

MSc Computer Science
Final Project

Fluid properties estimation and pressure drop sensing based on Coriolis flow sensor using machine learning

Romas Zubavičius

Supervisor:

dr. Mannes Poel (Chair / Supervisor)

dr.ir. Dennis Alveringh (Supervisor)

dr.ir. Jarno Groenesteijn (Committee member)

December, 2023

Department of Computer Science
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

Abstract

All forms of sensing devices have optimal performance conditions to achieve accuracy and reliability. Environmental conditions such as temperature affect the sensing elements of sensors, which results in a drop in accuracy due to non-ideal effects. Their effects are difficult to model due to the simplification of physics.

In this research, we focus on Coriolis Mass Flow Sensors (CMFS), which are microfabricated mass flow sensors. They are able to measure the true mass flow by detecting the inertial effects caused by a fluid flowing through a vibrating channel. The aim is to investigate the viability of machine learning to combine various sensing elements of a device to counteract the non-ideal effects.

We present statistical machine learning methods applied on a CMFS, to estimate mass flow, temperature, pressure drop, viscosity, and density as well as predict the measured fluid based on its electrical readings. The chip (1.2 cm^2) has been exposed to different combinations of temperatures, flows, and pressure for four different liquids (ethanol, water, isopropanol, acetone), and one gas (nitrogen). The sensing elements were sampled using a Data acquisition (DAQ) system at 250 kHz, and features were computed using a Fast Fourier Transform (FFT) based on its resonance frequency.

This thesis covers the classical machine learning and deep learning approaches. The classical machine learning approach covers linear regression, support vector regression with a non-linear kernel, and Gaussian process regression to estimate the fluid-state properties. Additionally, Naïve Bayes, Gaussian process and deep learning approach classifiers are employed to predict which fluid was measured based on its electrical readings.

The results for density and viscosity estimations show less fluid-state dependence using GPR on trained fluids, achieving an average error of 0.132 kg/m^3 (0.01%) and 0.62% for density and viscosity respectively, similarly, mass flow estimation combining sensing element features by polynomial features using linear regression achieves 2% accuracy, which is better by a factor of > 2 compared to its counterpart by directly estimating using phase difference, while even further improved by a non-linear method, improving performance by a factor of > 4 .

Keywords: microfluidics, Coriolis, mass flow sensors, machine learning, deep learning, regression, classification

Contents

1	Introduction and motivation	4
1.1	Outline	5
2	Background	6
2.1	Fluid dynamics	6
2.1.1	Density	6
2.1.2	Viscosity	7
2.2	Coriolis	8
2.2.1	Viscosity determination	11
2.2.2	Pressure sensing	11
2.3	Fourier Transform	11
2.3.1	Magnitude	13
2.3.2	Phase	13
2.4	Correlation and Cross-Correlation	14
2.5	Kernel method	14
2.5.1	Kernel trick	14
2.5.2	Radial Basis Function	15
2.6	Regression	15
2.6.1	Linear regression	16
2.6.2	Support Vector Regression	16
2.6.3	Gaussian Process Regressor	18
2.7	Classification	19
2.7.1	Naive Bayes	19
2.7.2	Gaussian Process Classifier	19
2.8	Deep learning fundamentals	20
2.8.1	Convolution	22
2.8.2	Gated Recurrent Unit	24
2.9	Model comparisons	25
2.10	Shapley Additive Explanations	25
2.11	Accuracy	27
2.11.1	R-squared	27
2.11.2	Mass flow	27
2.11.3	Temperature & Density	28
2.11.4	Pressure drop & Viscosity	28
2.11.5	Classification	28
3	Related work	29

4	Methodology	32
4.1	Measurement setup	32
4.1.1	Properties	33
4.2	Measurements	33
4.3	Feature extraction and analysis	34
4.4	Ground-truth values	35
4.4.1	Temperature label	35
4.4.2	Pressure drop label	35
4.4.3	Density and viscosity labels	36
4.5	Machine learning	36
4.5.1	Feature selection & processing	36
4.5.2	Data splitting	37
4.5.3	Data normalization	37
4.5.4	Validation	37
4.6	Deep learning	37
5	Data-Driven Device Characterization	40
5.1	Measurements	40
5.1.1	Labels	40
5.1.2	Raw data	41
5.2	Feature analysis	41
5.2.1	Frequency	42
5.2.2	Frequency and density	44
5.2.3	Phase difference	44
5.2.4	Magnitude	47
5.2.5	Offset	49
5.2.6	Pressure sensor	50
6	Fluid characteristics estimation	51
6.1	Mass flow estimation	51
6.1.1	Linear regression	51
6.1.2	Non-linear regression	55
6.2	Temperature	57
6.2.1	Linear models	57
6.2.2	Non-Linear models	58
6.3	Pressure drop	59
6.3.1	Linear models	59
6.3.2	Non-Linear models	61
6.4	Viscosity	62
6.4.1	Linear models	62
6.4.2	Non-Linear models	64
6.5	Density	69
6.5.1	Linear models	69
6.5.2	Non-Linear models	71
6.6	Fluid type classification	75
6.6.1	(Naive) Bayes	75
6.6.2	Gaussian Process	76
6.6.3	DeepSense	78
7	Discussion	82

8 Conclusion	84
8.1 Research sub-question 1	84
8.2 Research sub-question 2	85
8.3 Research sub-question 3	85
8.3.1 Main research question	85
A Appendix	86

Chapter 1

Introduction and motivation

Nanotechnology is one of the most promising technologies of the past decade [1]. Regular daily life is surrounded by microelectronics, whether it is phones or cars, these systems contain various micro-scale devices e.g. accelerometers, gyroscopes, etc. These sensors are typically MicroElectroMechanical Systems (MEMS) and generally contain mechanical, optical or fluidic functional structures. For readout, these devices contain electrical components such as resistors and capacitors. MEMS are commonly made of silicon via front-end processes in the cleanroom. With the advancement in nanotechnology and miniaturization of electronic devices, many of these micro/nanodevices are being used in different sub-fields.

In the context of microfluidic flow sensors, they control and measure accurately very low flows, e.g. few micro or nano litres per minute [2]. This allows to study of fluids at a micro-scale, which is useful for applications in pharmaceutical, biomedical and many other fields [3, 4, 5]. It has been shown, that the thermal sensors are capable of measuring up to a few nl/min, however, the general problem is, that the measurements are highly dependent on liquid properties, and the sensors would need to be calibrated per liquid [6]. Another type of microfluidic sensor is called a Coriolis Mass Flow Sensor (CMFS). These sensors utilise the Coriolis force principle, in which the inertia caused by the fluid causes the sensor tubes to swing in proportion to the mass flow independent of liquid properties. This is further explained in Section 2.2. These devices' weakness is the resolution compared to the thermal sensors [7]. Alveringh et al. [8] have demonstrated that it can be improved with a factor of 50 before it reaches the fundamental limit.

In theory, Coriolis force returns a direct measure of mass flow independent of fluid properties. In practice, this is not the case. The use of CMFS to determine mass flow can introduce errors due to the simplification of underlying physics which affects the mechanics of the device. This is highlighted in recent research which shows microfabricated CMFS with integrated pressure sensors (upstream and downstream) to estimate viscosity via the Hagen-Poiseuille, in which the viscosity error ranges from 0-12% depending on mass flow rate and pressure [9]. Moreover, the material properties of the sensing elements are affected by pressure and temperature [10, 11]. These relationships are difficult to accurately capture with the conventional methods mentioned in Section 2.2. Thus the goal is to characterize the device, specifically MEMS-based microfabricated CMFS, in terms of what functionality is it able to produce by combining sensing components of the device, and whether is it able to reach the performance of an already established device on the market.

This project focuses on exploring the limitations, additional functionality, and improved performance of microfabricated Coriolis sensors via a data-driven approach. The main research question for this project is:

*How **viable** is a machine learning application on MEMS-based Coriolis flow sensors to estimate mass flow, temperature, pressure, viscosity and predict the type of fluid?*

where **viable** is in terms of reaching similar levels of performance (i.e. accuracy) as a high-performance device on the market, thus creating a societal impact with the new presented methods. To answer this main research question, there are several sub-questions:

- To what extent in terms of accuracy ¹, linear and non-linear models are able to combine capacitive and resistive signals measured from Coriolis Mass Flow Sensor to estimate mass flow, pressure, density, temperature and viscosity?
- To what extent can different fluids be distinguished from each other using measured sensing elements at different measurement parameters² using probabilistic models?
- Is one of the state-of-the-art methods (DeepSense) able to capture the non-linearity's better compared to the linear/non-linear models?

The focus is the data part from the measured sensor, specifically determining the characteristics of the measured fluid, such as density, and viscosity, and estimating mass flow, pressure, and temperature. Designing and manufacturing the device according to the results is out of the scope of this project.

1.1 Outline

Chapter 2 covers the background necessary to follow the thesis. The first part covers the basic principles of fluid dynamics, the acquisition of theoretical fluid properties which depend on pressure and temperature, as well as covering microfabricated CMFS. Later sections cover the basic theory of used machine-learning methods, as well as definitions of used accuracy measures for individual estimation.

Chapter 3 reviews the current state of machine learning applications in microfluidics, fluid characteristics estimations, as well as deep learning signal processing implementations.

Chapter 4 explains the methodology used, covering the measurement setup and its properties, as well as the type of produced signals by the setup. The latter parts cover the methodology for feature extraction and processing, in addition to acquiring additional labels.

Chapter 5 dives into Data-Driven Device Characterization, analyzing and understanding the measured signals of the CMFS device and its response to fluid properties. This chapter acts as a data analysis as a pre-requisite in applying machine learning methods.

Chapter 6 covers in-depth machine learning applications on measured signals to estimate fluid characteristics, each section containing an estimation of the individual target in question - Mass flow, temperature, pressure, viscosity, density and determining what fluid was measured.

Chapter 7 discusses the outcome, its limitations and drawbacks.

Chapter 8 concisely answers research questions.

¹Accuracy in terms of IEC 60601-2-24 standard defined in background section

²At varying mass flow, temperature and pressure

Chapter 2

Background

This chapter covers the background knowledge necessary to follow the thesis.

Section 2.1 covers basic principles of fluids dynamics, with emphasis on the physical properties of fluids we want to estimate, e.g. density, viscosity

Section 2.2 covers the physics fundamentals of Coriolis-based microfluidic devices, as well as the integrated pressure sensor.

Sections 2.3 - 2.6 contain techniques and their theory which is used to process data signal readings into a different representation form.

Section 2.11 contains quantitative evaluation metrics used to assess the accuracy of estimations with its corresponding model.

2.1 Fluid dynamics

Since this work aims to estimate the physical properties of fluids, some basic fluid physics is discussed. This section covers two principles of fluid dynamics, namely fluid density, and how to estimate theoretical values with temperature dependence, as well as fluid viscosity, and its theoretical value estimation.

2.1.1 Density

Density is one of the physical properties of a fluid which will be estimated using CMFS sensor signals.

It is defined as *the mass contained in a unit volume* [12]. Mathematically speaking, it is defined as

$$\rho = \frac{m}{V} \quad (2.1)$$

where m is the mass, and V is the volume, with ρ being the density, usually given in kg/m^3 . It is used as a fluid property which characterizes mass in a fluidic system. It varies between fluids, as well as depending on the state of the fluid, such as temperature and pressure. Liquids are nearly incompressible, and generally, pressure dependence is neglected, while temperature dependence is more significant. In contrast to gasses, which are more compressible, hence their density is more sensitive to pressures.

Temperature dependence for fluid density is called thermal expansion, where matter changes its volume as a response to temperature. To describe this phenomenon, there is a coefficient of thermal expansion, which relates the change of volume with respect to the change in temperature. In the general case, it is defined as

$$\alpha_v = \frac{1}{V} \left(\frac{\partial V}{\partial T} \right)_p \quad (2.2)$$

This assumes constant pressure, indicated by subscript p. To simplify the relation, we will use [13]:

$$B_{iq} = a(1 - T/T_c)^c \quad (2.3)$$

Where a and c are regression coefficients, unique for each chemical compound, T is the temperature, and T_c is the critical temperature, such as a condition where liquid and vapour can coexist. This B_{iq} coefficient is denoted as the thermal expansion coefficient which depends on temperature, with change per °C. Gasses depend much less on temperature, however, they are much more compressible. Dependence on pressure is similarly defined - as temperature changes, volume as a response to pressure changes, and is defined as:

$$\beta_T = -\frac{1}{V} \left(\frac{\partial V}{\partial p} \right)_T \quad (2.4)$$

Where p is the pressure, the temperature is assumed to be constant. Similarly to temperature, in this case, the compressibility of liquids is much smaller in comparison to gasses. Calculation of compressibility factor of gasses is calculated with Z-Factor [13]. However, due to the nature of the Coriolis device, i.e., the spring constant is dependent on pressure, further characterization is out of scope.

To acquire theoretical density values for fluids, with the temperature dependence, they can be acquired via an exponential equation [14].

$$\rho_L = AB^{-(1-T/T_c)^n} \quad (2.5)$$

where A, B, T_c and n are liquid parameters, unique for each chemical compound. Using these parameters, the density values are acquired over the temperature range T . The drawback of this acquisition is the inability to include pressure dependence and similarly, does not work for gasses. Instead of this regression formula, the theoretical values are obtained from [15] with the same principle, which generates density values over a range of temperature and pressure. These values are based on [16, 17, 18, 19, 20, 21, 22], and the mentioned accuracy typically is $< 1\%$, but often better than 0.1%

2.1.2 Viscosity

Viscosity is another fluid property which will be estimated using CMFS signals.

Dynamic viscosity is the resistance to deformation. Newton's law of viscosity defines the relationship between shear stress and velocity. It is given as

$$\eta = \frac{\tau}{\gamma} \quad (2.6)$$

Where τ is the shear stress in fluid given by $\frac{F}{A}$, where F is the force, and A is the area, and γ is the shear rate $\frac{\partial u}{\partial y}$, where u is the fluid velocity relative to y , the plate. The units correspond to (mass/length)/time: $\frac{kg}{m \cdot s}$, which is equivalent to pressure multiplied by time (Pa·s), given in pascal-second. Similarly, the units can be given in Poise (P) from a centimetre-gram-second system of units (GCS), where $1 \text{ Pa} \cdot \text{s} = 10 \text{ P}$, and centipoise, $10^{-3} \text{ Pa} \cdot \text{s} = 1 \text{ cP}$. Viscosity can be approximated using regression coefficients as a function of temperature [13]. For gas, it is given as

$$\eta_G = A + BT + CT^2 \quad (2.7)$$

Where η_G is the viscosity of the gas, given in $10^{-7} \text{ Pa} \cdot \text{s}$, T is the temperature in Kelvin, and A, B, C coefficients are chemical compound regression coefficients, unique per gas. Similarly for viscosity of liquids as a function of temperature, is given as

$$\log_{10} \eta_L = A + \frac{B}{T} + CT + DT^2 \quad (2.8)$$

Where η_L is the viscosity of the liquid, given in mPa.s, and similarly as gasses, A, B, C and D are chemical compound coefficients, with temperature T given in kelvin. Likewise, the viscosity dependence on temperature and pressure for fluids may be generated using the same tool [15] as mentioned in the previous section. This accuracy is mentioned to be <5%, but often better than 2 %.

Kinematic viscosity is the ratio of dynamic viscosity and fluid density. Mathematically it is defined as

$$\nu = \frac{\eta}{\rho}$$

Where η is dynamic viscosity, and ρ is density. It also can be related to the Hagen-Poiseuille equation, derived for nonideal fluid dynamics. This equation works under the assumption that it is a Newtonian incompressible fluid. This equation describes pressure drop in relation to viscosity. It is given as follows:

$$\Delta P = \frac{8\eta LQ}{\pi R^4} \quad (2.9)$$

Where ΔP is the pressure drop, η dynamic viscosity, L length of the tube, Q the volumetric flow rate, and R the radius of the tube.

2.2 Coriolis

This section discusses the main device used in this research, with its fabrication steps discussed by J.Groenesteijn et al. [23].

When the moving mass of the flow passes through the tube, it generates Coriolis force. This is illustrated in Figure 2.1b. These forces are directly proportional to the mass flow. J. Haneveld et. al [24] expressed this in an equation

$$\vec{F}_c = -2 \cdot L_y \cdot (\vec{\omega}_{am} \times \vec{\Phi}_m) \quad (2.10)$$

Where Φ_m is the mass flow inside the tube, ω_{am} is the angular velocity, and L_y is the width of the tube. To measure these signals, two readout capacitors are implemented. The drive axis (twist mode) is present when the device is excited at its resonance frequency ω_T , illustrated in Figure 2.1a. This frequency can be modelled as second order system [9], which depends on the geometry characteristics of the tube:

$$\omega_T = \sqrt{\frac{k_{\text{eff}}}{m_{\text{eff}}}} \quad (2.11)$$

Where k is the effective modal spring constant and m is the effective modal mass. To measure the displacement of the tube, it is measured with capacitive combs. The twist capacitive signal was expressed by A.C de Oliveira et al.[25] as:

$$C_T = \hat{C}_T \sin(\omega_T t) \quad (2.12)$$

where \hat{C}_T is the amplitude of the twist component. The sense axis (Swing mode) appears as a result of mass flowing through the tube (see Figure 2.1b), which is caused by Coriolis force. Consequently, this changes the signal

$$C_S = \hat{C}_S \cos(\omega_T t) \quad (2.13)$$

where \hat{C}_S is the amplitude of the swing component. The swing and twist modes can be written as composite signals, where $\hat{C} = \sqrt{\hat{C}_T + \hat{C}_S}$, which follows:

$$C_1 = C_S - C_T = -\hat{C} \cdot \sin(\omega_T t - \varphi) \quad (2.14)$$

$$C_2 = C_S + C_T = \hat{C} \cdot \sin(\omega_T t + \varphi) \quad (2.15)$$

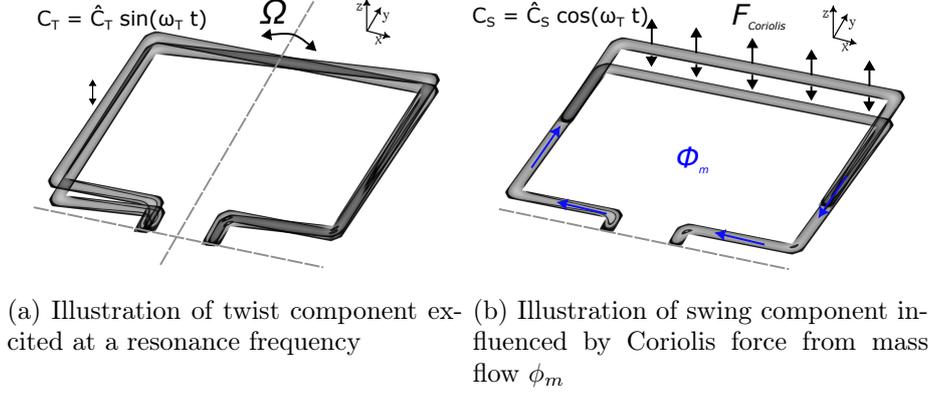


Figure 2.1: Working principle of Coriolis sensor [25] [26]

The phase shift of the capacitive signal output between C_1 and C_2 appears to be only dependent on mass flow, and independent of pressure or density. This can be deduced for small phase differences

$$\Delta\varphi = 2 \cdot \arctan\left(\frac{\hat{C}_S}{\hat{C}_T}\right) \quad (2.16)$$

The phase difference between a driving signal, and one readout capacitive signal also are able to measure mass flow, however, the sensitivity is around twice as less compared to the phase differences of two capacitive readout signals [25]. Moreover, additional non-linear behaviour might appear on the signals, introducing harmonics, which depend on pressure and temperature [26]. This behaviour is illustrated in Figure 2.2b.

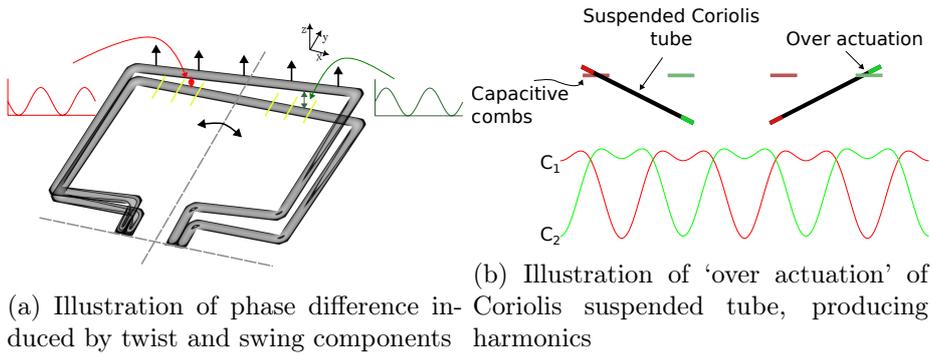


Figure 2.2: Illustration of the twist and the swing component induced by the Coriolis force due to a mass flow ϕ_m (a), with 'over actuation' the readout having secondary harmonic output (b)

To actuate, and read out the signals, there are small metal tracks on the Coriolis tubes. An additional track may also be used as a temperature-dependent track (resistor), which

allows for to calculation of more accurate temperatures and a temperature drop while mass flow cool-down the tube. This is shown in Figure 2.3.

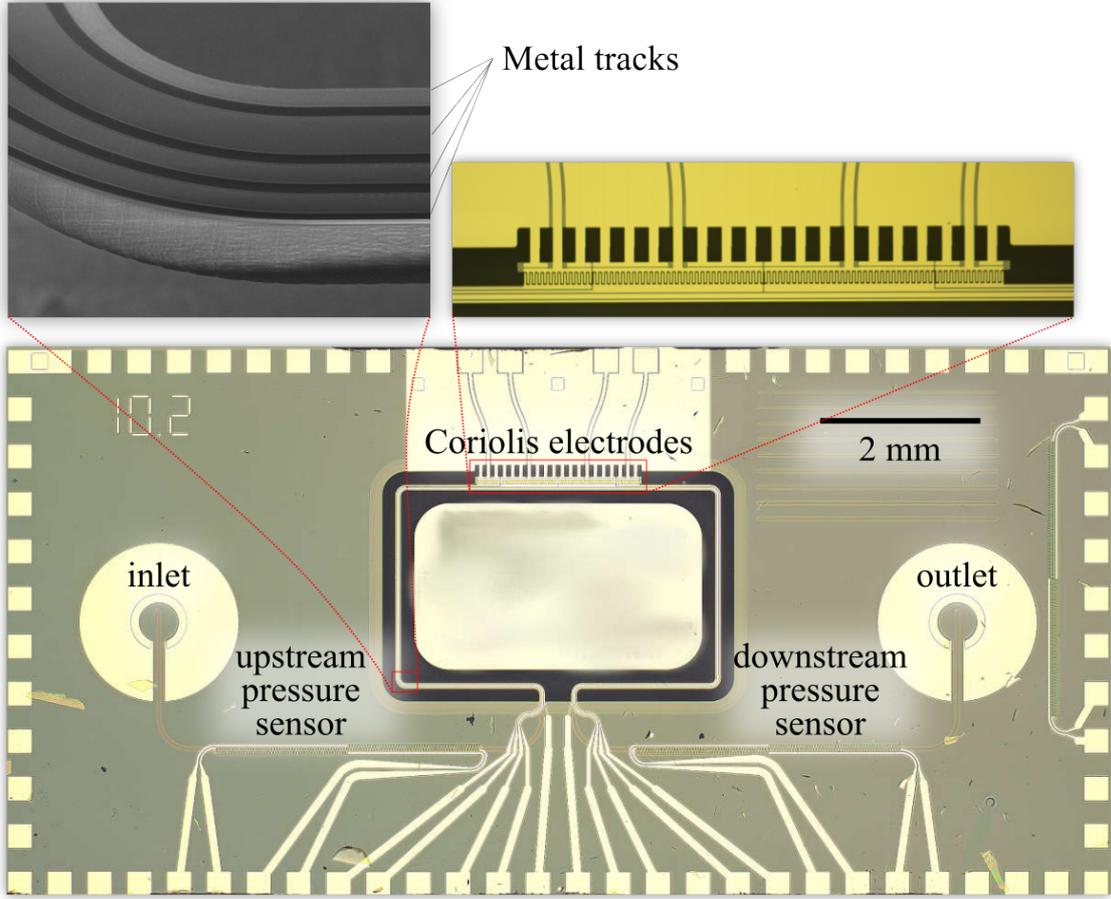


Figure 2.3: Overview microscopy images of Coriolis Mass Flow Sensor, with zoomed in part of metal tracks, and Coriolis electrodes.

Furthermore, the frequency is very dependent on density, and somewhat on pressure [27]. With the swing mode, the resonance frequency of all modes depends on the density of the fluid. Following Equation 2.11 it can be expanded in terms of the mass of the fluid and pressure-dependent spring, resulting in

$$\omega_{\text{fluid}} = \sqrt{\frac{k(P)}{\rho_{\text{fl}}(P)V_{\text{ch}} + m_{\text{ch}}}} \quad (2.17)$$

where the pressure-dependant spring constant is expressed as $k(P)$, ρ_{fl} as the fluid density, V_{ch} inner volume of the channel, and m_{ch} as the mass of the channel itself. Moreover, the stiffness of the tube, as well as the fluid density both depend on temperature. Since gas is compressible, $\rho_{\text{fl}}(P)$ depends on pressure for gasses, and for liquids, pressure is almost negligible. Due to the relation of the spring constant and density, a higher resonance frequency would imply a smaller density of the fluid.

In general, the Coriolis mass flow sensor achieves $4.7^\circ/(\text{g/h})$ on nitrogen. The frequency instability is less than 1mHz over an hour [25].

2.2.1 Viscosity determination

The fluid composition can be determined by pressure, mass flow rate, density and viscosity. As mentioned before, this sensor measures both mass flow and density. To find the viscosity, it can be related to mass flow and pressure drop over the channel [9]. As mentioned in the Viscosity section 2.1.2, this is given by the Hagen-Poiseuille equation. Due to the CMFS measuring mass flow, instead of the volumetric flow (Q), the calculated viscosity would correspond to kinematic viscosity. Dividing kinematic viscosity by density yields dynamic viscosity. The direct equation for kinematic viscosity for CMFS is given by

$$\frac{\pi R^4(P_1 - P_2)}{8L\phi_m} = \nu \quad (2.18)$$

where P_1 is upstream, and P_2 downstream pressures, ν is the kinematic viscosity, ϕ_m mass flow rate, L is the channel length, and R is the effective radius. From the previous section, we have that $\Delta\varphi \propto \phi_m$ and further it can be seen that $\frac{\Delta P}{\Delta\varphi} \propto \nu$.

2.2.2 Pressure sensing

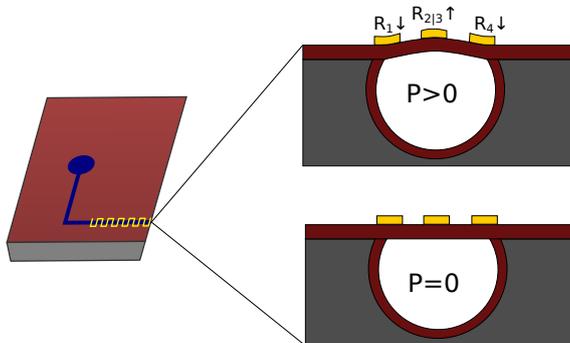


Figure 2.4: Illustration of on-chip pressure sensor. Adapted from [28]

Furthermore, there are pressure sensors integrated into the device. These pressure sensors are thin-film gold resistors deposited on top of the channel, connected in a Wheatstone bridge. Due to pressure, the channel deforms, resulting in two resistors elongating with increasing pressure, and the other two compressing (see Figure 2.4). This deformation causes a change in the resistance, which is then measured by the Wheatstone Bridge. This can be expressed by the equation:

$$V_{out} = \left(\frac{R_2}{R_1 + R_2} - \frac{R_4}{R_3 + R_4} \right) V_{in} \quad (2.19)$$

where R_1 to R_4 are resistors connected in a Wheatstone Bridge formation. To increase the sensitivity, the resistors are placed so that R_1 and R_4 resistance decreases, and R_2 , R_3 resistance increases due to pressure.

The pressure sensors achieve a sensitivity of $4\mu\text{V bar}^{-1}$ for pressures ranging from 0 to 1 bar, and it can handle at least 10 bars [27].

2.3 Fourier Transform

This section is based on *The Scientist and Engineer's Guide to Digital Signal Processing* [29]. Fourier transforms are commonly used to convert signals, e.g. time-dependent sound

signals, or spatial-dependent image signals (pixel values) to an output which describes the frequencies present in that signal. The transform also holds the other way around, i.e. it is invertible: $\mathcal{F} : \mathbb{C}^n \mapsto \mathbb{C}^n$, frequency components can be transformed back to its time representation. The Fourier transform is complex-valued for each output frequency, and it denotes the amplitude and phase at that particular frequency. Commonly, the Discrete Fourier Transform is used as measurements in a finite sequence of equally-spaced samples. It is defined as:

$$X_k = \sum_{n=1}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn} \quad (2.20)$$

where x_n is a sample, and $e^{\frac{i2\pi}{N}kn}$ a complex sinusoid, at a frequency $\frac{k}{N}$. This comes from the definition $re^{ix} = r(\cos(x) + \sin(x)i)$. The transform can be seen as an infinite series of the sum of sine waves (or in the case of discrete transform, the sum of $N - 1$ sine waves, where N are the available samples).

Nyquist sampling theorem

To accurately reproduce a signal, the sampling rate of the measurement must be at least twice its frequency.

Spectral leakage

Leakage results in additional frequency components. This occurs due to the signal having discontinuities as a consequence of finite windowing of the data.

Frequency bins

Bins are related to the sampling rate, which represents how ‘accurately’ it is able to explain the measurement. Suppose the sampling rate is 10 kHz, and it is measured for one second. Then, the frequency bins are 1 Hz, containing information for each frequency, this is a fraction of the sampling rate, and available samples

$$\text{bin}_{\text{width}} = \frac{\text{samplingFreq}}{\text{Samples}}$$

If there are fewer samples than the sampling frequency, the frequency resolution will be worse, and other frequency bins will interfere with the result.

Harmonics

Fundamental harmonics usually have the highest magnitude, with the lowest frequency in the frequency domain, and they correspond to the main signal (excluding 0 Hz component). Harmonics are secondary outputs of the system, corresponding to non-linearities in the readout, which may be intentionally produced. An example of signal with harmonics can be seen in Figure 2.5. Fundamental harmonic is usually denoted as ω , and the second and third harmonics, 2ω , 3ω correspondingly. This follows the property, that the second harmonic is twice the frequency of the first harmonic.

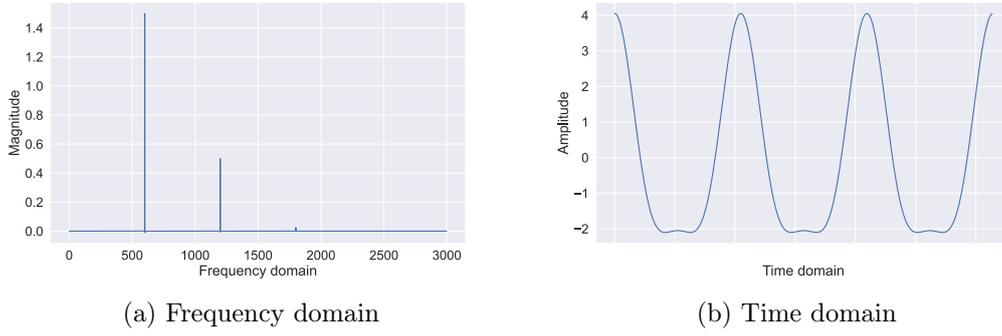


Figure 2.5: Illustration of frequency and time domain representation using example function $3 \cos(\omega t) + \cos(2\omega t) + 0.05 \cos(3\omega t)$

2.3.1 Magnitude

The magnitude of individual harmonic (i.e. sine wave) may also be found via DFT. This corresponds to taking the magnitude of the signal at every frequency, given as

$$A_k = \frac{1}{N} \sqrt{\text{Re}(X_k)^2 + \text{Im}(X_k)^2} \quad (2.21)$$

Where X_k is the complex value from DFT, and N are the number of samples. This returns a magnitude spectrum.

Fundamental component

To find the fundamental component, it consists of finding the frequency in which the magnitude is highest. This, however, excludes $k=0$, as it is the offset component of the signal.

$$k_\omega = \arg \max_k A_k$$

Where k_ω is the frequency bin at the fundamental frequency.

