*fSDcnn(i,:), 1*fSDcnn(i,:));
%    hold on
end

plot(0:100,0:100,'k','linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Predicted mass flow rate (g/s)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6)],'T=17°C, RH=56%','T=19°C,
RH=54%','T=21°C, RH=52%','T=23°C, RH=50%','T=25°C, RH=49%','T=27°C,
RH=47%','Location','NorthEastOutside');
```

277

```matlab
lgd.FontSize = lgdSize;
title(lgd,'Ambient conditions')
set(gca,'FontSize',FontSze)
set(gca,'linewidth',lineW)
pbaspect([1 1 1])
set(gca,'XTick',0:5:60);
set(gca,'YTick',0:5:60);
axis([0 max(FuniqLabels2)+5 0 max(FuniqLabels2)+5])
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\cnn.tif','tif');




FigH = figure('Position', get(0, 'Screensize'));

for i=1:size(Xfr,1)
    p(i) =
plot(XfrAct(i,:),FerrCNN(i,:),symb(i),'linewidth',lineW,'MarkerSize',MarkerSz
e);
    hold on
end

plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow rate (g/s)','FontSize', FontSze)
ylabel('Relative error (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6)],'T=17°C, RH=56%','T=19°C,
RH=54%','T=21°C, RH=52%','T=23°C, RH=50%','T=25°C, RH=49%','T=27°C,
RH=47%','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Ambient conditions')
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'linewidth',lineW)
set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:5:500);
axis([0 max(FuniqLabels2)+5 min(FerrCNN(:))-2 max(FerrCNN(:))+2]);
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\cnnErr.tif','tif
');




FigH = figure('Position', get(0, 'Screensize'));
```

```matlab
for i=1:size(Xfr,1)
    p(i) = plot(XfrAct(i,:),
fNSTDcnn(i,:),symb(i),'linewidth',lineW,'MarkerSize',MarkerSze);
    hold on
end

plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual mass flow Rate (g/s)','FontSize', FontSze)
ylabel('Normalized standard deviation (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6)],'T=17°C, RH=56%','T=19°C,
RH=54%','T=21°C, RH=52%','T=23°C, RH=50%','T=25°C, RH=49%','T=27°C,
RH=47%','Location','NorthEastOutside');
lgd.FontSize = lgdSize;
title(lgd,'Ambient conditions')
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'linewidth',lineW)
set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:5:500);
axis([0 max(FuniqLabels2)+5 0 max(fNSTDcnn(:))+2]);
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\IEEE_TIM\Figures\cnnNSTD.tif','ti
f');

%%----------------------------------------------------------------
```

Appendix 5 Program for Coal/Biomass Ratio Measurement

```matlab
%************************************************************************
%Programmer's name: Faisal Abbas
%Date: 12/03/2021

%Purpose: This program takes the post-processed sensor data and applies the
machine learning models such as ANN, SVM and CNN for the measurement of coal
and biomass ratios with the coal/biomass/air mixture
%************************************************************************


load X:\PhD_DataSets\Flow_Meas_Coal_Biomass\AllFlowData_Coal_Biomass_RAW



%% -------Training and Testing of Classifier with ANN Coal/Biomass Ratio----

hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize);
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};
net.divideFcn = 'dividerand';
net.divideMode = 'sample';
net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 0/100;

% net.trainFcn = 'trainbr';  % Bayesian regularization
% net.trainFcn = 'traingdx';
net.trainFcn = 'trainlm';

net.trainParam.epochs=5000;

net.performFcn = 'mse';  % Mean squared error
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
  'plotregression', 'plotfit'};
[CnetANN1,tr] = train(net,trainFeatureANN',trainLabel');


% outputs = net(inputs);
% errors = gsubtract(targets,outputs);
% performance = perform(net,targets,outputs)

CTestLabelsANN = (CnetANN1(testFeatureANN'))';
CTrainLabelsANN = (CnetANN1(trainFeatureANN'))';

CerrANN=(CTestLabelsANN-testLabel)./testLabel*100;

testTrainLabel = [testLabel ; trainLabel];
CTestTrainLabelsANN = [CTestLabelsANN ; CTrainLabelsANN];
ZtestTrainLab = [ZtestLab ; ZtrainLab];
%%-----------------------------------------------------------------------
```

```matlab
%%-------------Plots with ANN Coal/Biomass Ratio---------------------------


