Acoustic Flow Rate Estimation in Urban Sewage Systems

Masterthesis in Computer Science
Submittet by
Johannes Schmidt

Rheinischen Friedrich-Wilhelms-Universität
Bonn

December 2023

Ich versichere, dass ich diese Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die Zitate kenntlich gemacht habe.

I hereby declare that this thesis is my own work and that no sources or tools other than those cited were used.

Bonn, 29.12.2023 .....

Date Signature

First Supervisor: Prof. Dr. Frank Kurth
Second Supervisor: Jun-Prof. Dr. Florian Bernard
Advisor: Kevin Wilkinghoff

Abstract

This thesis investigates the potential of utilizing audio signals captured by microphones within sewer pipes to estimate sewage flow rates during heavy rainfall. Focusing on Gelsenwasser's need for flow rate information to enhance their sewer management, this study delves into data collected from experimental setups. By examining literature on fluid sound emissions and related fields, it aims to answer the research question: "Is it possible to estimate water flow in urban sewage systems using acoustical signals?". Through comprehensive analysis of experimental data and methodical exploration of acoustical signals within urban sewage systems, this study validates the feasibility of estimating water flow rates using acoustic signals. Challenges include noisy recordings, flow turbulence, and limited data availability. Experiments employ both manual feature investigation and machine learning techniques. Findings reveal that a combined approach does not significantly outperform machine learning alone, while predicting flow rates varies within the range of its values and with the available training data. Moreover, domain shifts affect precision, indicating acoustic flow rate estimation best suits long-term, location-specific monitoring systems. Despite potential benefits for hydraulic simulations and flood detection, further investigation into data generation and system calibration is necessary for practical implementation. This work represents an initial step toward developing real-world flow rate estimation algorithms for sewer systems, emphasizing the need for more data and improved methodologies to enhance accuracy and applicability in mitigating flood risks.

Acknowledgement

First of all wuerde ich gerne Frank Kurth und Kevin Wilkinghoff fuer die Fuehrung und Unterstuetzung bei der erstellung der Arbeit bedanken. Auch danke an Fabian Fritz, der mir den Trick mit der remote conection via vscode gezeigt hat. Danke an Kai X fuer das Script um die Rechner nachts laufen zu lassen, weil sie sonst ausgegangen waeren. Die Hupenbosonen (Especially Rick Oerder) fuer den Support auf meinem Weg zum Research Scientist, VDA Vorlesung fuer die nicen Tools fuer die Visualisierung von Daten, INS Admins fuer die Linux expertiese und toleranz waehrend der heissen Phase der Thesis, Hamza fuer die inspiration wie man chatGPT nutzen koennte um sein Progremieren effizienter und strukturierter zu gestalten, Felix Boes fuer die inspiration wie man im flow bleibt beim arbeiten, Jan Hamaeckers dafuer, dass er mich im Bachelor 'dafuer bezahlt hat' micht eigenstaendig in state-of-the-art ML einzuarbeiten. Danke an Prof. Dr. Simon Stellmer dafuer, dass er mich darin bekraeftigt hat nach einer spezialisierung im Bachelor eine andere fuer den Master zu waehlen. Danke an Thomas Erben fuer die Leitung des Kures 'EDV fuer Physiker' und fuer das Buch "Einführung in Unix/Linux für Naturwissenschaftler" (obwohl der Kurs so irrelevant fuer das Physik studium scheint, hat es mich moeglicherweise - was das gelernte Handwerkszeug anbelangt - am meisten gepraegt). Zum Thema essentieles Handwerkszeug lernen muss ich mich auch bei CyberChris (und den anderen aus dem Python Vortgeschrittenen Kurs), Felix Boes, Johannes und Clelia Albrecht bedanken fuer die unschlagbaren C und Python vertiefungskurse, die - wie ich spaeter erfuhl - inspiriert waren von den Kursen von Lars Wallenborn und Jesko Hüttenhain (wovon ich die Ehre hatte von einem von den beiden bei einem Hinderniss-Halbmarathon in Gent fuer eine kleine Strecke getragen zu werden. Danke an der Stelle noch mal an Christopher Voss fuer die Gelegenheit und fuer seine 'Investition' und sein Vertrauen in mich bzgl. meiner Eignung in seine/Ralphs/Achis Fussstapfen zu treten). Diese 'Nebenangebote' bzw. Einfuehrungs-/Vertiefungskurse von Studenten/Doktoranten sind fuer mich mit die wertvollsten Veranstaltungen an Unis ueberhaubt! Danke an alle die einen Teil dazu beitragen, dass sowas stattfinden kann! Last but not least, meine (locker 60 koepfige) Familie. Alles was ich tue ist das resultat der Liebe die ueber meine Familie in mich reingeflossen ist. Ich liebe euch alle!

Contents

1 Introduction

Societies worldwide face the ongoing challenge of efficiently managing their infrastructure systems, including the crucial task of sewage management, particularly during periods of heavy rainfall. The primary objective of this thesis is to explore the potential audio data captured by microphones installed within the sewer pipes to accurately determine the flow rate of sewage systems during such rain events. Currently, the feasibility of achieving precise measurements using this approach is uncertain. However, if successful, this methodology could offer significant cost savings compared to traditional, more expensive sensor-based systems. Furthermore, an improved monitoring system would provide the means to detect and respond to flood events promptly, which have become a growing concern in NRW/Germany since the flood events that occured in summer 2021.

By investigating the viability of leveraging audio signals for flow measurement and flood detection in sewage systems, this study aims to contribute to the advancement of infrastructure management, ensuring the efficient functioning of vital systems, particularly during rainfall and potential flooding.

1.1 Motivating the research question

This research project addresses the specific needs of a company called *Gelsenwasser* responsible for managing sewer pipes and sewage flow in the German city Gelsenkirchen. Amongst others their task is to perform hydraulic simulations, which needs to be provided with flow rate information which is not available to them.

For this task they have build an experimental setup for collecting acoustic data which can be used for developing algorithms performing flow rate estimation. Developing these algorithms is the goal of this thesis, so it can be viewd as a scientific embedding of the task provided by Gelsenwasser, i.e. we make an effort to solve their this, but also investigate closely related interessting problems that become apparent.

When searching for literature one finds evidence that sound emitted by fluids contains information about its flow rate [16, 40, 3]. The precision of those estimators depends on many factors. Two important ones are 'how much noise is contained in the recording' and 'how turbulent is the flow'.

This circumstance gives raise to the following research question: "Is it possible to estimate the flow rate of water in urban sewage systems by measuring and processing acoustical signals recorded in sewer pipes?". To answer this question, the following sub-questions will be investigated:

- Are the decisions and realisations concerning the experimental setup for recording acoustical data and using it later in production made by Gelsenwasser promising?

- Are the surroundings too noisy/chaotic for training and applying a deterministic model? Can an automated filtering of the noise help with improving performance?
- How to measure and compare performances? How to decide if it is good enough for practical applications? Obviously this depends on the application in mind, but what underlying assumptions will we chose for this thesis such that the results are of practical use for Gelsenwasser? They are interested in detecting heavy rain events, so the first objective is than: Can we differentiate regular and heavy rain events? How well can we do that? Where to set the decision boundaries?

1.2 Data availability

When the project of this thesis started some data was already available. It was provided by Gelsenwasser, since they already worked on acoustically estimating the flow rate in urban sewage systems for a few years before we joined working on it. The data was gathered without our input and feedback, so their decisions implicitly influence the direction of this work, since the data will be the basis for all of our experiments.

From the beginning of this work there was the labeled data of three rain events from 2021. We were provided with further data in Autumn 2023, where one of them turned out to be not usable, due to some technical problems during recording. The other one being recorded at a second location, allowing for evaluating performance changes under domain/location shifts of the developed model[1].

It is worth noting that the above mentioned objective of Gelsenwasser to differentiate regular from heavy rain events needs data for both types of events. In the end we only had one event of heavy flow. So it is to be expected that the analysis (especially the data driven analysis) of those events will be less robust and reliable.

1.3 Relying Hypotheses

Since there is no literature for this exact scenario, the literature for related fields was reviewed. We attempted to examine some of the hardware related decisions like using cheap microphones instead of expensive ones and applying automatic gain control, but also reviewed the results of experiments conducted in other comparable fields and what methods/algorithms were used. Furthermore we checked how water resource technicians usually do their monitoring.

In the literature most of the work had the capabilities to produce more data when it was needed and they had the capabilities to modify the surroundings to reduce the amount

[1]This supplementary data was not announced by Gelsenwasser in advance, so we did not take them into account for the initial planing phase. This is why there are two sections for experiments, since the new data opened up more possibilities for experiments previously not possible.

of noise. While both is not the case here, which suggests worse performance as the one achieved in the literature, their results can be used as a first benchmark in terms of expectable performance.

So in the body of this work we try to state as explicitly as possible on what hypothesis the literature relies on and carefully decide if for our application the same assumptions can be made. If it is the case, comparable experiments are performed with the hope of reaching comparable results.

1.4 Experimental design

The experiments performed here aim to explore different approaches and techniques. On the one hand we performed experiments focusing on manual investigation of human understandable features and on the other hand we used machine learning methods (whose decision finding processes are in general not interpretable by humans.).

When the limitations of these approaches for solving the task provided by Gelsenwasser were reached, we started exploring different objectives, like the already mentioned investigation of domain shifts. For that we staid within similar techniques as the previous experiments to maintain a degree of comparability.

1.5 Outline

The rest of this thesis is structured in the following way: Section 2.1 will present the initial motivation of Gelsenwasser, to precisely define the objective of this thesis. Then, Section 3 will review the literature and set the direction of the experiment design. After that Section 4 will define and refer to the theoretical background of the used and proposed methods followed by the experiments aiming at solving the task provided by Gelsenwasser, Section 5. After that Section 6 will continue with experiments focusing on certain flow rate regions and domain shifts . In the end Section 7 will summarise our conclusions complemented by Section 8 drawing an outlook for further investigations. Section 9 is the appendix, here mainly used for storing images that would otherwise interrupt the flow of reading when kept in the main text.

2 Gelsenwasser

Gelsenwasser (GW)[2] is the company that raised the research question. The following section presents the motivation that led to the investigations made in this thesis and the data acquisition of experimental data carried out by GW.

2.1 Motivation

Lets tell the story of how the question of using acoustic signals for flow estimation came up, because it already gives insights about what the developed technology needs to be used for later.
It will also help to deviate from the business goal and focus on the research question and explain why the data available for experiments is the way it is.

Gelsenwasser works on hydraulic calculations for sewer networks. These calculations aim to simulate the sewage system under load. In this case *load* refers to a large amount of liquids flowing through the network. In their calculations they try to determine where jams are likely to occur and what consequences this has for the entire city (e.g. to predict where water accumulates). That way they can locate potential problem areas in the infrastructure. This may be helpful for preventing problems by informing the cities about it so they can either work on their infrastructure directly, or are better informed in moments of emergency. One particular application would be a warning system for early detection of flood events so work forces can react accordingly.

For that simulations to work properly they need a digitized version of the entire sewer network. Usually their clients (cities, local authorities, communities) maintain datasets containing information about those systems, however with a different focus. Such datasets contain the location, length and diameter of the tubes and shafts, where they come together and other features. These features are captured to estimate the quality of the network such that their customers know when and where to renovate and improve on the system, since that drives expenses.

However, for hydraulic calculations three quantities are necessary: Position (where lies the tube and in what direction does it go), diameter of the pipe, and its slope. According to GWs own information, the slope is never present in those datasets. The reason for that may be that it is irrelevant for the state of the system. Cities, etc. care about when and where they have to repair parts of the network. Slope does not seem to correlate with this property a lot, so they do not make the afford to collect it. This complicates the work of GW. Without that information their calculations seem to be not very useful.

This leaves GW and many others with insufficient amount information to properly do

[2]Their official website: `https://www.gelsenwasser.de/`

hydraulic simulations. So it is up to them to collect the data themselves. Scanning through the entire network is highly impractical. For that they need to build up a team that drives to every man hole and measures slopes. To do this, streets need to be blocked, which has to be communicated and planned with the city. Moreover, note that even going into every man hole will not cover the entire system, but it is the closest one can get. Since scanning the network manually is not a satisfying option, they need another way to determine the slope.

2.1.1 Importance of the slope

The reason why the slope is so important is because the velocity of the liquids flowing highly depends on the slope. It is intuitive that a steep slope would allow for more flow. This of course saturates as soon as the tube fills up.

So another way of estimating the slope is by measuring the velocity of the liquids, which in this text will be referred to as flow rate. Since most of the time the situation is calm and under control, measuring the flow rate is only possible during rainfall. This measured flow rate can then be compared to the one predicted by the simulation. This way the simulation can be re-calibrated and improved over time when it deviates from the measurement.