2.3.2 Phase

Since the output from Fourier contains the magnitude and phase information of each frequency, the phase can be computed for each frequency, resulting in a phase spectrum.

$$\varphi_k = \arctan \left(\frac{\text{Im}(X_k)}{\text{Re}(X_k)} \right) \quad (2.22)$$

Where φ_k returns phase for each frequency component.

Phase difference

If two signals have the same frequency, the phase difference can be found by computing by simply taking the absolute difference between phase spectrums.

$$\Delta\varphi = |\varphi_{1k} - \varphi_{2k}|$$

Similarly, to find the fundamental signal phase shift, it is in the k_ω frequency bin.

2.4 Correlation and Cross-Correlation

Given that signals are sinusoidal, the cross-correlation method allows finding the time-lag τ between the two signals directly, which then can be converted into the angle, full expression is given as:

$$\theta = \arccos \left(\frac{\sum_{n=0}^{N-1} x[n]y[n]}{\sqrt{\sum_{n=0}^{N-1} (x[n])^2 \sum_{n=0}^{N-1} (y[n])^2}} \right) \quad (2.23)$$

Where x and y are two distinct sinusoidal signals, and the output θ returns the phase difference $\Delta\varphi$ between the two sinusoidal signals. This avoids computing DFT to find phase differences and does not require as many samples. In the most simple terms, this calculates the angle between two vectors, which is given as

$$\cos \theta = \frac{a \cdot b}{\|a\| \|b\|} \quad (2.24)$$

Where $\|\cdot\|$ is the ℓ^2 norm. However, this approach does not work accurately if higher harmonics are present. Alternatively, the equation inside the bracket can be seen as a Pearson correlation:

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} \quad (2.25)$$

Where $\text{cov}(X,Y)$ is the covariance between two signals, and σ_X and σ_Y are their standard deviations. The correlation metric returns values between $-1 < \rho < 1$. The expanded form :

$$\rho_{X,Y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (2.26)$$

The numerator of the ρ metric shows what is the joint variability between two data columns, where \bar{x} and \bar{y} are the means of the respective random variables.

Interpretation of this ρ coefficient was catalogued by Rodgers and Nicewander [30] with thirteen ways to interpret this function. Some of them are the Standardized slope of the regression line, the ratio of two means, the geometric mean of the two regression slopes, and so on. The metric gives some intuition on how much the data depends on each other (only linearly).

2.5 Kernel method

This section discusses the kernel methods used to process data. These methods are useful in kernel-based algorithms, as they allow linear algorithms to become nonlinear by implicitly mapping data to feature space.

2.5.1 Kernel trick

Kernel methods, also known as kernel tricks allow implicit mapping of data into high-dimensional feature space. It works by mapping input space \mathcal{X} to the feature space \mathcal{H} , via a nonlinear mapping $\kappa(\cdot) : \mathcal{X} \mapsto \mathcal{H}$. This approach translates the nonlinearity of the data to a linear feature space, in which linear methods for regression and classification can be applied. To find the right mapping $\kappa(\cdot)$ is not so easy, and general nonlinear mappings are proposed for their specific purposes, such as wavelet kernel for time series prediction

[31], sigmoid kernel for nonlinear classification [32], and many others [33, 34]. These methods were first used to solve binary classification problems via the so-called kernel trick applied to Support Vector Machines (SVM) [35]. These kernel tricks were applied to other algorithms, e.g. kernelized ridge regression [36], where the input to an algorithm is a kernel (preprocessed data). Due to its universality, the Radial Basis Function (RBF) kernel is discussed.

2.5.2 Radial Basis Function

This kernel has numerous applications for regression, classification and neural networks due to its universality. It is defined as:

$$\kappa(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (2.27)$$

where $X \in \mathbb{R}^{n \times m}$, and $\|x_i - x_j\|$ is the squared Euclidean distance between features. It may also be seen as a similarity function [37]. This method transforms the input space into feature space by computing the inner products, which by the definition of exponential, can be expanded in terms of basis functions. Intuitively, the kernel can be rewritten as

$$\begin{aligned} \kappa(x_i, x_j) &= e^{-\frac{(x_i - x_j)^T(x_i - x_j)}{2\sigma^2}} \\ &= e^{-\frac{(x_i^T x_i - 2x_i^T x_j + x_j^T x_j)}{2\sigma^2}} \\ &= e^{-\frac{x_i^T x_i + x_j^T x_j}{2\sigma^2}} e^{x_i^T x_j} \end{aligned} \quad (2.28)$$

which by definition, the Taylor series expansion is used for e^x

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad (2.29)$$

The transformation output can be represented in a matrix

$$\kappa(x_i, x_j) = \begin{bmatrix} \kappa(x_1, x_1) & \kappa(x_1, x_2) & \dots & \kappa(x_1, x_n) \\ \kappa(x_2, x_1) & \kappa(x_2, x_2) & \dots & \kappa(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(x_n, x_1) & \kappa(x_n, x_2) & \dots & \kappa(x_n, x_n) \end{bmatrix} \quad (2.30)$$

where $x_i \in \mathbb{R}^{1 \times m}$ represents input space with m features (generally $n > m$) and the transformation results in a feature space, $\mathcal{H} \in \mathbb{R}^{n \times n}$.

For the Gaussian Process Regressor, a slightly different RBF kernel is used:

$$\kappa(x_i, x_j) = \sigma^2 \exp\left(-\frac{\|x_i - x_j\|^2}{2l^2}\right) \quad (2.31)$$

where σ^2 is the amplitude of the function, and l is the length-scale, in which increasing the hyperparameter, points further away are more correlated.

2.6 Regression

This section covers regression algorithms used in this research.

2.6.1 Linear regression

Regression techniques are used to predict certain targets, (dependent variables), based on features (independent variables). Linear models of degree one, are in the form

$$f(x) = w_0 + w_1x_1 + \dots + w_px_p + \epsilon \quad (2.32)$$

Where w_0 is the intercept, w_1 to w_p are the weights corresponding to features x_1 to x_p , with some noise ϵ . The benefit of linear regression is that weights are easily interpretable, and give direct relation to which features impact the prediction the most.

Optimization of weights w for ordinary linear regression can be done directly via the least squares method called the Ordinary Least Squares (OLS):

$$\min_w \|Xw - y\|_2 \quad (2.33)$$

The double bars represent the ℓ_2 norm, where $\|x\|_2 = \sum_i^n \sqrt{x_i^2}$. This is a direct optimization in terms of Mean-Squared-Error (MSE), which is simply defined as the squared residuals $\mathbb{E}[(Y - \hat{Y})^2]$, where \hat{Y} are the estimated values. Minimizing w corresponds to taking the derivative of $f(x) = (Xw - y)^T(Xw - y)$ w.r.t w , and solving for w , yields $w = (X^T X)^{-1} X^T y$. There are several variations of linear regression using least squares. Such methods are called Weighted Least Squares (WLS) and Generalized Least Squares (GLS). Determining which method to use depends on the data, whether it is homoscedastic, i.e. variance of the residuals is constant, or heteroscedastic, where the variance of the residuals varies. When it can be reasonably assumed that data is homoscedastic, OLS is the Best Linear Unbiased Estimator (BLUE).

To interpret the predictions, P-values can be calculated for each independent variable. The P-value is a quantitative measure of how likely it occurred due to chance and generally determines the feature's statistical significance. The easiest way to calculate the P-value is via t-statistic. It is given as

$$t_{statistic} = \frac{\hat{w} - w}{S_{\hat{w}}} \quad (2.34)$$

Where \hat{w} are the estimated coefficients, and w is the hypothesis being tested (usually $w = 0$). The denominator is defined as

$$S_{\hat{w}} = \frac{\sqrt{\frac{1}{n-p} \sum_{i=1}^n (y_i - \hat{y}_i)^2}}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}} \quad (2.35)$$

Where $n - p$ signifies p degrees of freedom, $y_i - \hat{y}_i$ are the residuals, and the denominator is the total sum of squares. The t-value is a statistical method to calculate P-values given a known number of samples [38].

2.6.2 Support Vector Regression

Similar to linear regression, support vector regression fits data linearly, however, it fits given an error margin ϵ . Instead of a single linear fit, support vectors are fitted for the loss-sensitive region, illustrated in Figure 2.6 given by $f(x) \pm \epsilon$ hence the regressor loss is only affected if the estimate is not within the ϵ margin. Linearly, the goal is to find a function which models the relationship between input and output, given as $f(x) = w^T x + b$,

where x is the input data, and w , the weights. Given constraints of loss sensitive region, the optimization is formally given as:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \ell_{\epsilon}(f(x_i) - y_i) \quad (2.36)$$

where C is the regularization constant, and ℓ_{ϵ} is the ϵ -insensitive loss function. The ϵ region can be defined as

$$\ell_{\epsilon}(z) = \begin{cases} 0 & \text{if } |z| \leq \epsilon \\ |z| - \epsilon & \text{otherwise} \end{cases} \quad (2.37)$$

Since it is possible that no function can satisfy the constraint of having all residuals within the ϵ -error margin, slack variables are introduced for each point. With slack variables ξ_i and $\hat{\xi}_i$ (potential support vectors), we have

$$\min_{w,b,\xi_i,\hat{\xi}_i} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m (\xi_i + \hat{\xi}_i) \quad (2.38)$$

$$\begin{aligned} \text{s.t. } f(x_i) - y_i &\leq \epsilon + \xi_i, \quad \xi_i \geq 0, \\ y_i - f(x_i) &\leq \epsilon + \hat{\xi}_i, \quad \hat{\xi}_i \geq 0, \quad i = 1, 2, \dots, m. \end{aligned} \quad (2.39)$$

which is known as the primal formula. To solve this optimization problem, the Lagrangian dual formulation is applied. This means constructing a function by introducing nonnegative multipliers for each observation. Solving the optimization problem yields

$$f(x) = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) x_i^T x + b \quad (2.40)$$

where $\hat{\alpha}_i$ and α_i are Lagrange multipliers satisfying Karush–Kuhn–Tucker (KKT) conditions, which are given as

$$\begin{aligned} \alpha_i (f(x_i) - y_i - \epsilon - \xi_i) &= 0, \\ \hat{\alpha}_i (y_i - f(x_i) - \epsilon - \hat{\xi}_i) &= 0, \\ \alpha_i \hat{\alpha}_i &= 0, \quad \xi_i \hat{\xi}_i = 0, \\ (C - \alpha_i) \xi_i &= 0, \quad (C - \hat{\alpha}_i) \hat{\xi}_i = 0 \end{aligned} \quad (2.41)$$

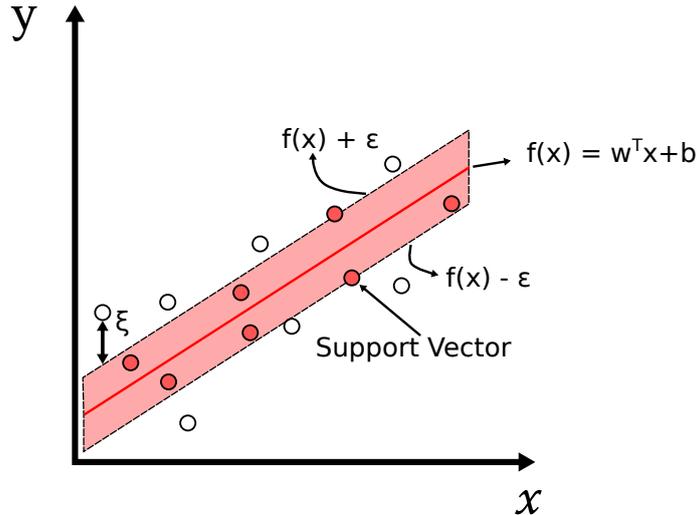


Figure 2.6: Illustration of Support Vector Regression

Kernelized SVR

To fit non-linear data, input space can be projected into higher dimensional feature space, and similarly, weights applied for that space. From linear formulation as in Equation 2.40 by substituting the kernel function in that equation, it results in

$$f(x) = \sum_{i=1}^m (\hat{a}_i - a_i) \kappa(x, x_i) + b \quad (2.42)$$

This allows for the same optimization to kernelize the algorithm, allowing for non-linear regression. More in-depth theory for Support Vector regression can be found in [39, 40, 41].

2.6.3 Gaussian Process Regressor

Another regression method vastly different from regular regression methods is a Gaussian Process Regressor (GPR) [42]. It is non-parametric, based on Bayesian learning, and is considered a non-linear model. This method is fully characterized by its mean and covariance functions and can be written as

$$f(x) \sim \mathcal{GP}(m(x), \kappa(x, x')) \quad (2.43)$$

where $m(x)$ is the mean value function, and $\kappa(x, x')$ is the covariance function (kernel). The covariance implies a distribution over functions, hence, theoretically, it is fitted over an infinite number of functions. This has the advantage of producing confidence bounds on the prediction. The derivation of finding the regression function is based on Multivariate Gaussian Distribution, where conditional distribution is based on training data, and new points based on approximated function are given as

$$P(f_* | f, X, X_*)$$

This approximated function (mean of functions) given by conditional distribution is represented as

$$\bar{f}_*(x) = \kappa(X_*, X) (\kappa(X, X) + \sigma_n^2 I)^{-1} y \quad (2.44)$$

where X_* is a new input (generally test data) used in $\kappa(X_*, X)$ and is projected into the feature space, relative to training data. Similarly, $\kappa(X, X)$ is the training data projection.

The $\sigma_n^2 I$ represents an additional predictive variance component on the diagonal covariance matrix, where I is the identity matrix. The κ matrix is explained in the previous Section 2.5.2. Likewise, the covariance function, defining the spread of functions is given as

$$\text{cov}(f_*) = \kappa(X_*, X_*) - \kappa(X_*, X)(\kappa(X, X) + \sigma_n^2 I)^{-1} \kappa(X, X_*) \quad (2.45)$$

The most basic aspect of estimating the new predictions is matrix operations using non-linear kernels, fundamentally modelling predictive distribution. Further readings of the theory, can be found in [42, 43, 44].

2.7 Classification

Several classification methods are discussed in this section.

2.7.1 Naive Bayes

It is a conditional probability model, which assigns probabilities for each outcome given data. The model can be read as the probability of a class given data x

$$p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)} \quad (2.46)$$

where $p(C_k)$ is the prior probability for each class, $p(x)$ is the evidence and $p(x|C_k)$ class-conditional probability. The ‘Naive’ part comes in the assumption that features are independent, which allows to use of conditional independence property.

$$P(C_k|x_1, \dots, x_n) \propto P(C_k) \prod_{i=1}^n P(x_i|C_k) \quad (2.47)$$

With the assumption that data is Gaussian distributed, a simplified overview of the Gaussian Naive Bayes can be seen as determining the mean μ_k and standard deviation σ_k of X for an individual class. For a new test point X_* , the likelihood of being in class C_k can be evaluated using the Gaussian assumption

$$p(x = x_*|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x_* - \mu_k)^2}{2\sigma_k^2}} \quad (2.48)$$

this then can be used to compute the probability of each class given data as described in equation 2.46. Further reading for more formal definitions and empirical study of Naive Bayes can be found in [45].

2.7.2 Gaussian Process Classifier

Similarly to the regression case, the Gaussian process classification is a probabilistic method *which assumes that all the data points are generated by a latent function with some Gaussian noise* [42]. However, due to the discrete nature of labels for classification, likelihood functions are intractable. Laplace approximation is used as a standard to approximate the non-Gaussian joint posterior analytically. The latent function assumes the GP prior directly on $f(x)$ as described in equation 2.43. This latent function is ‘squashed’ through a

function, which maps the values between 0 and 1, and it is given as a sigmoid function to produce a probabilistic prediction

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.49)$$

There are two main steps to produce inferences for a new test point. First is computing the posterior distribution of the latent function outputs given as

$$p(f_*|X, y, x_*) = \int p(f_*|X, x_*, \mathbf{f})p(\mathbf{f}|X, y)d\mathbf{f} \quad (2.50)$$

and the second step is computing probabilistic distribution given as

$$\bar{\pi}_* = p(y_* = +1|X, y, x_*) = \int \sigma(f_*)p(f_*|X, y, x_*)df_* \quad (2.51)$$

However, as mentioned previously, this equation is intractable due to discrete labels ($y \in \{0, 1\}$). To approximate the Gaussian, Laplace's method is used to the posterior of $p(\mathbf{f}|X, y)$, with the second order Taylor expansion, it is approximated as

$$q(\mathbf{f}|X, y) = \mathcal{N}(\mathbf{f}|\hat{\mathbf{f}}, A^{-1}) \propto \exp\left(-\frac{1}{2}(\mathbf{f} - \hat{\mathbf{f}})^T A(\mathbf{f} - \hat{\mathbf{f}})\right) \quad (2.52)$$

where $A = -\nabla\nabla \log p(\mathbf{f}|X, y)_{\mathbf{f}=\hat{\mathbf{f}}}$ and $\hat{\mathbf{f}} = \arg \max_{\mathbf{f}} p(\mathbf{f}|X, y)$. These parameters can be found using Newton's method, and once they are calculated, the Laplace-Gaussian approximation is

$$q(\mathbf{f}|X, y) = \mathcal{N}(\hat{\mathbf{f}}, (K^{-1} + W)^{-1}) \quad (2.53)$$

where $W = \nabla\nabla \log p(y|\mathbf{f})$, and K is the covariance matrix based on RBF kernel. To apply this method for a multi-class classification, where $C > 2$, **one versus rest** method can be used. This corresponds to converting the problem to binary classification such as [Class 1] vs [Class 2, 3, 4, etc.]. For further reading and algorithm definitions, this is described in [42].

2.8 Deep learning fundamentals

Deep learning generally refers to a neural network which is more complex and 'deeper' than a simple neural network. This complexity is mainly denoted by depth, and its representation learning capability, i.e., automatic feature extraction, often requiring much more data to train. More formal definitions for deep learning and convolutional neural networks can be found in [46, 47]. A simple illustration of an artificial neural network can be seen in Figure 2.7. Where the inputs to the network are denoted as x_i , and for all inputs, it computes a sum of weighted inputs, i.e. for a single red circle, it performs

$$b + \sum_{i=1}^N w_i x_i$$

The output of these operations is then used in activation functions to introduce non-linearities.

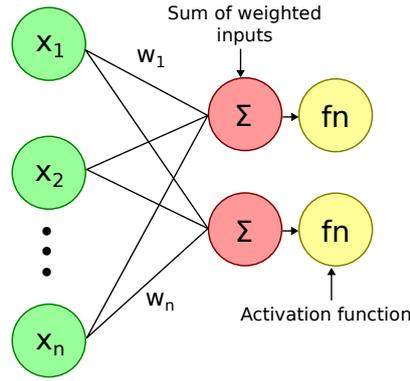


Figure 2.7: Illustration of a simple neural network

With deep learning, the output of activation functions is then further used as inputs in the network, computing again the sum of weighted inputs and again applying activation functions. Each hidden layer has its own weights which need to be optimized. The ‘deeper’ network is illustrated in Figure 2.8.

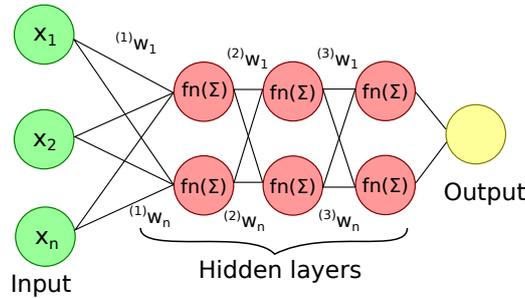


Figure 2.8: Illustration of deep learning network with hidden layers

Common activation functions used in these networks are Rectified Linear Units (ReLU), and it is given by a very simple equation

$$\text{ReLU} = \begin{cases} x & x \geq 0 \\ 0 & 0 < x \end{cases} \quad (2.54)$$

where x is the output of the sum of weighted inputs, which directly makes the output 0 if the values are negative. Likewise, Leaky ReLU may be used, which instead of returning a value of 0 when it is negative, is scaled down by a value, e.g. $0.01x$. These activation functions with a combination of many hidden layers, are able to model complex systems.

The optimization happens through backpropagation, in which the network is optimized with respect to a chosen loss function (similarly known as a cost function). An illustration of backpropagation is shown in Figure 2.9.

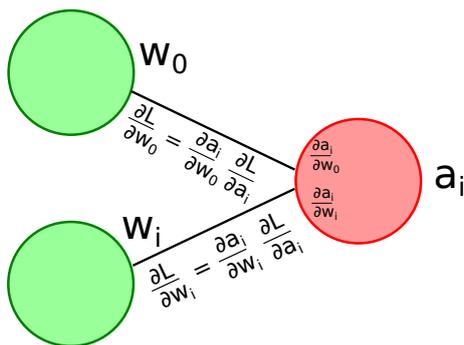


Figure 2.9: Illustration of backpropagation

Where L is a loss function chosen for a specific problem, i.e. regression or classification. For a regression problem, a simple Mean-Squared-Error could be used as a loss function, i.e.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

where y_i is the ground truth label, and \hat{y}_i is the prediction from the network. For classification problems, a widely used loss is cross-entropy, also known as log loss. It is often used with a combination of a softmax activation function, where the values of the sum of weighted input at the output layer are ‘squished’ between values 0 and 1, denoting the most likely ‘class’ the input belongs to. The softmax is defined as:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.55)$$

Where z_i is the output of the sum of weighted inputs for the i -th class. The loss function cross-entropy is defined as

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & y = 1 \\ -\log(1 - h_\theta(x)) & y = 0 \end{cases} \quad (2.56)$$

Where $h_\theta(x)$ is the predicted probability, e.g. the output of the softmax function and y denotes whether it was correctly classified. This function output value will decrease, the more accurate the probability is with predictions.

Commonly, the architecture of a network signifies how many hidden layers are in the network, and the width of individual hidden layers. Many architectures are proposed for their tasks, e.g. for image classification, object detection, autoencoders (mapping input to a lower dimension representation), etc.

2.8.1 Convolution

Convolutional Neural Networks (CNN) are a type of neural network which specializes in processing data with locality. They are very popular in image classification and detection as they mimic the visual cortex. They can similarly be applied to 1D data, with time interval representations.

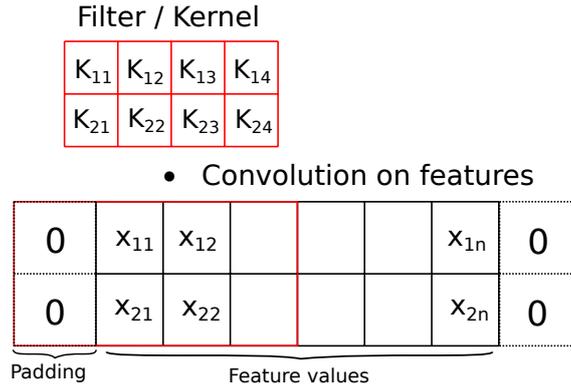


Figure 2.10: Illustration of a convolution operation on sequential data

The Kernel, or filter (if it consists of multiple kernels) moves through the image, or sequential data with a *stride* which can be specified as a hyperparameter, and performs the dot product of the feature values and the kernel, producing a convoluted feature map. To preserve the dimensions of the convoluted output, padding is used. For a $(m \times m)$ feature values, and a kernel of $(n \times n)$, the dimensions of the resulting output, would be $(m - n + 1) \times (m - n + 1)$, when padding is used, it adds zeros to the border of the input, and it becomes $(m + 2p) \times (m + 2p)$, with p being the number of layers of zeros added. Since convolutions are linear operations, after every convolution, an activation function is used to introduce non-linearities to the predictions For example, a single convolution performed on sequential data as shown in Figure 2.10

$$K_{11} \cdot 0 + K_{12} \cdot x_{11} + K_{13} \cdot x_{12} + K_{14} \cdot x_{13} + K_{21} \cdot 0 + K_{22} \cdot x_{21} + K_{23} \cdot x_{22} + K_{24} \cdot x_{23}$$

which produces a single value, for that particular window of features, and with a chosen stride, it moves through the features, convolving similarly. The kernel has K_{ij} values, which are adjusted with back-propagation techniques, allowing for the network to adjust its feature extraction depending on the context. Hence, fundamentally it works as a neural network, with the exception that produced feature maps have local impact from the sequential data, hence does not use all inputs directly as a sum of weighted inputs.

Regularization

The most basic regularization technique in neural networks is dropout layers. It nullifies the contribution of a random feature value with a probability p , set in the architecture. This helps the networks generalize better, and avoid overfitting. Furthermore, the more popular choice of regularization is batch normalization. This technique rescales and recenters the output of the convolution layer, before the activation function. The normalization is given as

$$\hat{x}^{(k)} = \frac{x^{(k)} - \mathbb{E}[x^{(k)}]}{\sqrt{\text{Var}[x^k]}}$$

where $\hat{x}^{(k)}$ are features with k dimensions. Moreover, introducing scaling and shifting is given by

$$y^{(k)} = \gamma^{(k)} \hat{x}^{(k)} + \beta^{(k)}$$

where γ and β are pair of parameters. This has nice properties, such as recovering original values. Moreover, this regularization technique speeds up the algorithm learning process and makes convergence much faster [48].

2.8.2 Gated Recurrent Unit

Another type of a very specialized neural network introduces an internal state inside the network. This network originated from Recurrent Neural Networks which were meant to handle sequential data, however, due to the RNN having ‘short-term memory’, the Long Short-Term Memory (LSTM) network was developed, which excelled in capturing long-term dependencies. This similarly had a drawback of being computationally intensive, thus Gated Recurrent Unit (GRU) was developed. It achieves comparable performance to LSTM while being faster to train. The main advantage of recurrent neural networks is the capability to process sequential data (e.g. time series) and model its temporal dynamics. An illustration of a ‘building block’ of a GRU is shown in Figure 2.11. The σ shown in the figure represents a sigmoid function, which forces the values to lie between 0 and 1. The function is defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

this function in the reset gate (R_t) controls the importance (how much to remember) of the previous state, while the update gate (Z_t) controls how much of the old state to remember. They are given as

$$\begin{aligned} R_t &= \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r) \\ Z_t &= \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z) \end{aligned} \tag{2.57}$$

where W are weights, with subscripts defining weights for feature inputs and hidden layers. For an input $X_t \in \mathbb{R}^{n \times d}$ the feature weights are $W_{xr}, W_{xz} \in \mathbb{R}^{d \times h}$ with d representing the number of features for a single input, and h number of hidden units. Weights for the hidden units are $W_{hr}, W_{hz} \in \mathbb{R}^{h \times h}$ with the bias parameters consequently being $b_r, b_z \in \mathbb{R}^{1 \times h}$. The output dimensions of the reset and update gates are $R_t, Z_t \in \mathbb{R}^{n \times h}$

To compute a new candidate hidden state, the reset gate is used in the tanh function, given as

$$\tilde{H}_t = \tanh(X_t W_{xh} + (R_t \odot H_{t-1}) W_{hh} + b_h) \tag{2.58}$$

where the symbol \odot is a Hadamard product, defined as an element-wise product. Similarly as previously, $W_{xh} \in \mathbb{R}^{d \times h}$ and $W_{hh} \in \mathbb{R}^{h \times h}$ are additional weights parameters for the candidate hidden state. To produce the final hidden state, with the effects of the update gate, it is given as

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t \tag{2.59}$$

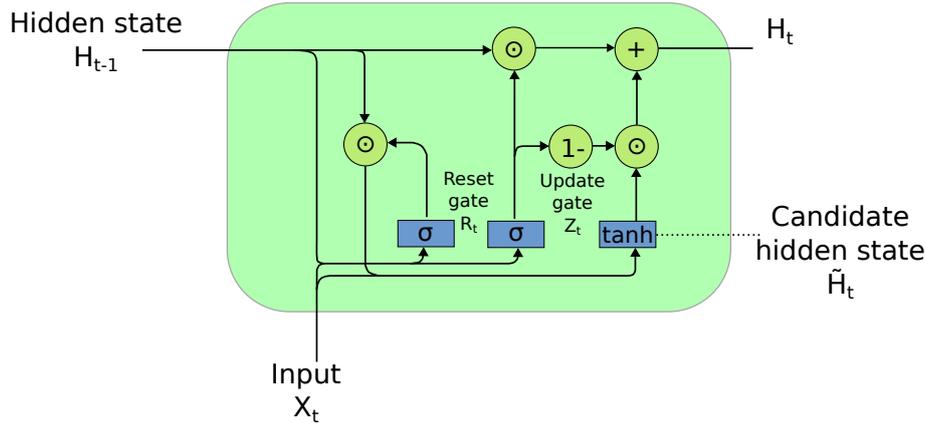


Figure 2.11: Illustration of a Gated Recurrent Unit. Adapted from [46]

2.9 Model comparisons

The chosen models for this thesis are based on simplicity, i.e. linear regression and Naive Bayes, as well as more complicated kernel methods (SVR, GP) which allow to introduction of non-linearities by dot products between features. Moreover, a deep learning method based on DeepSense architecture is used to test its capabilities in processing Coriolis data. Some of the advantages of the methods are summarised in Table 2.1.

Method	Advantage	Model
Linear regression	Simplest case of a linear combination of features. Fast and easy to implement.	Section 2.6
Support Vector Regression	Has a lot of flexibility in terms of kernels, and hyperparameters to tune the model. Practically can act as a linear, ridge and lasso regression due to its regularization parameter. Can be used for classification (SVM variant).	Section 2.6.2
Gaussian Process	Models predictive distribution of data, as well as produces confidence intervals in the estimation. Can be used for regression and classification.	Section 2.6.3, 2.7.2
Naive Bayes	One of the simplest classification methods, calculating the probability of a class given data with independence assumption between predictors.	Section 2.7.1
Deep Learning	Automatic feature extraction, resulting in the ability to handle large and complex data	Section 2.8

Table 2.1: Model comparisons

2.10 Shapley Additive Explanations

This section is based on definitions from [49]. The method is based on cooperative game theory, which determines feature impact on the prediction. It is a handy tool to interpret non-linear machine learning models, based on their global feature interactions. SHAP is based on Shapley values, where the individual feature is considered as a ‘player’ in a game, and it tells how to fairly distribute the ‘payout’ among the features. Shapley value

calculation for a feature is based on marginal contribution across all possible coalitions, where the coalitions are between various feature input variables ('players'). In the most basic case, calculating the individual feature effect of a linear model

$$\hat{f}(x) = w_0 + w_1x_1 + \dots + w_px_p \quad (2.60)$$

the contribution for the j -th feature with $j = \{1, \dots, p\}$ is given as ϕ_j

$$\phi_j(\hat{f}) = w_jx_j - \mathbb{E}[w_jX_j] = w_jx_j - w_j\mathbb{E}[X_j] \quad (2.61)$$

resulting in a feature effect w_jx_j subtracted from the average effect $w_j\mathbb{E}[X_j]$. Generally, this definition is written as

$$\phi_j(f) = \sum_{S \subseteq \{1, \dots, p\} \setminus \{j\}} \frac{|S|!(p - |S| - 1)!}{p!} (f(S \cup \{j\}) - f(S)) \quad (2.62)$$

where $f(S)$ marginalizes over features that are not in the set S

$$f(S) = \int \hat{f}(x_1, \dots, x_p) d\mathbb{P}_{x \notin S} - \mathbb{E}_x(\hat{f}(X)) \quad (2.63)$$

This method works for both regression and classification. Moreover, if non-linear kernels are used (Section 2.5), the KernelSHAP method is applied to convert back to the original feature space, and compute the feature effects in the same process. This method has been used with Gaussian Processes [50]. This thesis's most used model explainability is via a SHAP summary plot. This plot combines feature importance with feature effects, where the feature importance is calculated as

$$I_j = \frac{1}{n} \sum_{i=1}^n |\phi_j^{(i)}| \quad (2.64)$$

Example

As a basic example, assume we want to estimate a person's weight given age, height and average walking speed. Applying a regression model on these features, we can have a general interaction overview by using SHAP. As illustrated in Figure 2.12, in general, taller people would be heavier, hence the model output impact showcases that it is the most significant feature. Similarly, average walking speed has a very small influence, however, it also indicates that heavier people tend to walk slower (positive impact on the model), while faster walking indicates a lighter person. Each dot represents a person with its respective measurement impact on the model.