FigH = figure('Position', get(0, 'Screensize'));

clear Xfr XfrAct CerrANN SD SDtot

CuniqLabels2 = unique(testTrainLabel);
FuniqLabels2 = unique(ZtestTrainLab);

for j=1:length(FuniqLabels2)
    X11 = CTestTrainLabelsANN(find(ZtestTrainLab==FuniqLabels2(j)),:);
    X22 = testTrainLabel(find(ZtestTrainLab==FuniqLabels2(j)),:);

 %   XfrAct(i,:)= CuniqLabels2;

  for i=1:length(CuniqLabels2)

           if(~isempty(find(X22==CuniqLabels2(i))))
             ind123 = find(X22==CuniqLabels2(i));
             XfrAct(i,j) = mean(X22(ind123));
             Xfr(i,j) = mean(X11(ind123));
                 if(XfrAct(i,j)==0)
                     XfrAct(i,j) = XfrAct(i,j)+5;
                     Xfr(i,j) = Xfr(i,j)+5;
                 end

             CerrANN(i,j) = (Xfr(i,j) - XfrAct(i,j))*100/XfrAct(i,j);
             cSDann(i,j) = std(X11(ind123));
             cNSTDann(i,j) = std(X11(ind123))*100./(CuniqLabels2(i)+1);
             cSDann_err(i,j) = std((X11(ind123)-
X22(ind123))*100./X22(ind123));

                % Total Concentration Measurement ---------------------


           end

             SDtot(i) =
std(CTestTrainLabelsANN(find(testTrainLabel==CuniqLabels2(i)),:));
    end

end

NRMSE(1,2) = (1/mean(trainLabel))*rms(CTrainLabelsANN-trainLabel)*100;
NRMSE(2,2) = (1/mean(testLabel))*rms(CTestLabelsANN-testLabel)*100;

nstdd = cNSTDann(:,2:2:11);
NSTD(1,2) = mean(nstdd(:));
nstdd = cNSTDann(:,1:2:11);
NSTD(2,2) = mean(nstdd(:));
```

281

```matlab
symb = ['*' ; 'o' ; '+' ; 'v' ; 's' ; 'x' ; 'd' ; '^' ; 'p' ; '<' ; '>'];
ratio =
[{'0/100','10/90','20/80','30/70','40/60','50/50','60/40','70/30','80/20','90
/10','100/0'}];


for j=1:size(Xfr,2)
    for i=1:size(Xfr,1)
        if(XfrAct(i,j)~=0)
                p(j) =
plot(XfrAct(i,j),Xfr(i,j),symb(j,:),'linewidth',lineW,'MarkerSize',MarkerSze)
;
                hold on
        end
    end
end

plot(0:110,0:110,'k','linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual coal/biomass ratio (%)','FontSize', FontSze)
ylabel('Predicted coal/biomass ratio (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9),p(10),p(11)],'1.67',
'2.50','4.50','6.25','9.50','12.50','17.50','22.67','27.00','32.33','36.50','
Location','NorthEastOutside');
title(lgd,'Solids flow rate (ml/s)')
lgd.FontSize = lgdSize;
title(lgd,'Solids flow rate (ml/s)')
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
set(gca,'xtick',[CuniqLabels],'xticklabel',ratio)
set(gca,'ytick',[CuniqLabels],'yticklabel',ratio)
pbaspect([1 1 1])
% set(gca,'XTick',0:5:60);
% set(gca,'YTick',0:5:60);
axis([0 max(CuniqLabels)+5 0 max(CuniqLabels)+5])
xtickangle(45)
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\FERIA\Figures\ann.tif','tif');




FigH = figure('Position', get(0, 'Screensize'));

for j=1:size(Xfr,2)
      p(j) =
plot(CerrANN(:,j),symb(j,:),'linewidth',lineW,'MarkerSize',MarkerSze);
      hold on
end
```

```matlab
plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual coal/biomass ratio (%)','FontSize', FontSze)
ylabel('Relative error (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9),p(10),p(11)],'1.67',
'2.50','4.50','6.25','9.50','12.50','17.50','22.67','27.00','32.33','36.50','
Location','NorthEastOutside');
title(lgd,'Solids flow rate (ml/s)')
lgd.FontSize = lgdSize;
title(lgd,'Solids flow rate (ml/s)')
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
set(gca,'xtick',[1:length(CuniqLabels)],'xticklabel',ratio)
% set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:2:500);
axis([0 length(CuniqLabels)+1 min(CerrANN(:))-2 max(CerrANN(:))+2]);
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\FERIA\Figures\annErr.tif','tif');