Apart from being dependent on the weather, another drawback of this technique is that taking measurements inside of sewer pipes is difficult and thus expensive. According to GW, a single sensor (e.g. *CSM Korrelations-Keilsensor* by *NIVUS GmbH*) costs roughly between five and seven thousand euros. The act of mounting them is of similar order of magnitude. This is not feasible for many kilometers, since they expect a need of 100 to 300 sensors.

Such sophisticated commercially available sensors offer a precision that is often not necessary for calibrating the simulations, usually they provide a precision of several liters per second.
In the moment of interviewing for this thesis GW used three categories of states a pipe can be in: Dry weather condition (no rain), average rain condition and roughly filled to the top, during heavy rain events.
This rough classification already improves their hydraulic calculations significantly, which proofs the exuberance of high precision instruments for this particular application. If higher precision is needed, it can be iteratively improved further by refining those categories, depending on the accuracy of the sensor and needs for the calibration.

For the reason stated above, companies like GW are interested in cheap sensors that only roughly estimate the flow rate. So the cost of the sensor itself should be low as well as mounting them. They say it is even within the resources if some of those sensors are calibrated by a more expensive sensor or if the sensor needs a longer duration (years) to calibrate itself over time.

2.1.2 Filling level is not enough

One idea might be to use cheap distance measuring devices based on contact less radar emission and detection. Measuring only the filling level is not enough. This section will explain why.

Measuring the filling level can be done densely in sewage networks, because such sensors are cheap. They rely on radar or ultrasound and are often used for bulk materials or water levels in general. Depending on the application they are indeed used in sewer systems, however, for flow rate this is not enough and here is why:
It is possible to have backlog in sewer pipes, since there is not just water going through them. Such jams cause higher levels but not higher flow rates. In fact it could actually stagnate entirely and have zero flow at all. Although detecting jams is also an important source of information when monitoring the capacity utilization of a sewage system, it will not help in measuring flow rates.

GW proposed to utilize acoustic sensors to differentiate between filled pipes with flow and filled pipes without flow. It is intuitively clear that those two events would sound differently. This brought up the idea of combining the filling level measurement with measuring the sound of it.

For the further discussion it is important to differentiate between two objectives one might have in mind when designing acoustic systems for flow rate estimation:

- The first objective is to use the acoustic data in combination with other sensors like the ones using the fill level or other sensors that measure the amount of rain that falls on the ground.
- The second is to ask the question if and how well measuring only acoustical signals can be used for estimating flow rates in sewer pipes.

In this work the focus lies on the second objective as a research question. It is to be expected that adding information by gathering more data and performing sensor fusion might improve the performance. However, this question was not studied here. Furthermore, this work will not go into detail how the above mentioned early warning system could be set up utilizing the acoustic sensor network. This objective would be out of scope for this work.

The next section will present the experiments related to measuring acoustical signals for flow rate estimation that have been conducted by GW. This will serve as an introduction into the dataset available for the experiments carried out in this thesis.

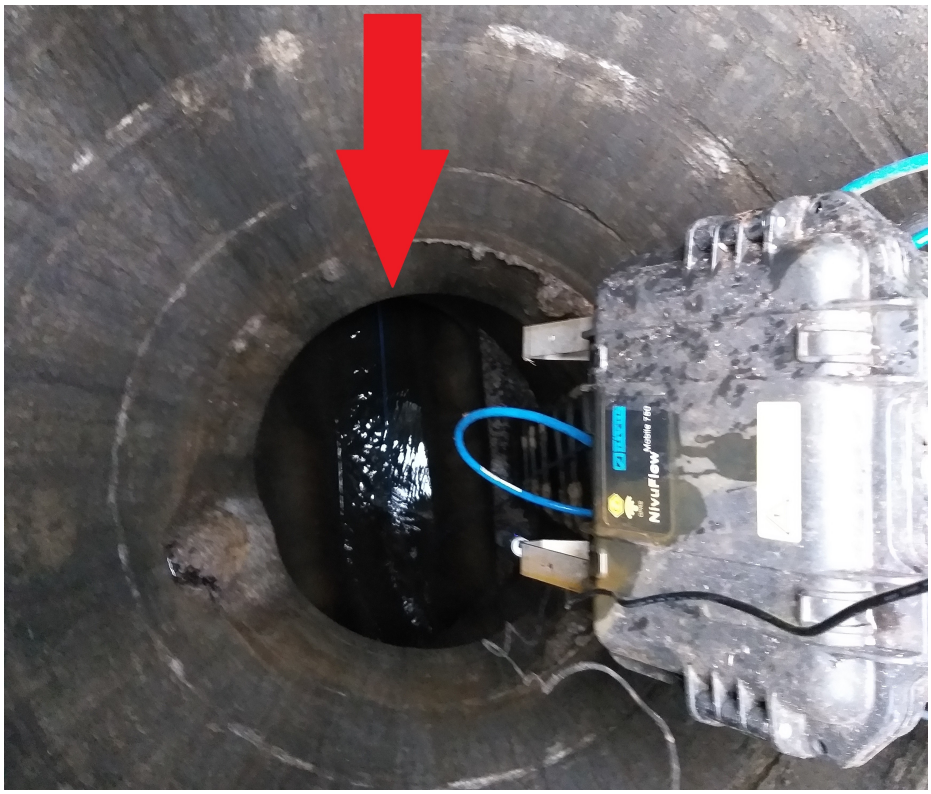

Figure 2.1: NivusFlow measuring device inside a sewer pipe.

2.2 Data collection/description

2.2.1 Experimental setup

GW already did some experiments by measuring sound inside of sewer pipes. The purpose of their experiments was mainly to generate labeled data, to investigate the feasibility of using acoustic sensor measurements for estimating flow rates. In this case labeling refers to measuring with two devices at the same time. The second sensor was an expensive industry standard for measuring the flow rate.

The device used by them is called *NivuFLow Mobile*, a mobile solution form *NIVUS GmbH* for flow monitoring inside partially and entirely filled pipes. It consists of a waterproof box that contains the battery and electricity and the sensor. In Figure 2.1 one can see a box connected to the sensor with a blue cable. In Figure 2.2 one can see how the sensor is mounted inside the sewer pipe within the liquid flow, being connected to a blue cable. More details on the expensive sensor can be found in Section 3.1.3.

Next the prototype of GW for an acoustic flow rate sensor will be presented. Figure 2.3 shows how they look like from the outside.

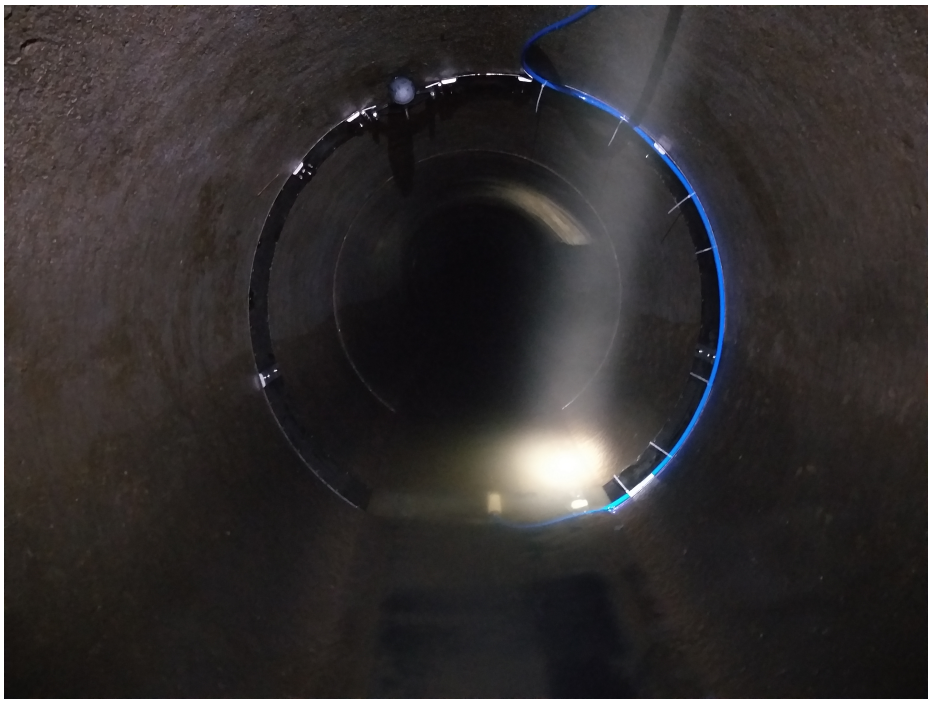

Figure 2.2: Sensor inside the sewer pipe connected to the NivuFlow Mobile Box.

The sensor was build by GW in collaboration with an external company, which has more experience with building prototypes.
Their prototype consists of a low-cost microphone that costs less then one euro inside a small pipe to protect if from damage caused by external forces (dirt, wetness, hitting against some obstacle). This small pipe is connected by a cable (in Figure 2.3 both are grey) to another box (in Figure 2.3 the black one) that contains the electronics and power supply. The power is supplied by a lead battery and is designed to provide the electronics with power for up to three weeks. The cable that connects the microphone to the box has a length that allows to directly place the box below the man hole. The box can be directly placed below the man hole, while the microphone is close to the flowing water.

The electronics consists of a Raspberry Pi Zero with a SD Card, an analog to digital converter (ADC) and a LoRaWAN chip for Wireless communication.
This setup is designed to prepare more functionality that could be used in future experiments. At the time the measurements available for this thesis were collected, the Raspberry Pi was only used to receive the data coming from the ADC, which was connected to the microphone and storing it directly on the SD card in a .wav format.

The **LoRaWAN Chip could further be used** for remote configuration, control and data

Figure 2.3: Boxes containing all the electrical elements.

transmission. The Raspberry could be idle the entire time and only start measuring (and supplying power to the ADC) after a remote signal was send. LoRaWAN could also be used to synchronise the time between the ground truth measuring device and the Raspberry and/or other Raspberries. It could even be used to send the predicted flow rate, so some server would collect the data and store and/or analyse it. The last scenario is often called edge computing in the Internet of Things (IoT) scene. Since LoRa uses only a small bandwidth it could only send features online, the raw data would be too much.

GW took several places into consideration for mounting their prototype. In the end they chose a larger pipe with a diameter of one meter since they were interested in heavy rain events. Sewer pipe systems can be thought of branches of trees. They consist of many smaller pipes that come together to bigger ones, which finally ends in a single big one. For the measurements they decided for a bigger one, because here many branches already came together so the probability to observe high flow rates is higher.

The process of mounting involves preparation. A street needs to be blocked for that time. Such activities have to be communicated beforehand with the city council. GW has build five prototypes. To have some redundancy while measuring they mounted three of those boxes each a single sewer pipe, see Figure 2.4. One observation was that it does not make a large difference when one compares the signals of different devices. Even though the exact location, especially the distance to the flowing water, of the microphones is different. This means that the process of mounting does not require much detailed adjustments.

When the experiments were made GW noticed that the Raspberry shuts down after 3 to

Figure 2.4: Three prototypes inside the man hole.

4 days. This is surprising, because it is supposed to last up to three weeks. They assume that the Raspberry has some automatic shutdown functionality that is supposed to recognise the final discharge voltage of its power source to initiate a proper shutdown to prevent data lost of failures. The assumption now is that the Raspberry is a bit too sensitive and gets triggered to early. However, this is only an assumption, there might be a different reason why this happens.

GW gathered data from 11.07.21 to 28.12.22 (1.5 years). For that period of time they have continuous measurements from the NivuFlow device and some measurements made by the prototype. In Figure 2.7 one can see this history. The green and yellow boxes indicate events where measurements from the prototype are available. For this thesis only the first three of them were provided: (14.07.2021, 28.07.2021, 28.09.2021). These three cases already cover all relevant scenarios: $> 200\,l/s$, $< 200\,l/s$, the special $1000\,l/s$ case and almost no flow. The last scenario is covered by all the recordings, since the recording sample start with quiet conditions until the rain/flow event occurs and it ends in quiet conditions again.

In Figure 2.7 one can also see that relevant events are rare. There is only one event with $1000\,l/s$ and only five events with equal or higher $400\,l/s$. This sparsity of relevant data already hints at problems in the training process later on.

Figure 2.5: Gws box inside the sewer pipe. This is one way the data was collected.

2.2.2 Dataset description

Next the data available for experimentation is presented: The next section in part summarizes the findings found in the lab course [19] where I already worked on a subset of the data, but with the objective of detecting anomalies as a noise extraction method.