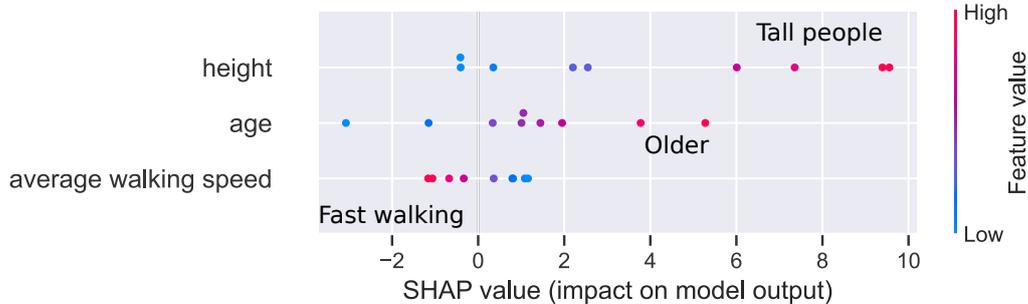


Figure 2.12: SHAP example

2.11 Accuracy

This section defines accuracy in terms of mass flow, i.e. how accurate is the mass-flow estimate. The final thesis will include and define accuracy in terms of classification, i.e. fluid-type prediction.

2.11.1 R-squared

A simple statistical metric which gives general intuition of how well the method performs. It measures the proportion of explained variance [51], and the closer the value is to 1, the better the model performance. It is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (2.65)$$

where y_i is the ‘true’ value, and \hat{y}_i is the model prediction of y_i , which the numerator can simply be seen as the sum of squared residuals. In the denominator, we have \bar{y} representing the average observation of y , and its squared difference represents the proportional variance of the data. Generally, this returns values between 0 and 1, but because the model can perform arbitrarily worse, negative values can appear, i.e. prediction is worse than \bar{y} .

2.11.2 Mass flow

To align the results with the standards in the industry, mass flow accuracy uses an adapted version of *Trumpet curve* according to IEC 60601-2-24 standard [52]. Instead of the x-axis showing the averaging time window, it shows the flow rates measured by the flow reference and the y-axis shows the error between what is set by the flow controller, and the measured flow rate [53]. Two types of errors - Reading (RD) and Full Scale (FS) are mainly used. RD is expressed as the error between the actual value, and the reading in percentage, while FS is expressed relative to the maximum scale in percentage. This is generally used to show the maximum error an instrument can have, i.e. $\text{MaxError} = x\%RD + y\%FS$, as shown in Figure 2.13.

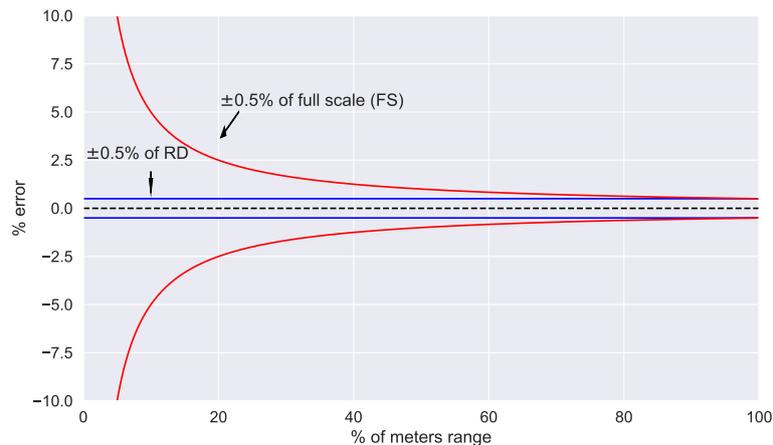


Figure 2.13: Accuracy definition of the measurements. Adapted from [54]

2.11.3 Temperature & Density

To understand the estimation error on temperature, we will use Mean Absolute Error (MAE), which is the absolute difference between the prediction, and the value given as a truth. This is given as

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.66)$$

where \hat{y}_i is the estimated value at i-th sample, and y_i is the ‘true’ value. This results in interpreting the average error in °C. Similarly, density values may also be expressed with MAE, to have the errors in kg/m³

2.11.4 Pressure drop & Viscosity

For the estimation performed on pressure drop, we will use Mean Absolute Percentage Error (MAPE). This value can be seen as the average relative estimation error to the ‘true’ value and is given as

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (2.67)$$

Similarly, we will use MAPE as a way to express the errors of estimations of viscosity. The values are too small to have to quick and accurate compared to using MSE or MAE.

2.11.5 Classification

There are several common metrics which are used to evaluate the performance of classification models. The first metric is the most known one, which simply measures the total accurate predictions.

$$\text{Accuracy} = \frac{\text{tp} + \text{tn}}{\text{total}} \quad (2.68)$$

where tp are the true positives, and tn are the true negatives. The drawback of just using accuracy, it does not take into account the balance of samples. Other metrics such as Recall and precision complement the accuracy score, as they take into account class predictions. Recall is defined as:

$$\text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \quad (2.69)$$

which measures what fraction we correctly predict as positive of all positive records, and it might be also called the sensitivity. Similarly, precision is defined as

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \quad (2.70)$$

which measures how many positives were correctly predicted. The small differences in the denominator emphasize whether to capture false positives or false negatives of the model.

One of the most useful metrics which showcase the overall performance is a Confusion matrix, which originates from a system confusing two classes. For binary cases, the diagonals showcase the percentage of correctly classified positive or negative classes, which works the same for multiple-class classification. It also shows for each class the proportion of correctly predicted classes.

Chapter 3

Related work

This chapter presents and summarizes related literature to this project. The main focus of reviewing similar literature was to survey machine learning approaches on similar types of data, as well as what has been done previously on Coriolis-based sensors to improve fluid mass flow estimations. Likewise, machine learning applications on density and viscosity estimations were surveyed for micro/nanofluids. To find the prior work, Google Scholar, Arxiv, Connected Papers and Semantic Scholar were used. The main keywords used for the search were: *coriolis, machine learning, deep learning, microfluidics, viscosity, density, liquid characteristics, signal processing*

As of 2023, most of the available research which implements a microfluidic measurement system to analyse fluid via a machine learning approach applies vision-based analysis for liquid droplets. These approaches focus on enhancing the performance of point-of-care tests. These tests are usually medical diagnostic tools which do not necessarily need to be in a medical environment [55]. However, there are several articles which focus on improving CMFS by applying mathematical methods to derive phase differences of Coriolis-based sensors. The importance of accurately extracting the phase differences is highlighted in [56]. This is highly important, as under the Coriolis effect yielded by fluid flow, the signal phase difference is proportional to the mass flow. This phase difference is a true measure of the mass flow, independent of fluid, however, due to non-ideal sensor effects, and environmental and measurement instrument noise, may introduce errors. One such method known as the Hilbert transform on Coriolis signals has been presented by Lin et al. [57], where the signals were enhanced by using a filter to eliminate white noise, and employing the Hilbert transform to find the phase difference. Similarly, Hou et al. [58] used Lagrangian interpolation to find the zero crossing and calculate the time difference between two signals to find the phase difference. Likewise, Tu et al. [59] presented an improved version of the standard method using DFT to estimate the phase difference, by adjusting the recursive relation of the Discrete Fourier Transform (DFT) and improving the convergence rate.

Besides purely mathematical methods, an article by Zhang et al. [60] focuses on improving the mass flow accuracy of CMFS by processing vibration signals via a machine learning approach, which may reduce the dependence on fluid states, and increase the performance. High flow Coriolis flow meter was used, in which the sample flow range was within 5-25 kg/min, and the Coriolis vibration frequency of around 300 Hz, sampled at 6000 Hz. The resulting dataset had 8448 samples, with a ratio of 10:1:1 for training, validation and testing set. The temperatures and pressures are not mentioned in the article. They applied Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN) and Artificial Neural Network (ANN) on Coriolis flow meters as a method to process the vibra-

tion signals and determine fluid mass flow. The input data was a standardized sequence of vibration (sinusoidal) signals. Similarly, Batch Normalization was applied to add regularization. The models were evaluated based on Mean Squared Error (MSE) criteria, and the results have shown that LSTM (single layer) performed best, with MSE of 0.0546 on the test set, with close results of RNN, with MSE of 0.0873, and ANN performed worst, with MSE of 6.8338. The best-performing models are far superior to a calibrated meter used as a reference, which had an MSE of 0.12.

Similarly, ML algorithms have been used previously to estimate fluid physical properties using physical quantities and conceptual numbers used to describe fluids. An article by Adun et al. [61] applied Support Vector Regression (SVR), Gaussian Process Regression (GPR), ANN and Boosted Tree Regression (BTR) to estimate the thermophysical properties of nanofluids. Their focus was to predict the viscosity of several hybrid nanofluids using experimental data acquired from the literature. The dataset contained 700 data points, with physical parameters, and data split into a common ratio for training (70%) and testing (30%). As an input to the algorithms, the features were Temperature (T), volume fraction (VF), density (NPD), nanoparticle size (NPS), and mixture ratio (MR). The reported performance shows very accurate results, with the best-performing method as GPR, reaching an R-squared of 0.999998 and Mean Absolute Error (MAE) of 0.008, while the worst-performing method was BTR, reaching an R-squared of 0.86 and MAE of 0.06. Furthermore, an article by Rezaei et al. [62] estimated the viscosity of gas condensate by an ensemble approach. This approach combines multiple algorithms to give improved results of estimation. The algorithms used were SVR, K-nearest neighbours (KNN), Radial Basis Function (RBF) neural network, and ANN optimized by Bayesian Regularization and Levenberg-Marquardt. The dataset contained 1370 data points, which were comprised of physical parameters, such as temperature, pressure, and gas-condensate composition, given in mole % of N₂, CO₂, C₁ to C₁₂ and the molecular weight of C₁₂₊. Data was split into 80/20 for training and testing respectively. Authors reported the performance of these models using Average Absolute Percent Relative Error (AAPRE) which is analogous to Mean Absolute Percent Error (MAPE), and the best-performing model was SVR, with an error of 4.95%, while the ensemble method with two models: SVR and K-NN, had a minor performance boost, and reduced the error to 4.83%.

For the signal processing approach, one of the state-of-the-art methods has been proposed by Yao et al. By applying CNN and RNN on time series data to do regression and classification in a unified way [63]. They focused on accelerometer data, combining multiple sensors to e.g. estimate car position, or recognize human activity. The proposed method *DeepSense* uses frequency representation of the features. The first layers of CNN, approximate sensing quantity within a time interval, and the following layers of RNN approximate across time intervals. According to the authors, their architecture might be the first one that possesses the capability for modelling temporal relationships and fusing multi-modal sensor inputs. The evaluation is based on data from phones consisting of 17,500 miles for driving data, such as accelerometer, gyroscope, and magnetometer, and their frequency responses are used in the model. This resulted in a mean absolute error of 40.43 ± 5.24 meters. Similarly, the same architecture applied to human activity recognition, e.g. sitting, standing, walking, climbing up, and biking, is able to reach an accuracy of 94.2 %. The architecture more in-depth will be discussed in methodology Chapter 4.6.

The most relevant literature was found when searching specifically on periodic time series data processing. An applicable to this project article seems to be *DeepSense*, in which the proposed method deals with combining multiple sensor data in a unified way to do regression and classification, which fits the theme of this project. Furthermore, some

attempts have been made to process Coriolis flowmeter devices' signals, to estimate mass flow. To the best of our knowledge, there is only a single available article published that processes these signals only to estimate the mass flow via a deep learning approach.

Chapter 4

Methodology

The methodology is split into four parts: (i) an overview of the measurement setup and how were the measurements performed, (ii) feature extraction and analysis, (iii) acquiring theoretical values for labels - density, viscosity, as well as an approximation for temperature and pressure drop, (iv) classical machine learning approach, and deep learning approach.

4.1 Measurement setup

The measurement setup used for experiments was custom-built at the University of Twente [64]. A visual schematic of the setup can be seen in Figure 4.1. The Coriolis device - Device-Under-Test (DUT) along with its electronics is stationed in an incubator, which allows having stable temperature, as well as vary it. The fluid used for experiments is in a reservoir, in which then nitrogen is applied to pressurise the system, while the degasser removes the nitrogen from the liquid. The controllers for setting the experiment parameters also measure mass flow, pressure and temperature which act as the target for the DUT. An actual overview image of the setup can be seen in the appendix, Figure A.2.

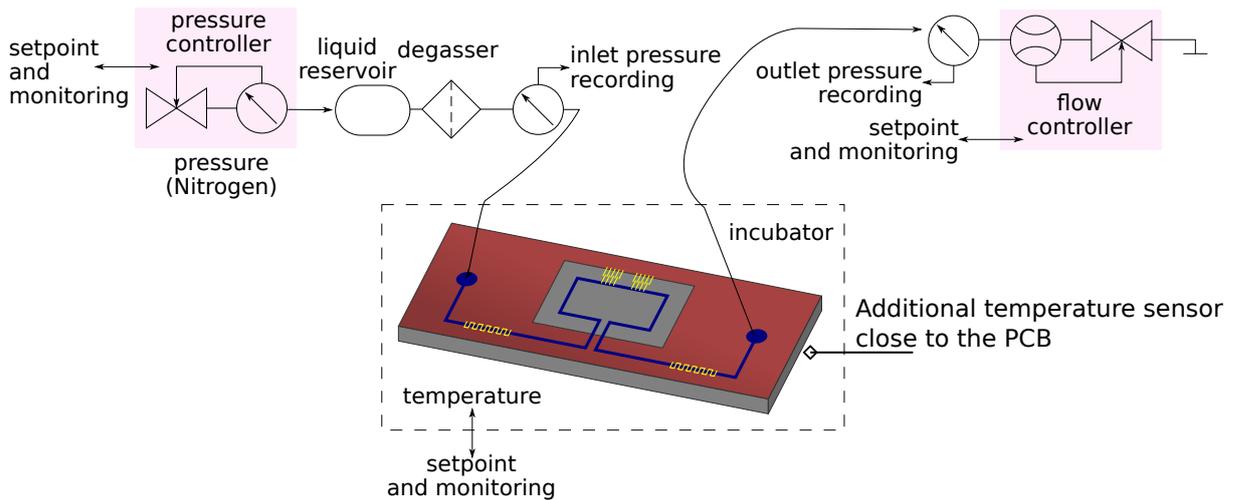


Figure 4.1: Illustration of measurement setup [64]

The measuring principle of the DUT is shown in Appendix A. Theory with respect to amplifiers, and filters are out of scope for this project. The Data Acquisition (DAQ) process is used to sample the signals and the Analog-to-digital converter (ADC) of the DAQ converts analog signals into numerical values which allow for further analysis.

4.1.1 Properties

The flow controller is a mini CORI-FLOW ML120V21 manufactured by Bronkhorst. This controller is of the same principle as the discussed Coriolis, with a different readout method, and instead of a micromachined microchannel, it is a conventionally machined metal tube. It has an accuracy to measure mass flow of $\pm 0.2\%$ for liquids, $\pm 0.5\%$ for gas (in terms of RD), and zero flow stability of $< \pm 10\text{mg}/h$. The incubator, which is used to control the temperature is MEMMERT IPP55, with a temperature accuracy of $\pm 0.1^\circ\text{C}$, and the additional temperature sensor (PT100) measuring temperature locally at chip-level, with an accuracy of $\pm 0.1^\circ\text{C}$. The pressure sensor, is EL-PRESS P-602CV, with an accuracy of $\pm 0.5\%$, and lastly, the DAQ is NI PCI-6143, with 16 bits. In practical situations due to noise, effective bits may be closer to 14. These accuracies directly correspond to how accurate are the ground truth values. The accuracy of these instruments is summarised in Table 4.1

Table 4.1: Measurement instruments

Measurement	Instrument	Accuracy
Temperature	MEMMERT IPP55	$\pm 0.1^\circ\text{C}$
Temperature _{pcb}	PT100	$\pm 0.1^\circ\text{C}$
Mass flow	CORI-FLOW ML120V21	$\pm 0.2\%$ (RD) for liquids, $\pm 0.5\%$ (RD) for gas
Pressure	EL-PRESS P-602CV	$\pm 0.5\%$ (FS)
DAQ	NI PCI-6143	16 bits

4.2 Measurements

Measurements were performed with varying mass flow, pressure and temperature parameters. The parameters were changed linearly. The temperature range was from 15 to 35 $^\circ\text{C}$, pressure from 4 to 6 bar(a), and mass flow from 0 to 5 g/hr. Temperature step change was set to 5 $^\circ\text{C}$, pressure set to 0.5 bar(a) and similarly, mass flow to 0.5 g/hr. A single measurement at a specific measurement parameter contains reference sensor readings: pressure input to the vessel (P_{in}), mass flow (ϕ_m), and the temperature ($T_{\text{incubator}}$) of the incubator.

To achieve a stable temperature, mass flow and pressure, for each measurement, a ‘settling time’ was set according to the reference readings. The settling time depends on the variance of reference readings, e.g. variance should be $< 0.05\%$ within a time window (30 mins). For each measurement at stable temperature, mass flow, and pressure, there are $3 \times (250\text{k})$ sampled readings using DAQ. Capacitive signal readings (C1 and C2) and actuation signal (D) were sampled at 250 kHz. This results in a single experiment containing time-based samples $X_p \in \mathbb{R}^{3 \times 250k}$, where X_p is the data for the p-th experiment. Additionally, one-time readings at the stable parameters are saved from the CMFS: temperature-dependent resistive track (R_{track}), resistive pressure sensors (P_1 and P_2). One-time readings such as temperature (T_{pcb}) with the addition of (R_{track}) to calculate the fluid cooling down effect on the tube, and mass flow (ϕ_m) readings can be used as a ‘ground truth’. The measured signals of CMFS are illustrated in Figure 4.2.

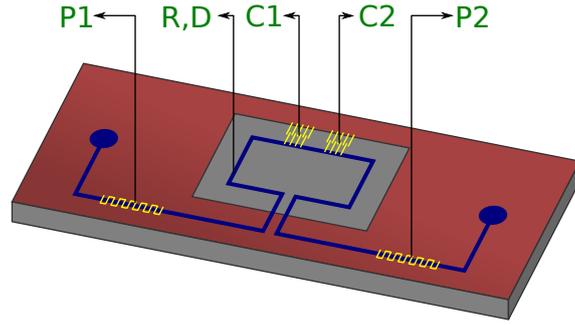


Figure 4.2: Illustration of measured parts of the CMFS.

4.3 Feature extraction and analysis

Since signals were sampled at 250 kHz, performing further analysis and correlating them to the ground truth requires a method to summarise the behaviour of the signal. One of the main methods is FFT.

Features acquired via the FFT approach are from the first fundamental harmonic. Since the signals were in the form of $A\sin(\omega t + \varphi) + o$, for capacitive signal 1 and 2 (shortened to C1 and C2), the offset C1 implies o signal shift in y-axis from the capacitive signal 2, similarly as capacitive signal 1. The same logic holds for magnitude C1 and magnitude C2. The phase differences are denoted by $\Delta\varphi_{C1}$ and $\Delta\varphi_{C2}$, which is analogous to the phase difference between the actuation signal and the corresponding capacitive signal, while $\Delta\varphi$ symbolizes the phase difference between $\Delta\varphi_{C1}$ and $\Delta\varphi_{C2}$.

Features from FFT are used in the ‘classical’ machine learning approach. An overview of the feature extraction can be seen in Figure 4.3

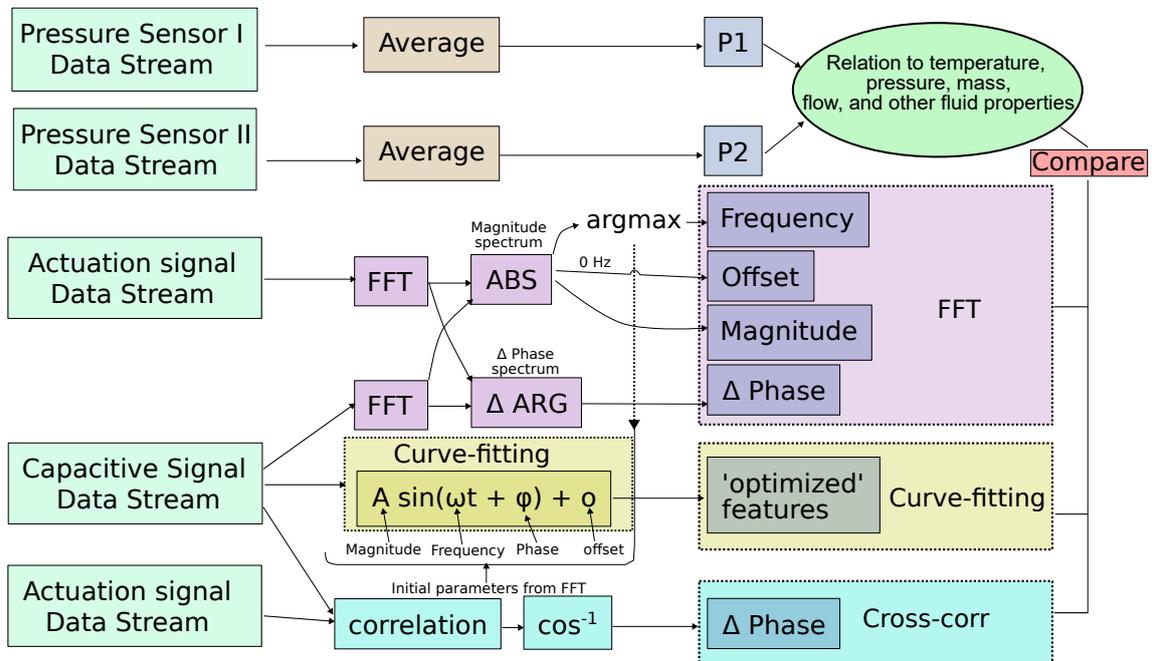


Figure 4.3: Feature processing methodology and correlations relative to fluid properties

4.4 Ground-truth values

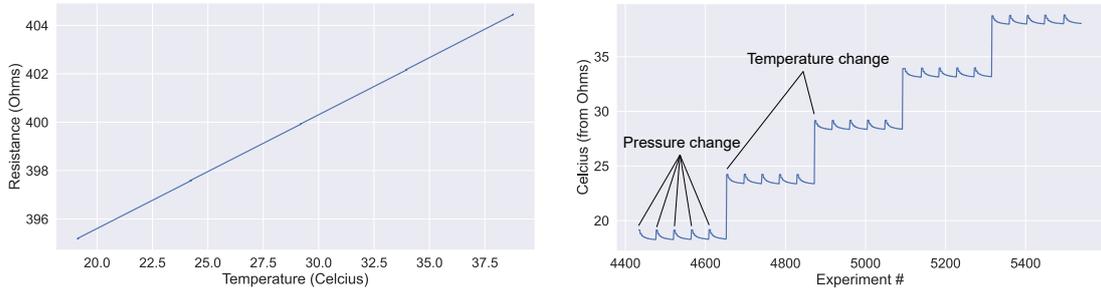
Accurate values to be used as a label, such as viscosity, density and fluid temperature are not present in the current measurements or were inaccurate. The data set as labels contained mass flow, pressure, temperature (PCB), and the liquid measured. However, measurements of the temperature were locally at the chip level, and the pressure values were the pressurised vessel readings. This ignores the pressure drop between the tubing and the device, as well as across the device. Although the liquid type was mentioned in the dataset, viscosity was not measured, as well as more accurate density measurements were not directly available.

4.4.1 Temperature label

By using temperature-dependent resistive tracks mentioned in the background section, together with the reference temperature of the PT100 sensor, we estimate the more accurate temperature to be used in our data as a label and further analysis. Performing linear fit at no flow:

$$T_{R_{\text{track}}} = \alpha R_{\text{track}} + T_{R0}$$

where R_{track} are the temperature-dependant resistance track readings. Performing a linear fit using coefficient α (ohms per °C) with temperature offset T_{R0} , at experiments with mass flow, we will get a temperature drop from the fluids (see Figure 4.4a and 4.4b). This results in the resistive track sensitivity of 2.12 ohms per °C. This neglects the temperature offset of the tube, hence the true temperature may be hotter by a couple of °C.



(a) Temperature-dependant resistive track measurements at varying temperature and no mass flow (b) Including measurements with mass flow, and applying α and T_{R0} coefficient to estimate temperature at varying mass flow

Figure 4.4: Modelling temperature drop

4.4.2 Pressure drop label

To have a ground-truth value to train and evaluate pressure-drop estimation, we utilise curve fitting using electrical readings from the pressure sensors. While pressure input is included in the dataset, as well as pressure sensors are present on-chip, the pressure drop reading across the tube is not present in the measurements. This was estimated by linear fit, including temperature dependence. For individual liquids, measurements with no mass flow were used, since the pressure in the system should be constant. The readings from pressure sensors (P_1 and P_2) were fitted to equal the pressure of the input (since $P_1 = P_2$ should result in 0 pressure drop), and the coefficients applied with measurements including mass flow. This is calculated as:

$$P_{1_{\text{est}}} = \alpha_1 P_1 + \beta_1 T_{R_{\text{track}}} + P_{10}$$

$$P_{2_{\text{est}}} = \alpha_2 P_2 + \beta_2 T_{R_{\text{track}}} + P_{20}$$

where P_1 and P_2 are pressure sensor readings, $T_{R_{\text{track}}}$ temperature based on temperature-dependant resistive track, and $\alpha_{1/2}$, $\beta_{1/2}$ are coefficients with the assumption that the behaviour is linear with P_{10} and P_{20} pressure offsets. This results in an estimated pressure drop

$$\Delta P_{\text{est}} = (P_{1_{\text{est}}} - P_{2_{\text{est}}})$$

4.4.3 Density and viscosity labels

Fluid density and viscosity are sensitive to pressure and temperature changes. To obtain the ground-truth values as closely as possible to pressure and temperature changes, we use the estimated temperature using resistive track values as a ‘true’ temperature, and the applied pressure in the system. Using the estimated temperature, as well as the pressure, we apply theory from background subsections 2.1.1 and 2.1.2 to obtain the theoretical values of density and viscosity to use as ground-truths. Measured fluids and their characteristics are summarised in Table 4.2.

Table 4.2: Fluids measured, and their theoretical characteristics

Liquid	Density kg/m^3 (at 20°C, 4 bar)	Dynamic viscosity mPa.s (at 20°C, 4 bar)
Ethanol	806.4	1.146
Acetone	796.5	0.318
IPA	810.8	2.191
H2O	998.4	1.018
Gas	Density kg/m^3 (at 20°C, 4 bar)	Dynamic viscosity mPa.s (at 20°C, 4 bar)
N2	4.602	0.0174

4.5 Machine learning

This section covers the steps used before training classical machine learning models

4.5.1 Feature selection & processing

The first crucial step in classical machine learning is to select features and/or do feature engineering to increase performance. Feature engineering is used to find better representations of features, by performing non-linear transformations [65].

Each section in the results covers its unique transformations for the task. The feature selection was based on correlation metrics and analysis of feature dependence on fluid, temperature, pressure and mass flow for individual tasks, i.e. mass flow, density, viscosity and other estimations. The collinearity between features (when one feature can explain the relationship of another) was minimized as much as possible. The effects of collinearity are unfavourable, which affects the explainability of the model, as the weights can be shifted between two collinear features, as well as it may introduce higher prediction errors in certain regions [66, 67]

4.5.2 Data splitting

In general, data splitting is used to test whether the model can generalize well on unseen data. This avoids overfitting data, e.g. memorizing without reliably estimating new untrained data.

For this research, the ratio of data is 30% for training and 70% for testing. This is due to expected linearities, which it should generalize very well. Similarly, simple random sampling is used across all measurements without any of the advanced methods. However, for some of the estimations (specifically density and viscosity) in the result section, performance is tested on an untrained fluid.

4.5.3 Data normalization

Another essential step before training classical machine learning is the data normalization/standardization pre-processing step. Due to various scales of features, the algorithms would give uneven contributions to individual features [68] if data was not properly pre-processed. By applying methods to transform the data, e.g. normalize, it has the advantage of speeding up the learning process of algorithms, as well as increasing performance.

Many normalization and standardization techniques are proposed, such as Z-score, Pareto scaling, Power transformation, Min-Max, etc. [69]. For this project, a Min-Max normalization method is applied. It has been shown that Min-Max outperforms Z-score for classification tasks using k-NN [70], moreover, it provides the benefit of each feature having the same scale, the weight for the individual feature should directly represent its importance in linear regression tasks. The Min-Max normalization is given as

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (4.1)$$

4.5.4 Validation

The validation of performance is based on the test set, which includes 70% of data. Performance metrics are mentioned in Section 2.11, and vary per property estimation. Mass flow estimation, uses Full-Scale (FS) accuracy, denoting the maximum error, temperature and density uses Mean Absolute Error (MAE) to directly interpret error in given units, while pressure drop and viscosity use Mean Absolute Percentage Error (MAPE).

The performance of the estimations are compared to one of the state-of-the-art devices to determine viscosity and density in a paper by H.Bissig et al. [71], as well as to similar measurement principle of high-performing devices on the market for mass flow estimation comparison, in particular one from Bronkhorst[®] mini CORI-FLOW ML120V21 (mentioned in Table 4.1).

4.6 Deep learning

Compared to classical machine learning, deep learning has a different approach. Usually, it requires much more data to achieve good performance, however, the benefit of doing so, it learns its feature representations without explicit feature engineering [72]. For this method, feature selection was not done, and the only pre-processing step was applying FFT to acquire the frequency representation of data. The features were not explicitly normalized, and the input training data were real and imaginary components $X_p \in \mathbb{R}^{4 \times 15k}$ where X_p represents data of the p-th experiment, containing 4 real and imaginary components of C1 and C2 sampled signals, however only first 15000 frequency components were taken,

as the most information up to 3ω (3 times the fundamental frequency) is stored. The architecture of this deep learning approach was based on Yao et al. DeepSense [63] as faithfully as possible, which can be used for both regression and classification. An adapted illustration of this architecture can be found in Figure 4.5. Individual layer output is denoted as $X^{(k)}$.

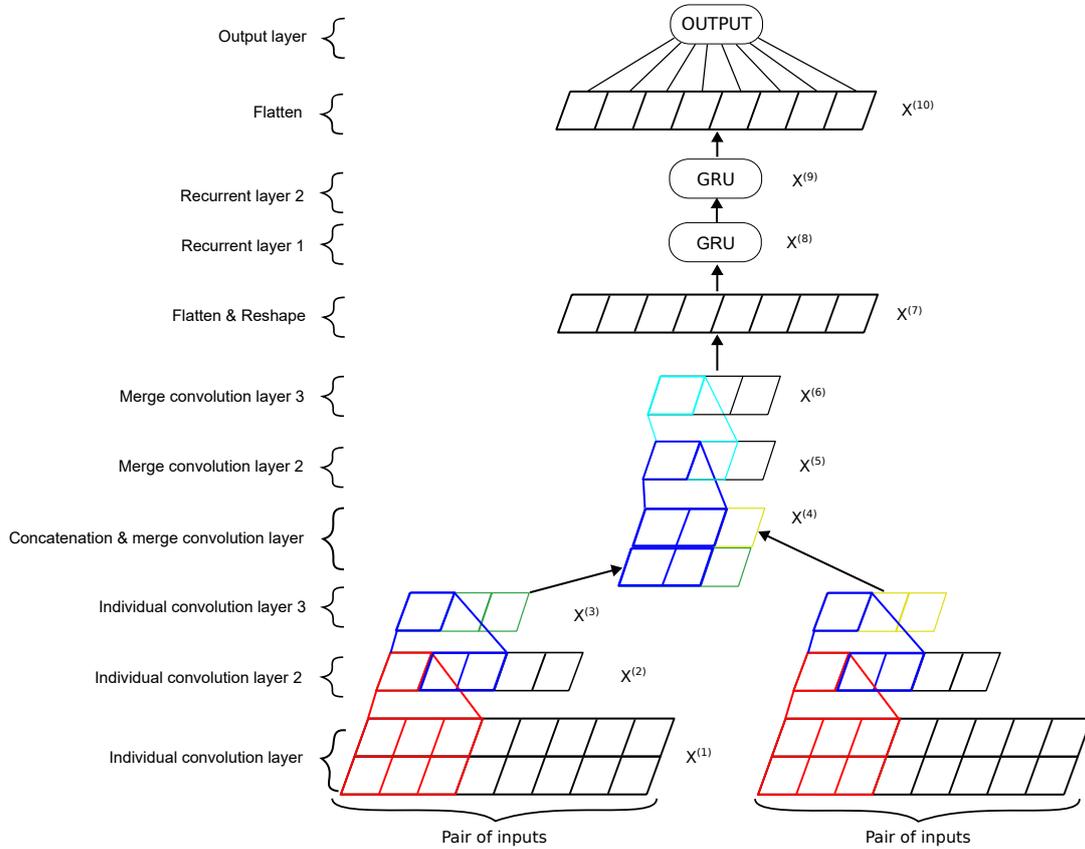


Figure 4.5: Illustration of DeepSense architecture used for Deep Learning. Adapted from [63].