FigH = figure('Position', get(0, 'Screensize'));

for j=1:size(Xfr,2)
    p(j) =
plot(cNSTDann(:,j),symb(j,:),'linewidth',lineW,'MarkerSize',MarkerSze);
    hold on
end

plot([0:max(CuniqLabels2)+20],zeros(1,length([0:max(CuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Coal/Biomass ratio','FontSize', FontSze)
ylabel('Normalized standard deviation (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9),p(10),p(11)],'1.67',
'2.50','4.50','6.25','9.50','12.50','17.50','22.67','27.00','32.33','36.50','
Location','NorthEastOutside');
title(lgd,'Solids flow rate (ml/s)')
lgd.FontSize = lgdSize;
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
set(gca,'xtick',[1:length(CuniqLabels)],'xticklabel',ratio)
```

```matlab
% set(gca,'YTick',-500:2:500);
axis([0 length(CuniqLabels)+1 0 max(cNSTDann(:))+2])
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\FERIA\Figures\annNSTD.tif','tif')
;


%%-------------------------------------------------------------



%% ------Training and Testing of Classifier with SVM Coal/Biomass Ratio------

Fmdl =
fitrsvm(trainFeatureSVM,trainLabel,'KernelFunction','rbf','KernelScale',20,'S
tandardize',true,'IterationLimit',10000, 'DeltaGradientTolerance',1e-6,
'GapTolerance',1e-6,'Epsilon',0.05)%'PolynomialOrder',3);
CTestLabelsSVM=predict(Fmdl,testFeatureSVM);

CTrainLabelsSVM = predict(Fmdl,trainFeatureSVM);

CerrSVM=(CTestLabelsSVM-testLabel)./testLabel*100;
Diff = CTestLabelsSVM-testLabel;
CerrSVM2=(CTrainLabelsSVM-trainLabel)./trainLabel*100;
Diff2 = CTrainLabelsSVM-trainLabel;

testTrainLabel = [testLabel ; trainLabel];
CTestTrainLabelsSVM = [CTestLabelsSVM ; CTrainLabelsSVM];
ZtestTrainLab = [ZtestLab ; ZtrainLab];
%%-----------------------------------------------------------------------



%%-----------------Plots with SVM Coal/Biomass Ratio-------------------


FigH = figure('Position', get(0, 'Screensize'));

clear Xfr XfrAct CerrSVM SD SDtot

CuniqLabels2 = unique(testTrainLabel);
FuniqLabels2 = unique(ZtestTrainLab);

for j=1:length(FuniqLabels2)
    X11 = CTestTrainLabelsSVM(find(ZtestTrainLab==FuniqLabels2(j)),:);
    X22 = testTrainLabel(find(ZtestTrainLab==FuniqLabels2(j)),:);

 %   XfrAct(i,:)= CuniqLabels2;

  for i=1:length(CuniqLabels2)

        if(~isempty(find(X22==CuniqLabels2(i))))
```

```matlab
            ind123 = find(X22==CuniqLabels2(i));
            XfrAct(i,j) = mean(X22(ind123));
            Xfr(i,j) = mean(X11(ind123));
                if(XfrAct(i,j)==0)
                    XfrAct(i,j) = XfrAct(i,j)+5;
                    Xfr(i,j) = Xfr(i,j)+5;
                end

            CerrSVM(i,j) = (Xfr(i,j) - XfrAct(i,j))*100/XfrAct(i,j);
            cSDsvm(i,j) = std(X11(ind123));
            cNSTDsvm(i,j) = std(X11(ind123))*100./(CuniqLabels2(i)+1);
            cSDsvm_err(i,j) = std((X11(ind123)-
X22(ind123))*100./X22(ind123));