The dataset comprises information that was gathered over the course of four days measured at one location. Audio recordings were taken every ten minutes, with each recording lasting for a duration of 10 seconds. Additionally, for some of these days, the flow rate was measured every five minutes. Each folder contains a maximum of 420 .wav files that have a sampling rate of 48 kHz. In total, there are 1169 .wav files that occupy approximately 1 gigabyte of space.

A random subset of data was inspected in order to obtain an initial impression of it. The corresponding audio was listened to and their spectra were examined. Figure 2.8 displays some of the spectra that were chosen carefully.

It was observed that some of the audio files were corrupted by white noise, either for short-term interruptions or for the entire ten-second period. There were also very short transients in the signal, and the number of these peaks varied from file to file. Some files contained only a few of these peaks, while in others it was difficult to distinguish them from pure white noise.

Figure 2.6: How the recording setup looked from outside the sewer pipe.

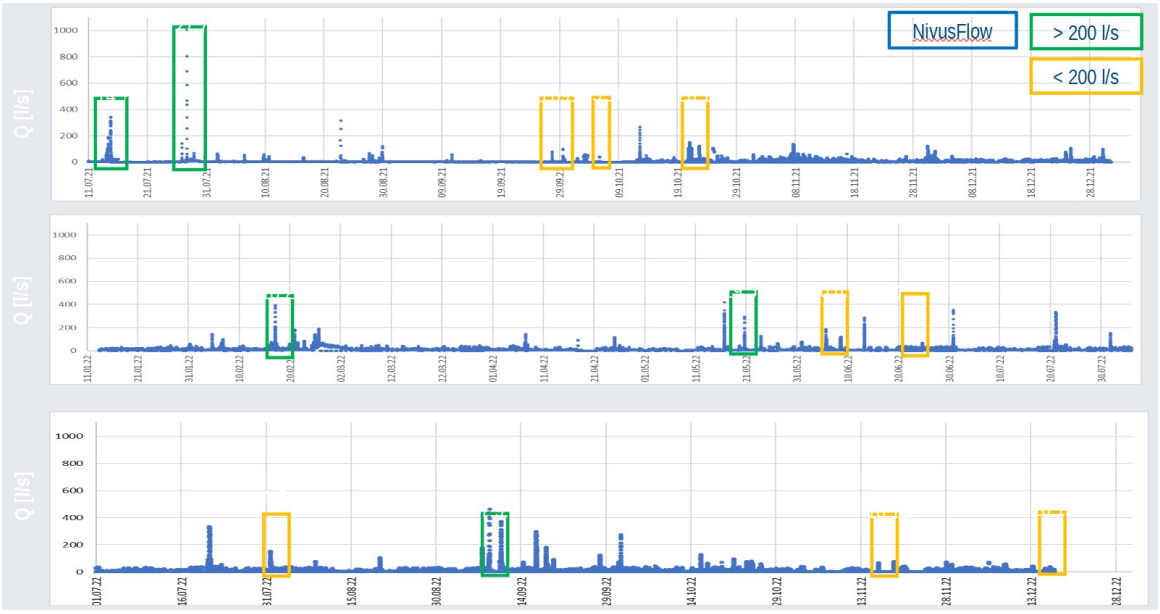

Figure 2.7: The measuring history of the prototype and the NivuFlow device. The green and yellow boxes indicate events where measurements from the prototype are available.

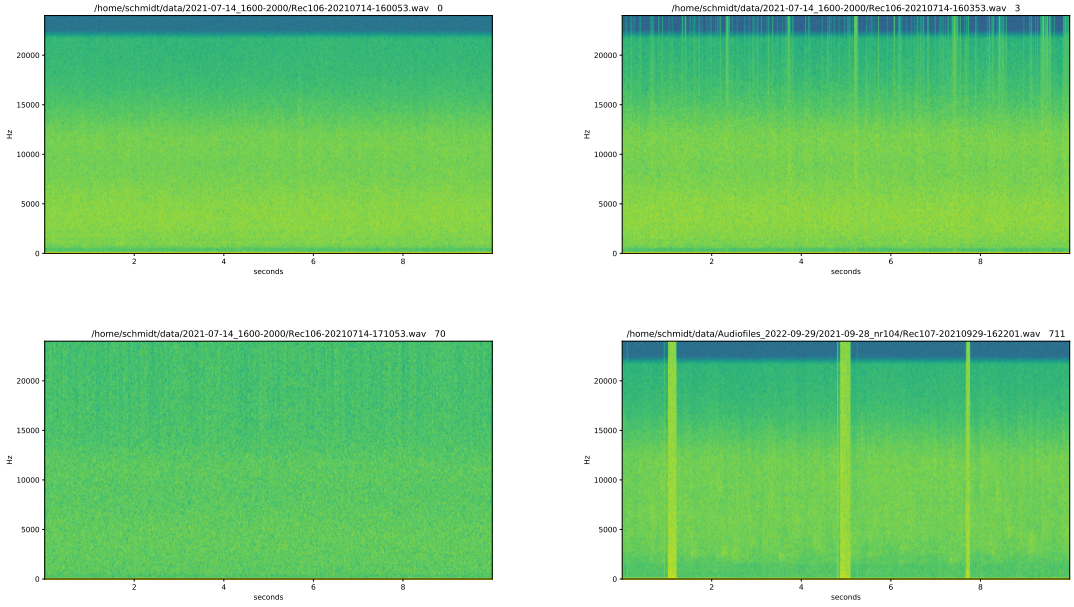

Figure 2.8: Some examples for noise contained in the dataset. upper left: no noise, upper right: many crackle sounds, lower left: close to white noise, lower right: short term white noise events.

Interestingly, the presence of these peaks and white noise events could be visually detected by the fact that much energy of the signal was contained in frequencies above 20 kHz. This suggests that these events were not directly related to the sounds transmitted by the liquid flow, as normal recordings do not contain any energy above 20 kHz. However, it is possible that they were indirectly related to it, as they could be caused by clipping during the recording. As a result, the occurrence of these distortions could be used as an indicator of high flow rate. More Discussion on this later.

In addition to the white noise distortions, there were other short-term events present in the data, such as drip sounds, car wheels, car brakes, and even a siren. The corresponding spectrograms of them are depicted in Figure 2.9. However, most of the samples remained constant during the entire 10 seconds. Furthermore, there were some subtle artifacts in the recordings, such as a constant received signal below 20 Hz or the application of compression with a release time of around half a second for loud short-term events. It was also observed that the drip sounds were almost entirely covered by the recordings of one particular measuring day, which could be due to some specific object that got stuck in the pipe and, therefore, exhibit a different characteristic sound in general instead of a rare occurring event in that case. This suggests that the acoustic situation will vary strongly between different locations, and a thorough calibration may be necessary for each location.

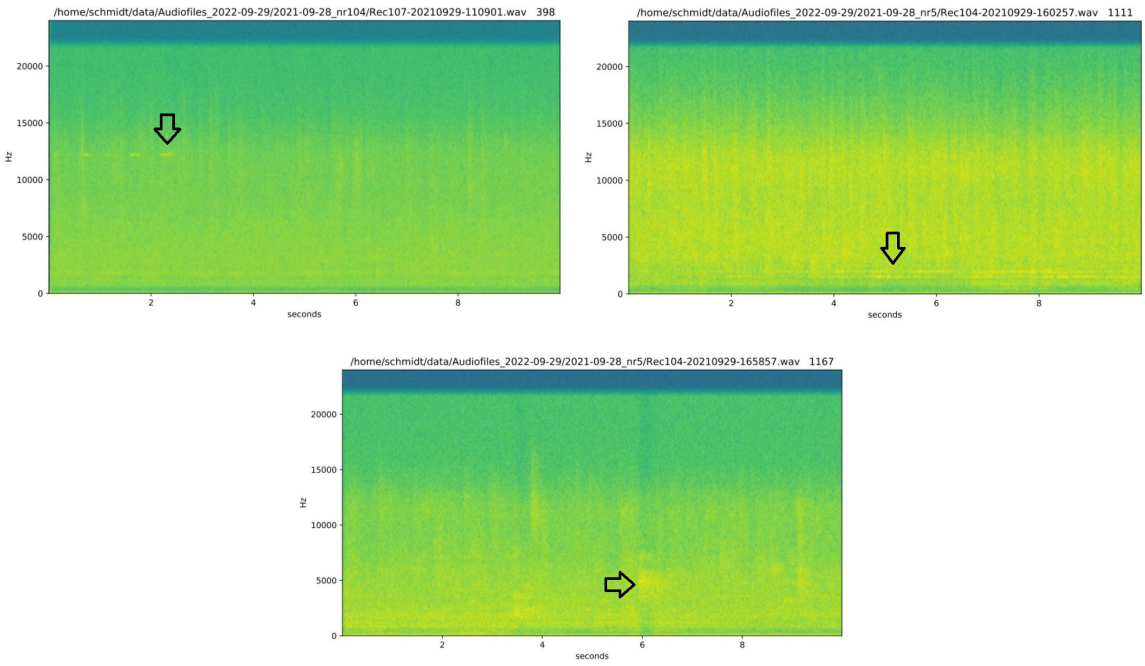

Figure 2.9: Some examples for noise in the dataset, caused by external sound sources. upper left: break of a car, upper right: siren (see the bottom part), bottom: click sounds (there one can notice the auto-gain).

2.3 Data Overview and Assumptions

In this section, we will provide an overview of the ground truth data used in our study, in the following called Nivus data, because Nivus is the name of the company that has build it. We have access to data from five distinct events, recorded at two different locations. Note that until now we only showed three events and one location. This is because two events were provided much later.

2.3.1 Nivus Data

The following list enumerates the flow rate recordings made by the Nivus device during these five events:

1. Event 1, which occurred on 29.09.2021, starting at 11:01 at location Ueckendorfer_Str 2.10.
2. Event 2, which occurred on 14.07.2021, starting at 23:01 (Ueckendorfer_Str) 2.10.
3. Event 3, which occurred on 27.07.2021, starting at 16:46 (Ueckendorfer_Str) 2.10.
4. Event 4, which occurred on 12.08.2023, starting at 14:00 at a different location (Holtkamp) 2.11.
5. Event 5, also occurring on 12.08.2023, starting at 14:00 again at the first location (Ueckendorfer_Str) 2.11.

A few observations can be made from Figure 2.10 and 2.11, which provide insights into the characteristics of the recorded events:

- Different maximum values can be observed. Event 1 exhibits a peak flow rate of 100 l/s, while event 3 reaches 1000 l/s. This discrepancy in peak values suggests variations in the intensity of the rainfall events.
- The recorded events also differ in terms of their durations, both in the number of data points used for recording (400 data points) and the duration of high-flow moments. This variation may be indicative of the temporal characteristics of the events.
- Furthermore, the curves representing these events exhibit diverse patterns. Some events display longer periods of increased activity, suggesting sustained high flow rates, while for others, this high flow period lasts for only a few minutes. This divergence in the course of the curves may hint at variability in the responses of the studied system to different rainfall events.

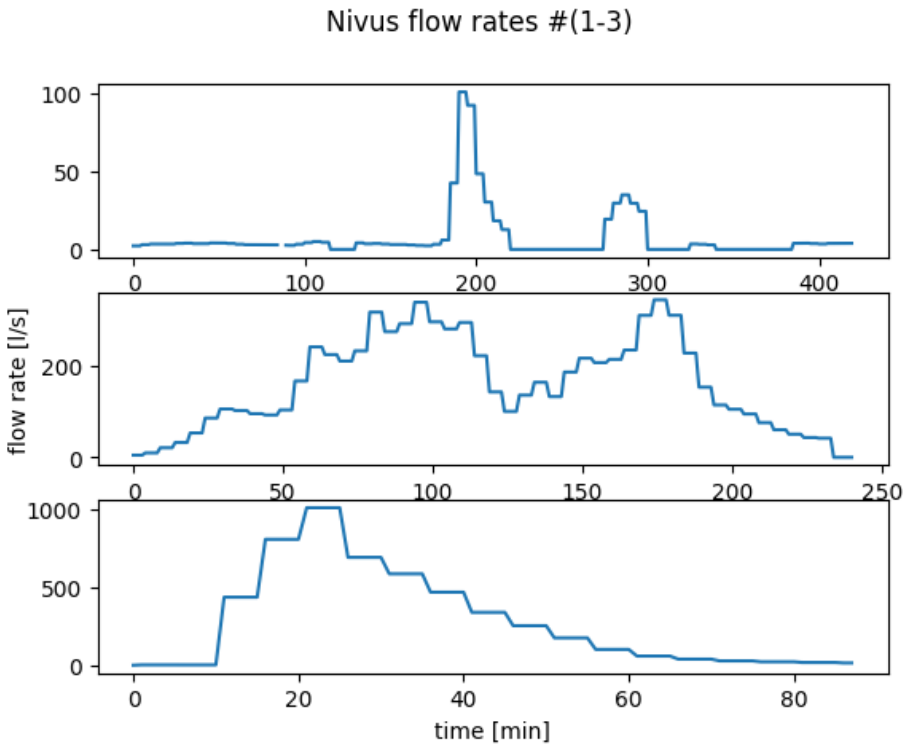

Figure 2.10: Flow rates for Event 1, 2, 3.