The pair of inputs on the left and right sides of the architecture are symmetrical. The input on the left side contains the C1 real part and C2 real part, in which the network learns the patterns from real parts of the capacitive signal, and convolves them together, while the right side inputs are C1 and C2 imaginary components. The input dimensions were adjusted within the network, such that the dimensions are $1 \times 3 \times 2 \times 5000$, representing (channel \times 3 Frequency bins \times 2 frequency components \times 5000 frequencies). The channel is an extra dummy dimension to produce feature maps, while 3 frequency bins are simply dividing 15000 frequency components into separate layers, i.e. first frequency bin layer contains frequencies 0-5000, while the second 5000-10000 and third 10000-15000, and frequency components containing 2x real/imaginary components of C1 and C2 signals. The role of frequency bins is to build up on the recurrence of frequencies, such that a Gated Recurrent Unit could be applied to learn the dependencies between the frequency ranges. The input of frequency bins is illustrated in Figure 4.6. The input data is processed via convolutions. Individual convolution layer contains regularization functions such as dropout, batch normalization and activation functions. Each layer contains the same structure, however, parameters such as stride, kernel size and padding vary between some layers. An illustration of a convolution layer composition can be seen in Figure 4.7.

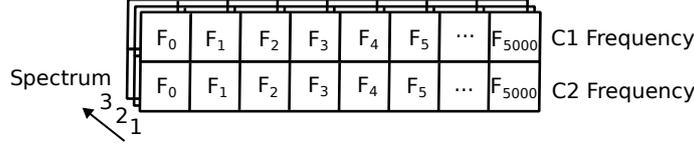


Figure 4.6: Illustration of input dimensions to the DeepSense architecture

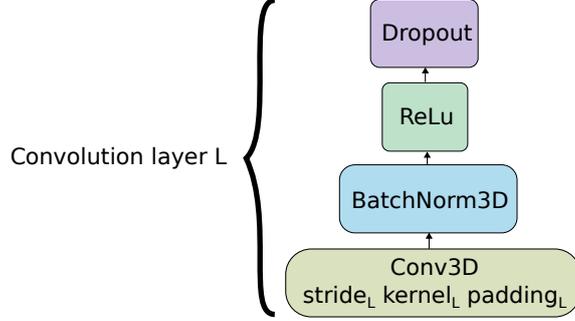


Figure 4.7: Illustration of a single Convolutional layer composition

The first convolution layer has a kernel size of (1,2,18), where 1 denotes how many frequency bins to convolve directly, 2 represents the pair of frequency inputs, and 18 convolves in a width of 18 frequency components. Stride results in how the kernel steps through the input, with the first layer having a stride of (1,2,6). Equivalently to kernel size, it steps through frequency components, e.g. assuming the convolution starts from F_0 , and ends at F_{18} , then with stride, the next convolution begins from F_6 , and ends F_{24} . The convolution parameters for each layer are summarized in Table 4.3. The activation function was ReLU for every layer, as well as batch normalization, normalizing over the three frequency bins. A dropout of 0.8 was used to have regularization in the network.

Table 4.3: Convolution layer summary

Layer	Kernel size	Stride	Padding
Individual convolution layer 1	(1, 2, 18)	(1, 2, 6)	valid
Individual convolution layer 2 & 3	(1, 1, 3)	(1, 1, 1)	valid
Merge convolution layer 1	(1, 2, 8)	(1, 1, 1)	same
Merge convolution layer 2	(1, 2, 6)	(1, 1, 1)	same
Merge convolution layer 3	(1, 2, 4)	(1, 1, 1)	same

The individual convolution layers from the article are twice as small compared to this implementation. The original paper approach contained real and imaginary components in a row, alternating between (Real, Im, Real, Im, ... etc.), and the FFT values were based on non-overlapping time samples. Hence, the input contained $X \in \mathbb{R}^{t \times m}$, where t was the time-dependent samples and m number of features per time. This input was processed on a 1D convolution, with kernel size (1,18), however, due to our split of real and imaginary components, we process twice the amount of features. The output layer of the architecture was the same as the original paper, two stacked GRU units were used with 120 hidden layers. In terms of regression tasks, due to the time dependence, this architecture was not adjusted for our problem. However, for classification, CrossEntropy was used as a loss function.

Chapter 5

Data-Driven Device Characterization

This chapter begins by showcasing measured signals, and the data set used in this research. The purpose is to understand the relations between measured signals and analyse each feature individually to assist in interpreting the estimation results.

Section 5.1 showcases the available experiments, and two types of data; measured signals via DAQ, and the corresponding one-time readings.

Section 5.2 covers correlations between features, which were acquired at the resonant frequency (ω), and the corresponding one-time readings of labels. Analogously, it covers more in-depth individual feature analysis and its linearities.

5.1 Measurements

This section covers two types of measured data - measurements from the controllers, i.e. labels in the data set, and measured signals from the capacitive and resistive sensing elements, i.e. raw data.

5.1.1 Labels

As mentioned in Chapter 4.2, the measurements were performed with varying parameters (mass flow, pressure and temperature) linearly. There were a couple of differences in performed measurements between fluids. Nitrogen was measured from 0 g/hr to 1.5 g/hr with steps of 0.3 g/hr due to its very low density. Water measurements similarly had different mass flow steps, measuring at steps of 1 g/hr. The rest of the fluids were measured with steps of 0.5 g/hr. An overview of the performed measurements of acetone can be seen in Figure 5.1 illustrating the sequential measurement process. The labels as ground truth were taken from measurements acquired from instruments summarized in Table 4.1. They are the pressure input (P_{in}), mass flow (ϕ_m), temperature ($T_{incubator}$) as well as the fluid name. Ground truth labels acquired via theoretical methods mentioned in Chapter 4.4, containing the temperature of the suspended tube ($T_{R_{track}}$), pressure drop (ΔP_{est}), density (ρ_{fl}) and dynamic viscosity (η).

In general, this data set contains 2130 Ethanol, 1125 IPA, 1107 Acetone, 617 H2O and 561 N2 measurements.

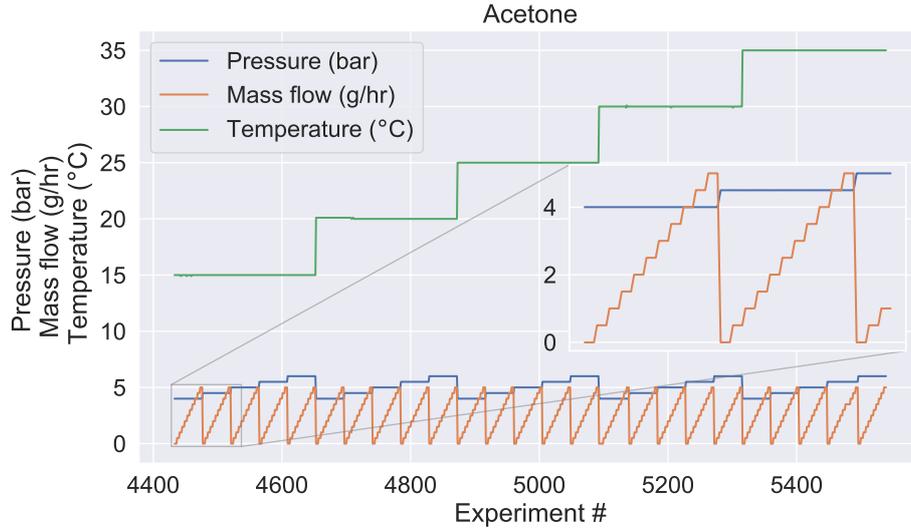


Figure 5.1: Performed Acetone measurements with varying temperature, pressure and mass flow, showcasing the structure of experiments

5.1.2 Raw data

As mentioned in Section 4.2, there are 3 dimensions of raw data representing a single measurement, and with the addition of pressure sensors (P_1 and P_2), as well as temperature-dependant resistive track (R_{track}), it can be expanded to 6 dimensions. The 3 dimensions represent actuation signal, and capacitive readouts measured for one second synchronously at 250 kHz, and can be seen in Figure 5.2a. Pressure sensors having a negative output and a one-time reading stretched to match the sampling length can be seen in Figure 5.2b.

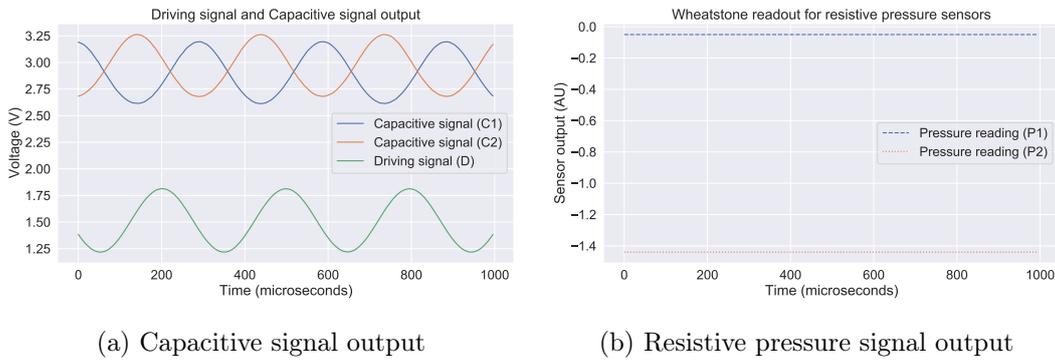


Figure 5.2: Sample of a time series data of a single experiment, showcasing raw data signals

5.2 Feature analysis

Due to high amounts of raw data, extracting features using certain techniques, e.g. DFT requires a lot of computation resources. While the resources were not an issue, this limits the efficient processing of such datasets with non-consumer-friendly machines (e.g. >150 GB RAM). As a workaround, DFT was attempted on a truncated data set - using only 5k samples out of 250k. This results in inadequately capturing small differences between the measurements, as it loses the resolution of small differences in frequencies.

The following analysis performed on the features attempts to capture relations to dependent variables - the parameters of the measurements which were mass flow, pressure, temperature and the fluid. As a way to correlate different fluids with respect to differences between measurements, instead of using fluid names, we use density ρ instead. These variables were correlated with respect to the extracted features such as the amplitude of the signal actuation (D) and capacitive signals (C1 and C2), resonance frequency (F_ω) and phase differences ($\Delta\varphi$). Correlations of acquired features with respect to the ground-truth labels (target) are summarised in Table 5.1. The initial linear correlation shows that the values correspond to theory: density and frequency of the signal have a linear relationship, and resistive pressure sensors have a linear correlation of 0.77, however, the current iteration of this specific design had pressure drift, which means the correlation might not be accurate across all fluids directly. Moreover, since these correlations contain all measurements and all liquids altogether, it does not show more complex relationships to viscosity. Mass flow is very correlated with the $\Delta\varphi$, although the correlation of $\Delta\varphi_{C1}$ and $\Delta\varphi_{C2}$ seems quite low due to the Nitrogen inclusion, which will be expanded in following Section 5.2.3.

Table 5.1: Feature correlation with respect to labels. Coefficients of $|r| > 0.5$ are highlighted

Features	Mass Flow	P_{in}	T_{pcb}	$T_{R_{track}}$	Density	Viscosity
Magnitude D	-0.003	-0.006	0.019	0.020	-0.003	-0.075
Magnitude C1	0.019	-0.004	-0.866	-0.869	0.299	0.336
Magnitude C2	0.040	-0.004	-0.770	-0.774	0.378	0.339
R_{track}	0.045	-0.009	0.999	0.999	-0.042	-0.265
Offset D	-0.110	0.013	-0.892	-0.869	-0.008	0.264
Offset C1	0.072	-0.009	0.999	0.997	-0.038	-0.268
Offset C2	0.073	-0.008	0.999	0.997	-0.035	-0.266
Phase C1	0.016	0.009	-0.011	0.007	0.014	-0.001
Phase C2	0.016	0.009	-0.011	-0.011	0.021	0.013
Frequency (F_ω) ¹	-0.320	0.020	0.012	0.016	-0.992	-0.483
P_1	0.402	-0.528	0.598	0.583	0.005	0.054
P_2	-0.114	0.774	-0.621	-0.617	-0.061	0.135
$P_1 - P_2$	-0.237	0.703	0.635	0.627	-0.041	-0.062
$\Delta\varphi_{C1}$	0.792	0.176	0.175	0.154	-0.321	-0.306
$\Delta\varphi_{C2}$	0.786	-0.038	-0.034	-0.061	0.824	0.326
$\Delta\varphi$	0.995	0.074	0.102	0.074	0.23	0.032

5.2.1 Frequency

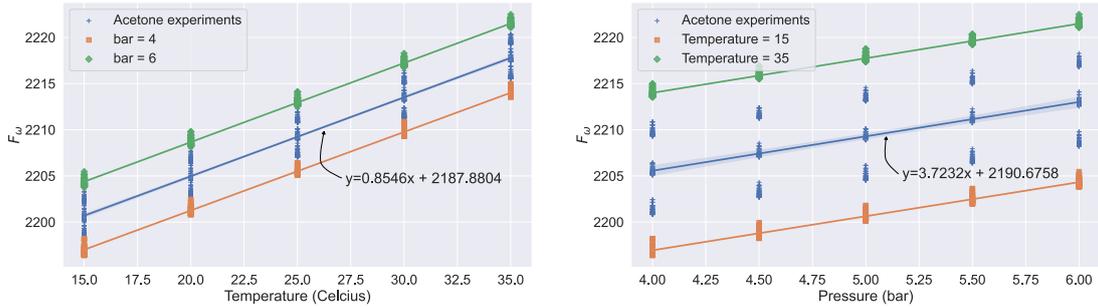
The acquired resonance frequencies (F_ω) of the capacitive signal from individual measurements using DFT have been compared to the 'refined' frequency from the curve-fitted signal. Applying DFT on the dataset, returns frequency bins of 1 Hz, not allowing to see fractional frequency. This has been observed via curve fitting the sinusoidal signal with Equation 2.14, which returns a frequency of 3367.43 Hz while DFT returns 3367 Hz. However, using curve-fitted frequency may result in introducing noise from the instrument itself.

¹Actuation frequency as well as the resonance frequencies of C1 and C2 are identical

Further analysing the dependence of frequency in terms of fluid characteristics, from theory it is known that frequency is highly dependent on the density of the fluid, however, pressure and temperature have an influence as well. This can be observed in Figure 5.4. This impact is reasonably correlated to fluid dynamics, i.e. the relation of thermal expansion changes density, and pressure has more impact on gasses due to its compressibility. Empirically it has been observed that the temperature has less impact compared to the pressure in frequency change, e.g. temperature for Acetone changes by ≈ 0.86 Hz per $^{\circ}\text{C}$, while pressure influences by ≈ 3.6 Hz per Bar. This is due to the dependence of pressure on the spring constant $k(P)$ denoted in Equation 2.17. The spread of experiments in Figure 5.3a is due to pressure dependence, while the spread in Figure 5.3b shows the temperature dependence. To capture this relationship, we use

$$F_{\omega_e} = \alpha_T \cdot T_{\text{Rtrack}} + \beta_p \cdot P_{\text{in}} + F_0 \quad (5.1)$$

where T_{Rtrack} is the temperature of the suspended tube (based on Section 4.4.1), and P_{in} is the pressure input to the device. Performing linear fit using coefficients α_T and β_p , we get the so-called temperature and pressure coefficients, with frequency offset F_0 . With this equation, we try to capture the relationship of pressure and temperature effect per liquid on frequency. This is summarized for individual fluid in Table 5.2. B_{iq} is the thermal expansion coefficient at 20°C . This coefficient was calculated using the regression formula mentioned in Equation 2.3 with regression coefficients taken from [14], and the correlation between fitted temperature coefficients and thermal expansion coefficient is > 0.99 , however, the sample size (different fluids) is quite low to determine the significance. The explanatory power of frequency using independent variables such as temperature and pressure is quite high. Acetone achieves R^2 of 0.998, however, it achieves the lowest explanatory power on Nitrogen. Furthermore, the temperature coefficient may indicate some fluid properties, while the pressure coefficient might be too dependent on the spring constant to derive useful properties directly, and would require first finding the spring constant value itself.



(a) Frequency dependency on the temperature of the fluid. The fitted middle line represents the average temperature influence on frequency. (b) Frequency dependency on the pressure applied. The fitted middle line represents the average pressure influence on frequency.

Figure 5.3: Frequency dependency

¹At 20°C and 4 bar(a)

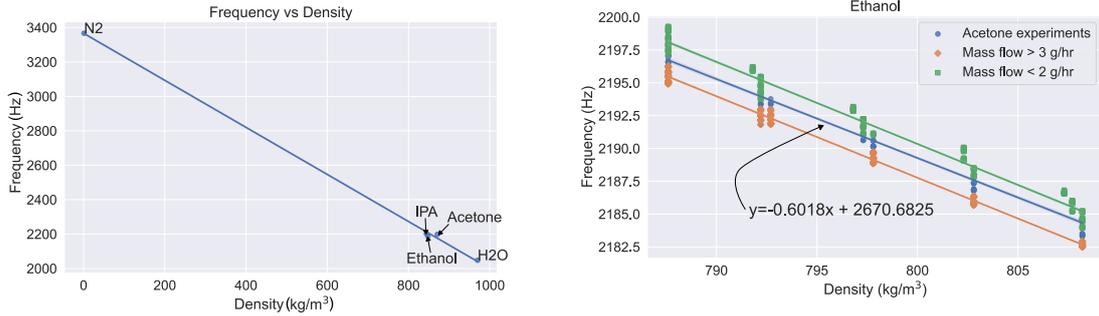
²thermal expansion coefficient

Table 5.2: Frequency dependency on temperature and pressure

Fluid	Pressure coef	Temperature coef	Density ¹	B_{iq} coefficient ²	R^2
Ethanol	3.499 ± 0.206	0.659 ± 0.065	806.4 kg/m^3	$1.159 \cdot 10^{-3}$	0.932
Acetone	3.704 ± 0.094	0.869 ± 0.030	796.5 kg/m^3	$1.458 \cdot 10^{-3}$	0.998
IPA	3.294 ± 0.266	0.652 ± 0.084	810.8 kg/m^3	$1.095 \cdot 10^{-3}$	0.904
H2O	3.498 ± 0.231	0.161 ± 0.073	998.4 kg/m^3	$3.715 \cdot 10^{-4}$	0.892
N2	2.727 ± 0.227	-0.062 ± 0.072	4.662 kg/m^3	-	0.838

5.2.2 Frequency and density

According to theory, frequency is inversely proportional to the square root of the density, and in the general sense, density is determined by frequency as shown in Figure 5.4a however, by itself, it's not an accurate measure of fluid density. The influence of temperature and pressure affects the frequency. Such dependence can be visualised in Figure 5.4b, where pressure is set to constant, and the frequency varies by ≈ 5 Hz. Estimating density, as a direct linear measure of frequency, on average, can make an error of $\approx 15 \text{ kg/m}^3$, depending on mass flow, pressure and temperature. This is due to the complex relationship of temperature, pressure and mass flow affecting the density of the fluid, as well as the device, e.g. pressure and temperature changes as mass flow changes.



(a) Frequency and density, linearity illustration (b) Ethanol at constant pressure and varying temperature, illustrating small fluid state dependence (mass flow) on frequency.

Figure 5.4: Frequency and density linearity

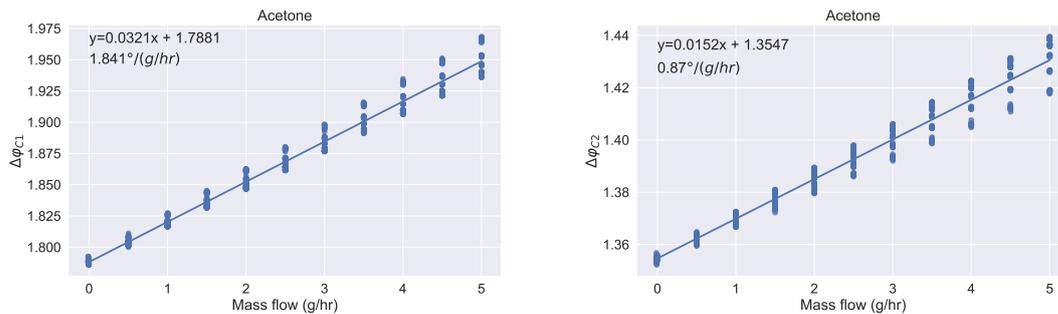
5.2.3 Phase difference

As explained in Section 2.2, the phase difference between the capacitive signal is linearly dependent on mass flow. However, observing the data points with varying temperature and pressure shows a spread of $\Delta\varphi_{C1}$ and $\Delta\varphi_{C2}$ per mass flow in g/hr. It is important to extract the $\Delta\varphi$ which would be as linear as possible to the mass flow. Three different methods were tested to extract the phase difference.

Comparison between the methods is made via a linear fit ($\phi_m = \alpha\Delta\varphi + o$), which explains the sensitivity of how much phase difference between measured capacitive signal α flow coefficient in terms of radians per g/hr, with zero flow offset. Analysing the differences in coefficients between methods: DFT, cross-correlation and curve fitting do not result in substantial differences on average. Coefficients vary by $\pm 10^{-4}$ or $\approx 0.005^\circ$, and the offset by $\pm 10^{-3}$, or $\approx 0.05^\circ$. However, the variance of phase difference for mass flow measurement

is much higher for cross-correlation and curve fitting methods. One of the main reasons could be the harmonics present in the signal, as the variance is even more evident when applying the cross-correlation directly between C1 and C2 signals.

Proceeding with DFT as the best-performing method to acquire phase difference, individual phase difference, i.e. phase difference between actuation and capacitive readout signals shows higher variance at higher mass flows, as seen in Figure 5.5a and 5.5b. This variance is removed if the difference between phase differences is taken, as shown in Figure 5.6.



(a) Capacitive signal phase difference between the driving signal and C1 readout (b) Capacitive signal phase difference between the driving signal and C2 readout

Figure 5.5: Sensitivity of phase difference relative to the mass flow

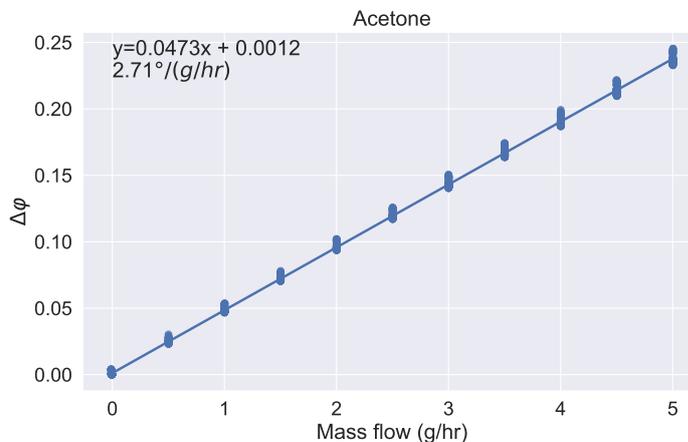


Figure 5.6: $\Delta\varphi$ calculated as the phase difference between capacitive signals $|\varphi_{C1} - \varphi_{C2}|$ and plotted relative to mass flow.

The sensitivity can be fitted for each fluid separately, to see whether fluid characteristics influence phase difference. The influences could be the thermal conductivity of the fluid, e.g. how fast it could cool down the tube to affect the sensing element. One of the noticeable effects is density, which affects the sensitivity of the sensor. Table 5.3 summarizes the sensitivity for each fluid. This shows that the phase difference and the offset did not change per liquid substantially. These differences can be explained due to the imbalanced dataset per fluid, where some measurements had fewer data points at lower or higher temperatures (i.e. the same measurement sweep may have been performed multiple times at a certain temperature). The main difference can be observed for Nitrogen, as clearly the

offset for zero mass flow is different, as well as phase difference is more sensitive to a mass flow change. According to $\Delta\varphi$ coefficients, the sensitivity correlates to denser fluids being less sensitive (e.g. water compared to acetone), in contrast to nitrogen, being the lowest density gas, with the most sensitivity. All coefficients and their offsets are significant, with a P-value much lower than <0.001 . A summary plot of sensitivities is shown in Figure 5.7, at a constant temperature, and constant pressure. Since Nitrogen was measured at only up to 1.5 g/hr, the plot shows only $\Delta\varphi$ for ≤ 1 g/hr. At constant temperature and pressure, liquids correspond to the same sensitivity, hence they overlap.

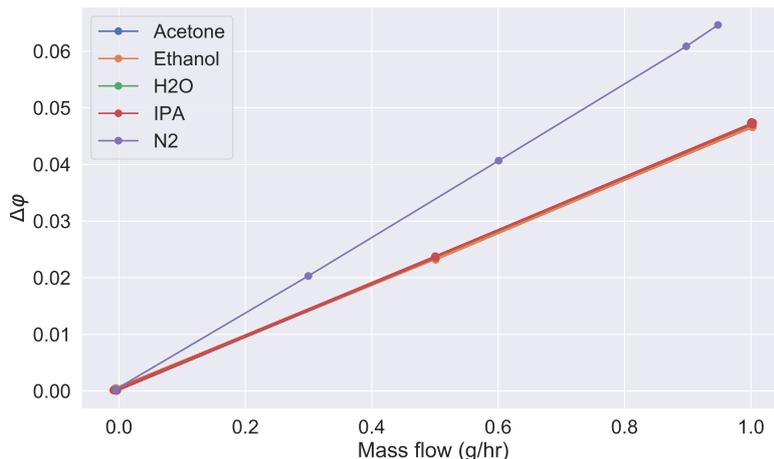


Figure 5.7: $\Delta\varphi$ sensitivity of all fluids at constant temperature and pressure. The figure shows $\Delta\varphi$ to ≤ 1 g/hr.

Table 5.3: Comparison $\Delta\varphi_{C_i}$ sensitivity to mass flow

Fluid	Coef. $\Delta\varphi_{C_2}$	Offset C2	Coef. $\Delta\varphi_{C_1}$	Offset C1	Coef. $\Delta\varphi$	Offset $\Delta\varphi$
Ethanol	0.0151	1.356	0.0323	1.786	0.0473	0.0007
Acetone	0.0152	1.354	0.0321	1.788	0.0473	0.0012
IPA	0.0147	1.355	0.0333	1.785	0.0479	-0.0
H2O	0.0146	1.370	0.0297	1.77	0.0441	0.0005
N2	0.0223	1.269	0.0459	1.876	0.0681	0.0038

Assuming a perfect linear relationship of $\Delta\varphi$ to mass flow for all liquids, we should expect closer to 1% Full-Scale (FS) accuracy (including errors from the measurement setup), however, without compensating temperature, or pressure, it achieves an FS accuracy of approximately 4.5 %, as shown in Figure 5.8. Moreover, the errors are not normal distributions, which indicates that fluid properties affect mass flow calculations. One of such discussed properties is the fluid density, which results in mass flow being over-estimated. This is also seen in the same figure, starting at 20%, and every additional 20% ¹ of flow meters full range, an additional skew in the distribution.

¹Water was measured at a step size of 1 g/hr

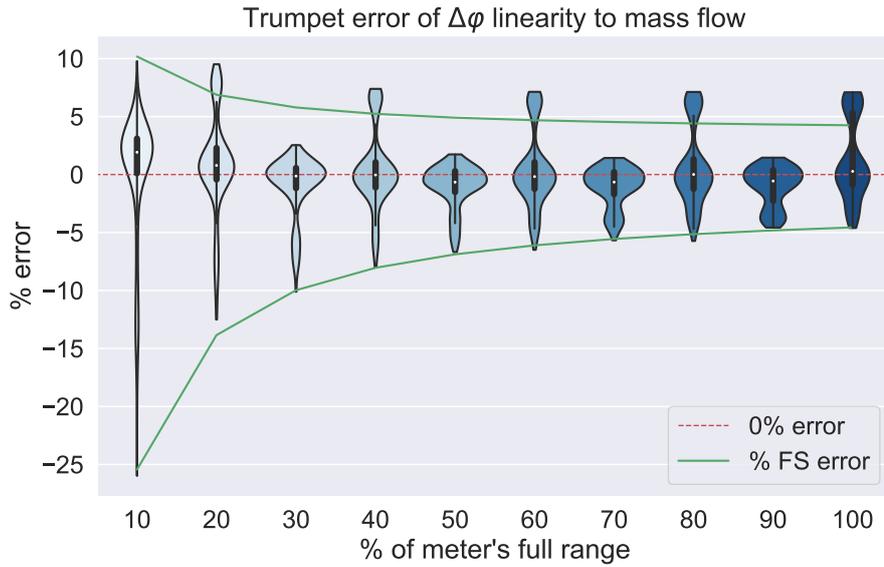
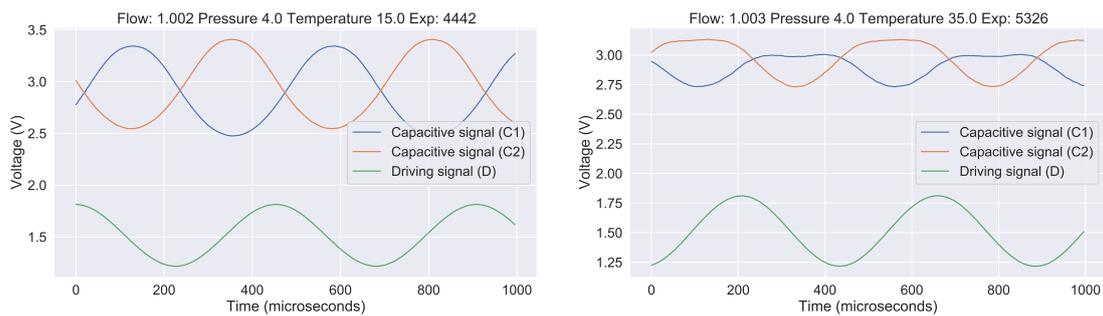


Figure 5.8: (FS) accuracy curve including water, isopropanol, acetone and ethanol. Excluding nitrogen due to its different sensitivity.

5.2.4 Magnitude

According to the initial linear correlations, the magnitude of the capacitive signal had some linear correlation to temperature, while the magnitude of the driving signal, had no linear correlation to any of the dependent variables. This of course is logical, as the driving signal is simply actuation, independent of the measurement parameters (mass flow, pressure, fluid), although temperature may affect the wires, it seems the hardware internally compensates for this change. However, at higher temperatures, it can be observed that capacitive signal produces harmonics, this can be seen in Figure 5.9 at a constant temperature and pressure, and for a small sample size.



(a) Capacitive signal output at low temperature (15 °C) with no visible deformations (b) Capacitive signal output at higher temperature (35 °C) with visible deformations

Figure 5.9: Capacitive signal outputs at two different temperatures (15 °C and 35 °C), with the same pressure and mass flow readings. Capacitive signal readouts at higher temperatures (b) have a ‘flattened’ signal, compared to the same measurement conditions, except at lower temperature (a).

Modelling only the fundamental magnitude relation of capacitive signal measurement

using magnitude from DFT, two models were attempted. One is a quadratic relationship in only in terms of suspended tube temperature, where A_C is the magnitude of the capacitive signal:

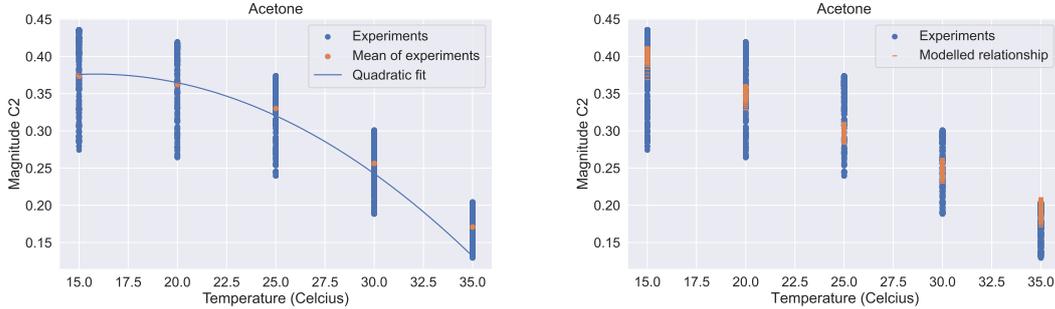
$$A_C = a(T_{R_{\text{track}}} - s)^2 + o \quad (5.2)$$

where $T_{R_{\text{track}}}$ is the temperature of the suspended tube, a , s and o are the quadratic term coefficients. This relation can be seen in Figure 5.10. The second model utilises environmental conditions of the measurement, i.e. temperature, pressure, mass flow and resonance frequency. This is formalized in an equation as:

$$A_C = (\alpha_t \cdot T_{R_{\text{track}}} + \alpha_\phi \cdot \phi_m + \alpha_f \cdot F_\omega + \alpha_p \cdot P_{in}) + A_0 \quad (5.3)$$

where $T_{R_{\text{track}}}$, P_{in} , ϕ_m and F_ω are the experiments temperature, pressure, mass flow and frequency correspondingly, and coefficients α_t , α_ϕ , α_f , and α_p correspond as relationship weight to these measurement conditions.