% Total Concentration Measurement ----------------------


        end

            SDtot(i) =
std(CTestTrainLabelsSVM(find(testTrainLabel==CuniqLabels2(i)),:));
    end

end

NRMSE(1,2) = (1/mean(trainLabel))*rms(CTrainLabelsSVM-trainLabel)*100;
NRMSE(2,2) = (1/mean(testLabel))*rms(CTestLabelsSVM-testLabel)*100;

nstdd = cNSTDsvm(:,2:2:11);
NSTD(1,2) = mean(nstdd(:));
nstdd = cNSTDsvm(:,1:2:11);
NSTD(2,2) = mean(nstdd(:));


symb = ['*' ; 'o' ; '+' ; 'v' ; 's' ; 'x' ; 'd' ; '^' ; 'p' ; '<' ; '>'];
ratio =
[{'0/100','10/90','20/80','30/70','40/60','50/50','60/40','70/30','80/20','90
/10','100/0'}];

for j=1:size(Xfr,2)
    for i=1:size(Xfr,1)
        if(XfrAct(i,j)~=0)
            p(j) =
plot(XfrAct(i,j),Xfr(i,j),symb(j,:),'linewidth',lineW,'MarkerSize',MarkerSze)
;
            hold on
        end
    end
end

plot(0:110,0:110,'k','linewidth',lineW,'MarkerSize',MarkerSze);
hold on
```

```matlab
xlabel('Actual coal/biomass ratio (%)','FontSize', FontSze)
ylabel('Predicted coal/biomass ratio (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9),p(10),p(11)],'1.67',
'2.50','4.50','6.25','9.50','12.50','17.50','22.67','27.00','32.33','36.50','
Location','NorthEastOutside');
title(lgd,'Solids flow rate (ml/s)')
lgd.FontSize = lgdSize;
title(lgd,'Solids flow rate (ml/s)')
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
set(gca,'xtick',[CuniqLabels],'xticklabel',ratio)
set(gca,'ytick',[CuniqLabels],'yticklabel',ratio)
pbaspect([1 1 1])
% set(gca,'XTick',0:5:60);
% set(gca,'YTick',0:5:60);
axis([0 max(CuniqLabels)+5 0 max(CuniqLabels)+5])
xtickangle(45)
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\FERIA\Figures\svm.tif','tif');




FigH = figure('Position', get(0, 'Screensize'));

for j=1:size(Xfr,2)
      p(j) =
plot(CerrSVM(:,j),symb(j,:),'linewidth',lineW,'MarkerSize',MarkerSze);
      hold on
end


plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual coal/biomass ratio (%)','FontSize', FontSze)
ylabel('Relative error (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9),p(10),p(11)],'1.67',
'2.50','4.50','6.25','9.50','12.50','17.50','22.67','27.00','32.33','36.50','
Location','NorthEastOutside');
title(lgd,'Solids flow rate (ml/s)')
lgd.FontSize = lgdSize;
title(lgd,'Solids flow rate (ml/s)')
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
set(gca,'xtick',[1:length(CuniqLabels)],'xticklabel',ratio)
% set(gca,'XTick',0:5:60);
```

```matlab
set(gca,'YTick',-500:2:500);
axis([0 length(CuniqLabels)+1 min(CerrSVM(:))-2 max(CerrSVM(:))+2]);
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\FERIA\Figures\svmErr.tif','tif');




FigH = figure('Position', get(0, 'Screensize'));

for j=1:size(Xfr,2)
     p(j) =
plot(cNSTDsvm(:,j),symb(j,:),'linewidth',lineW,'MarkerSize',MarkerSze);
     hold on
end

plot([0:max(CuniqLabels2)+20],zeros(1,length([0:max(CuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Coal/Biomass ratio','FontSize', FontSze)
ylabel('Normalized standard deviation (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9),p(10),p(11)],'1.67',
'2.50','4.50','6.25','9.50','12.50','17.50','22.67','27.00','32.33','36.50','
Location','NorthEastOutside');
title(lgd,'Solids flow rate (ml/s)')
lgd.FontSize = lgdSize;
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
set(gca,'xtick',[1:length(CuniqLabels)],'xticklabel',ratio)
% set(gca,'YTick',-500:2:500);
axis([0 length(CuniqLabels)+1 0 max(cNSTDsvm(:))+2])
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\FERIA\Figures\svmNSTD.tif','tif')
;

%%----------------------------------------------------------------


%% ------------- Dataset formating for CNN ------------------------
Fi =100; %Factor which accounts for image dimension
Feature=zeros((size(X,1)/Fi) , Fi, NS);
Flabels = zeros((size(X,1)/Fi),1);
for j=1:(size(X,1)/Fi)

    XX = X(((j-1)*Fi+1):((j-1)*Fi+Fi) , :);
    Feature(j,:,:) = XX;
```

287

```matlab
        Flabels(j,1) = Y((j-1)*Fi+1);
        CtrainLab(j,1) = Z((j-1)*Fi+1);
    end