- Notably, the last recording appears to be a flattened version of the one above, with different maximum values. According to GW this phenomenon is not unusual and can be attributed to the influence of precipitation characteristics on the channel hydrograph's shape. Depending on flow times and the size of the sewer pipes , different levels of damping (stretching of the curve) or intensity (larger catchment area leading to higher amplitude) may occur. This variability is influenced by the specific location in the network where the rainfall runoff is being measured.
- Lastly, it is worth noting that the last two recordings share the same datetime in the database. According to GW, this is likely because these recordings were indeed measured simultaneously in two different locations.

Overall, these observations underline the complexity and variability of hydrological responses to different rainfall events, emphasizing the need for a nuanced understanding of the system's behavior under various conditions.

2.3.2 Data Consistency

We encountered certain challenges regarding data consistency, particularly in the synchronization of timestamps:

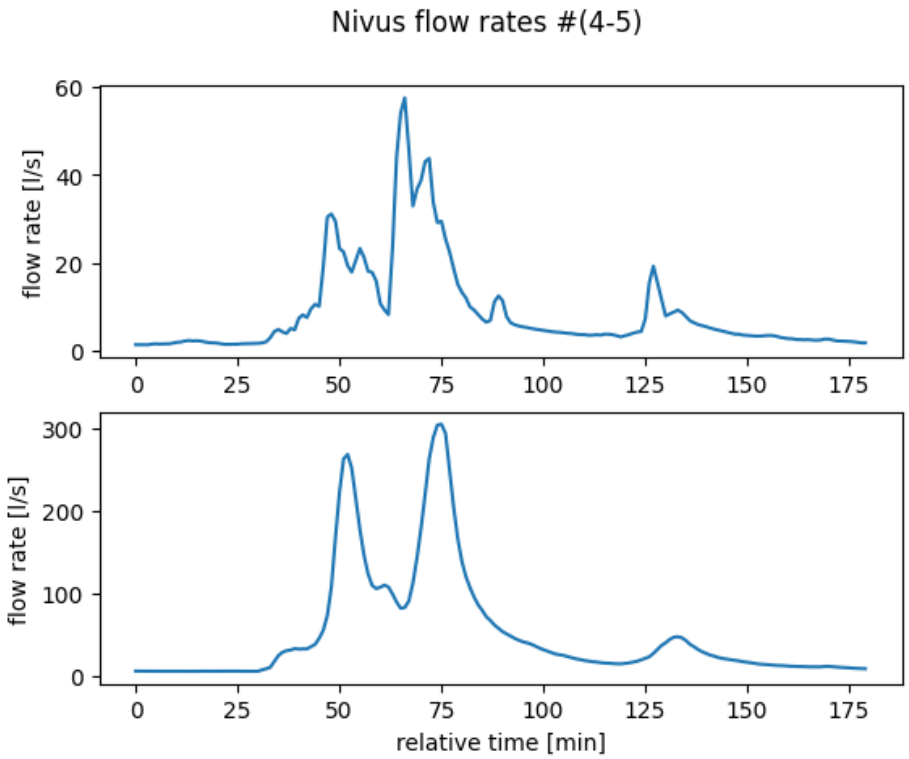

Figure 2.11: Flow rates for Event 4, 5.

- Inconsistencies in the naming of event dates were noted. The folder dates containing audio recordings did not always match the dates mentioned in the corresponding CSV file names, leading to some confusion in the labeling of events.
- Another inconsistency was observed in the number of audio files compared to the number of rows in the CSV files. Some events had more recordings or labels than others, indicating potential synchronization issues.

Despite these labeling inconsistencies, the data source assured us that the assignments in the CSV files are accurate. While the labeling may be confusing, the data's quality and the experiments performed on it are not significantly affected. It'is important to note that acquiring data of this nature is challenging, and efforts were made to ensure the data's reliability. The main takeaway from this is that initial care scales better than fixing problems like this afterwards.

The interpretation of the data presents some challenges, primarily because we lack information about rainfall events from other sensors or sources. Therefore, it is not always clear if rainfall was the sole cause of the observed events. For example, the first event's cause remains uncertain. In contrast, the second event on 14.07.2021 is attributed to a well-known flood event in NRW/Germany [12]. The third event, with a flow rate of 1000 l/s,

is likely associated with rainfall, as this magnitude exceeds what household and industrial discharges could produce at that location.

3 Literature Review

This section will give the necessary background needed to understand the problem at hand, what has already been done by other researchers and findings from similar research fields.

To the best knowledge of the author, at the time of writing this thesis there was neither work published in journals/conferences nor preprints directly addressing the task of estimating flow rates in sewer pipes using acoustical data. However, there were two closely related questions with existing literature to look for to learn more about this problem:

What challenges do water resources technicians, infrastructure and utility companies face when they need to monitor their systems?

and

Are there other applications, where acoustic sensors are used to capture the sound of liquids flowing for estimating flow rates?

In the following, these two questions will be investigated in two separate sections followed by a third one with other related work.

3.1 Challenges for infrastructure and utility companies

The information presented in this section is mainly drawn from a *Water Environment Federation* Fact Sheet [41], an article by *Short Elliott Hendrickson Inc.* [25] and from meetings with experts from Gelsenwasser[3]. In the following, explicit citations are only made when those three sources deviated in point of view from each other or when another resource was used.

3.1.1 Why flow estimation in sewer pipes

Monitoring the flow plays a vital role in assessing and characterizing flow conditions in sanitary sewer collection systems. The real-time utilization of this data has become increasingly crucial for decision-making, optimization, maintenance, and regulatory compliance within the industry. Tackling flood events is another crucial motivation. Monitoring the behavior of sewer systems under load can help to identify and locate trouble zones early and prevent some of the damage. Depending on the exact infrastructure it might also help during a flood event. However this is of less relevance, because during a flood the biggest thread is on the street.

The exact purpose of monitoring is highly application-dependent. Usually there are clear objectives and expectations causing the investment of installing monitoring hardware, such as preventing flood and damage or examining the condition of a sewer system.

[3]Note that much of their situation has already been presented in section 2.1

Depending on the purpose, one needs to choose between a permanent or temporary setup. For the temporary setups, the timing and duration has to be determined. For example a temporary setup aimed at monitoring the flow rate during rainfall is best suited during seasons with higher probability of rain. Another relevant parameter is the placement of the devices. How many sensors shall be used or how dense shall the sewer system be covered by sensors.

It is an industry-wide standard that rainfall is measured separately using dedicated systems or external services/companies. Getting this right increases the quality of the data. This is especially true for the monitoring of so-called *rainfall derived infiltration and inflow* (RDII). Infiltration refers to the unintended entry of external water into sanitary sewer systems due to structural deficiencies or aging. Inflow refers to the unintentional entry of external water into sanitary sewer systems from sources such as rain water, private property drainage, or other external sources, often times abbreviated by I/I.

The next section will depict a concrete example of wastewater analysis. It will show how monitoring the flow rate can be used as a tool in this context.

3.1.2 Flow Monitoring and Analysis in Sewer Systems, Colombia

Carlos-Alexis Bonilla-Granados et al. published their wastewater analysis in a journal called Respuestas [6]. The main objective of their research is to estimate the factors specific to the sanitary sewerage system of San José de Cúcuta, a city in Columbia. They measured and computed them and discovered that the used model overestimates the capacity needed. Although those factors and their main findings are not directly relevant for this thesis, their work is presented here for two reasons: First, to get a first impression of the order of magnitude for the properties one deals with, when working with wastewater systems. Second, to show a concrete example of how monitoring flow rates in sewer pipe systems can solve real world problems. Readers more interested in the motivation behind such investigations are encouraged to read their paper. Here only the circumstances are outlined.

The selected area of investigation in Cúcuta was roughly $1.7\,km^2$. Wastewater coming from this area flows through the pipe where the measuring devices are installed. This order of magnitude for the area that captures the rain can roughly be considered a similar order of magnitude for the data from GW. The monitoring in Cúcuta was conducted for 19 weeks, 24 h per day. For the measuring an ultrasonic flow meter with data logger was used. For more information about such devices see Section 3.1.3. The flow was measured in units of $[m^3/h]$.

Figure 3.1 shows the total value of the aforementioned monitoring data of wastewater flow in the sewage system. For this image they decided to show the highest value of the day. The days with significantly higher values correspond to rainfall events. The frequency of occurring rainfall events highly depends on the geographical location of the city. However, the estimation of roughly 10 out of 150 days (so 15 %) is of a similar order of magnitude for

regions managed by GW. Also note that events where the system is under high load (more than twice as much flow as the ground level) are very rare (1 out of at least 150 events, so less then 1 % of the days). Hence, it is to be expected that algorithms that rely on data recorded at events of (possibly) maximum load of the system might take a long time to get calibrated, because such data is rare.

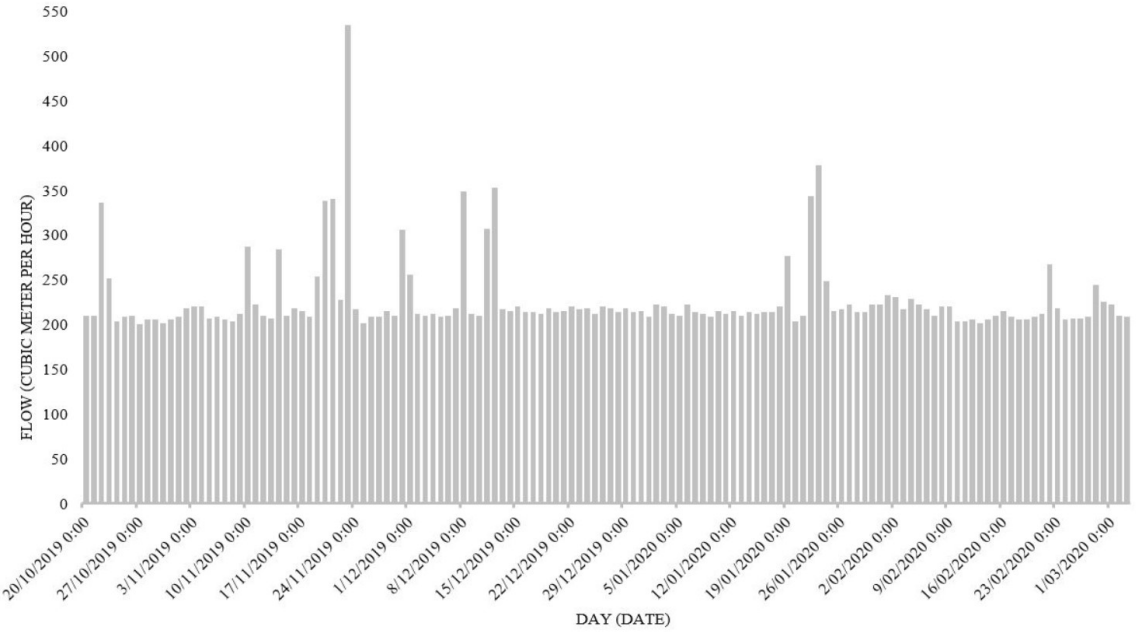

Figure 3.1: Monitoring of wastewater flow (max value of the day) in the sewage system of Cúcuta during the measurement period. Source of image:[6]

Figure 3.2 shows the daily variation of wastewater flow in the sewage system during dry weather conditions. Again the exact values will deviate depending on the location, but the relative slope of change in value and and variance between the different days will look similar.

In summary, good monitoring of one city can simplify setting up a similar system in another sity. This is especially true for flood events, since they happen relatively rarely per city (at least for German cities it is less than once every 10 years). It is typical to try to estimate properties that cannot be measured due to absence of information. More conclusions are drawn in the end of Section 3.1.