The spread of the experiments influenced the fitting significantly to get accurate estimates of the coefficients and did not fit nicely per individual fluid. Applying the first model (eq. 5.2) to fit temperature dependence is shown in Figure 5.10a. The line fits the mean of the spread quite well, however at 35 °C it fails to fit. In an attempt to explain the spread of the magnitude, involving parameters such as temperature, pressure, mass flow and frequency (using eq. 5.3) to explain capacitive signal magnitude did not show significant results (see 5.10b). This relationship on average per liquid explains 70% of the spread (R^2 score of 0.7), with frequency having the most impact.



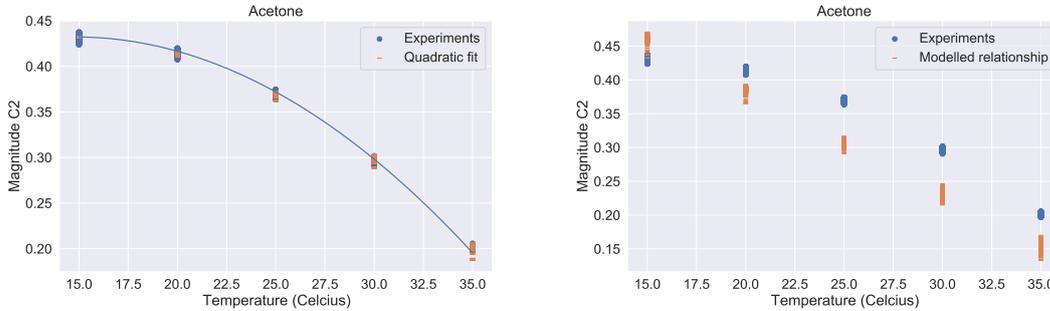
(a) Estimating magnitude relation to temperature in terms of quadratic fit using DFT magnitude
 (b) Modelling magnitude relation to temperature, mass flow, pressure and frequency using DFT magnitude

Figure 5.10: Capacitive signal magnitude (acquired via DFT) relation to temperature, and the fits which try to explain the non-linear behaviour

Using magnitude acquired from DFT has quite a big spread between experiments for a constant temperature. This is because harmonics affect the fundamental signal magnitude components. The driving signal magnitude does not have any dependence on the temperature, and the electronics itself attempts to keep it stable, hence it was not further analysed.

In an attempt to reduce harmonic ‘noise’ from the magnitude acquired via DFT, curve-fitted magnitude is used instead. This has shown the spread is much more stable at a constant temperature, as well as exhibits the same non-linear behaviour. Modelling the temperature dependence using Equation 5.2 shows quite an accurate fit across all fluids ($R^2 > 0.99$), and the fit for Acetone can be seen in Figure 5.11a. Using the relationship of temperature, pressure, mass flow and frequency again on the noise-reduced magnitude,

does not explain the spread of the magnitude per experiment (see Figure 5.11b). These results indicate that the fundamental signal magnitude has a quadratic relationship to temperature (and mass flow due to the cooling effect on the tube), and is independent of pressure, however dependant on fluid properties. Coefficients of the quadratic relation in terms of temperature are summarised in Table 5.4.



(a) Estimating magnitude relation to temperature in terms of quadratic fit using curve fitted magnitude
 (b) Modelling magnitude relation to temperature, mass flow, pressure and frequency using curve fitted magnitude

Figure 5.11: Capacitive signal magnitude (curve fitting) relation to temperature, and the fits which try to explain the non-linear behaviour

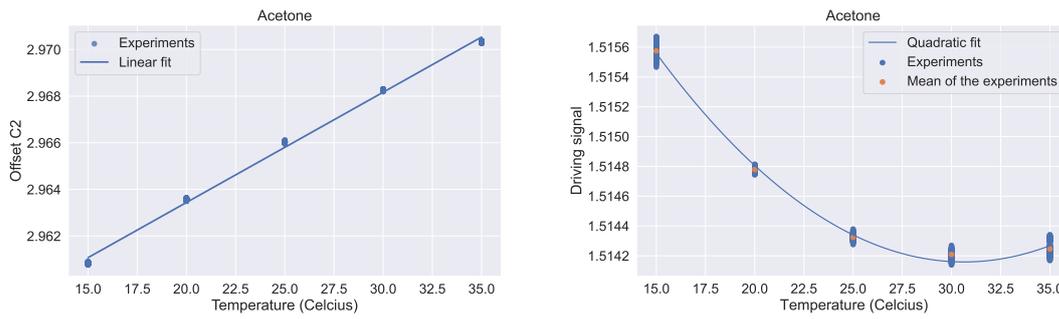
Table 5.4: Magnitude quadratic coefficients using curve fitted magnitude, with fitted coefficients a , s , o for C2 and C1 capacitive signals

Fluid	a_{C2}	s_{C2}	o_{C2}	a_{C1}	s_{C1}	o_{C1}
Ethanol	$-5.58 \cdot 10^{-4}$	18.016	0.439	$-5.01 \cdot 10^{-4}$	12.24	0.469
Acetone	$-5.68 \cdot 10^{-4}$	18.066	0.432	$-4.89 \cdot 10^{-4}$	12.44	0.457
IPA	$-5.88 \cdot 10^{-4}$	18.204	0.437	$-5.17 \cdot 10^{-4}$	12.88	0.464
H2O	$-7.06 \cdot 10^{-4}$	19.352	0.482	$-6.62 \cdot 10^{-4}$	15.34	0.500
N2	$-4.11 \cdot 10^{-4}$	20.201	0.289	$-3.76 \cdot 10^{-4}$	15.99	0.295

5.2.5 Offset

Similarly, as the magnitude of the signal, the offset of the capacitive signal is more linear to temperature. This relationship can be modelled with a simple linear equation ($y = ax + b$), where a denotes the temperature sensitivity as shown in Figure 5.12a. Offset dependence on the temperature is very minimal. Capacitive signal 2 changes $\approx 4.73 \cdot 10^{-4}$ (V) or ≈ 0.473 (mV) per $^{\circ}\text{C}$. This change seems independent of other parameters, e.g. fluid, pressure, and mass flow. The same behaviour is apparent on the offset of the Capacitive signal 1, with a sensitivity of $\approx 4.65 \cdot 10^{-4}$ (V) or ≈ 0.465 (mV) per $^{\circ}\text{C}$. This is unlikely to have much of an influence on the sensor.

Interestingly, the offset of the driving signal has a non-linear relation to temperature. This may be due to the current decreasing with temperature and electronics compensating for the change. The fitted coefficients uses the Equation 5.2, with coefficients being $\approx 5.724 \cdot 10^{-6}$ for linear term, ≈ 1.514 shift, and ≈ 30.630 offset [Figure 5.12b]



(a) Estimating C2 signal offset with relation to the temperature with linear fit (b) Estimating driving signal offset with relation to the temperature with quadratic fit

Figure 5.12: Capacitive signal offset (a) linear dependence to temperature, and the fit which explains the relation. Driving signal offset (b) non linear behaviour, and its non-linear fit explaining the behaviour

5.2.6 Pressure sensor

This iteration of pressure sensors has a linear temperature dependence and introduces offset to the readings as shown in Figure 5.13. The offset seems to be around 0.2 (scaled voltage units) per 5 °C.

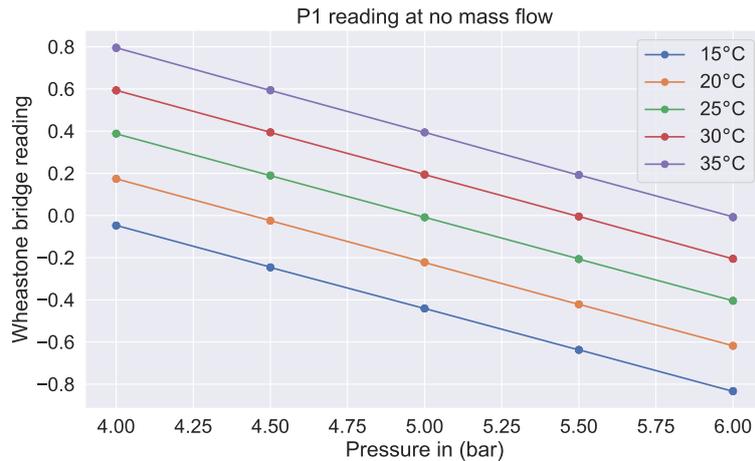


Figure 5.13: Temperature dependence on pressure sensor readings. The figure is based on no mass flow measurements, for constant pressure on the system. The y axis is a scaled value of the wheatstone bridge reading.

Chapter 6

Fluid characteristics estimation

This chapter covers the results of fluid characteristics estimations using a linear combination of features, which are most interpretable physically, as well as including non-linear methods, such as Support Vector Regression (SVR) with Radial Basis Function kernel (RBF), and Gaussian Processes. The estimation is performed on the mass flow of the fluids, temperature, pressure drop, viscosity, density, and classifying what fluid was being measured.

6.1 Mass flow estimation

This section covers mass flow estimation of various fluids using linear and non-linear methods. The first part of mass flow estimation is based on the expected linearity of the device, hence a linear combination of features is used to estimate the mass flow. The second part is based on non-linear methods, which mainly constitute Radial Basis Function Kernel (RBF) and Support Vector Regression to include robustness to outliers.

6.1.1 Linear regression

Linear models depend on the assumption that the relationship between the dependent and the independent variable is linear. The choice of features is based on the correlation metric with respect to the mass flow. This was analysed in Section 5.2, where the main features are $\Delta\varphi_{C_i}$, P_1 and Frequency (F_ω). The feature $\Delta\varphi_{C_i}$ with subscript i describes whether it is the phase difference between actuation signal and capacitive signal 1, or 2 or the phase difference between both capacitive signals, named simply as $\Delta\varphi$. Although the linear correlation to the mass flow of phase differences between actuation and capacitive signals is quite low, this is due to nitrogen having different properties, however, the difference between the two signals compensates for this linearity. To estimate the mass flow more accurately, we also can include R_{track} to compensate for temperature. The linear correlation between features is also quite low. Colinearity occurs between $\Delta\varphi_{C_i}$ features, where $\Delta\varphi$ has correlation with $\Delta\varphi_{C1}$ of 0.84, and between $\Delta\varphi$ and $\Delta\varphi_{C2}$, with correlation of 0.73. Interestingly, the correlation is quite low between $\Delta\varphi_{C1}$ and $\Delta\varphi_{C2}$. Similarly, frequency has a correlation of -0.82 with $\Delta\varphi_{C2}$. This is summarised in Table 6.1, where coefficients with $|r| > 0.5$ are in bold.

Although $\Delta\varphi_{C_i}$ features should be quite linear to each other, it seems the linearity is influenced by some physical properties. This could be due to uneven stress on the geometry of the suspended tube, i.e. on the left and right parts of the tube due to pressure and mass flow.

Table 6.1: Linear colinearity between features

Features	$\Delta\varphi_{C1}$	$\Delta\varphi_{C2}$	$\Delta\varphi$	R_{track}	P_1	F_ω	Mass flow
$\Delta\varphi_{C1}$	1	0.25	0.84	0.15	0.44	0.3	0.79
$\Delta\varphi_{C2}$	0.25	1	0.73	-0.06	0.20	-0.82	0.78
$\Delta\varphi$	0.84	0.73	1	0.074	0.43	-0.24	0.99
R_{track}	0.15	-0.06	0.07	1	0.58	0.02	0.04
P_1	0.44	0.20	0.43	0.58	1	-0.01	0.40
F_ω	0.3	-0.82	-0.24	0.02	-0.01	1	-0.32

We apply weights to individual features to estimate mass flow with the linear model. Three linear models were used:

Linear model 1, is described as:

$$y = \beta_0 + \beta_1\Delta\varphi_{C1} + \beta_2R_{\text{track}} + \beta_3F_\omega + \beta_4P_1$$

Linear model 2, is described as:

$$y = \beta_0 + \beta_1\Delta\varphi_{C2} + \beta_2R_{\text{track}} + \beta_3F_\omega + \beta_4P_1$$

Linear model 3, is described as:

$$y = \beta_0 + \beta_1\Delta\varphi + \beta_2R_{\text{track}} + \beta_3F_\omega + \beta_4P_1$$

The main difference between the models is which fundamental linear relationship we use to estimate the mass flow, i.e. which $\Delta\varphi_{Ci}$. This allows us to see the feature differences with respect to the mass flow. Including all features would not result in a higher performance, as well as multi-colinearity between features affects interpretability.

Between the three tested models, simple linear relations to compensate for the difference between gas and liquids, as well as adjust the estimation with temperature, frequency and pressure coefficients, shows that there are minor differences between $\Delta\varphi$ with respect to estimating gas altogether. The phase difference between two capacitive signals shows the best performance in regards to linearity, as shown in Table 6.2.

Table 6.2: Linear models R^2 scores, trained on every fluid, evaluated for individual fluid only from the test set (70%). The red color denotes the nitrogen inclusion effect

Fluid	R^2 (Model 1)	R^2 (Model 2)	R^2 (Model 3)
Acetone	0.995 (-0.001)	0.990 (-0.001)	0.999 (-0)
Ethanol	0.994 (-0)	0.989 (-0.001)	0.999 (-0)
Water	0.992 (-0.02)	0.989 (-0.01)	0.996 (-0.005)
Isopropanol	0.993 (-0.002)	0.986 (-0.001)	0.999 (-0)
Nitrogen	0.802	0.691	0.799

Linear estimations including Nitrogen do not work due to the phase difference sensitivity between gas and liquid. Gasses appear to be much more sensitive as analysed in Section 5.2.3. Interestingly, using $\Delta\varphi_{C1}$ as the main linear feature, it tends to underestimate the mass flow as seen in Figure 6.1a, while in contrast, $\Delta\varphi_{C2}$ tends to overestimate mass flow as seen in Figure 6.1b. These differences are nullified if $\Delta\varphi$ is applied to estimate the mass

flow as shown in Figure 6.2. Same performance is reached as Model 3, if features $\Delta\varphi_{C1}$ and $\Delta\varphi_{C2}$ are used together, instead of $\Delta\varphi$.

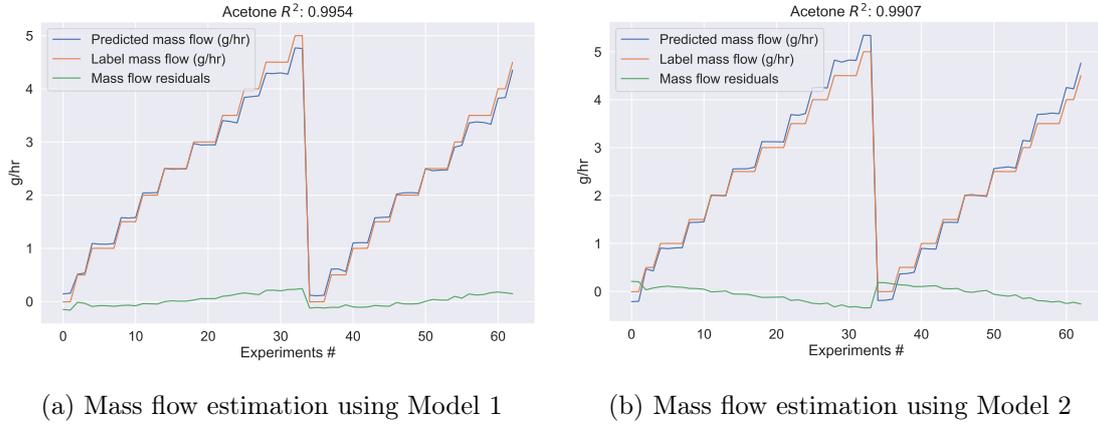


Figure 6.1: Differences in estimating mass flow using $\Delta\varphi_{C1}$ and $\Delta\varphi_{C2}$

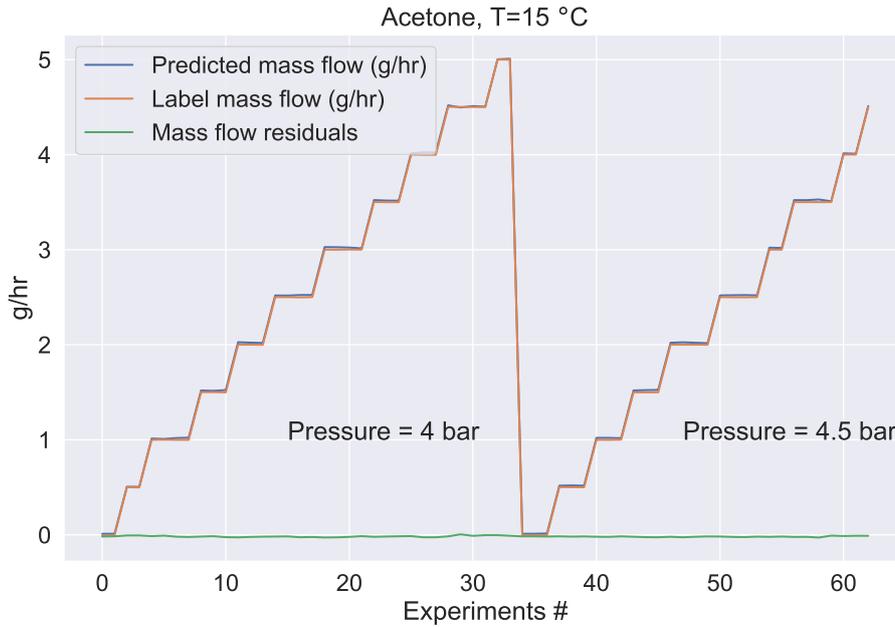


Figure 6.2: Mass flow estimation using Model 3

Weight differences between the models show the influence of frequency and temperature correction as shown in Table 6.3. Including Nitrogen in the training, there is more emphasis on frequency correction, since the gas has a much higher frequency due to its density compared to fluids. The majority of the features are very significant, with a P-value much lower than 0.05. The exception is the P_1 feature, on model 1, with a P-value of 0.041, and for model 2, P_1 is insignificant.

The best-performing linear model with $\Delta\varphi$ achieves FS accuracy of $\approx 2.5\%$, and the trumpet error curve is shown in Figure 6.3. This already outperforms the conventional method of using just the phase difference by a factor of 2.

Table 6.3: Linear model weights (excluding nitrogen)

Weights	Model 1	Model 2	Model 3
$\beta_1 (\Delta\varphi_{Ci})$	6.238	6.516	5.261
$\beta_2 (R_{\text{track}})$	-0.414	0.590	-0.080
$\beta_3 (F_\omega)$	-0.840	1.0778	0.198
$\beta_4 (P_1)$	-0.042	0.0012	-0.063

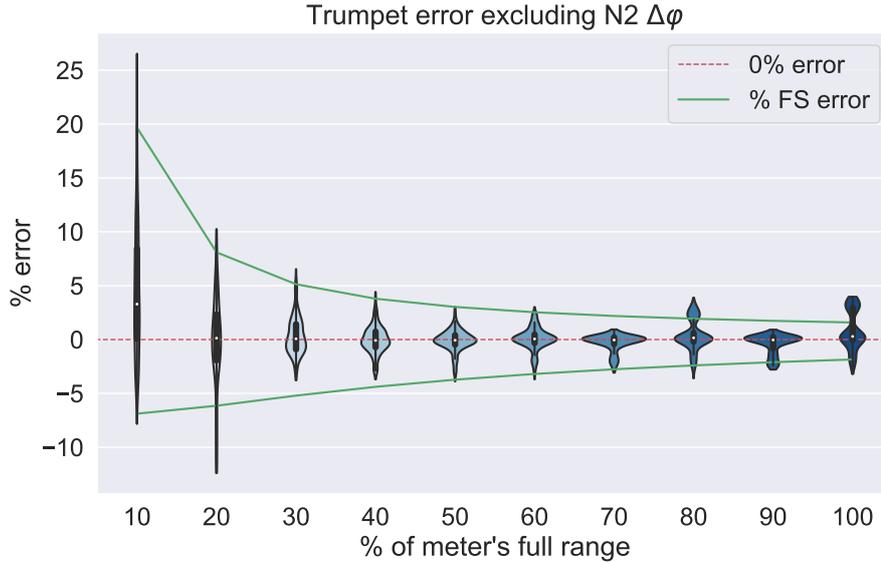


Figure 6.3: (FS) accuracy curve including water, isopropanol, acetone and ethanol from the testing dataset (70%)

Polynomial features

To further investigate non-linearities in the sensing elements, we can apply a polynomial transformation to our feature space, acquiring additional features such as $[\Delta\varphi^2; \Delta\varphi \cdot R_{\text{track}}; \Delta\varphi \cdot F_\omega; \Delta\varphi \cdot P_1; R_{\text{track}}^2, \dots; P_1^2]$ which returns additional 10 features. By selecting one polynomial feature, and testing the performance on Nitrogen, while training on the entire dataset, we can find the most significant polynomial feature to estimate more accurately, as well as keeping the complexity of the model low. Such brute force method found that the polynomial feature of $\Delta\varphi \cdot F_\omega$ is the most significant to estimate mass flow, which compensates for $\Delta\varphi$ sensitivity. This relationship physically can be seen as the fluid density effect on mass flow, as by definition, density is the mass contained in a unit volume. Unfortunately, the colinearity between $\Delta\varphi$ and the new polynomial feature is quite high (0.989), however, without $\Delta\varphi$, the mass flow estimation performs sub-optimal. We define a model:

Linear model 4

$$y = \beta_0 + \beta_1 \Delta\varphi + \beta_2 (\Delta\varphi \cdot F_\omega) + \beta_3 R_{\text{track}} + \beta_4 P_1$$

This model outperforms model 3 by a small margin, however, allows it to be a much more accurate estimate of Nitrogen, achieving a R^2 score of 0.993. The model with these polynomial features achieves an FS accuracy of $\approx 2.6\%$, however, if we do not take into

account the Nitrogen, it is closer to $\approx 2\%$ across all fluids. The features are all significant, with a P value of < 0.001 . The FS accuracy trumpet curve is shown in Figure 6.4.

Table 6.4: Linear regression coefficients with most significant polynomial features (including nitrogen)

Feature	Weight
$\beta_1 (\Delta\varphi)$	8.691
$\beta_2 (\Delta\varphi \cdot F_\omega)$	-3.467
$\beta_3 (R_{\text{track}})$	-0.091
$\beta_4 (P_1)$	0.0382

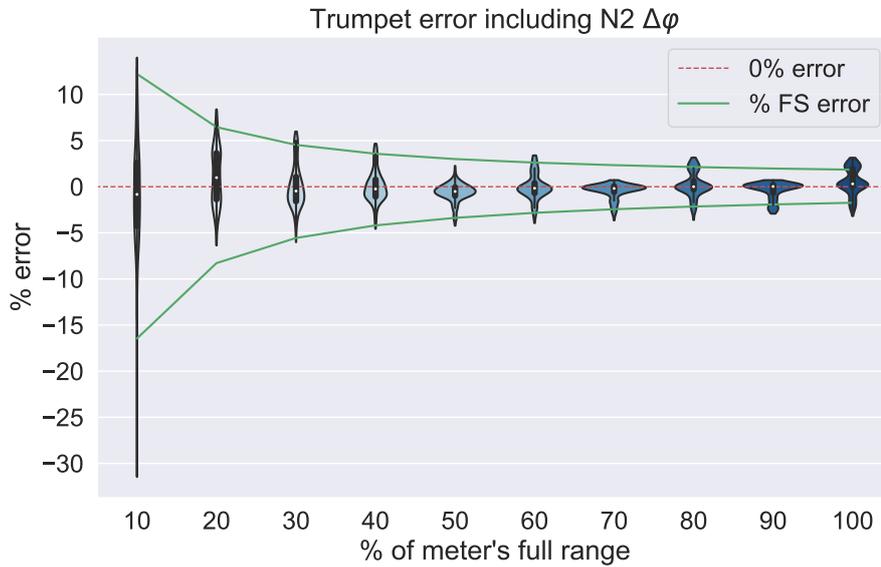


Figure 6.4: (FS) accuracy curve of a linear model including water, isopropanol, acetone, ethanol and nitrogen from the testing dataset (70%)

6.1.2 Non-linear regression

To allow much more non-linearity in estimations, as well as introduce hyperparameters, which help adjust the model, we apply the Support Vector Regression algorithm. Additionally, with the Radial Basis Function kernel, we project the feature space into infinite dimensional space. The same features are used as the best-performing linear model (excluding polynomial features) - $\Delta\varphi$, F_ω , R_{track} and P_1 . Polynomial features are excluded because the RBF projects in high-dimensional space for us directly, and we avoid colinearity between features. Since the correlations between features are quite low, SHAP values are interpretable for the estimations. Practically all fluid mass flows are explained with $0.999 R^2$, with the exception of Nitrogen, which has an R^2 of 0.998. However, this non-linear model achieves FS accuracy of $\approx 1\%$, which is influenced by the accuracy of the device we are testing, and the accuracy of the ground-truth. The trumpet error curve, showcasing FS accuracy is shown in Figure 6.5.

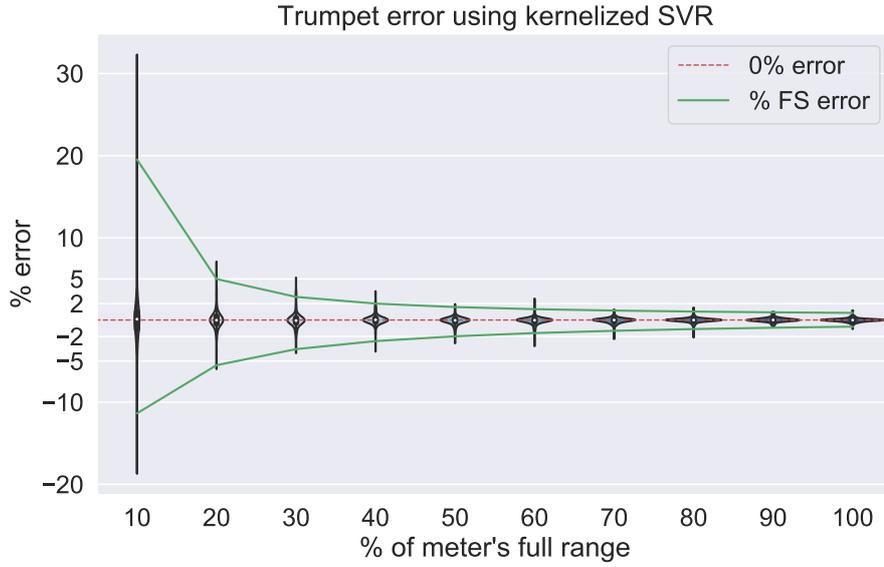


Figure 6.5: (FS) accuracy curve of a non-linear model including water, isopropanol, acetone, ethanol and nitrogen from the testing dataset (70%)

Feature significance for the non-linear model, shown in Figure 6.6, describes that $\Delta\varphi$ is used as the main feature to estimate the mass flow, frequency compensates for Nitrogen mass flow estimation, capturing $\Delta\varphi \cdot F_\omega$ relation, Rtrack has minor influence to compensate for temperature, and P_1 compensates at higher pressures.

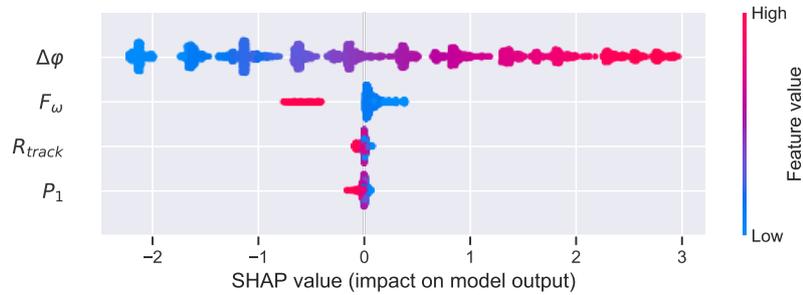


Figure 6.6: Non-linear mass flow estimation using SVR, feature significance based on Shapley values

Table 6.5: Mass flow estimation Summary of models (FS) accuracy. Red denotes Nitrogen inclusion effect on accuracy

Fluid	Linear model (poly features)	Non-linear model	$\Delta\varphi \propto$ mass flow
Acetone	$\approx 2.2\%$ (2.8%)	$\approx 1\%$ (1.1%)	$\approx 4.5\%$
Ethanol	$\approx 2.6\%$ (3.1%)	$\approx 1\%$ (1.1%)	$\approx 4.5\%$
Water	$\approx 1.6\%$ (3.1%)	$\approx 0.6\%$ (0.84%)	$\approx 7\%$
Isopropanol	$\approx 1.8\%$ (2.2%)	$\approx 1.3\%$ (1.1%)	$\approx 4\%$
Nitrogen	$\approx 1.8\%$	$\approx 1.8\%$	-

6.2 Temperature

Temperature can easily be estimated using one of the capacitive signal offsets. Taking a Fourier transform of C1 or C2, the 0 Hz offset present in the frequency domain is linearly sensitive to temperature. This temperature linearity was also analysed in Section 5.2.5. To add some more complexity, we aim to estimate the temperature drop, i.e. how much does the fluid cool down the suspended capacitive tube, with a goal to expand the analysis on relationships to temperature. The temperature drop label was acquired using linear fit on a resistive track, which is explained in the methodology section 4.4.1. The drawback of temperature estimation is that it was performed in laboratory conditions, and it is uncertain how well it would work in real conditions.

6.2.1 Linear models

The most linear features to temperature, as well as correlations between the features, are present in the table below. There is no significant difference between the two capacitive signal offsets, magnitude C1 is more linear to temperature, and the colinearity between magnitude C2 is quite high. Pressure sensors also hold some linear relation to temperature, as was shown in Section 5.2.6, the temperature introduces offset in the pressure reading.

Table 6.6: Linear colinearity between features

Features	Magn. C1	Magn.C2	Offset C1	OffsetC2	P_1	P_2	$T_{R_{track}}$
Magn. C1	1	0.98	-0.86	-0.85	-0.51	0.51	-0.87
Magn. C2	0.98	1	-0.76	-0.76	-0.46	0.44	-0.77
Offset C1	-0.86	-0.76	1	0.99	0.59	-0.62	0.99
Offset C2	-0.85	-0.76	0.99	1	0.59	-0.62	0.99
P_1	-0.51	-0.46	0.59	0.59	1	-0.84	0.58
P_2	0.51	0.44	-0.62	-0.62	-0.84	1	-0.61
$\Delta\varphi$	-0.02	-0.01	0.10	0.10	0.43	-0.12	0.07

However, excluding the mass flow feature from the estimation, it does not estimate temperature drop based on mass flow. Hence, for the final linear model, we include $\Delta\varphi$ as a feature, and exclude P_2 .

Defining the **linear model** as :

$$y = \beta_0 + \beta_1\Delta\varphi + \beta_2o_{C1}$$

Table 6.7: Linear regression coefficients with most significant polynomial features

Feature	Weight
$\beta_1 (\Delta\varphi)$	-0.7598
$\beta_2 (\text{OffsetC1})$	20.3670

Temperature set points are mainly estimated by offset, and the temperature drop from the flow is corrected by $\Delta\varphi$. Since the temperature drops similarly across the fluids, it is quite consistent. A sample of temperature drop estimation is shown in Figure 6.7 for Acetone.