CuniqLabels = unique(Z);

clear XX
%% --------------------------------------------------------------------



%% ---------Data Segregation Based on Output Labels--------------------

trainNum = CuniqLabels(2:2:length(CuniqLabels));
testNum = CuniqLabels(1:2:length(CuniqLabels));

TrainCases = length(trainNum);
TestCases = length(testNum);

trainFeature=[];
CtrainLabel=[];
ZtrainLab = [];
for i=1:length(trainNum)
    trainFeature=[trainFeature ; Feature(find(CtrainLab==trainNum(i)),:,:)];
    CtrainLabel = [CtrainLabel ; CtrainLab(find(CtrainLab==trainNum(i)))];
    ZtrainLab = [ZtrainLab ; Flabels(find(CtrainLab==trainNum(i)))];
end


testFeature=[];
CtestLabel=[];
ZtestLab=[];
for i=1:length(testNum)
    testFeature=[testFeature ; Feature(find(CtrainLab==testNum(i)),:,:)];
    CtestLabel = [CtestLabel ; CtrainLab(find(CtrainLab==testNum(i)))];
    ZtestLab = [ZtestLab ; Flabels(find(CtrainLab==testNum(i)))];
end


% CtrainLabel = (trainLabel*0.000001) ./ ((3.14*(0.05/2)^2)*22)*100;
% CtestLabel = (testLabel*0.000001) ./ ((3.14*(0.05/2)^2)*22)*100;


% %%-------------------------------------------------



%% ---------Training and Testing of Classifier with Method 1------

validSampRatio = 0.25; % percentage of training data that can be used as
validation data
trainSamp = permute(trainFeature,[2 3 1]);
```

```matlab
trainSamp =
reshape(trainSamp,size(trainSamp,1),size(trainSamp,2),1,size(trainSamp,3));
FtrainSampAll = trainSamp;
trainLab = CtrainLabel;

FtestSamp = permute(testFeature,[2 3 1]);
FtestSamp =
reshape(FtestSamp,size(FtestSamp,1),size(FtestSamp,2),1,size(FtestSamp,3));


idx3=[];
caseLen = 0;
for i=1:length(trainNum)
    idx2 =
randperm(length(find(CtrainLabel==trainNum(i))),floor(validSampRatio*length(f
ind(CtrainLabel==trainNum(i)))));
    idx3 = [idx3 (idx2 + caseLen)];
    caseLen = caseLen + length(find(CtrainLabel==trainNum(i)));
end

validationSamp = trainSamp(:,:,:,idx3);
validationLab = trainLab(idx3);

trainSamp(:,:,:,idx3)=[];
trainLab(idx3)=[];
% ZtrainLab(idx3)=[];

numResponses = size(trainLab,1);
featureDimension = size(trainSamp,1);



layers = [      % Tuning this one, decreased by 00
    imageInputLayer([Fi (NS) 1])

    convolution2dLayer(4,2,'Padding','same') %Adding stride making
performance worst
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2) %Increasing stride rate improving
efficiency

    convolution2dLayer(4,2,'Padding','same')
    batchNormalizationLayer
    reluLayer
    %maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(4,2,'Padding','same')
    batchNormalizationLayer
    reluLayer
    %maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(8,8,'Padding','same')
    batchNormalizationLayer
```

289

```matlab
    %maxPooling2dLayer(2,'Stride',2)
    reluLayer

    convolution2dLayer(8,4,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(8,2,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(16,8,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(16,4,'Padding','same')
    batchNormalizationLayer
    reluLayer

    convolution2dLayer(16,2,'Padding','same')
    batchNormalizationLayer
    reluLayer

%     convolution2dLayer(32,8,'Padding','same')
%     batchNormalizationLayer
%     reluLayer
%
%     convolution2dLayer(64,32,'Padding','same')
%     batchNormalizationLayer
%     reluLayer
%
%     convolution2dLayer(64,16,'Padding','same')
%     batchNormalizationLayer
%     reluLayer
%
%     convolution2dLayer(64,8,'Padding','same')
%     batchNormalizationLayer
%     reluLayer
%
%     convolution2dLayer(128,32,'Padding','same')
%     batchNormalizationLayer
%     reluLayer


    softmaxLayer
    dropoutLayer(0.1)
    fullyConnectedLayer(1)
    regressionLayer];


miniBatchSize  = 6; %Lower batch size means high number of iterations (lower
value is better)
validationFrequency = floor(numel(trainLab)/miniBatchSize);
options = trainingOptions('sgdm', ...
```