3.1.3 Measuring equipment

Following the assessment of [41], here is a non-exhaustive list of currently used methods depicting some of their strengths and weaknesses:

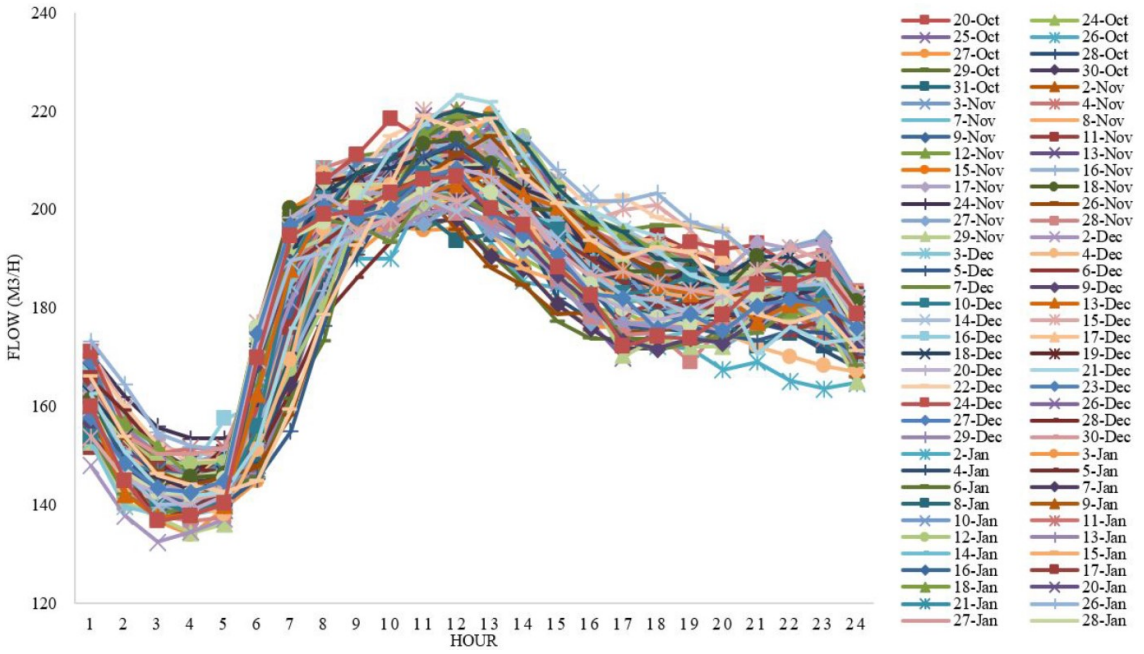

Figure 3.2: Daily variation of wastewater flow in the sewage system of Cúcuta during dry weather conditions in the measurement period. Source:[6]

Ultrasonic sensors are widely employed for non-contact flow rate measurements. The underlying technology is different from device to device. Some rely on the Doppler effect, some on Time-of-Flight effects and some identify the trajectory of particles floating in the water and estimate the velocity of the liquids by estimating the velocity of the particles. GW used a device that utilises the third approach. Advantages of those sensors are their high accuracy and their flexibility in terms of the sizes and materials of the pipes they are mounted in. Also they never have direct contact with the wastewater, and thus require less maintenance. However, their limitation is that with high turbulences and therefore bubbles inside the pipe the accuracy drops. Furthermore, they are typically the most expensive sensors.

Electromagnetic flow meters, also known as magmeters, are commonly used devices. These sensors operate based on Faraday's law of electromagnetic induction, which states that when a conductive fluid flows through a magnetic field, a voltage is induced that is proportional to the flow velocity. They are best suited for the largest pipes in use and also reach high accuracy. Their biggest drawback though is that they are particularly difficult to set up and don't operate well on fluids having conductivity smaller than $20\,m\Omega/cm$.

Pressure sensors rely on measuring the pressure difference across the pipe and then calculating the flow velocity based on Bernoulli's principle and pipe characteristics. They are

cost efficient, simple to install and require low maintenance as they consist of few moving parts. Limitations are that their accuracy depends on flow conditions and pipe material to a greater amount than the other methods presented. This results in those sensors being unpractical in some environments

The cost of these sensors lies in order of magnitude of a couple of thousand euros, with ultrasonic sensors usually being the most and pressure sensors being the least expensive. There are more techniques (such as so-called bubbler or float sensors), but there is no point in discussing them in the same amount of detail, because they do not offer any benefits over the other sensors discussed above. In practice, setups often combine multiple types of sensors to fit individual needs. Note that the quality of the data highly depends on the execution of the building/mounting process. Proper installation and calibration of the equipment by qualified staff following manufactures recommendations and industry standards is key. Once this is complied, regular setup inspections are scheduled and executed. The goal is to detect erroneous configurations (e.g. synchronisation/clock errors or battery issues) early. The data is gathered for data analysts. Often companies use so called *geographical information systems* for combining all this information to spot overall trends of the network.

Sometimes companies are only interested in the level of the water and not in its velocity. However, this is not our objective here, since hydraulic calculations cannot be made by level alone. For more details on why that is, see Section 2.1.

This brings us to the main point of this section: To the best knowledge of the author, there is no industry standard for cheap - low accuracy - sensors (less than a thousand euro) to have a rough estimator for flowing activities inside sewer pipes. It is still to be decided if those can potentially help satisfying the listed needs of water resources technicians, infrastructure and utility companies. As discussed above, collecting the necessary amount of data for data-driven algorithms takes a long time and thus is often infeasible, especially for the task of flood prevention. This likely is the reason why hydraulic simulations are made in the first place.

Moreover, the listed equipment has to be installed either when building the pipe or with quite some effort afterwards by mounting devices at the bottom of the pipe, while ensuring that there is no or almost vanishing flow. The proposed prototype by GW only needs to be mounted right underneath a manhole cover. This makes it very easy to install and maintain the sensors.

While using theories, models and simulations in the absence of information is a common approach in water management investigations, there was no investigation found that the usage of many cheap sensors combined with few high quality ones fails to improve the situation. So utilising many acoustical low-cost sensors might indeed be a promising strategy, or at least a new niche. Also note, that some of the problems with sophisticated methods remain a challenge in acoustical flow meters. Those are battery live and the question of

how to transmit the data from the sensor to the controlling station.

3.2 Estimating flow rates using acoustic signals

Lets first investigate into the following key question:

Is there evidence that estimating flow rates using acoustic signals is indeed possible with sufficient accuracy[4]

There are publications investigating this question as well as groups developing systems for different applications. In the following those are quickly reviewed such that their main findings have enough context for our investigation to draw connections.

3.2.1 Relying Hypotheses

Jacobs He et. al. [16] investigated the question if an mathematical model can relate recorded sound of water flowing through a tap with the corresponding flow rate. The actual flow rate was measured separately by a typical volumetric measuring device as a ground truth for developing the model.

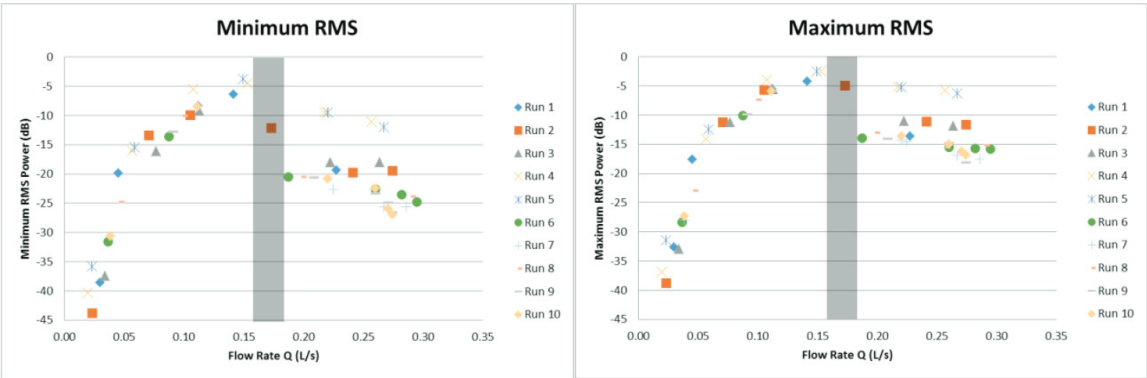

Figure 3.3: Results of thir sound signal amplitude analyses. The moment of saturation is marked by the grey vertical bar. Source of image: [16]

To be more precise, the experimental setup was an outside tap with a pickup microphone - manufactured for acoustic guitars - mounted underneath it. The water coming out of the tap was forwarded by a hose pipe, so the sound recorded was the sound of the water going through the tap. There were no splash sound from the water hitting the ground involved. Furthermore the authors made special effort to filter out every unwanted external sources of sound as for example barking dogs and detecting on- and off-set of the flowing event.

[4] Again, what can be considered "sufficient" depends on the objective in mind. For our investigations the classification into three classes (dry weather, rain, heavy rain) as mentioned in Section 2.1 is enough.

They have compared peak amplitude (in dB) and root mean squares (RMS) as amplitude features first. Frequency specific features computed by the Fourier transform were investigated too. For the amplitude they already discovered that the relation between the amplitude of the signal and the flow rate saturates at some point. This discontinuity was between $0.16\,l/s$ and $0.18\,l/s$. This saturation behaviour of the amplitude can be seen in Figure 3.3, with the grey vertical bar marking it.

The discontinuity caused ambiguous behavior in the amplitude. Therefore, one can conclude, with an application in sewer pipes in mind that treading all frequencies the same and averaging over them might not be the most suitable feature.

For this reason they put their main attention onto five frequencies, also called modus. This requires some effort and involves trial and error.
In Figure 3.4 (left plot) one can see one of those features. Discovering this saturation effect, motivated features that describe what is happening with higher flow rates.

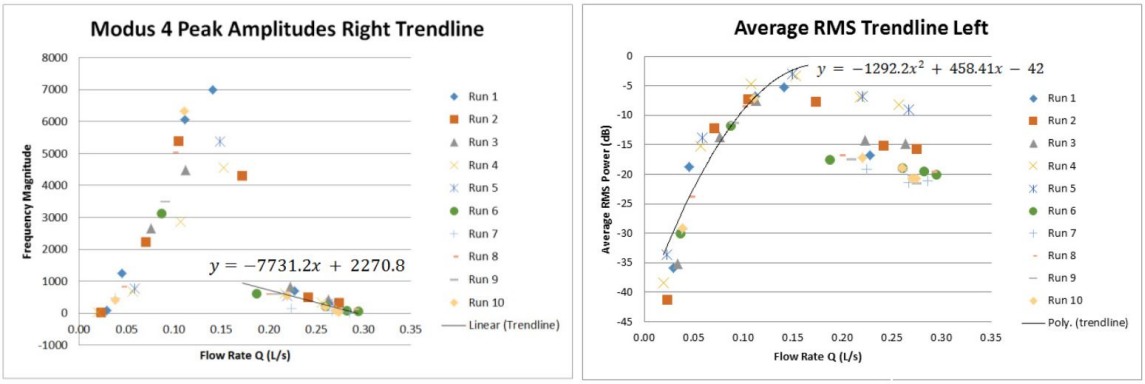

Figure 3.4: Trend lines used to estimate flow rate with DISFLOREM. Source of image: [16]

In the end they came up with multiple models, here only the one they called DISFLOREM is reviewed. It is the only one presented here, because that is the one they showed with larger detail. This model is depicted in Figure 3.5. In short, this method is described in table 3.1.

They achieved an average error of 15 % when results were verified against five independently recorded data points. This means, their prediction misses the ground truth by 15 % on average. In total they collected a dataset of 60 recorded sound signals, so they used roughly 8 % of their data for testing. Therefore we can keep the following in mind for our own investigation: It seems reasonable to assume that in the context of the sewer pipes the performance will be at best as good as the performance of this water tap experiment, because we still have noise in the data and they do not have noise in the data.

1.	For a given audio sample compute the amplitude of the entire recording as average RMS (RMS_{Ave}) and the peak amplitude of the frequency 180 Hz (modus 1 : A_{M1}) and 7999 Hz (modus 4 : A_{M4})
2.	Check the range in which the values A_{M1} and A_{M4} lie in.
3.	Depending on those values solve a linear or quadratic equation for the flow rate Q

Table 3.1: Rough steps of the DISFLOREM method by [16]. The decision tree is shown in Figure 3.5.