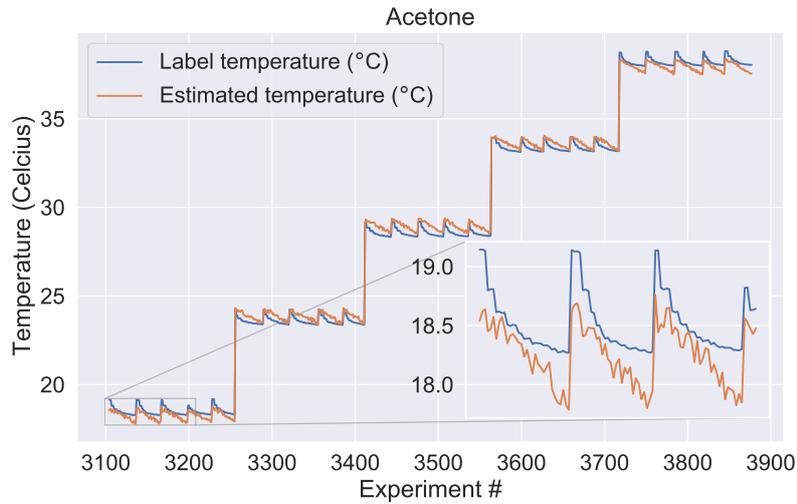


Figure 6.7: Sample of linear temperature estimation, and the cooling down based on mass flow of Acetone

The temperature linearly drops down for the estimation, due to the linear model. It can also be observed, that at temperature ranges between 24 - 34 °C, the model overestimates the temperature drop, while at the lowest, and the highest temperatures, it underestimates. On average, the temperature error per fluid is 0.29°C (or 0.33°C if excluding $\Delta\varphi$). Similarly, the explained variance is consistent across the fluids. R^2 coefficient is 0.997. Small variations could be explained by fluid thermal properties.

6.2.2 Non-Linear models

Including non-linear dependence, with only two features - $\Delta\varphi$ and offset C1, the temperature estimations are much more accurate. Explained variance (R^2) is 0.999 for all fluids

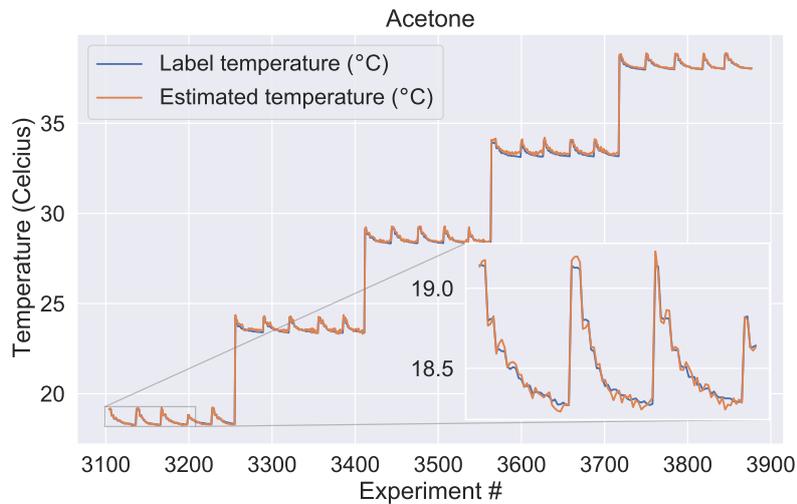


Figure 6.8: Sample of SVR-based non-linear temperature estimation, and the cooling down based on the mass flow of Acetone

Since there is no correlation between the two features, SHAP feature significance values give an accurate representation, which corresponds to the linear case - at high mass flow, there is a small negative drop in temperature, while at low, and no flow, temperature increases.

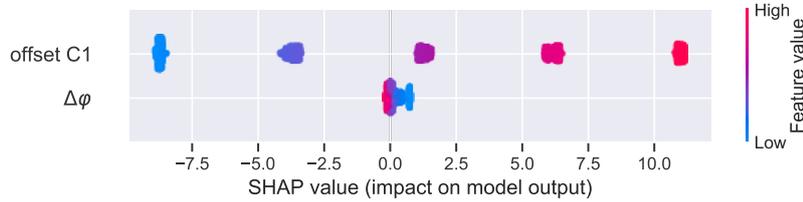


Figure 6.9: Non-linear individual feature impact on model output

The general estimation error (mean absolute error) for both linear and non-linear approaches is summarized in Table 6.8. In general, the offset feature can determine the general system temperature and the temperature drop can be approximated using mass flow dependant feature. However, using temperature-dependant resistive track is more accurate, as well as more efficient.

Table 6.8: Temperature estimation summary

Fluid	(Linear) Average error °C	(Non-linear) Average error °C
Acetone	0.27	0.06
Ethanol	0.28	0.10
H2O	0.33	0.17
IPA	0.28	0.06
N2	0.29	0.09

6.3 Pressure drop

A Coriolis-based sensor has already integrated pressure sensors on the chip, however, it is interesting whether using the capacitive signals itself, it would be possible to estimate the pressure drop. Similarly, this section acts as an additional analysis in terms of analysing linear and non-linear physical relations. The drawback of this estimation is that an accurate pressure drop label does not exist, hence it was approximated with the approach mentioned in the methodology section 4.4.2. Moreover, acetone was not included in this estimation, as the pressure sensors were not sensitive enough to measure this liquid.

6.3.1 Linear models

We begin by looking at the most linearly correlated features with respect to the pressure drop. Acetone and nitrogen measurements were not included due to low sensitivity, and more complex physical relations to gasses.

The two separate capacitive phase differences between the actuation signals have the highest correlation coefficient, as the pressure sensor is sensitive to flow. This correlation would similarly hold for $\Delta\varphi$ itself, however logically it would be to have two separate ends

Table 6.9: Linear colinearity between features

-	F_ω	$\Delta\varphi_{C1}$	$\Delta\varphi_{C2}$	R_{track}	Magn. C1	Magn. C2	Pressure drop
F_ω	1	0.02	-0.40	0.10	-0.19	-0.22	0.26
$\Delta\varphi_{C1}$	0.02	1	0.85	0.23	-0.22	-0.22	0.76
$\Delta\varphi_{C2}$	-0.40	0.85	1	-0.07	0.11	0.11	0.66
R_{track}	0.10	0.23	-0.07	1	-0.90	-0.82	-0.17
Magn. C1	-0.19	-0.22	0.11	-0.90	1	0.98	0.10
Magn. C2	-0.22	-0.22	0.11	-0.82	0.98	1	0.05

of the tube measured. The magnitudes of the capacitive signal do not hold any significant correlation to the pressure drop.

Based on the colinearity Table 6.15 Chosen features are phase difference ($\Delta\varphi_{C2}$), resonance frequency (F_ω) due to its pressure dependence, and R_{track} for temperature compensation, as pressure drop is related to the viscosity of the fluid, which is again sensitive to temperature.

Defining **Linear model** as:

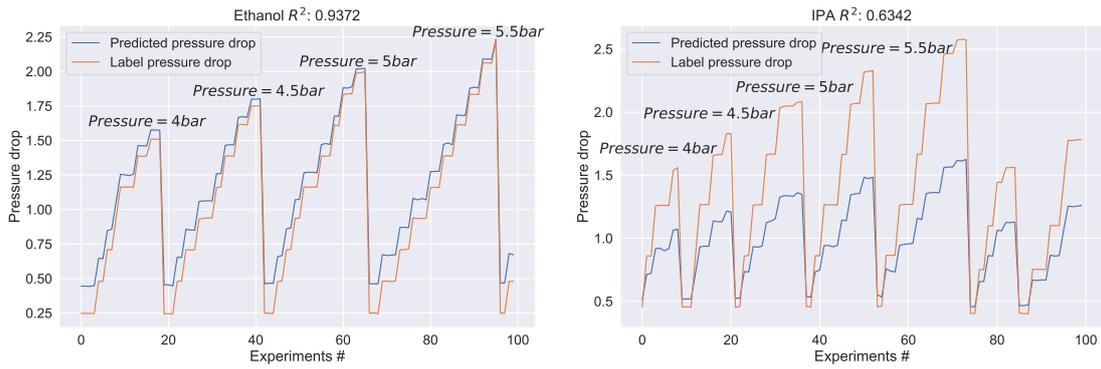
$$y = \beta_0 + \beta_1\Delta\varphi_{C2} + \beta_2F_\omega + \beta_3R_{track}$$

Similar performance can be achieved by adding additional phase ($\Delta\varphi_{C1}$) difference, however, the increase in performance is not significant, and the feature significance using P-value shows that $\Delta\varphi_{C1}$ is not significant. Using both frequency and $\Delta\varphi_{C2}$ as a feature achieves best results, with features being significant. The feature weights are summarized in Table 6.10.

Table 6.10: Linear regression coefficients with most significant polynomial features

Feature	Weight
β_1 ($\Delta\varphi_{C2}$)	2.391
β_2 (F_ω)	1.135
β_3 (R_{track})	-0.260

The performance of pressure drop estimation seems to be dependent on fluids. Ethanol achieves a R^2 score of 0.93, seemingly accurately estimating it based on capacitive tube phase difference, while isopropanol tends to underestimate the pressure drop, achieving R^2 of 0.63. This is most likely due to the different viscosities of fluids, and linearly it overfits ethanol due to some imbalance in the training data set. Water is the worst-performing, and has an R^2 score of 0.50. Sample estimations are shown in Figures 6.10a and 6.10b. The model achieves MAPE of 20%, 42.8% and 22.5% for ethanol, water and isopropanol respectively.

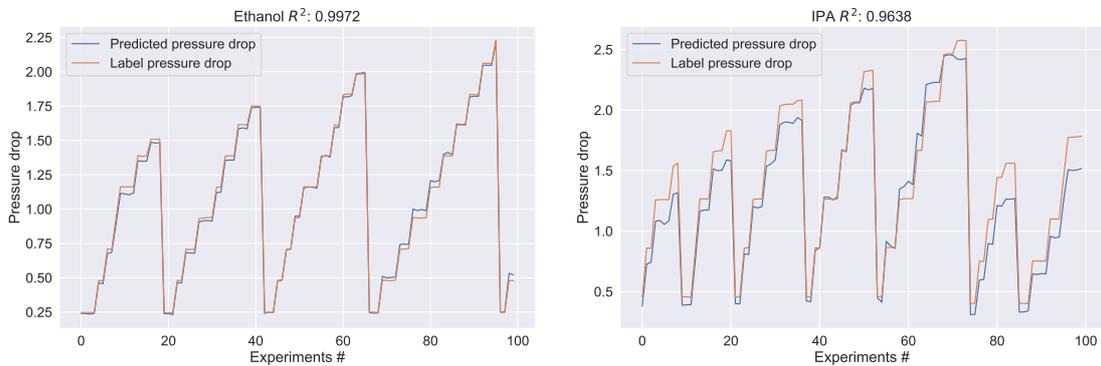


(a) Small sample of Ethanol linear pressure drop estimation, based on capacitive signals (b) Small sample of Isopropanol linear pressure drop estimation, based on capacitive signals

Figure 6.10: Pressure drop visualisations on a small sample of measurements. At a constant pressure input, measurements contain varying mass flow parameters.

6.3.2 Non-Linear models

Applying the non-linear approach, with RBF and SVR, the pressure drop estimates are more accurate for individual fluids. The model achieves R^2 of 0.99 for ethanol and water, with worst-performing isopropanol at 0.96. The MAPE values are 2.3%, 1.5% and 7.9% for ethanol, water and isopropanol accordingly.



(a) Small sample of Ethanol non-linear pressure drop estimation, based on capacitive signals (b) Small sample of IPA non-linear pressure drop estimation, based on capacitive signals

Figure 6.11: Non-linear pressure drop estimation

Similarly to the previous section, because of a small linear relation between features, the SHAP value indicates which feature values have what influence on the estimations (see Figure 6.12). The phase difference due to being proportional to mass flow affects the pressure drop linearly. Temperature track shows at lower temperatures, the higher the pressure drop, while higher temperature, decreases the pressure drop. Frequency feature may affect the main differences between the liquids. Table 6.11 summarizes the errors for each liquid per model.

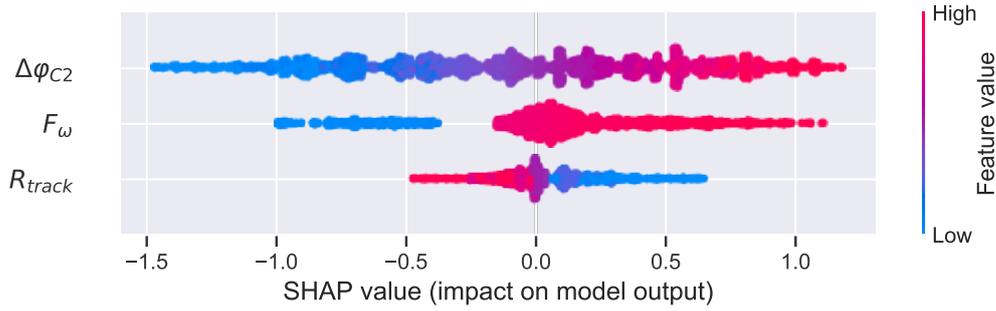


Figure 6.12: SHAP figure indicating feature importance and impact to estimation

Table 6.11: Pressure drop estimation summary of models (MAPE)

Fluid	Linear model	Non-linear model
Acetone ¹	-	-
Ethanol	20%	2.3%
Water	42.8%	1.5%
Isopropanol	22.5%	7.9%

6.4 Viscosity

Viscosity can be estimated with a mathematical Hagen Poiseuille equation, however, it has been shown that the accuracy is very dependent on mass flow, as well as other inaccuracies due to the simplification of physics, i.e. it would be very challenging to include exact geometry of the Coriolis suspended tube, such as rounded edges. Using the Hagen-Poiseuille equation on this data set to estimate viscosity, on average it produces 7.2%, 13.2% and 5.4% error, for ethanol, water and isopropanol respectively. This error also depends on temperature, pressure and mass flow. For estimations performed with linear and non-linear models, we take out Acetone and nitrogen measurements, as well as filter mass flow to ≥ 0.5 g/hr, since measurements with no flow do not yield a pressure drop.

6.4.1 Linear models

Choosing the most linear feature to viscosity shows that pressure sensors by themselves do not have a linear relation to viscosity, while frequency has the highest correlation coefficient to viscosity. Due to the colinearity between frequency and $\Delta\varphi_{C2}$, the latter feature is not used in the model. Due to the compressible nature of gas, for the linear model, nitrogen is not used.

Although the linear correlation of pressure sensors to viscosity is very low, based on physics, they are important to estimate the viscosity. This was also observed empirically, that without including the pressure sensors, linear model viscosity estimation performs much worse. Furthermore, the magnitudes of C1 and C2 are not statistically significant to estimate the viscosity based on P-values. Pressure sensors are linearly correlated with each

¹Pressure sensor readings very insensitive

Table 6.12: Linear colinearity between features

Features	F_ω	$\Delta\varphi_{C1}$	$\Delta\varphi_{C2}$	R_{track}	Magn. C1	Magn. C2	Viscosity
F_ω	1	0.36	-0.85	0.02	-0.31	-0.39	-0.64
$\Delta\varphi_{C1}$	0.36	1	0.14	0.17	0.27	-0.30	-0.36
$\Delta\varphi_{C2}$	-0.85	0.14	1	-0.04	0.28	0.35	0.49
R_{track}	0.02	0.17	-0.04	1	-0.85	-0.75	-0.32
Magn. C1	-0.31	-0.27	0.28	-0.85	1	0.98	0.42
Magn. C2	-0.39	-0.30	0.35	-0.75	0.98	1	0.43
P_1	0.05	0.52	0.28	0.58	-0.51	-0.44	-0.08
P_2	0.08	-0.13	-0.08	-0.61	0.50	-0.42	0.17

other, having a 0.84 Pearson coefficient. A simple linear combination of features does not outperform the classical approach of viscosity estimation. To resolve this, we can include Hagen-Poiseuille in the linear model, performing additional feature engineering for which we can use Equation 2.18. Since we do not care about the constants in linear regression, as they are multiplied by a scalar, we can use feature $\frac{\Delta P}{\Delta\varphi}$, which is analogous to the ratio between pressure drop and mass flow. Furthermore, the best performance is achieved while including both $\Delta\varphi_{C1}$ and $\Delta\varphi_{C2}$, although the colinearity affects the interpretation of accurate weights to colinear features. This indicates a very small dependence on individual sensing elements, replacing with $\Delta\varphi$ decreases the accuracy by 2%.

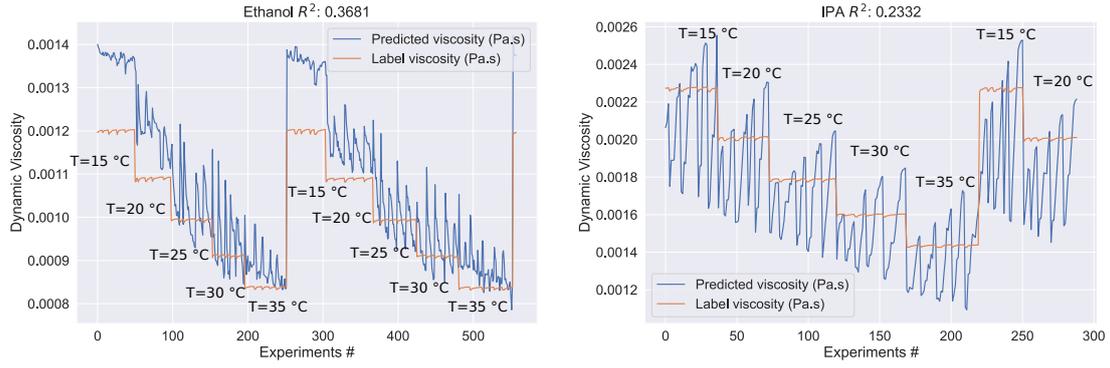
We define a **linear model**

$$y = \beta_0 + \beta_1\Delta\varphi_{C1} + \beta_2\Delta\varphi_{C2} + \beta_3\frac{\Delta P_r}{\Delta\varphi} + \beta_4R_{track} + \beta_5F_\omega + \beta_6P_1 + \beta_7P_2$$

Table 6.13: Linear regression coefficients with most significant polynomial features

Feature	Weight
$\beta_1 (\Delta\varphi_{C1})$	0.0005
$\beta_2 (\Delta\varphi_{C2})$	-0.0030
$\beta_3 (\frac{\Delta P_r}{\Delta\varphi})$	0.0002
$\beta_4 (R_{track})$	-0.0012
$\beta_5 (F_\omega)$	-0.0007
$\beta_6 (P_1)$	0.0058
$\beta_7 (P_2)$	0.0040

Chosen features have significant P-values of much lower than $P < 0.025$. Since the viscosity labels are extremely small, i.e. ranges from 10^{-5} to 10^{-3} , the linear combination of the features will also have very small weights, which might be challenging to interpret individual features' impact. Based on feature weights shown in Table 6.13, it can be seen that P1 and P2 features are the most significant. Linearly, the viscosity estimations are very sensitive to mass flow, and the best-performing fluid is Ethanol, with an R^2 of 0.36, shown in Figure 6.13a.



(a) Viscosity estimation on ethanol using a linear model. Showcasing fluid state dependence. (b) Viscosity estimation on IPA using a linear model. Showcasing fluid state dependence.

Figure 6.13: Viscosity estimation result of a linear model, plotting testing dataset (70%). At constant temperature, the measurements contain varying pressure and mass flow parameters.

The mean absolute percentage error for these estimations are 7.8%, 25.3% and 11.4% for ethanol, water and isopropanol respectively. If ΔP_{est} is used instead, which is pressure drop calculated with respect to the P_{in} , compensating for temperature, the performance is much better, resulting in 5.9%, 11.6% and 3.6% respectively.

6.4.2 Non-Linear models

Applying SVR with RBF kernel, the results outperform the linear method by a significant amount due to the feature space produced by the kernel. The best-performing fluid shown in Figure 6.14, water achieves R^2 of 0.99.

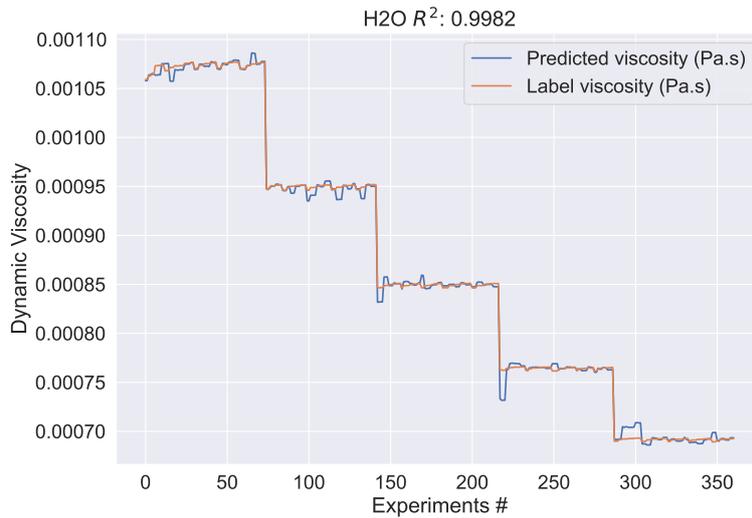


Figure 6.14: Viscosity estimation using non-linear kernelized SVR method. Showcases estimation with fluid state dependence.

With the selected features, there is a colinearity of 0.84 between P_1 and P_2 values. The feature significance of the selected features is shown in Figure 6.15. The values show some

consistency with physics - the higher the temperature (R_{track}), the less viscous the fluid, likewise, pressure readings have the most significance in the estimations, with a mostly linear relationship.

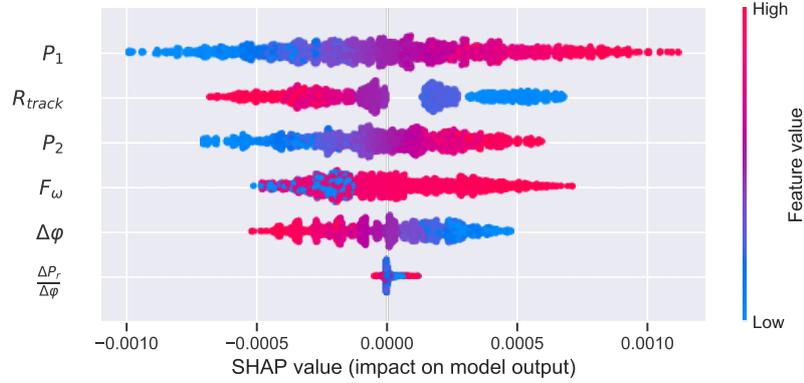
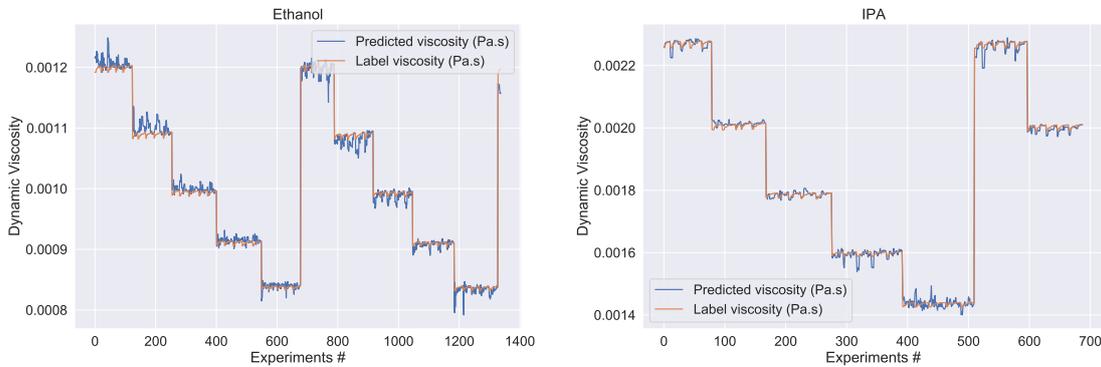


Figure 6.15: SHAP feature significance and impact on estimation

A second non-linear method is a Gaussian Process. Since it is a non-parametric method, it can fit non-linearly better due to its kernel hyperparameter length scale, involving covariance calculation with points further away. Applying the same features as previously for SVR, the results look very promising. All of the fluids have explained variance (R^2) of >0.99 , results are shown in Figure 6.16.



(a) Viscosity estimation using Gaussian Process on ethanol (b) Viscosity estimation using Gaussian Process on isopropanol

Figure 6.16: Gaussian Process applied on a trained liquid, and showcasing results on (70%) testing dataset

Feature significance shows very different model impact patterns compared to SVR. The temperature dependant track (R_{track}) and resonance frequency (F_ω) seems to have less impact while the pressure sensors seem to be much linear impact. This may be due to the SVR epsilon loss, as the maximum error in the test dataset is 4.35% compared to 5.56% of GPR. Although $\frac{\Delta P_r}{\Delta \varphi}$ should be a significant feature, as it is proportional to kinematic viscosity, the influence from using sensor readings directly is not so linear to viscosity. Difference between electrical readings ΔP_r and fitted readings ΔP_{est} is shown in Figure 6.18.

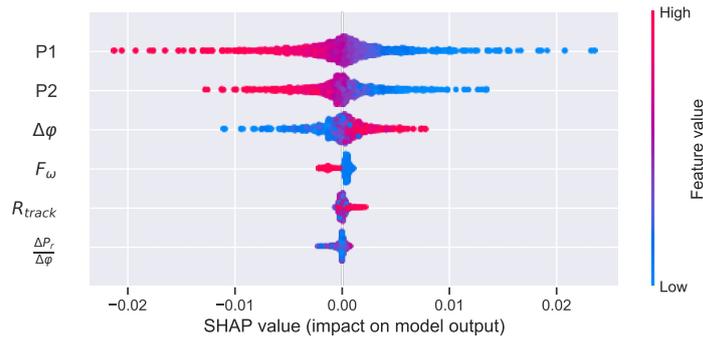
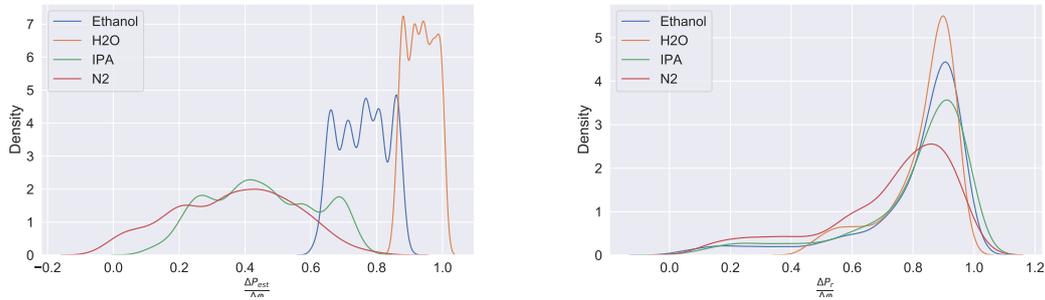


Figure 6.17: SHAP feature significance and impact on estimation

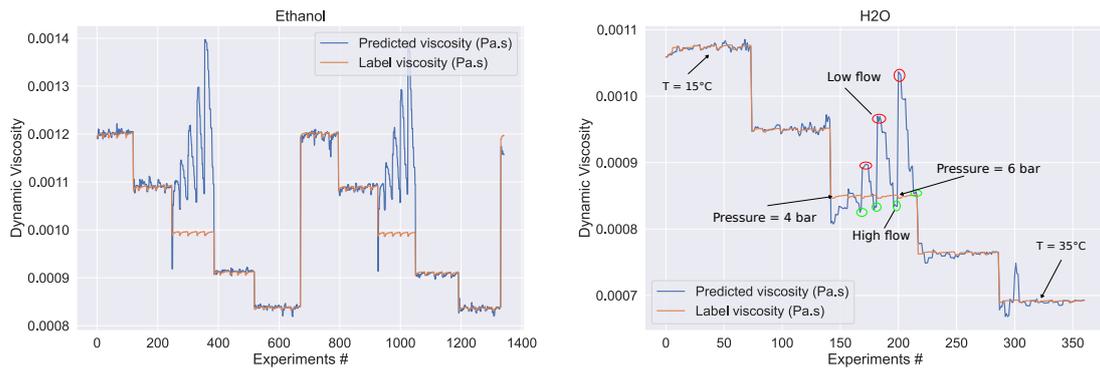


(a) Estimated pressure drop compensating for individual fluid. (b) Pressure readings directly from Wheatstone temperature showcasing distribution for individual fluid.

Figure 6.18: Differences between estimated pressure drop using pressure input as a reference, and the direct on-chip measurements

This approach works reasonably well for some liquids (e.g. IPA), even if a certain temperature range was taken out from the training data set. However, for the best accuracy, it has to interpolate between liquids, hence nitrogen was excluded. One such viscosity interpolation for temperature between 28 °C and 30 °C can be seen in Figure 6.19. The downside of excluding temperature range, it becomes very fluid-state dependent.

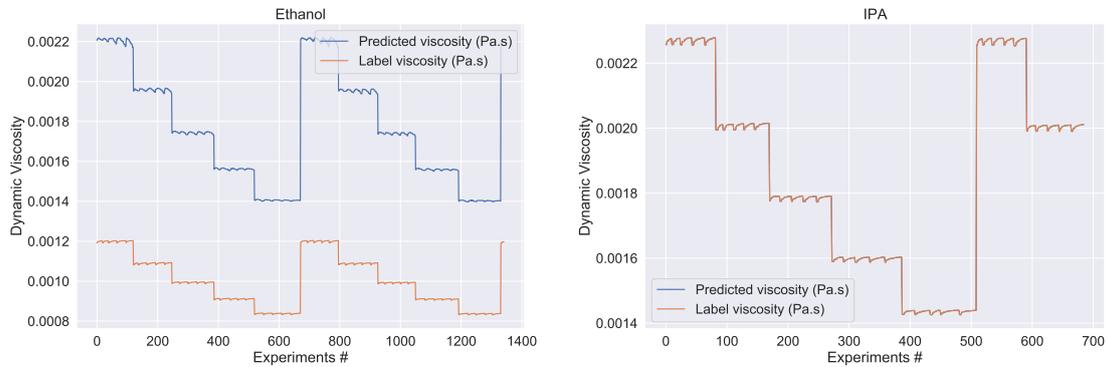
Ethanol had the largest error in terms of this interpolation, achieving a maximum error of 29% at very low flow (0.5 g/hr). On average, the estimation of the unseen temperature range is 16.8%, 5.5%, and 9.8% for ethanol, water and isopropanol respectively.



(a) Viscosity estimation on ethanol, on a range of (b) Viscosity estimation on isopropanol, on a temperatures, excluding 28-30 °C from training range of temperatures, excluding 28-30 °C from training completely.

Figure 6.19: Viscosity estimations showcasing fluid-state dependence on unseen temperature range

Another performance check is to exclude ethanol entirely from the training dataset and check the performance of the unseen fluid. Since ethanol viscosity is between water and isopropanol, theoretically it should interpolate viscosity between the trained viscosities. The result shows a huge offset in the estimated viscosity Figure 6.20a compared to the trained liquids Figure 6.20b. This also shows that it transformed into features which are completely fluid-state independent.



(a) Viscosity estimation on ethanol while completely excluding ethanol from training. (b) Viscosity estimation on isopropanol, while excluding ethanol, increasing accuracy on other trained fluids

Figure 6.20: Viscosity estimation showcasing fluid state independence on untrained fluid as well as trained.

This offset issue seems to be completely frequency-dependent. Which might be affected by the normalization method. Without normalizing or applying different techniques, it produces different results. Deriving a new scaling method instead of min-max might solve the offset. The current approach is to simply offset ethanol frequency by subtracting 100 Hz from its fundamental frequency. This results in very accurate viscosity estimation as shown in Figure 6.21 although the liquid has not been trained, it shows the potential of using a statistical approach to determine viscosity.

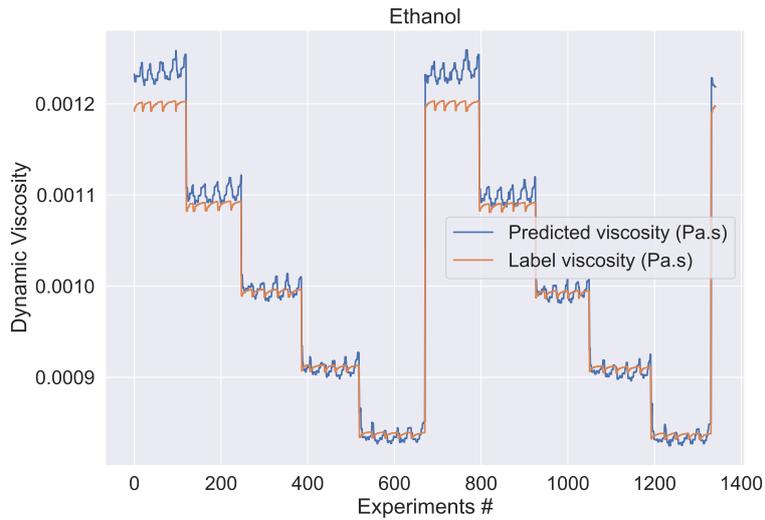


Figure 6.21: Viscosity estimation on ethanol while completely excluding ethanol from training. The frequency was offset by 100 Hz to solve the resulting viscosity offset.