```matlab
    'ValidationData',{validationSamp,validationLab}, ... % Tuning this one,
X100
    'MiniBatchSize',miniBatchSize, ...
    'MaxEpochs',30, ...
    'Momentum',0.95, ...
    'ValidationPatience', 20, ...
    'InitialLearnRate',0.000001, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropFactor',0.001, ...
    'LearnRateDropPeriod',10, ...
    'Shuffle','every-epoch', ...
    'ValidationFrequency',validationFrequency, ...
    'Plots','training-progress', ...
    'Verbose',false);

        Fnet = trainNetwork(FtrainSampAll,CtrainLabel,layers,options);


        CTestLabelsCNN = (predict(Fnet,FtestSamp));
        CTrainLabelsCNN = (predict(Fnet,FtrainSampAll));



CtestTrainLabel = [CtestLabel ; CtrainLabel];
CTestTrainLabelsCNN = [CTestLabelsCNN ; CTrainLabelsCNN];
ZtestTrainLab = [ZtestLab ; ZtrainLab];


%%----------------------------------------------------------------------


%%-----------Plots with CNN Coal/Biomass Ratio------------------

clear Xfr XfrAct CerrCNN SD SDtot

CuniqLabels2 = unique(CtestTrainLabel);
FuniqLabels2 = unique(ZtestTrainLab);

for j=1:length(FuniqLabels2)
    X11 = CTestTrainLabelsCNN(find(ZtestTrainLab==FuniqLabels2(j)),:);
    X22 = CtestTrainLabel(find(ZtestTrainLab==FuniqLabels2(j)),:);

 %   XfrAct(i,:)= CuniqLabels2;

  for i=1:length(CuniqLabels2)

        if(~isempty(find(X22==CuniqLabels2(i))))
          ind123 = find(X22==CuniqLabels2(i));
          XfrAct(i,j) = mean(X22(ind123));
          Xfr(i,j) = mean(X11(ind123));
            if(XfrAct(i,j)==0)
                XfrAct(i,j) = XfrAct(i,j)+5;
                Xfr(i,j) = Xfr(i,j)+5;
            end

        CerrCNN(i,j) = (Xfr(i,j) - XfrAct(i,j))*100/XfrAct(i,j);
```

291

```matlab
            cSDcnn(i,j) = std(X11(ind123));
            cNSTDcnn(i,j) = std(X11(ind123))*100./(CuniqLabels2(i)+1);
            cSDcnn_err(i,j) = std((X11(ind123)-
X22(ind123))*100./X22(ind123));

            % Total Concentration Measurement ----------------------


        end

            SDtot(i) =
std(CTestTrainLabelsCNN(find(CtestTrainLabel==CuniqLabels2(i)),:));
    end

end

NRMSE(1,2) = (1/mean(CtrainLabel))*rms(CTrainLabelsCNN-CtrainLabel)*100;
NRMSE(2,2) = (1/mean(CtestLabel))*rms(CTestLabelsCNN-CtestLabel)*100;

nstdd = cNSTDcnn(:,2:2:11);
NSTD(1,2) = mean(nstdd(:));
nstdd = cNSTDcnn(:,1:2:11);
NSTD(2,2) = mean(nstdd(:));


symb = ['*' ; 'o' ; '+' ; 'v' ; 's' ; 'x' ; 'd' ; '^' ; 'p' ; '<' ; '>'];
ratio =
[{'0/100','10/90','20/80','30/70','40/60','50/50','60/40','70/30','80/20','90
/10','100/0'}];

FigH = figure('Position', get(0, 'Screensize'));

for j=1:size(Xfr,2)
    for i=1:size(Xfr,1)
        if(XfrAct(i,j)~=0)
            p(j) =
plot(XfrAct(i,j),Xfr(i,j),symb(j,:),'linewidth',lineW,'MarkerSize',MarkerSze)
;
            hold on
        end
    end
end

plot(0:110,0:110,'k','linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual coal/biomass ratio (%)','FontSize', FontSze)
ylabel('Predicted coal/biomass ratio (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9),p(10),p(11)],'1.67',
'2.50','4.50','6.25','9.50','12.50','17.50','22.67','27.00','32.33','36.50','
Location','NorthEastOutside');
title(lgd,'Solids flow rate (ml/s)')
lgd.FontSize = lgdSize;
```