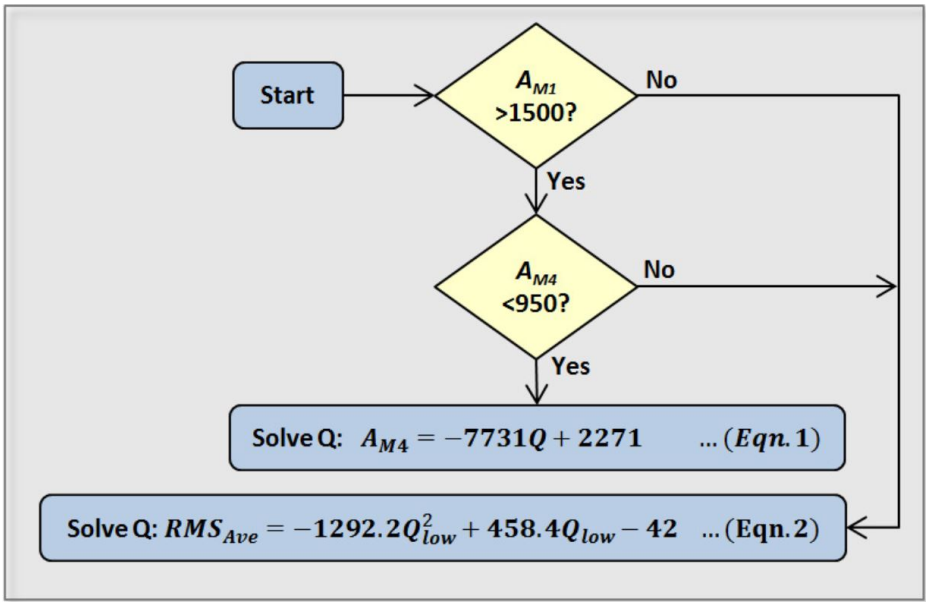

Figure 3.5: Decision tree of the DISFLOREM model. The meaning of the signs can be checked in Table 3.1 or in the original paper. The source of the equations can be see in Figure 3.4 Source of image: [16]

Also note that they did not investigate how the learned model performs, when a tap with different size, shape or material was used. It is to be expected that the performance drops, when circumstances change, because the emitted sound changes entirely. More evidence for that is discussed later in this section when we talk about the discontinuity in the model.

It is worth noting that this is not the first paper bringing up the question of estimating flow rates utilizing sound. They refer to a work that discovered a nearly quadratic relationship between the signal noise and flow rate in a pipe [9]. This however only holds within a specific region. A similar behaviour was expected and confirmed by [16].

This work gives a good starting point for our investigation in terms of expected accuracy, amount and quality of data necessary, what type of audio features to consider first and what kind of data driven methods/algorithms might serve as a starting point.

3.3 Related work on analysing acoustic signals for regression tasks

In this section there is more work presented that utilizes acoustic signals to estimate quantities from the source of emitting the sound. The purpose of this section is to get further inspiration on possibly promising methods to try. To this end, the literature in this section will be discussed briefly by presenting the main findings and relating them to the topic of this thesis.

3.3.1 Flow rate estimation for agricultural sprayer nozzles [40]

Their idea is to use microphones as flow meters for nozzle tips in agricultural sprayers. Under laboratory conditions they achieved an accuracy of 5 % relative RMSE. This would definitely be accurate enough for our application.

Another interesting finding was that the distance between the nozzle and the microphone did not change the accuracy a lot, once calibrated. That is also useful to know for our application, since this relaxes the constrains on the cable connecting the microphone to the box. There should not be much adjustment necessary. However, the change of distance required a re-calibration.

Another useful finding was that each nozzle requires its own calibration, so it is expected, that the sewer application will be similar. The calibration was even necessary, when they varied the distance. This also suggests, that when unmounting the device and mounting it again with a different location (for the most part height) of the microphone, this will negatively influence the accuracy.

In their study they put a special emphasis on normalizing the numerical values before the regression task, see Figure 3.6. The min and max values correspond to the minimum and maximum of the tested flow rates of the particular nozzle tip assessed, respectively. Additionally, during training, the min and max values were obtained, once again, individually for each assessed nozzle tip. This circumstance is an important detail to keep in mind during the development of the data processing pipeline.

The last relevant finding is the comparison of high- and low-end microphones. Their observation was that consistent results can be obtained when using a low-end microphones. When compared to more expensive high-end microphone there is no measurable improve-

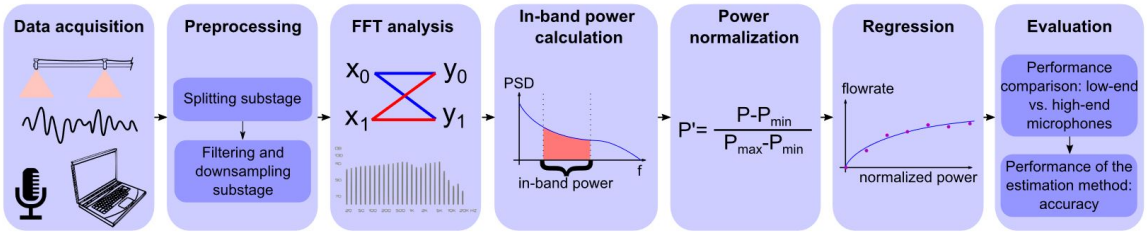

Figure 3.6: Overall block diagram summarizing the main processing stages performed in the study. Source of image: [40]

ment in accuracy. Therefore, there is no point in upgrading the microphone. Doing so will not improve the prediction accuracy. This of course cannot be generalised for every application, however, expensive microphones for the most part have higher prices, because they were optimized for a certain application. Often this is a linear frequency characteristic or optimization for recording voice or instruments. It may also be optimized for capturing very quiet sounds. There the noise-poor amplification of the signal is equally important. All those adjustments in the microphones properties do not seem to relate to the scenario of sewer pipes.

Indeed, the last observation turned out behave the same way for the context of sewer pipes, because this behaviour was confirmed by experiments done by GW.

3.3.2 Varying liquid jet stream onto a free surface [3]

This paper investigates the sound produced by water jets falling into a pool to predict the flow rate and trajectory. Two methodologies are explored: Utilizing machine learning models trained on audio features extracted from the collected sound to predict flow rate and trajectory, or directly acoustic parameters associated with spectral energy for flow rate trajectory estimation.

The paper compares the effectiveness of these approaches against the actual flow rate, demonstrating their alignment and accuracy in predicting flow rate trajectory.

Their investigations motivate that utilising machine learning methods without much preprocessing might be a promising strategy as well. However, note that their dataset is much larger than the one we have available and their dataset represents the underlying distribution well, since they can just produce more data if it is needed.

3.3.3 Acoustic vehicle speed estimation [53]

This study focuses on estimating vehicle speed using acoustic measurements from a single sensor. A novel feature, dependent on speed and derived from sound amplitude attenuation,

is introduced. This speed-dependent feature is extracted from audio signals and utilized as an input for a regression model aimed at speed estimation.

To facilitate this investigation, a dataset comprising annotated audio-video recordings of single vehicles passing by a camera at consistent known speeds has been collected, annotated, and made publicly available. The dataset encompasses 304 recordings captured in urban environments, featuring ten distinct vehicle types. The proposed method is both trained and evaluated on this compiled dataset.

Experimental results indicate the method's efficacy in accurately predicting the moment a vehicle passes by and estimating its speed, achieving an average error of 7.39 km/h. Discretizing speed into 10 km/h intervals, the proposed technique attains an average accuracy of 53.2 % for correct interval prediction and 93.4 % accuracy when allowing for misclassification within adjacent intervals. Moreover, the experiments highlight the substantial impact of sound disturbances, particularly wind, on acoustic speed estimation.

From this study we learn three main things for our investigation:

- First, keeping the noise decreases the accuracy, but only to some degree. This of course depends on the amount of noise, but it seems comparable to our situation.
- Second, discretizing the output into classes instead of using continuous values simplifies the task and allows for easier interpretability and comparability of results.
- And third, allowing for misclassification within neighboring classes, allows for ersimating the amount of classification errors made by the model.

3.4 Other Related Work

3.4.1 Low-Cost Home Activity Recognition by Fogarty et al. [10]

The following paper performs a classification task on raw audio, by only recognizing the duration of sound events. So it works with audio data, but does not perform any signal processing like Fourier transformations, still it is similar by utilizing many cheap and small sensors for monitoring. The objective was to develop a low cost sensor-based systems for monitoring water use activities in daily living. For this purpose, multiple low-cost microphones at locations of minimal systematical noise are used across the household (see Figure 3.7). The recordings were analysed according to the duration of disruptions in the sound signal.

They were able to identify the following scenarios with an accuracy here given in percent: 100 % of clothes washer usage; 95 % of dishwasher usage; 94 % of showers; 88 % of toilet flushes; 73 % of bathroom sink activity lasting ten seconds or longer; 81 % of kitchen

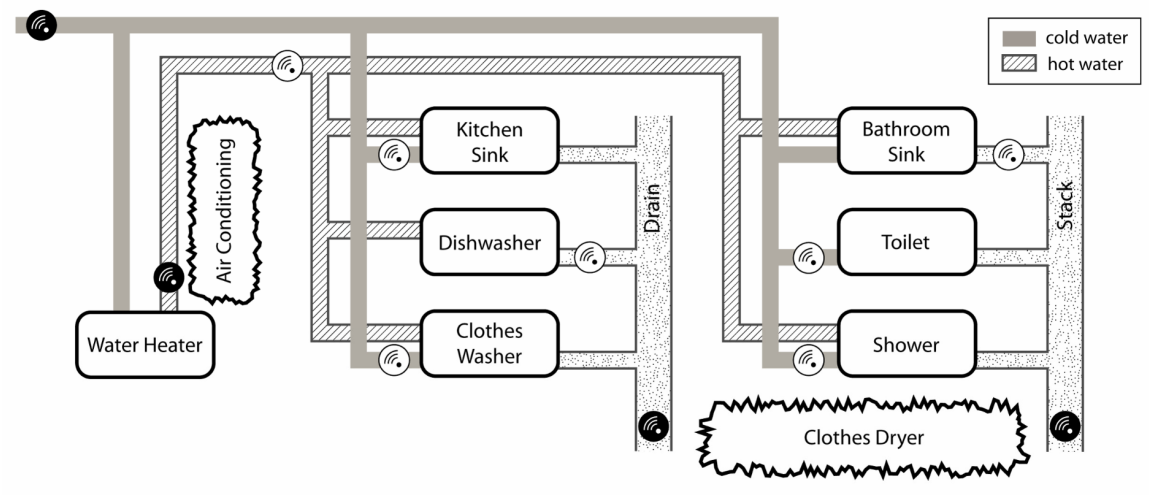

Figure 3.7: Scheme of the water pipes in the home. The air conditioning and clothes dryer are shown because they rattled nearby pipes, introducing noise that needs to be considered in analyses. The four shaded sensors are used for modeling activities, while the unshaded sensors are included only for validating their results. Source of image: [10]

sink activity lasting ten seconds or longer.

The work by Fogarty et al. [10] confirmed that sound could be used to obtain an indication of the water use. Even though clustering the duration of events and assigning them to previously known events does require much less knowledge about signal processing than estimating flow rates from the sound/timbre of a recording, this paper gives further evidence that there is useful information captured in audio recordings. It also suggests that the duration of an event contains information about it.

There is also the option of only using one sensor at a pipe just before entering the wastewater system, which was done by Hu et al. [1]. They achieved to identify four water-use activities: bathing, toilet flushing, cooking and clothes washing. The system could recognize about 70 % of those water-use activities. This and the previous work confirm that avoiding as much noise as possible pays off in the analysis afterwards.

3.4.2 Results of the Anomaly detection investigation

In Prior work, we investigated detecting anomalous data in the same dataset that is used in this thesis. There we mainly focused on detecting white noise, since there are some faulty interruptions in the data.

White noise detection worked well as an anomaly detection task, however as we will

discover later in this thesis, white noise is an important indicator when it comes to the prediction of very high flow rates.

At the point the anomaly detection investigations were made it was not clear what features will be used for measuring the flow rate so the amount of work that went into checking for events like the siren or the breaks will probably have little effects on predicting flow rates.

There we also noticed some auto-gain occurring. There was an event with a 'loud' click and one could clearly see that the entire signal became quiet for a moment and raised its gain again shortly after that, the same way it is done for vocal recordings in a studio with so called compression applied.

For the coming investigations we did not apply any of the developed anomaly detection methods, due to the reasons mentioned above.

3.5 Summary of main findings

There are attempts on predicting properties like velocity and flow rate from acoustic data. But there is no published work that tackles that problem for the specific situation of sewer pipes.

Other flow rate prediction investigations make hope that the goal of this thesis can be accomplished at least to some accuracy. The main differences between our situation and the ones reviewed are that they usually manually removed noisy data and they manually configured the gain control of the electronics. So we have to deal with white noise interrupt and the auto-gain, which requires for robustness of the methods.

The reviewed work utilizes a variety of methods. The most promising amongst them, seem to be manual investiation of the amplitude for specific frequency bands [9] and machine learning methods [3].

Although the manual investigations produce human understandable features and models - and in their case even perform well -, still since it is not using much automated feature extraction techniques, it is likely that some relationships between features remain undiscovered. Also in practice there is more noise exposure expected compared to most of the experiments revieved here.