The final result shows that the Gaussian Process can estimate viscosity very accurately, achieving $<2\%$ error on average, even while including nitrogen in the training data set. However, since Nitrogen is an extreme outlier with its characteristics, it achieves an average 112% error, while at higher temperatures it is even more. Overall, the GPR outperforms conventional approaches by a factor of ≈ 4 compared to the same dataset presented in [71]. In the worst-case performance, SVR outperforms GPR, however, on average, GPR has the best performance. A summary of temperature-dependent viscosity with estimated values using Gaussian processes regressor can be seen in Figure 6.22.

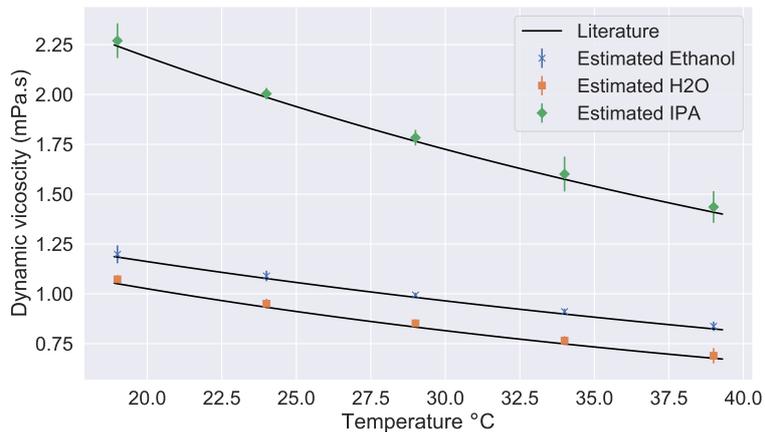


Figure 6.22: Estimated viscosity of the fluids

¹Insensitive pressure readings

²Hagen-Poiseuille estimation only works on non-compressible fluids

³Estimated using ΔP_{est}

⁴The better performance represents the offset correction

Table 6.14: Viscosity estimation summary of models (MAPE)

Fluid	Linear model	Non-linear(SVR)	Non-linear(GP)	Hagen-Poiseuille ³
Acetone ¹	-	-	-	-
Ethanol	8.0%	0.78%	0.65%	7.2%
Ethanol(unseen) ⁴	-	4.76 % / 74%	1.18% / 75.3%	-
Water	25.6%	0.39%	0.76%	13.2%
Isopropanol	11.4%	0.67%	0.47%	5.4%
Nitrogen ²	-	-	112%	-

6.5 Density

In theory, frequency should be proportional to the density of the fluid. However, the frequency is affected by temperature, pressure and fluid differently. Using frequency as directly proportional to density, on average, the absolute error is 17.4 kg/m³, 17.2 kg/m³, 81 kg/m³, 3 kg/m³ 11.4 kg/m³ for acetone, ethanol, water, isopropanol and nitrogen respectively.

6.5.1 Linear models

Similar features as the previous viscosity section are present to estimate density. As already known, frequency is the most linearly correlated to density, having a coefficient of -0.99 (this is due to lower density equaling higher frequency). Similarly, $\Delta\varphi_{C_i}$ features have some linear correlation.

Table 6.15: Linear colinearity between features

-	F_ω	$\Delta\varphi_{C1}$	$\Delta\varphi_{C2}$	R_{track}	Magn. C1	Magn. C2	Density
F_ω	1	0.30	-0.82	0.02	-0.27	-0.35	-0.99
$\Delta\varphi_{C1}$	0.30	1	0.25	0.15	-0.24	-0.27	-0.32
$\Delta\varphi_{C2}$	-0.82	0.25	1	-0.06	0.25	0.31	0.82
R_{track}	0.02	0.15	-0.06	1	-0.86	-0.77	-0.04
Magn. C1	-0.27	-0.24	0.24	-0.86	1	0.98	0.29
Magn. C2	-0.35	-0.37	0.31	-0.77	0.98	1	0.37

However, due to the effective spring constant of the suspended tube, the frequency is dependent on pressure. To include pressure, temperature and flow dependence, we will use features: $\Delta\varphi_{C1}$, R_{track} , P_1 , P_2 and frequency. Although magnitudes of the C1 and C2 signals have more correlation w.r.t density, the resulting estimation is very noisy, which is most likely due to harmonic dependence on the signal, as analysed in Chapter 5.2.4. Due to the high magnitude dependence on temperature, substituting with Rtrack should be sufficient. Similarly, $\Delta\varphi_{C2}$ has a high correlation to density, however, its colinearity with frequency affects the interpretation of the model. Although including it as a feature to estimate density improves the estimation on average by $\approx 4.8\text{kg}/\text{m}^3$, the weights of the features are mainly shifted between the frequency and $\Delta\varphi_{C2}$.

Defining **Linear model** as :

$$y = \beta_0 + \beta_1 F_\omega + \beta_2 P_1 + \beta_3 P_2 + \beta_4 \Delta\varphi_{C1} + \beta_5 R_{track}$$

Table 6.16: Linear regression coefficients

Feature	Weight
F_ω	-920.83
P_1	45.98
P_2	28.89
$\Delta\varphi_{C1}$	-31.79
R_{track}	-14.21

The chosen features all have $P < 0.001$ and are significant. The density estimation seems to be substantially sensitive to mass flow, as well as temperature, as seen in Figure 6.23. The weights of features show that frequency is the most dominant component. Since features are normalized in a range from 0 to 1, the frequency weight might seem not intuitive. The linear model has an offset (β_0) in the initial predictions, which results in density decrease with frequency increase. Likewise, the upstream pressure (P_1) has quite a substantial effect. At the highest pressure, it increases density by 45.98 kg/m³, however, it is reduced by temperature and mass flow. The overall performance of the linear model outperforms the Frequency \propto Density as an estimation by a small margin. The MAE values are 12 kg/m³, 16.8 kg/m³, 80.9 kg/m³, 6 kg/m³ and 12.3 kg/m³ for acetone, ethanol, water, isopropanol and Nitrogen respectively.

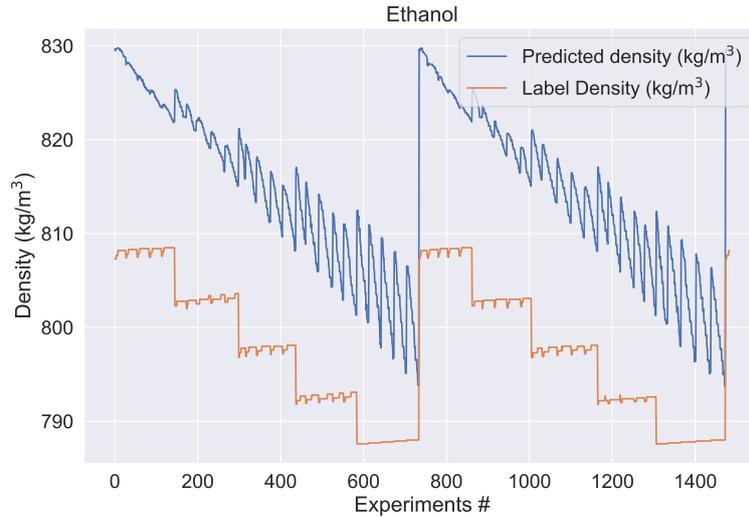


Figure 6.23: Linear model on fluid density estimation

Linearly, it seems there are many different influences to estimate accurately fluid density. Frequency and density vary quite a lot depending on its fluid state (mass flow, pressure, temperature), as well as fluid properties itself. This is shown in Figure 6.24, a contour plot showcasing three liquids which are very similar in density, however at the same frequency, the density may differ by > 20 kg/m³. Likewise, due to the different properties of gasses, the effect of frequency on nitrogen is opposite due to its compressibility property as shown in Figure 6.25a.

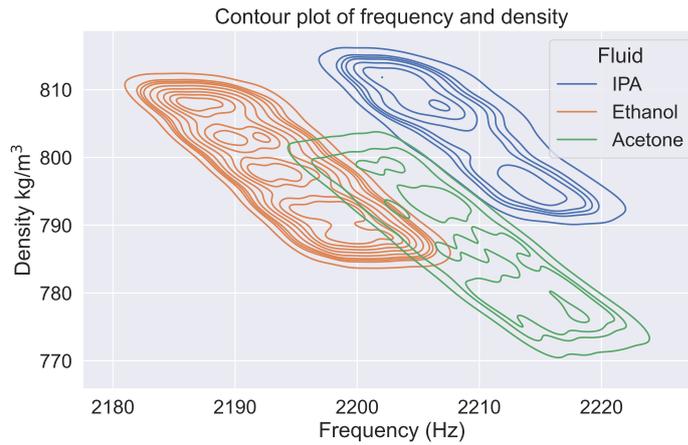
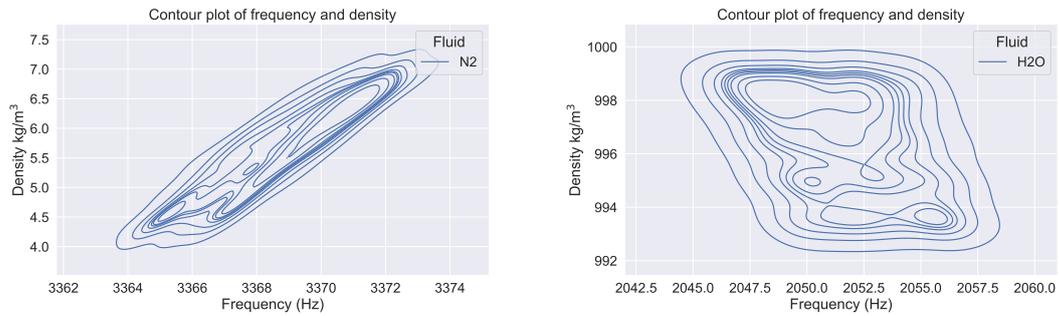


Figure 6.24: Similar density liquids, and their contour plots of frequency with respect to density



(a) Contour plot of nitrogen, containing frequency and density (b) Contour plot of water containing frequency and density

Figure 6.25: Contour plots of fluids very different from the rest

6.5.2 Non-Linear models

Similarly, as with other estimations, we re-use the same features as in the linear case and apply SVR with the RBF kernel to estimate the density. The results show very accurate predictions of density based on the given features [Figure 6.26].

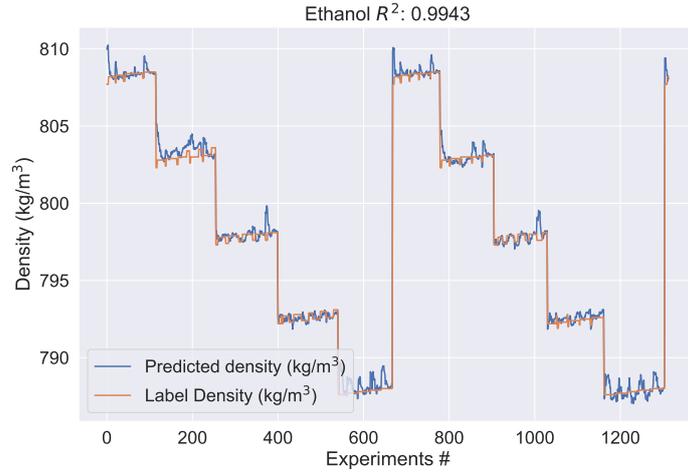


Figure 6.26: Density estimation on trained liquid using SVR

The fundamental feature as expected is frequency. The impact of pressure sensors, as well as mass flow, seem quite similar to each other, while temperature has quite a small effect. These feature impacts are similar to the linear case, where higher flow ($\Delta\varphi_{C1}$) decreases the density, while higher pressure increases. The overall performance is quite superior compared to the linear method. The density error for each fluid is 0.62 kg/m^3 , 0.34 kg/m^3 , 2.6 kg/m^3 , 1.34 kg/m^3 and 0.75 kg/m^3 for acetone, ethanol, water, isopropanol and nitrogen respectively. On average, SVR outperforms the linear method by 23 kg/m^3 . The feature's importance and impact on the estimation can be seen in Figure 6.27

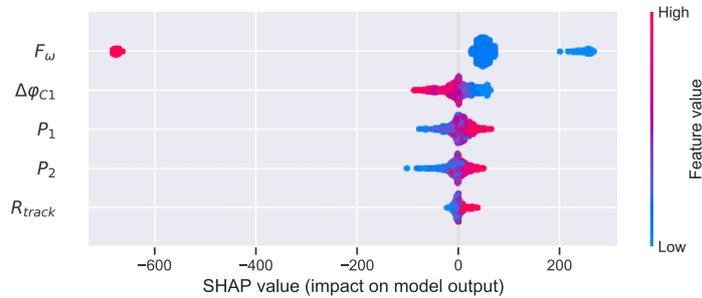
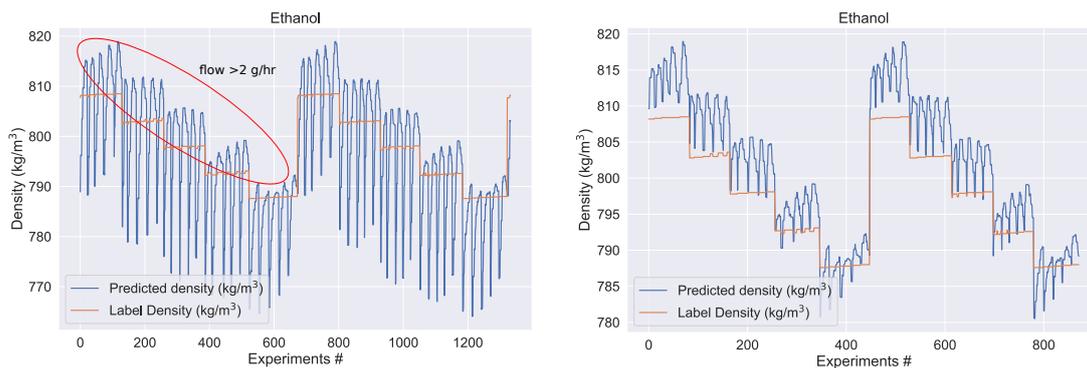


Figure 6.27: SHAP SVR with RBF feature importance

It is also interesting to see how it performs on unseen liquids, i.e. estimating properties of unknown liquids. Since there are two extremes - nitrogen and water, the estimation can be performed on liquids between this range. Applying the trained model which excludes ethanol, shows its high dependence on the fluid state, particularly mass flow. This estimation can be seen in Figure 6.28a. On average, it has an error of 7.2 kg/m^3 . However, if we exclude low mass flow ($< 2 \text{ g/hr}$), the performance is better, reaching an average error of 4.3 kg/m^3 , shown in Figure 6.28b



(a) Density estimation on untrained fluid (b) Density estimation on untrained fluid, filtered for mass flow of >2 g/hr

Figure 6.28: Density estimation on untrained fluid, showcasing fluid-state dependence

Applying the non-parametric technique - the Gaussian Process performs better than SVR on trained fluids. Overall average performance is increased by ≈ 1 kg/m³. One such sample estimation is shown in Figure 5.10b. The results for individual fluid achieve an error of 0.13 kg/m³, 0.09 kg/m³, 0.3 kg/m³, 0.18 kg/m³, 0.19 kg/m³ for acetone, ethanol, water, isopropanol and nitrogen respectively.

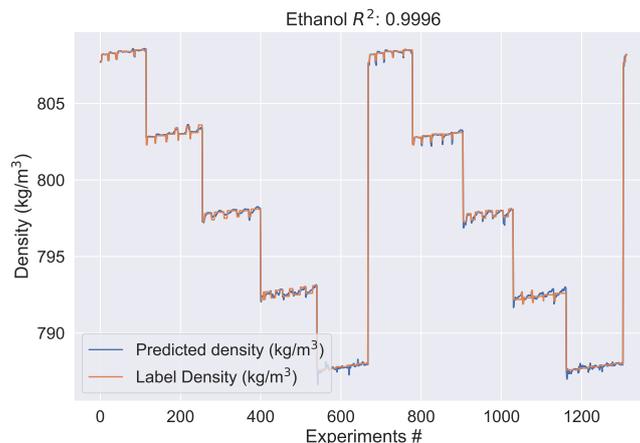


Figure 6.29: Density estimation on trained fluid using GP showcasing fluid-state independence

However, the performance of SVR is better for unseen fluids. Applying the same method for unseen liquid as in the SVR case, the result reaches an average error of 10 kg/m³. In contrast, to SVR's significant dependence on flow, the GPR results have different effects as shown in Figure 6.30.

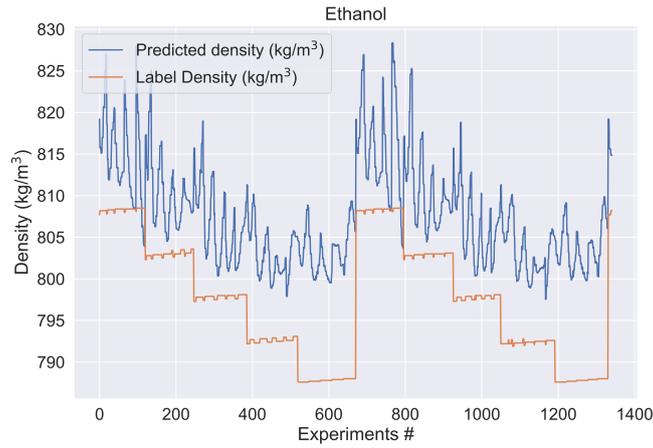


Figure 6.30: Density estimation on untrained fluid using GP

Feature significance shows a similar trend to the SVR case, however, pressure sensors have different properties in estimating density. Likewise, the mass flow has much less impact (see Figure 6.31).

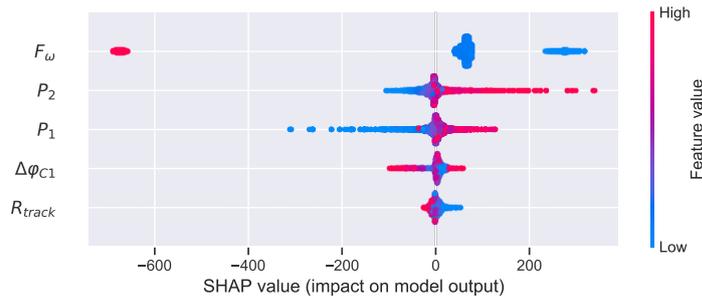
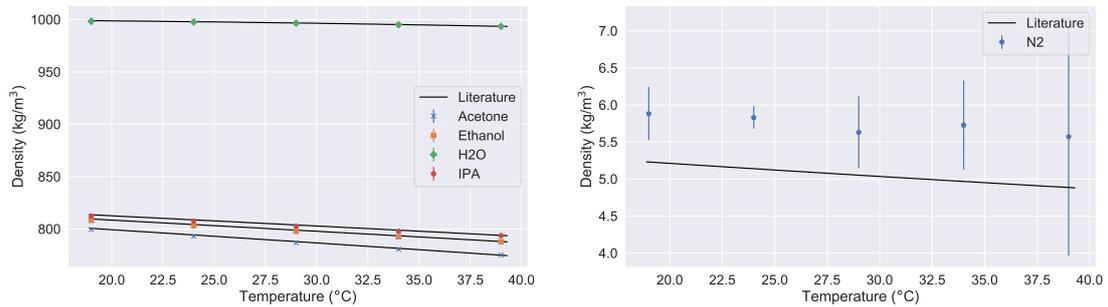


Figure 6.31: Gaussian Process feature importance



(a) Estimated density of the liquids using GPR (b) Estimated density of the gas using GPR

Figure 6.32: Density estimation on test data set (70%). The error plot represents maximum error.

¹The lower error is with higher mass flow (>2 g/hr)

Table 6.17: Density estimation models summary (MAE)

Fluids	Linear	Non-linear (SVR)	Non-linear (GP)	$F_\omega \propto$ Density
Acetone	12 kg/m ³	0.62 kg/m ³	0.13 kg/m ³	17.4 kg/m ³
Ethanol	16.8 kg/m ³	0.34 kg/m ³	0.09 kg/m ³	17.2 kg/m ³
Ethanol(unseen) ¹	-	4.3 / 7.2 kg/m ³	10.1 kg/m ³	-
Water	80.9 kg/m ³	2.6 kg/m ³	0.3 kg/m ³	81.8 kg/m ³
Isopropanol	6 kg/m ³	1.34 kg/m ³	0.18 kg/m ³	3.4 kg/m ³
Nitrogen	12.3 kg/m ³	0.75 kg/m ³	0.19 kg/m ³	11.4 kg/m ³

6.6 Fluid type classification

To determine what liquid was being measured, we can use probabilistic models to determine the measured fluid based on its response to the sensing elements.

6.6.1 (Naive) Bayes

One of the most straightforward probabilistic models is Naive Bayes. It assigns probabilities for each class (fluid) given data.

From the data analysis performed in Section 5.2.5, it is known that capacitive signal offset does not depend on any fluid and is proportional to temperature. Similarly, magnitude had some minor differences per fluid but was more sensitive to temperature. The intrinsic features which correspond mostly to different fluids were frequency, pressure sensors, and some fluid differences in $\Delta\varphi_{C1}$. Although $\Delta\varphi_{C2}$ has more correlation for density, it is quite colinear with frequency. For classification, we use frequency, P1 and P2 readings, temperature-dependant resistive track (Rtrack) and $\Delta\varphi_{C1}$. Applying the Naive Bayes approach, we can probabilistically predict which liquid is measured. The results obtained show an overall accuracy of 82.9%, with nitrogen and water perfectly classified. Recall scores of acetone, ethanol and isopropanol correspond to 61.1%, 93.1% and 67.3%, (of specific fluid correctly classified) while precision corresponds to 68.5%, 87.8% and 67.8% (correctly classified % of the time). However, very similar results can be seen from a simple Bayes classifier, classifying directly based on frequency. The total accuracy drops by 6.6%, while still correctly classifying water and Nitrogen. Including more fluid-dependent features increases accuracy on similar liquids. The best-performing model with Naive Bayes is shown in Figure 6.33.

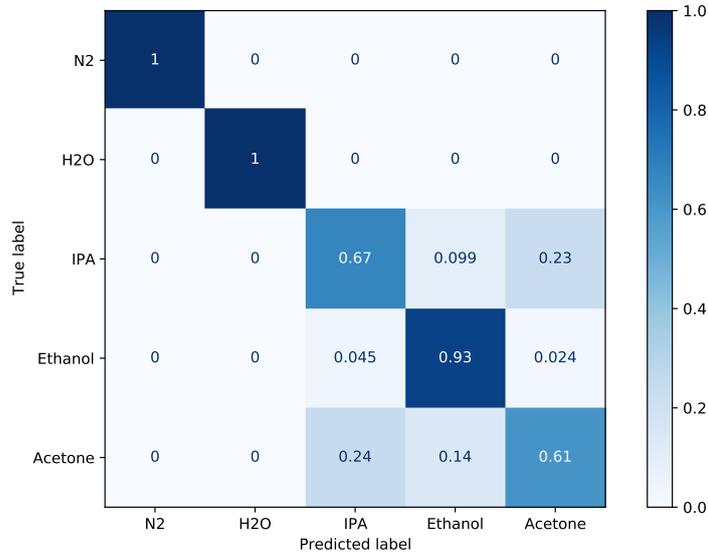
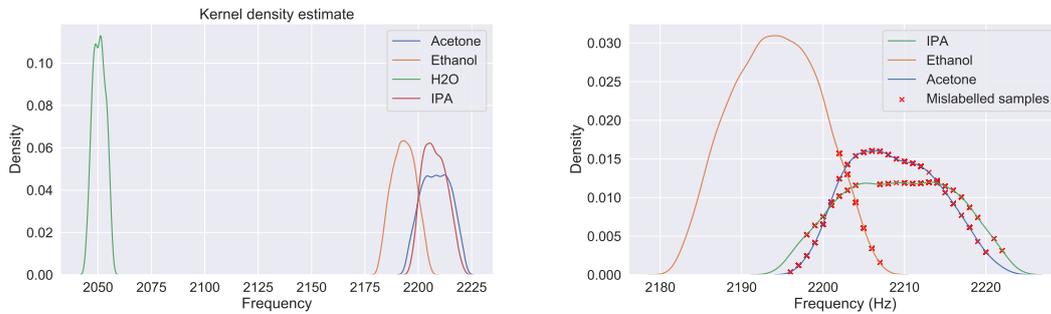


Figure 6.33: Naive Bayes confusion matrix

The reason for the performance is due to the similarity between acetone, ethanol and isopropanol densities. This results in very similar frequencies, which overlap, and in return misclassify fluids based on it. The frequency distribution of fluids (excluding nitrogen due to its very different frequency) is shown in Figure 6.34a, and Figure 6.34b shows the samples misclassified respectively to their density distribution.



(a) Density estimation of frequencies with different fluids (excluding Nitrogen) (b) Misclassified samples corresponding to liquid frequency distribution

Figure 6.34: Distribution of frequencies with respect to individual fluids, and misclassified samples with reference to the distribution

6.6.2 Gaussian Process

Applying the Gaussian process classifier, the performance is much superior with only a fraction of training data. The input features are the same as the Naive Bayes approach, using frequency, pressure (upstream/downstream) sensors, resistive track and $\Delta\varphi_{C1}$. These features were not normalized to keep the distance between fluid-specific measurements far apart.

The accuracy of the models varying how much data is used to train the model is shown

in Figure 6.36a. With only 1% of data (55 samples), the performance is similar to that of Naive Bayes. The samples consist of measurements of 23, 11, 10, 6, and 5 of ethanol, isopropanol, acetone, water and nitrogen respectively. While only a few measurements of water and nitrogen were included, they were classified correctly without misclassifying, and the confusion matrix looks similar to that of Naive Bayes, with ethanol also being correctly classified 100% of the time. Using 5% of training data, there is not much difference between using more for training the models as shown in Figure 6.36a. It achieves 99.5% accuracy, with few misclassifications on similar liquids in terms of densities (isopropanol and acetone). The confusion matrix where only 5% of the data was used is shown in Figure 6.36b

In total, there were 24 misclassifications, which from a frequency perspective, most of the samples had overlapping frequencies. The samples which were misclassified are visualised in Figure 6.37, showcasing from which part of the distribution the samples were. The results show that using GPC, the results are completely fluid-state independent, and only a very small amount of data is required. The general model performance still mainly depends on frequency as a feature and pressure sensors similarly assist in determining the fluid based on its viscosity. The feature importance is shown in Figure 6.35.

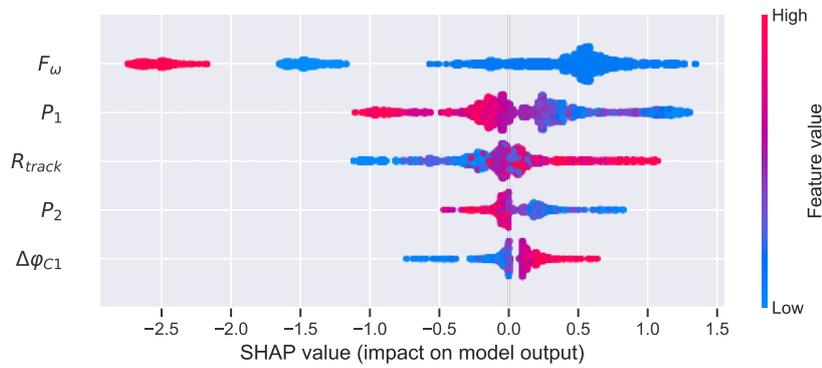
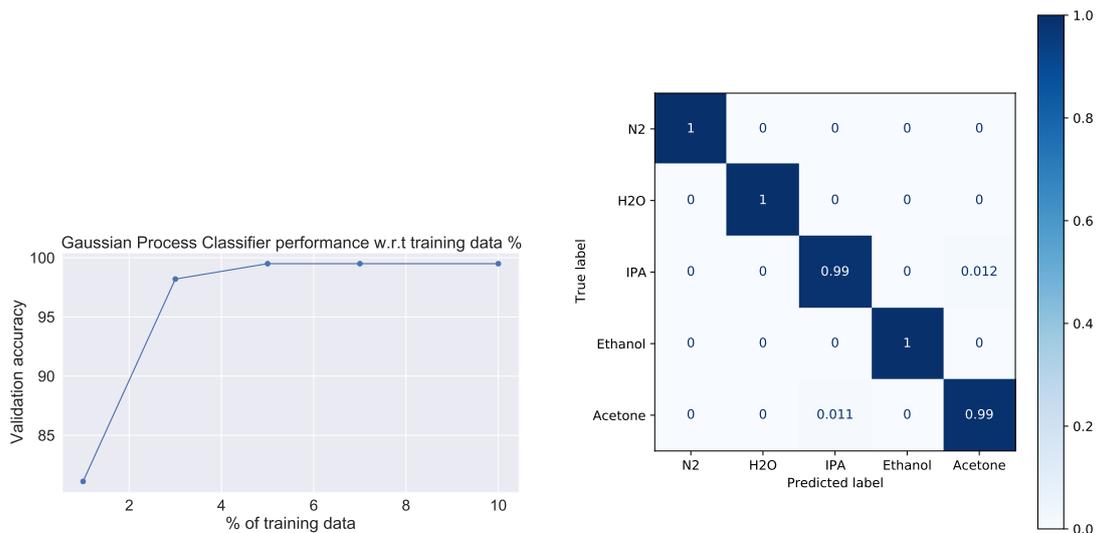


Figure 6.35: SHAP feature impact on the model



(a) Classification accuracy of Gaussian Process (b) Gaussian Process Classification confusion matrix varying % of training data.

Figure 6.36: Gaussian Process performance

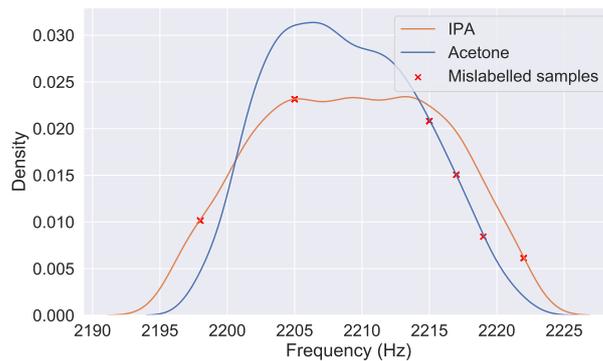


Figure 6.37: Gaussian Process misclassified samples respective to frequency distribution.

6.6.3 DeepSense

In an attempt to capture more features across a wide range of frequency components, we use the DeepSense deep learning architecture developed by Yao et al. The original paper uses real and imaginary features directly into the architecture, which they used to classify human activity based on accelerometer and gyroscope measurements. In our scenario, we have readings of capacitive signals, which similarly can use frequency components to classify the liquids. The best-performing model achieves an accuracy of 98.7%. In the input layer, it convolves real parts of the frequency components as [C1 real, C2 real] as the left side of the input, and equivalently, [C1 imag, C2 imag]. The confusion matrix results can be seen in Figure 6.38. It can be observed, that the main misclassified fluids were quite similar in their density properties, i.e. frequency components were overlapping.

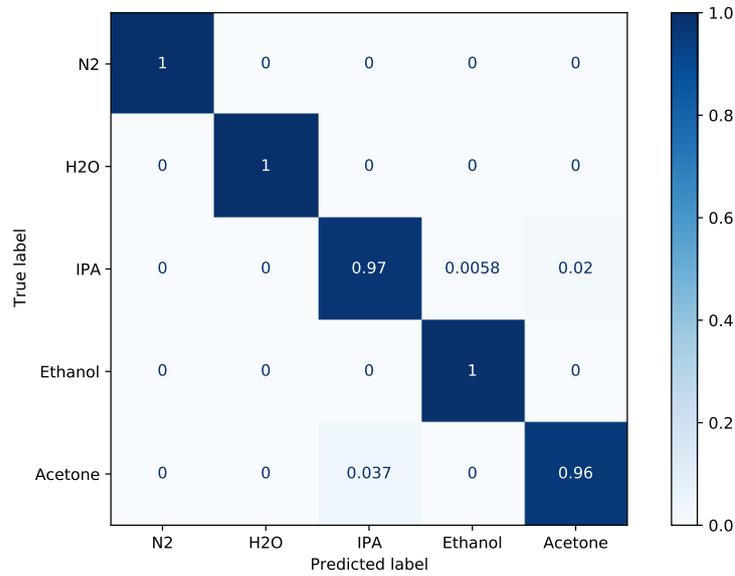


Figure 6.38: DeepSense classification results summarised in confusion matrix

Varying the number of feature maps (filters) against the training data shows that with more filters, it requires much less training data to achieve close to 92% accuracy, still outperforming the simple Bayes classifier based on frequency data. With more training data, 16 filters achieve comparable performance, with the advantage of faster training. This can be seen in Figure 6.39

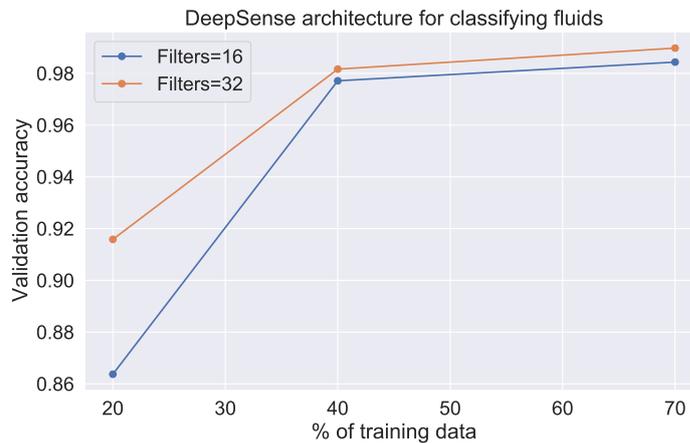


Figure 6.39: DeepSense architecture

The architecture output for the left side and right side of the convolutions can be seen in Figure 6.40. Using real and imaginary components directly in the architecture seems to compress data, and find impulse responses for resonant frequencies. The peaks vary depending on the resonant frequency of the fluid, such as the nitrogen peak is around 600 on the x-axis.