```matlab
title(lgd,'Solids flow rate (ml/s)')
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
set(gca,'xtick',[CuniqLabels],'xticklabel',ratio)
set(gca,'ytick',[CuniqLabels],'yticklabel',ratio)
pbaspect([1 1 1])
% set(gca,'XTick',0:5:60);
% set(gca,'YTick',0:5:60);
axis([0 max(CuniqLabels)+5 0 max(CuniqLabels)+5])
xtickangle(45)
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\FERIA\Figures\cnn.tif','tif');




FigH = figure('Position', get(0, 'Screensize'));

for j=1:size(Xfr,2)
        p(j) =
plot(CerrCNN(:,j),symb(j,:),'linewidth',lineW,'MarkerSize',MarkerSze);
        hold on
end


plot([0:max(FuniqLabels2)+20],zeros(1,length([0:max(FuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Actual coal/biomass ratio (%)','FontSize', FontSze)
ylabel('Relative error (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9),p(10),p(11)],'1.67',
'2.50','4.50','6.25','9.50','12.50','17.50','22.67','27.00','32.33','36.50','
Location','NorthEastOutside');
title(lgd,'Solids flow rate (ml/s)')
lgd.FontSize = lgdSize;
title(lgd,'Solids flow rate (ml/s)')
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
set(gca,'xtick',[1:length(CuniqLabels)],'xticklabel',ratio)
% set(gca,'XTick',0:5:60);
set(gca,'YTick',-500:2:500);
axis([0 length(CuniqLabels)+1 min(CerrCNN(:))-2 max(CerrCNN(:))+2]);
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\FERIA\Figures\cnnErr.tif','tif');
```

```matlab
FigH = figure('Position', get(0, 'Screensize'));

for j=1:size(Xfr,2)
    p(j) =
plot(cNSTDcnn(:,j),symb(j,:),'linewidth',lineW,'MarkerSize',MarkerSze);
    hold on
end

plot([0:max(CuniqLabels2)+20],zeros(1,length([0:max(CuniqLabels2)+20])),'k','
linewidth',lineW,'MarkerSize',MarkerSze);
hold on

xlabel('Coal/Biomass ratio','FontSize', FontSze)
ylabel('Normalized standard deviation (%)','FontSize', FontSze)
%title('CNN','FontSize', 22)
lgd=legend([p(1),p(2),p(3),p(4),p(5),p(6),p(7),p(8),p(9),p(10),p(11)],'1.67',
'2.50','4.50','6.25','9.50','12.50','17.50','22.67','27.00','32.33','36.50','
Location','NorthEastOutside');
title(lgd,'Solids flow rate (ml/s)')
lgd.FontSize = lgdSize;
% pbaspect([1.5 1 1])
set(gca,'FontSize',FontSze)
set(gca,'fontname',FontStyle)
set(gca,'linewidth',lineW)
set(gca,'xtick',[1:length(CuniqLabels)],'xticklabel',ratio)
% set(gca,'YTick',-500:2:500);
axis([0 length(CuniqLabels)+1 0 max(cNSTDcnn(:))+2])
hold off

saveas(FigH,
'C:\Users\fa326\OneDrive\My_Research_Papers\FERIA\Figures\cnnNSTD.tif','tif')
;

%%----------------------------------------------------
```

Publications and Dissemination

1. **F. Abbas**, L. Wang, and Y. Yan, "Mass Flow Rate Measurement of Pneumatically Conveyed Solids Through Multi-modal Sensing and Data Driven Modelling", IEEE Transactions on Instrumentation and Measurements, vol. 70, pp. 1-16, Aug. 2021

2. **F. Abbas**, L. Wang, and Y. Yan, "Mass Flow Rate Measurement of Solids in a Pneumatic Conveying Pipeline in Different Orientations", Measurement: Sensor, vol. 10-12, pp. 1-12, Nov. 2020.

3. **F. Abbas**, Y. Yan, and L. Wang, "Mass flow measurement of pneumatically conveyed solids through multi-modal sensing and machine learning," 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Dubrovnik, Croatia, 25-28 May 2020.

4. **F. Abbas**, L. Wang, and Y. Yan, "Concentration Measurement of Pulverized Coal and Biomass in a Pneumatic Conveying Pipeline Through Multi-Modal Sensing and Machine Learning", The 1st FERIA Conference, the European Conference on Fuel and Energy Research and Its Applications, Nottingham, UK, 6-8 September 2021