Our work combines a few edge cases relevant for practical application that have not been sufficiently explored in previous research: Insufficient amount of training data due to the high costs of creating more data, noise contained in the data and varying recording conditions inside the data. All of this might often be the case in practice. There is a trend towards many small/cheap sensors that collectively perform better than one expensive sensor [8] .

This can be seen for cameras and the entire Internet of Things movement. This work might open up a new potential field of application for that philosophy.

So despite GWs application, from a research point of view this thesis will also consider the difficulty of collecting data in real world environments, investigating the robustness of the methods presented above.

Furthermore this work will also make an effort to combine manual models on human understandable features with machine learning methods. This way of combining these two methods might improve the performance of the presented work as well.

4 Theoretical Background

In this section, we look into the theoretical foundations and concepts that underpin the analysis and methodologies used in this thesis. We expect the reader to have a basic understanding of the methods and concepts related to signal processing tools, particularly the theory behind the applied techniques, such as the Short-Time Fourier Transform (STFT)[5]. This assumption is based on the knowledge covered in the courses MA-INF 2113 - Foundations of Audio Signal Processing and MA-INF 2212 - Pattern Matching and Machine Learning for Audio Signal Processing, led by Prof. Dr. Frank Kurth. Similarly, familiarity with machine learning concepts, including Intelligent Learning and Analysis Systems: Machine Learning (ILAS-ML) and technical neural networks (tNN), is assumed.

4.1 Common Features in Audio Analysis

In the context of the application treated in this thesis, signal processing (in our case particularly feature extraction) primarily serves to reduce irrelevant information or to highlight relevant information from the underlying signal. Everything is calculated by means of the signal, but due to the sampling rate, the signal usually has a dimension that is very high and complicates statistical analyses [44].

This work focuses on exploring and applying common features in audio analysis. We do not provide in-depth discussions of all concepts, but rather give short introductions with just enough theory so it becomes clear how they fit into the investigations made here.

For readers seeking to refresh or deepen their understanding of these concepts, we recommend the following resources:

1. Fundamentals of Music Processing, Second Edition by *Meinard Müller*[33].
2. An Introduction to Audio Content Analysis, Second Edition by *Alexander Lerch*[26].
3. Introduction to Audio Analysis: A MATLAB Approach by *Theodoros Giannakopoulos and Aggelos Pikrakis*[13]

These resources serve as references for gaining a comprehensive understanding of the foundational concepts in audio analysis, which is not a prerequisite but definitely helps to comprehend the subsequent sections of this thesis.

Core Principle	Feature [unit]	Explanation
STFT		Short-time Fourier transform (32 bins)
	Pitch [Hz]	Pitch tracking on thresholded parabolically-interpolated STFT.
	PSD / Magni [Hz-dB]	Power spectral density / Magnitude of STFT
Spectral Features		Based on STFT and Pitch.
	center_freq [Hz]	Compute the spectral centroid.
	spec_bandwidth [Hz]	Compute p'th-order spectral bandwidth.
	spec_contrast [dB]	Compute spectral contrast.
	spec_flatness [dB]	Compute spectral flatness.
	zero_cross [counts]	Compute the zero-crossing rate.
	rms [amplitude]	Compute root-mean-square (RMS) value for each frame.

Table 4.1: List of Features

4.2 Signal Processing Features

In table 4.1, we present a list of core signal processing features that were examined in this study. These features serve as fundamental building blocks for our audio analysis and provide valuable insights into the characteristics of the data. In the following sections, we will investigate some of these features more in-depth, providing detailed explanations and analysing their relevance to the research conducted in this thesis.

4.2.1 Pitch Estimation Algorithms

Pitch estimation algorithms, also known as Pitch Detection Algorithms (PDAs), are computational techniques designed to estimate the pitch or fundamental frequency of an oscillating signal[42]. These signals can include digital recordings of speech, musical notes, or tones. The primary objective of pitch estimation is to determine the frequency of the dominant periodic component within the signal. This estimation can be achieved through various computational methods, considering both the time domain and the frequency domain, or a combination of both. In this work the implementation of the librosa python library (librosa.piptrack) was used[27].

Pitch estimation algorithms play a crucial role in a variety of domains, including phonetics, music information retrieval, speech coding, and musical performance systems. However, the specific demands placed on these algorithms may vary depending on the appli-

[5]There is a nice introduction in Fourier Analysis and the Short-Time Fourier Transform with particular focus on digitization and its application for Audio in the book *Fundamentals of Music Processing* by *Meinard Müller* (Chapter 2) [33]. The lectures MA-INF 2113 - Foundations of Audio Signal Processing and MA-INF 2212 - Pattern Matching and Machine Learning for Audio Signal Processing, led by Prof. Dr. Frank Kurth covered this material.

cation context. In our concrete application, since the data intrinsically has no particularly dominant frequency in it (white noise like signal), we would a priorily not suppose that this feature contains useful information. Still, we include the pitch feature in our investigations to verify this and in order to be more comparable to the first investigations that were done by GW on this application and where the feature was used [11].

It seems plausible that traditional pitch estimation methods are not suitable for white noise-like data, and they may not yield meaningful results. Instead, one should focus on alternative acoustic features and analysis techniques to gain insights into the unique characteristics of sewer pipe sounds and acoustics. In the experimental section there will be some short discussion on these features.

4.2.2 Spectral Flatness

Spectral flatness is a measure used in digital signal processing to characterize an audio spectrum. Spectral flatness is typically measured in decibels, and it provides a way to quantify how much a sound resembles a pure tone, as opposed to being noise-like[20].

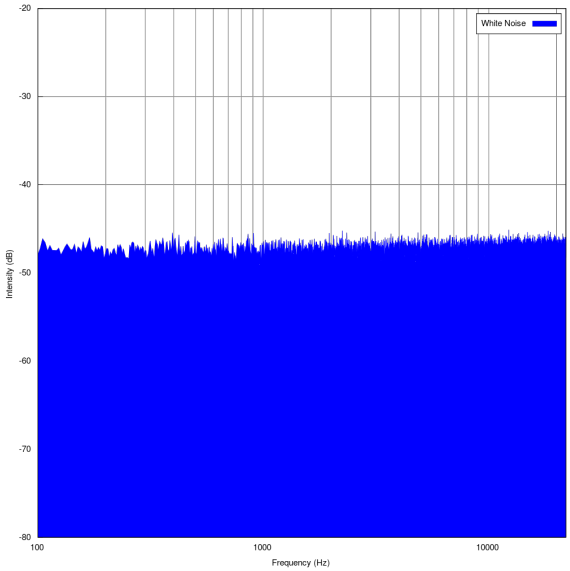

Figure 4.1: Maximum spectral flatness (approaching 1) is achieved by white noise. Source of image: [49]

The term "tonal" in this context refers to the amount of peaks or resonant structures in a power spectrum, as opposed to a flat spectrum of white noise (spectral flatness is sometimes also called tonality coefficient). A high spectral flatness (approaching 1.0 for white noise) indicates that the spectrum has a similar amount of power in all spectral bands, making it sound similar to white noise. In contrast, a low spectral flatness (approaching 0.0 for a pure

tone) indicates that spectral power is concentrated in a relatively small number of bands, resulting in a sound similar to a mixture of sine waves [7].

Mathematically, spectral flatness is calculated by dividing the geometric mean of the power spectrum by the arithmetic mean of the power spectrum:

$$\text{Flatness}[x] := \frac{\sqrt[N]{\prod_{i=0}^{N-1} x(i)}}{\frac{1}{N}\sum_{i=0}^{N-1} x(i)}$$

where $x = (x(0), ..., x(N-1))$ is a finite real-valued signal of length N and $x(i)$ represents the magnitude of bin number i. The result is often converted to a decibel scale for reporting, with a maximum of 0 dB and a minimum of $-\infty$ dB.

Spectral flatness can also be measured within a specified sub-band rather than across the whole band. It has applications in various domains, including audio processing and signal analysis. In this work, we will explore the significance of spectral flatness in the context of estimating flow rates.

4.2.3 Center Frequency/Spectral Centroid and Spectral Bandwidth

Spectral centroid (In he experiments mostly denoted by the term *center frequency*) and *spectral bandwidth* can be understood as the first two (statistical) moments of the spectrum [22].

When one defines the normalized magnitude spectrum by

$$\tilde{x}(n) = \frac{|x(n)|}{\sum_{k\in\mathcal{K}_+} |x(i)|} \tag{4.1}$$

with $x(k)$ being the discrete Fourier spectrum, k the index corresponding to a certain frequency and $\mathcal{K}_+$ the set that contains only non-negative frequency indices, than the spectral centroid is defined by

$$C_f = \sum_{k\in\mathcal{K}_+} k\tilde{x}(k) \tag{4.2}$$

and the spectral bandwidth by

$$S_f^2 = \sum_{k\in\mathcal{K}_+} (k - C_f)^2 \tilde{x}(k)\,. \tag{4.3}$$

So, in simpler terms, the spectral centroid tells you where the "center" of all the frequency energy in a signal lies. If the spectral content of the signal is concentrated towards higher frequencies, the centroid will be higher. If most of the energy is at lower frequencies, the centroid will be lower.

It's a useful feature in audio analysis because it can help characterize the tonal brightness or darkness of a sound. For example, a high spectral centroid value might indicate a sound that's brighter or more treble-heavy, like a whistle or a cymbal, whereas a lower spectral centroid could correspond to a sound that's more bass-heavy, like a kick drum or a low-pitched rumble.

For the spectral bandwidth can than be imagined like the spread of the spectrum. For example a sine wave was a small spectral bandwidth compared to the sound produces by a vibrating string or white noise signals.

4.2.4 Zero Crossing

The zero crossing rate, here denoted as $v_{ZC}(n)$, is a fundamental low-level feature extensively employed in speech and audio analysis. It measures the number of sign changes in consecutive blocks of audio samples, offering valuable insights into the noisiness and periodicity of a signal[26]. It has been used for decades in speech and audio analysis (for example for the classification of percussive sounds[14]) due to its simple calculation. The zero crossing rate is calculated as follows:

$$v_{ZC}(n) = \frac{1}{|i_e(n) - i_s(n) + 1|} \sum_{i=i_s(n)}^{i_e(n)} |\text{sign}[x(i)] - \text{sign}[x(i-1)]|$$

with the sign function being defined by

$$\text{sign}[x(i)] = \begin{cases} 1, & \text{if } x(i) > 0 \\ 0, & \text{if } x(i) = 0 \\ -1, & \text{if } x(i) < 0 \end{cases}$$

and $i_s(n)$ being the start and $i_e(n)$ the end sample for a recording.

The output of the zero crossing rate, $v_{ZC}(n)$, falls within the range $0 \leq v_{ZC}(n) \leq 1$. A higher zero crossing rate indicates a signal with more frequent sign changes, suggesting the presence of noise or high-frequency content. Additionally, variations in the zero crossing rate across blocks can imply a lack of periodicity in the signal.

The zero crossing rate serves a dual purpose: It has been employed to gauge the signal's noisiness and to estimate its fundamental frequency. In our case it is interesting, because our data is noise like, and becomes even more close to white noise the higher the flow rates are. Therefore, it might serve as a promising feature.

4.3 Machine Learning Methods

This section will introduce concepts related to machine learning. The methods we used for the hybrid and the end-to-end methods.

4.3.1 Random Forest

Random forests have emerged as a prominent ensemble learning technique due to their robustness and versatility in handling various machine learning tasks. Comprising an ensemble of decision trees, each trained on a different subset of the dataset and making predictions independently, random forests harness the power of multiple learners to generate more accurate and stable predictions.

The fundamental building block of a random forest is the decision tree, a hierarchical structure that recursively partitions the feature space based on specific criteria, such as the Gini impurity or information gain, aiming to minimize uncertainty and maximize homogeneity within the resulting subsets. The Gini split criterion, measuring the impurity of a node by evaluating the probability of incorrectly classifying a randomly chosen sample, serves as a crucial metric in guiding the tree's node splitting process.

The interpretability inherent in decision trees contributes significantly to the appeal of random forests. By inspecting the individual decision trees within the ensemble, practitioners can glean valuable insights into the decision-making process, identifying key features and pathways that influence predictions. This interpretability fosters a level of transparency and understanding, enabling users to comprehend why certain predictions are made—a feature highly desirable in practical applications and model evaluations.

One compelling reason for utilizing random forests in hybrid methods is their inherent ability to balance interpretability and complexity. While decision trees provide a clear and intuitive framework for understanding predictions, the ensemble approach of random forests allows for increased model complexity and improved predictive performance by aggregating diverse tree-based models [17].