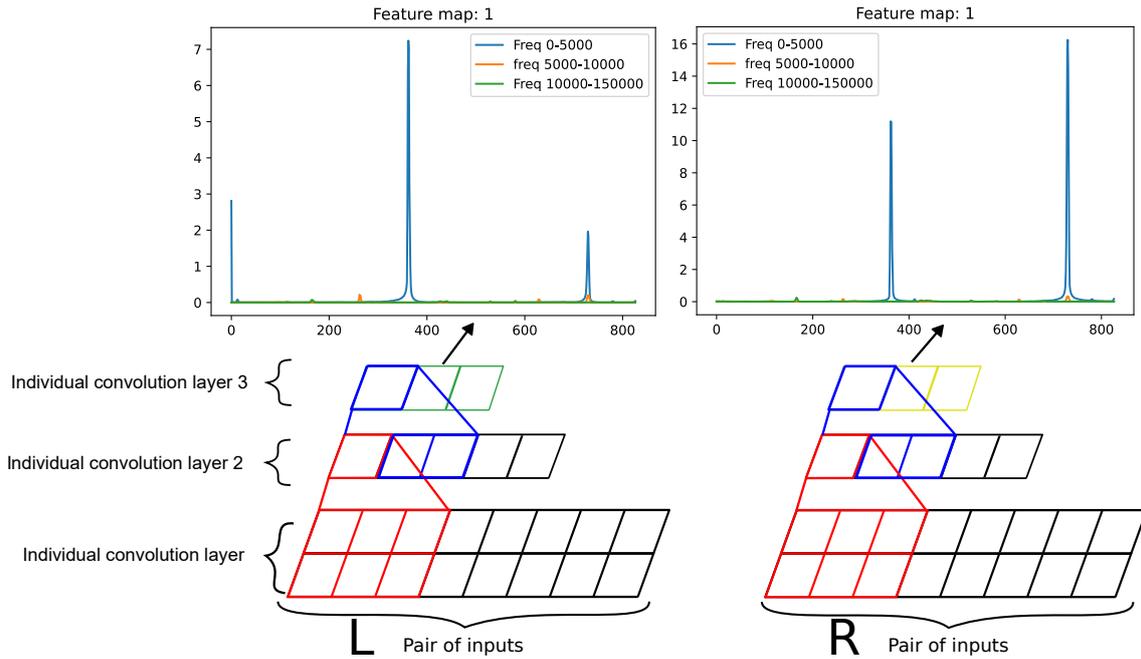
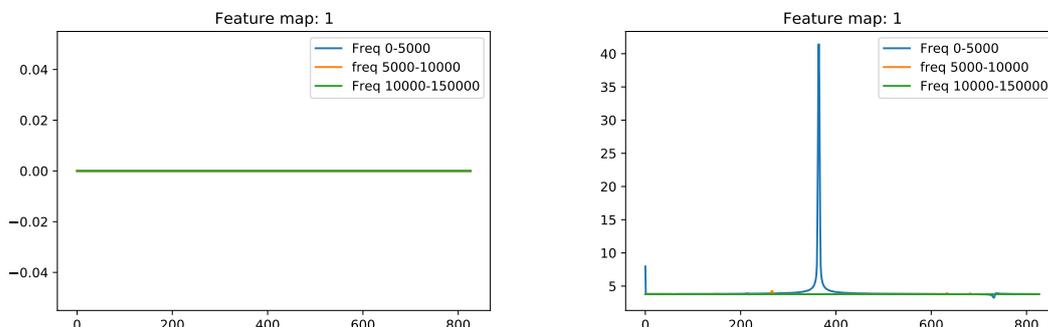


Figure 6.40: Visualised architecture output for a single measurement

Multiple orders of pairs of inputs were tested. This means that the individual convolution was performed on [C1 real, C2 real], [C1 real, C1 imag] on the left and right side as an input to the architecture. Similarly, magnitude and phase spectrums were also used as input, which was preprocessed by finding the magnitude with the square root, as well as phase with arctan operations. Using phase spectrum and magnitude spectrum directly such as [C1 magnitude, C2 magnitude] and [C1 phase spectrum, C2 phase spectrum] shows sub-optimal results. The convolutions on the phase spectrums seem to be unable to learn the fundamental features, and they converge to extremely small values in the output of the convolutions as shown in Figure 6.41a. This may be due to the fact that the phase spectrum with the non-linear arctan operation contains too much noise and finding resonance frequency is too complex, while magnitude spectrum convolutions similarly compress data as shown in Figure 6.41b, and the resonant peaks correspond to the best-performing approach.



(a) Phase spectrum converges to very low values (b) Magnitude spectrum finds the resonant frequency peak

Figure 6.41: Phase and magnitude convolution output from *Individual convolution layer 3*

If we exclude the phase spectrum and use magnitude spectrums instead, this results in very similar results to the Bayes classifier, with an accuracy of approximately 70%. This underperformance is caused by scaling magnitude values by $\frac{1}{N}$, where N is the number of samples (250kHz) which results in the majority of the frequency components closer to 0, while only harmonics have some magnitude value. Directly training on unscaled magnitude features, the model achieves an accuracy of 92%. This difference seems to be caused by feature map smoothness, as seen in the difference between Figure 6.41b and Figure 6.40, where using magnitude spectrum directly, has less 'noise' in the higher frequencies in contrast to convolving real and imaginary components it produces 'noisier' output, which increases accuracy. Similarly, the imaginary components assist in determining what liquid was being measured. This may be due to some small noise footprint from individual liquids.

Chapter 7

Discussion

This work aims to improve Coriolis-based microfluidic mass flow sensors by using a machine learning-driven approach, and demonstrates possibilities to enhance MEMS-based devices using machine learning. The current standard techniques rely on physical relations to estimate mass flow, pressure, temperature, density and viscosity. With our proposed methods, we combine sensing elements to compensate for fluid property differences. This project covers a lot of multi-disciplinary knowledge, which involves the physics of fluid dynamics and its properties, as well as the physics of the microfabricated Coriolis mass flow sensor which acts as a domain knowledge to apply machine learning methods to estimate fluid properties in question.

The main results demonstrate that using a resonance frequency in combination with phase difference achieves more accurate mass flow results in contrast to just applying phase difference directly. This is due to phase difference sensitivity having a physical relation to frequency. This is discussed in Section 6.1.1. Applying a non-linear method to estimate the mass flow, the error reaches approximately 1.2% Full-Scale, which still indicates there are non-linearities in the reading, which are not captured by the fundamental signal. Using frequency spectrum instead via a deep learning approach may increase the performance even further. Likewise, a statistical machine learning approach to determine viscosity and density shows less fluid-state dependence while only using a small sample of training data (30%). Moreover, these results outperform state-of-the-art methods presented by H.Bissig et al. [71] by a factor of > 4 . The results of density and viscosity estimation from this thesis have been submitted to the peer-reviewed, 37th International Conference on Micro Electro Mechanical Systems (IEEE MEMS 2024). Correspondingly, applying similar methods to classify what fluid was measured based on its electrical readings, achieves near-perfect accuracy while only using a small fraction of the data (5%) to train the model. Similarly, a deep learning approach to process Coriolis signals shows a lot of promise. By using the frequency spectrum, instead of summarized features, it can incorporate the non-linearities in the measurements. With the current approach, the deep learning approach has shown that it is able to determine the measured fluid, outperforming a simple Naive Bayes approach, moreover, the impulse responses in the architecture output, similarly show dependence on mass flow, temperature and pressure, which indicates a unified approach is possible.

This research still has many limitations and drawbacks. For example, the labels used as a 'ground truth' in like manner have accuracy to actual true values of every measurement due to equipment having accuracies such as 0.5 % FS for mass flow readings, with differences in the accuracy of gas and liquids. With the mass flow controller being outside of the highly controlled temperature environment, may introduce more errors. Errors

were introduced by simplification of modelled labels, such as pressure drop, and labels for density and viscosity as well as accuracies of the literature values. This is more obvious with nitrogen measurements, as its density is much more sensitive to pressure, in which the pressure changes with mass flow, introducing further errors. Likewise, there are drawbacks to chosen fluids: it contains two extremes in terms of densities - water being much denser than other liquids, and nitrogen gas, which has completely different properties from liquids, e.g. it is compressible, while isopropanol, ethanol and acetone have very similar densities. The last limitation is that only one CMFS was analysed, which does not represent the average performance of a device. Miniscule variations in resistivities/sensitivities affect the performance, which may result in a lot of work to implement a machine-learning approach as a commercial product for individual devices. This requires more validation to check the transferability of models. Many of these limitations and drawbacks are easily solvable, which includes more variety of fluids, as well as more devices to validate these results.

Chapter 8

Conclusion

This research aimed to improve CMFS via a data-driven approach, analysing measured electrical signals, and combining them to improve the general performance of the device (mass flow, pressure and viscosity sensing), as well as finding potential new methods, such as estimating pressure drop purely based on Coriolis vibration signals. The main research question for this project is given as:

*How **viable** is a machine learning application on MEMS-based Coriolis flow sensors to estimate mass flow, temperature, pressure, viscosity and predict the type of fluid?*

The following sub-questions aim to answer the extent of accuracy of the machine learning approach on CMFS.

8.1 Research sub-question 1

- *To what extent in terms of accuracy, linear and non-linear models are able to combine capacitive and resistive signals measured from Coriolis Mass Flow Sensor to estimate mass flow, pressure, density, temperature and viscosity?*

Mass flow

The best-performing linear model for mass flow achieves full-scale accuracy of $\approx 2.05\%$ if nitrogen (gas) is excluded, and $\approx 2.6\%$ if including gas. Implementing a non-linear method SVR with RBF, the performance reaches $\approx 1.2\%$ even if gas is included. In contrast, using $\Delta\varphi$ as a direct measure of mass flow achieves $\approx 5\%$. [Section 6.1]

Pressure

[NOTE: *Since no accurate label was available, the pressure drop was based on on-chip pressure sensors, with a reference to pressure input.*] Linear model performed on average with an error of approximately 23.9%, showing high fluid dependence. Similarly, with a non-linear model, the pressure drop estimation performed much better, reaching an average error of approximately 3.8 %, with less dependence on fluids. [Section 6.3]

Temperature

For temperature estimation, it was found that offsets of capacitive signals are highly linear to temperature, and it is already a good estimate of temperature, on average producing

an error of 0.33 °C. Applying a non-linear model, the temperature drop is highly accurate, reaching the limit of the accuracy of measurement instruments (0.1 °C). [Section 6.2]

Density

With a linear approach, the density on average achieves $\approx 25 \text{ kg/m}^3$ error, which is very similar to the performance of simply using frequency proportional to density. The best-performing non-linear approach using GPR achieves on average an error of $\approx 0.178 \text{ kg/m}^3$ or 0.01% relative error for trained liquids. [Section 6.5]

Viscosity

Linear viscosity estimation does not outperform the Hagen-Poiseuille approach and reaches an average error of 13% (depending on the fluid state). The best-performing approach using GPR achieves an average 0.39 % error on trained liquids, excluding nitrogen, due to its compressibility factor. Similarly, it shows the potential to have similar performance on untrained fluids, achieving a 1.18 % error with offset correction on ethanol. [Section 6.4]

8.2 Research sub-question 2

- *To what extent can different fluids be distinguished from each other using measured sensing elements at different measurement conditions using probabilistic models?*

Applying the Naive Bayes classifier, it achieves 83% accuracy, mainly misclassifying acetone and isopropanol, with the two fluids being classified 67% and 61.9% correctly. Classification using Gaussian processes, achieves near-perfect accuracy of 99.5% using only 5% of training data, whilst similarly only misclassifying a couple of cases of acetone and isopropanol measurements due to their similarities. [Section 6.6]

8.3 Research sub-question 3

- *Is one of the state-of-the-art methods (DeepSense) able to capture the non-linearity's better compared to the linear/non-linear models?*

DeepSense method results in an accuracy of approximately 98% if it learns on 40% of the measured fluids. This performance is comparable to the Gaussian Process, with similarly misclassifying acetone and isopropanol measurements due to their property similarity. [Section 6.6.3]

8.3.1 Main research question

Revisiting the main research question:

How viable is a machine learning application on MEMS-based Coriolis flow sensors to estimate mass flow, temperature, pressure, viscosity, density and predict the type of fluid?

The estimations on fluid properties show a lot of promise. Density and viscosity show less fluid-state dependence, and these results outperform accuracies mentioned in an article by H.Bissig et al. [71], which showcased state-of-the-art viscosity and density determination. Similarly, determining (classifying) what liquids were measured shows superb accuracy with only a fraction of the data used, while measurements contain very similar liquids.

Appendix A

Appendix

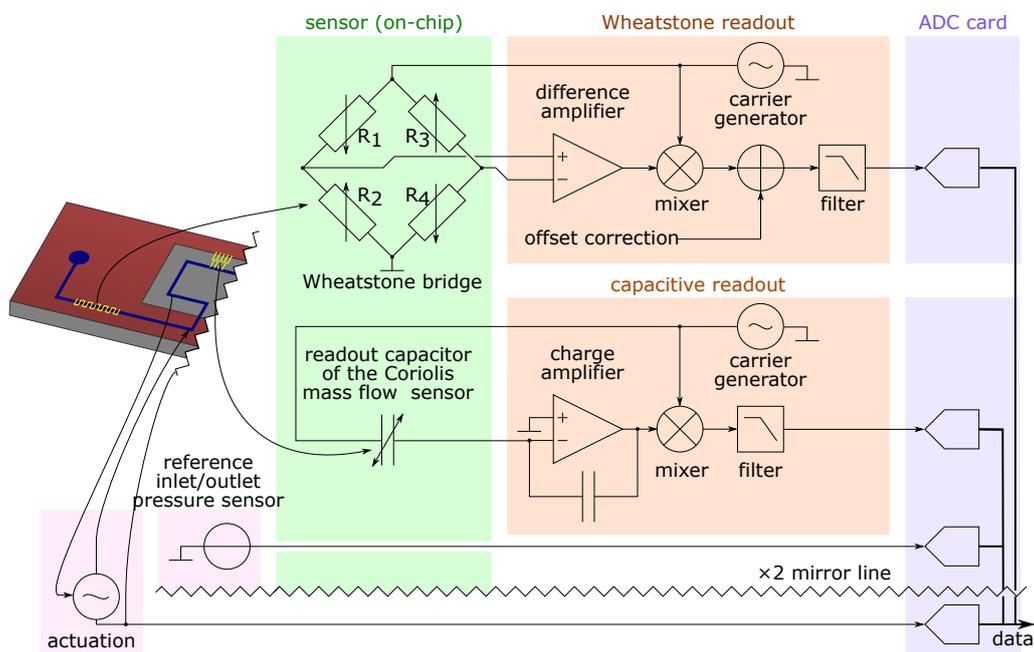


Figure A.1: Electronic interface overview of the DUT [64]

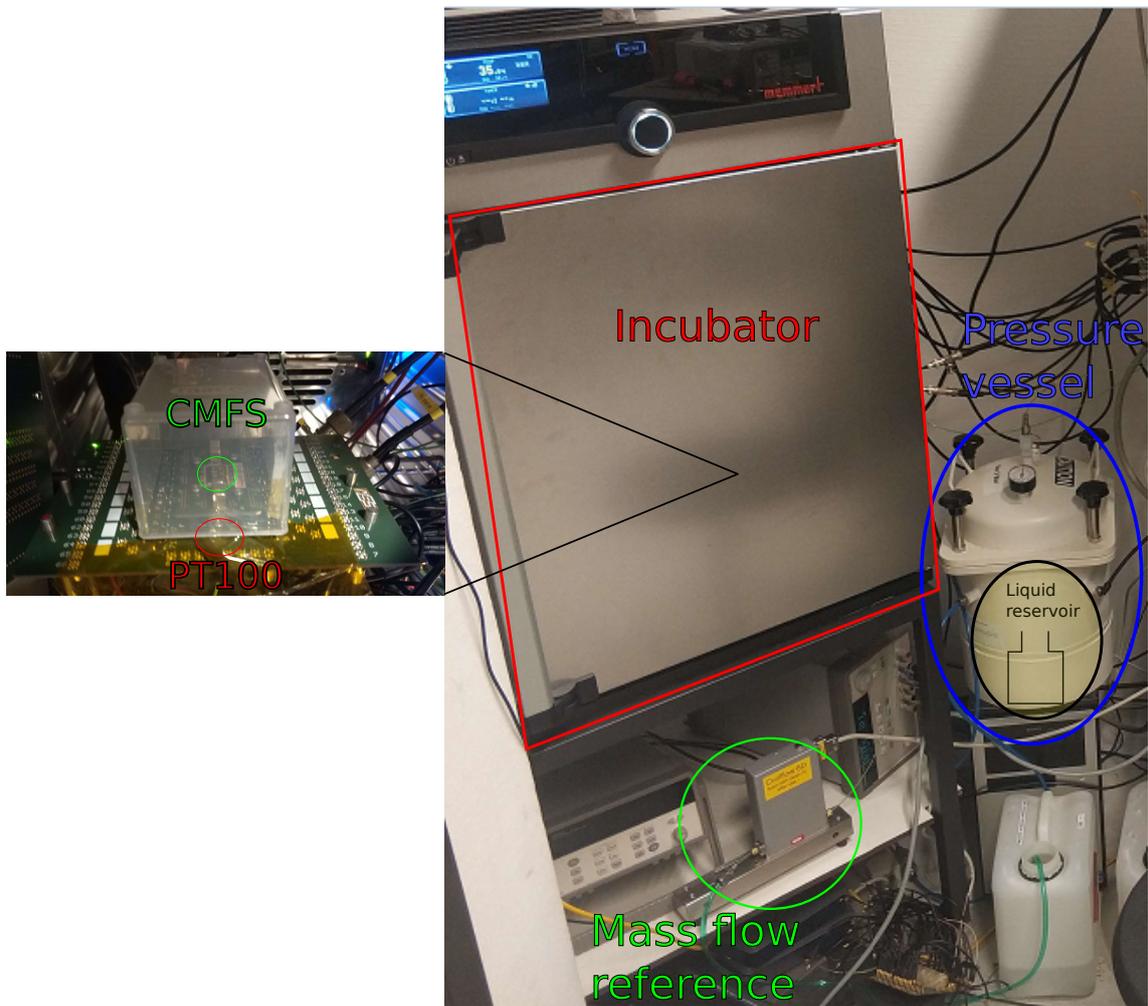


Figure A.2: Experimental measurement setup in a laboratory

Bibliography

- [1] Samer Bayda, Muhammad Adeel, Tiziano Tuccinardi, Marco Cordani, and Flavio Rizzolio. The history of nanoscience and nanotechnology: From chemical–physical applications to nanomedicine. *Molecules*, 25(1), 2020.
- [2] Marcel Dijkstra, T. Lammerink, R. Wiegerink, and M. Elwenspoek. Nano-flow thermal sensor applying dynamic w-2w sensing method. *Journal of The Optical Society of America B-optical Physics - J OPT SOC AM B-OPT PHYSICS*, 01 2006.
- [3] Sithara Radhakrishnan, Minu Mathew, and Chandra Sekhar Rout. Microfluidic sensors based on two-dimensional materials for chemical and biological assessments, 2 2022.
- [4] K.A.A. Makinwa and J.H. Huijsing. A wind sensor with an integrated low-offset instrumentation amplifier. In *ICECS 2001. 8th IEEE International Conference on Electronics, Circuits and Systems (Cat. No.01EX483)*, volume 3, pages 1505–1508 vol.3, 2001.
- [5] Neil Convery and Nikolaj Gadegaard. 30 years of microfluidics. *Micro and Nano Engineering*, 2:76–91, 2019.
- [6] R. J. Wiegerink, T. S.J. Lammerink, M. Dijkstra, and J. Haneveld. Thermal and coriolis type micro flow sensors based on surface channel technology. volume 1, pages 1455–1458, 9 2009.
- [7] Thomas Victor Paul Schut. *Towards a thermal noise limited micro-coriolis mass flow sensor*. PhD thesis, University of Twente, Netherlands, October 2021.
- [8] Dennis Alveringh, Remco J. Wiegerink, Jarno Groenesteijn, Remco G.P. Sanders, and Joost Conrad Lötters. Experimental analysis of thermomechanical noise in micro coriolis mass flow sensors. *Sensors and actuators. A: Physical*, 271:212–216, March 2018.
- [9] T. V.P Schut, D. Alveringh, W. Sparreboom, J. Groenesteijn, R.J. Wiegerink, and J. C. Lotters. Fully integrated mass flow, pressure, density and viscosity sensor for both liquids and gases. (January):3–6, 2018.
- [10] T. O’Banion. Coriolis: The direct approach to mass flow measurement. 109:41–46, 03 2013.
- [11] Fábio Ouverney Costa, Jodie G. Pope, and Keith A. Gillis. Modeling temperature effects on a coriolis mass flowmeter. *Flow Measurement and Instrumentation*, 76:101811, 2020.
- [12] Frank M White. *Fluid mechanics*. New York, 1990.

- [13] A. Kayode Coker. 3 - physical properties of liquids and gases. In A. Kayode Coker, editor, *Ludwig's Applied Process Design for Chemical and Petrochemical Plants (Fourth Edition)*, pages 103–132. Gulf Professional Publishing, Burlington, fourth edition edition, 2007.
- [14] Appendix c: Physical properties of liquids and gases. In A. Kayode Coker, editor, *Ludwig's Applied Process Design for Chemical and Petrochemical Plants (Fourth Edition)*, pages 827–862. Gulf Professional Publishing, Burlington, fourth edition edition, 2007.
- [15] Fluid property calculation. <https://www.fluidat.com>. Accessed: 2023-05-03.
- [16] John M. Prausnitz Robert C. Reid and Bruce E. Poling. *The properties of gases and liquids. Fourth edition.* 1987.
- [17] John M. Prausnitz Robert C. Reid and Thomas K. Sherwood. *The properties of gases and liquids. Third edition.* 1977.
- [18] *Gas encyclopaedia, l'Air Liquide.*
- [19] Matheson. *Gas data book, Sixth edition.*
- [20] *The handbook of chemistry and Physics. 70th edition.*
- [21] Carl L. Yaws. *Chemical Properties Handbook.* 1999.
- [22] Kabelac et al. Gnielinsky. *VDI-Wärme Atlas. Tenth edition.*
- [23] J.Groenesteijn et al. A versatile technology platform for microfluidic handling systems, part i: fabrication and functionalization. *Microfluidics and Nanofluidics*, 21:127, 07 2017.
- [24] J. Haneveld, T. S.J. Lammerink, M. J. De Boer, R. G.P. Sanders, A. Mehendale, J. C. Lötters, M. Dijkstra, and R. J. Wiegerink. Modeling, design, fabrication and characterization of a micro Coriolis mass flow sensor. *Journal of Micromechanics and Microengineering*, 20(12), 2010.
- [25] A. C. De Oliveira, T. V.P. Schut, J. Groenesteijn, Q. Fan, R. J. Wiegerink, and K. A.A. Makinwa. A MEMS Coriolis Mass Flow Sensing System with Combined Drive and Sense Interface. *Proceedings of IEEE Sensors*, 2019-Octob(2):2–5, 2019.
- [26] Dennis Alveringh, Remco J Wiegerink, and Joost C Lotters. Integrated Pressure Sensing Using Capacitive Coriolis Mass Flow Sensors. 26(3):653–661, 2017.
- [27] D. Alveringh, T. V.P. Schut, R. J. Wiegerink, W. Sparreboom, and J. C. Lotters. Resistive pressure sensors integrated with a coriolis mass flow sensor. *TRANSDUCERS 2017 - 19th International Conference on Solid-State Sensors, Actuators and Microsystems*, (1):1167–1170, 2017.
- [28] Dennis Alveringh. Integrated throughflow mechanical microfluidic sensors. 2018.
- [29] Steven W Smith et al. The scientist and engineer's guide to digital signal processing, 1997.
- [30] Joe Rodgers and Alan Nicewander. Thirteen ways to look at the correlation coefficient. *American Statistician - AMER STATIST*, 42:59–66, 02 1988.

- [31] Anestis Antoniadis, Efsthios Papanoditis, and Theofanis Sapatinas. A Functional Wavelet–Kernel Approach for Time Series Prediction. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(5):837–857, 10 2006.
- [32] Hsuan-Tien Lin and Chih-Jen Lin. A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods. *Neural Comput*, 3(1-32):16, 2003.
- [33] Luis Gómez-Chova, Gustavo Camps-Valls, Jordi Munoz-Mari, and Javier Calpe. Semisupervised image classification with laplacian support vector machines. *IEEE Geoscience and Remote Sensing Letters*, 5(3):336–340, 2008.
- [34] Christina Leslie, Eleazar Eskin, and William Stafford Noble. The spectrum kernel: A string kernel for svm protein classification. In *Biocomputing 2002*, pages 564–575. World Scientific, 2001.
- [35] Fernando Pérez-Cruz and Olivier Bousquet. Kernel methods and their potential use in signal processing. *IEEE signal processing magazine*, 21(3):57–65, 2004.
- [36] Vladimir Vovk. Kernel ridge regression. In *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pages 105–116. Springer, 2013.
- [37] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. A review of kernel methods in machine learning. *Max-Planck-Institute Technical Report*, 156, 2006.
- [38] Martin Krzywinski and Naomi Altman. Significance, P values and t-tests. *Nature Methods*, 10(11):1041–1042, November 2013.
- [39] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3), may 2011.
- [40] John Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.
- [41] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.
- [42] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press.
- [43] Jochen Görtler, Rebecca Kehlbeck, and Oliver Deussen. A visual exploration of gaussian processes. *Distill*, 2019. <https://distill.pub/2019/visual-exploration-gaussian-processes>.
- [44] Jie Wang. An Intuitive Tutorial to Gaussian Processes Regression, April 2022. arXiv:2009.10862 [cs, stat].
- [45] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.
- [46] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*. Cambridge University Press, 2023. <https://D2L.ai>.
- [47] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.

- [48] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.
- [49] Christoph Molnar. *Interpretable machine learning*. <https://christophm.github.io/interpretable-ml-book/>, 2023.
- [50] Ruiqiao Bai, Jacqueline CK Lam, and Victor OK Li. What dictates income in new york city? shap analysis of income estimation based on socio-economic and spatial information gaussian processes (ssig). *Humanities and Social Sciences Communications*, 10(1):1–14, 2023.
- [51] Alessandro Di Bucchianico. Coefficient of determination (r^2). *Encyclopedia of statistics in quality and reliability*, 2008.
- [52] Stefan Pleus, Ulrike Kamecke, Delia Waldenmaier, and Guido Freckmann. Reporting insulin pump accuracy: Trumpet curves according to iec 60601-2-24 and beyond. *Journal of Diabetes Science and Technology*, 13(3):592–596, 2019. PMID: 30311794.
- [53] T.H. Platenkamp, Wouter Sparreboom, G.H.J.M. Ratering, M.R. Katerberg, and Joost Conrad Lötters. Low flow liquid calibration setup. In *Proceedings of the Second International Conference on Microfluidic Handling Systems*, pages 165–168. University of Freiburg, October 2014. Conference date: 08-10-2014 Through 10-10-2014.
- [54] Flow meter accuracy & repeatability. <https://www.bronkhorst.com/int/blog-1/flow-meter-accuracy-repeatability/>. Accessed: 2023-05-03.
- [55] Christopher P Price. Point of care testing. *BMJ*, 322(7297):1285–1288, 2001.
- [56] Nan Chen, Shangchun Fan, and Dezhi Zheng. A Phase Difference Measurement Method Based on the Extended Kalman Filter for Coriolis Mass Flowmeters. *Measurement Science Review*, 19(2):48–52, April 2019.
- [57] W. Lin and J. . Zhao. A novel method based on hilbert transform for signal processing of coriolis mass flowmeter. *International Journal of Pattern Recognition and Artificial Intelligence*, 32(2), 2018. Cited By :5.
- [58] Qi-Li Hou, Ke-Jun Xu, Min Fang, Cui Liu, and Wen-Jun Xiong. Development of coriolis mass flowmeter with digital drive and signal processing technology. *ISA Transactions*, 52(5):692–700, 2013.
- [59] Y. Tu and H. Zhang. Method for cmf signal processing based on the recursive dtft algorithm with negative frequency contribution. *IEEE Transactions on Instrumentation and Measurement*, 57(11):2647–2654, 2008. Cited By :42.
- [60] Yanjin Zhang, Y. Liu, Zhendong Liu, and Weiping Liang. Developing a Long Short-Term Memory-based signal processing method for Coriolis mass flowmeter. *Measurement: Journal of the International Measurement Confederation*, 148:106896, 2019.
- [61] Humphrey Adun, Ifeoluwa Wole-Osho, Eric C. Okonkwo, Tonderai Ruwa, Terfa Agwa, Kenechi Onochie, Henry Ukwu, Olusola Bamisile, and Mustafa Dagbasi. Estimation of thermophysical property of hybrid nanofluids for solar Thermal applications: Implementation of novel Optimizable Gaussian Process regression (O-GPR) approach for Viscosity prediction. *Neural Computing and Applications*, 34(13):11233–11254, July 2022.

- [62] Farzaneh Rezaei, Mohammad Akbari, Yousef Rafiei, and Abdolhossein Hemmati-Sarapardeh. Compositional modeling of gas-condensate viscosity using ensemble approach. *Scientific Reports*, 13(1):9659, June 2023.
- [63] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek F. Abdelzaher. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. *CoRR*, abs/1611.01942, 2016.
- [64] D. Alveringh et al. To be published.
- [65] Max Kuhn and Kjell Johnson. *Feature engineering and selection: A practical approach for predictive models*. Chapman and Hall/CRC, 2019.
- [66] Emmanuel Lesaffre and Brian D. Marx. Collinearity in generalized linear regression. *Communications in Statistics - Theory and Methods*, 22(7):1933–1952, 1993.
- [67] Tormod Næs and Bjørn-Helge Mevik. Understanding the collinearity problem in regression and discriminant analysis. *Journal of Chemometrics*, 15(4):413–426, 2001.
- [68] Hadeel S. Obaid, Saad Ahmed Dheyab, and Sana Sabah Sabry. The impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning. In *2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)*, pages 279–283, 2019.
- [69] Dalwinder Singh and Birmohan Singh. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97:105524, 2020.
- [70] Henderi Henderi, Tri Wahyuningsih, and Efana Rahwanto. Comparison of min-max normalization and z-score normalization in the k-nearest neighbor (knn) algorithm to test the accuracy of types of breast cancer. *International Journal of Informatics and Information Systems*, 4(1):13–20, 2021.
- [71] Hugo Bissig, Oliver Büker, Krister Stolt, Emmelyn Graham, Leslie Wales, Andreia Furtado, Sara Moura, Zoe Metaxiotou, Seok Hwan Lee, Sabrina Kartmann, Jarno Groenesteijn, and Joost C. Lötters. In-line measurements of the physical and thermodynamic properties of single and multicomponent liquids. *Biomedical Engineering / Biomedizinische Technik*, 68(1):39–50, February 2023.
- [72] Shaveta Dargan, Munish Kumar, Maruthi Rohit Ayyagari, and Gulshan Kumar. A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning. *Archives of Computational Methods in Engineering*, 27(4):1071–1092, September 2020.