In this thesis, we delve into the utilization of random forests within hybrid methods and more, leveraging their interpretability and predictive power to enhance the performance and explainability of machine learning models in diverse domains.

4.3.2 Overview of used ML methods

At one point of the experiments (when we optimize classical performance utilizing machine learning methods, Section 5.5.1) we try out other machine learning methods other than the

random forest. When checking against other methods one ensures that the model has nothing intrinsically unsuitable for the problem at hand.

The alternative models are the following:

- Support Vector Machine (SVM)
- Gradient Boosting
- k-Nearest Neighbors
- Logistic Regression
- Multi Later Perception (MLP)[6]

Notably, this investigation is not the main focus here and ultimately revisiting the details is out the scope for this work. However, the reader interested in the definition and detail of those methods we can recommend the book *Pattern Recognition and Machine Learning* by *Bishop, Christopher M.* [5].

4.3.3 Mixup: A Data Augmentation Technique

Mixup is a data augmentation method proposed by Hongyi Zhang et al[52]. It is designed to expand the dataset by mixing both input and output data based on a mixing ratio sampled from the Beta distribution. This technique is used to improve the generalization performance of machine learning models, as it smoothens the decision boundary.

Mixup creates augmented data by mixing original data points. The mixing ratio determines the degree to which the data is mixed. The augmented data will have labels between 0.0 and 1.0, representing the interpolation between two original data points.

Motivated by the need for better generalization performance, mixup introduces a simple data augmentation routine. It constructs virtual training examples as linear interpolations of feature vectors and associated targets. The method can be summarized as follows:

$$\tilde{x} = \beta x_i + (1-\beta)x_j, \text{ where } x_i, x_j \text{are raw input vectors}$$
$$\tilde{y} = \beta y_i + (1-\beta)y_j, \text{ where } y_i, y_j \text{are label encodings}$$

Here, (x_i, y_i) and (x_j, y_j) are data samples and corresponding labels randomly drawn from the training data, and $\beta \in [0, 1]$. Mixup extends the training distribution by incorporating

[6] In this text we sometimes just call it neural network.

the knowledge that linear interpolations of feature vectors should lead to linear interpolations of the associated targets. It can be easily implemented with minimal computational overhead, yet is highly effective.[7]

4.4 Evaluation Measure

4.4.1 Accuracy

As a basic performance measure, the accuracy was defined in the following way

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} \mathbf{1}(\hat{y}_i = y_i) \tag{4.4}$$

with y being the prediction of the network on the entire dataset, $\hat{y}$ the according ground truth, n_{sampes} the number of samples and the identity function $\mathbf{1}(\cdot)$ results to 1 when the argument is true and 0 if false.

This measure was chosen, because it is easily implemented, interpreted and easy to debug. Note that this performance measure does not take precision and recall into account, i.e. depending on the balance of true and false samples in data learning a constant function might achieve a high accuracy. This issue was addressed by checking it manually afterwards.

4.4.2 Weighted F1 Score

In the realm of machine learning and classification tasks, the evaluation of a model's performance is a fundamental aspect of the analysis. Accuracy, the ratio of correctly predicted instances to the total instances, serves as a commonly employed metric. However, in the presence of class imbalance, where one class significantly outnumbers the other, accuracy alone may not provide an accurate representation of the model's effectiveness.

To address this limitation, the Weighted F1 Score emerges as a more nuanced performance metric. The Weighted F1 Score is an extension of the traditional F1 Score, which itself is the harmonic mean of precision and recall. Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive, while recall quantifies the fraction of correctly predicted positive instances out of all actual positive instances.

[7] Despite its simplicity, mixup has achieved state-of-the-art performance in various image classification datasets. The source code to replicate the CIFAR-10 experiments using mixup is available at `https://github.com/facebookresearch/mixup-cifar10`.

The F1 Score combines both precision and recall into a single metric, offering a balanced assessment of a model's ability to correctly classify positive instances [46]. However, in real-world scenarios where class imbalances are prevalent, it becomes crucial to consider not just the overall F1 Score but how it accounts for the imbalance between classes.

This is where the Weighted F1 Score comes into play. To ensure a more comprehensive evaluation, it takes into account the class distribution by assigning different weights to different classes. In situations where the minority class holds greater importance, the Weighted F1 Score assigns higher weight to that class, thereby emphasizing its correct classification [29].

The formula for the Weighted F1 Score $F1_w$ for an n-class problem is expressed as follows:

$$F1_w = \frac{\sum_{i=1}^{n}(w_i \cdot F1_i)}{\sum_{i}^{n} w_i} \tag{4.5}$$

Where:

$F1_i$ is the F1 Score for the ith class.
w_i is the weight/support assigned to the ith class.

By employing the Weighted F1 Score, we can tailor our model evaluation to consider the practical implications of class imbalance. It provides a balanced and weighted assessment that gives prominence to the minority class. In the context of our experiment, where imbalanced class distributions are prevalent, the Weighted F1 Score will yield insights that better align with actual applications, because it is more robust.

4.4.3 Confusion Matrix

The Confusion Matrix is a fundamental evaluation measure used in machine learning and classification tasks. It provides a tabular representation that summarizes the performance of a classification algorithm by displaying the predicted and actual classes of a dataset.

The matrix is structured into four compartments: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The rows represent the actual classes, while the columns signify the predicted classes.

Predicted Positive (TP)	False Negative (FN)
False Positive (FP)	Predicted Negative (TN)

The main diagonal of the matrix represents correct predictions (TP for positive class and TN for negative class), while off-diagonal elements indicate errors. TP and TN reflect correctly classified instances, whereas FP signifies false alarms (Type I error), and FN represents missed detections (Type II error).

The Confusion Matrix aids in computing various performance metrics such as accuracy, precision, recall, F1-score, and specificity. These metrics offer insights into the classifier's effectiveness, highlighting its strengths and weaknesses.

The Confusion Matrix enables a comprehensive assessment of classification model performance, assisting in making informed decisions regarding model optimization and improvement [45].

Note that this concept trivially extents to classification tasks of more than two classes. This gives more insights than just checking if the prediction failed or not, but also makes clear what class was actually chosen by the classifier.

4.5 Visualisation Technique

4.5.1 Kernel Density Estimation

Kernel Density Estimation (KDE) is a fundamental statistical technique used to estimate the probability density function of a continuous random variable based on a set of observed data points. Informally, KDE can be understood as a continuous, smooth version of histograms. Unlike traditional histograms, which provide a discrete representation of data, KDE offers a continuous and smooth estimate of the underlying data distribution.

KDE achieves this by placing kernel functions, typically Gaussian functions, at each data point and then summing them to create a smooth estimate of the probability density[8]. This process results in a curve that represents the likelihood of observing a data point at any given value along the variable's range.

KDE is particularly useful in situations where you have a set of data points and wish to gain insights into the underlying probability distribution. It is employed in various fields, including statistics, data analysis, and machine learning, for tasks such as data visualization and density estimation.

In this work it was primarily used to gain quick insights into the distribution of the data by studying its visualisation. An example can be seen in Figure 6.1

[8]For a rigorous definition, here is a nice source: [36]

4.5.2 Scatter Plot Matrix

A Scatter Plot Matrix (SPM) is a grid of scatter plots that visually displays relationships between pairs of variables in a dataset. It serves as a comprehensive tool for exploring multivariate data patterns and correlations.

The SPM comprises a symmetric grid of scatter plots, showcasing relationships between variables. Its symmetry allows for examining only the upper or lower triangular part of the matrix to understand all relationships. Along the diagonal, Kernel Density Estimation (KDE) plots illustrate the univariate distribution of each variable, aiding in understanding individual variable characteristics.

The scatter plots in the matrix reveal patterns such as linear relationships, clusters, outliers, and correlations between variables. Analysis involves observing trends, concentrations of points, or dispersion across plots, offering insights into the data's interrelationships.

To create an SPM, a dataset with multiple variables is required. Each variable represents a dimension, and the SPM visualizes relationships between these dimensions through scatter plots [34]. An example can be seen in Figure 9.2

5 Experiments

This section will apply the features and methods for analysing audio signals previously discussed in the theory section on the the GW data. As already mentioned in the theory section, working with audio always means working with large amounts of high dimensional data. In this work we deal with this fact in two ways.

I. Reduce the dimension as much as possible, by using human understandable low dimensional representations of the data, while still capturing as much information as possible

II. Make use of data driven machine learning methods, i.e. let a regression algorithm 'learn' a low dimensional representation of the data. Here we don't spend much time on monitoring the features, but focus on the networks capability of producing "good" predictions (supervised learning).

Note that the second approach requires much more data than the first one. We note that for the task addressed in this thesis the process of data aquisition is quite complex and dependent on external factors such as the weather and access to specialized recording equipment and sites. Due to these difficulties the amount of data available for ML methods was restricted by those external conditions. However, it is still interesting to see the performance of such methods on the given data. The goal is to find out whether the achieved accuracy will be sufficient to solve the task of predicting the water flow acoustically. Nonetheless, the conducted experiments reveal further problems one might tackle, namely choosing different decision boundaries than the ones requested by GW and analyzing the impact of changing acoustic conditions (domain shift). So in the end of this section there will be experiments deviating from the just defined goal.

The order of the presented experiments follows a structure that starts with the above proposed (**I**) method of manually designing processing pipelines end ends with the mentioned (**II**) method of applying machine learning. In between those two extremes will be several combinations of classical and machine learning methods. Obviously, this is not the most precise way to cluster analytical methods, since there are many ways to combine machine learning with manual processing, especially when it comes to combining them. For the sake of giving the reader some orientation, the experiments will be ordered by grouping them into the following three categories.

1. Classical Methods (human understandable representation and reasoning)
2. Hybrid Methods (mix between 1. and 3.)
3. End-to-End Methods (applying Machine Learning)

The following subsection will motivate the decision of combining classical and end-to-end methods. After that follows the presentation of the experiments.

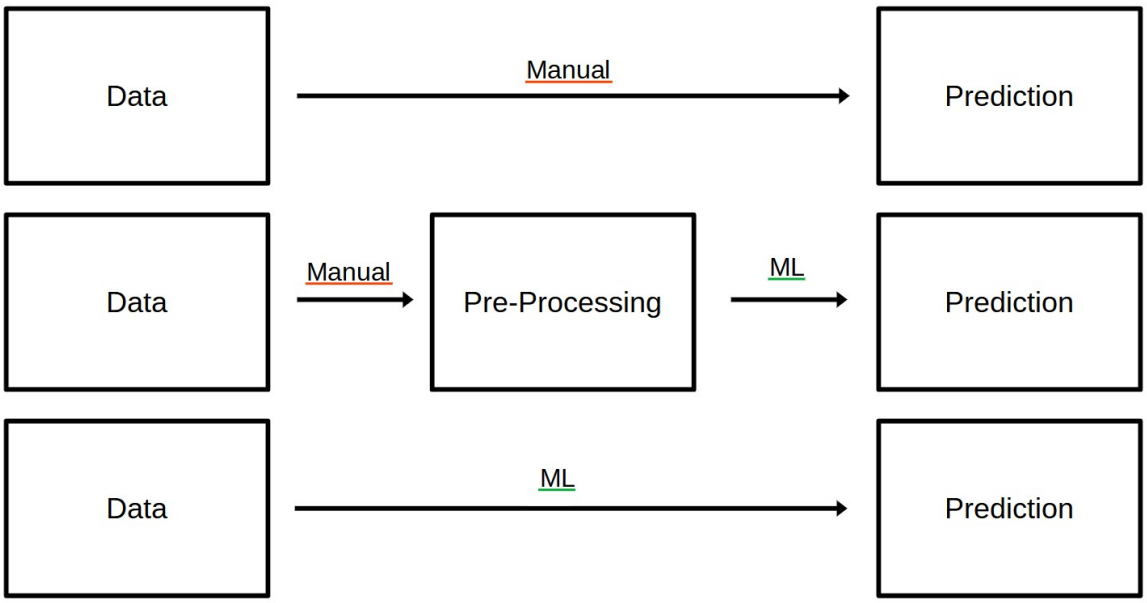

Figure 5.1: Schematic on how we the presentation of experiments is structured. 'Manual' refers to applying classical methods, and 'ML' refers to machine learning methods.

5.1 Leveraging Domain Knowledge Before Transitioning to Machine Learning

The data science and machine learning communities often go hand in hand. On one hand, the power of advanced machine learning algorithms and neural networks promises automated solutions with the potential to find patterns within data. On the other hand, the importance of domain expertise and interpretability cannot be overstated, especially when dealing with datasets of limited size.

The experimental section starts with manual decision-making and uses increasingly more machine learning techniques, with the goal of utilizing the full potential of domain knowledge before embracing the "black-box" nature of fully automated learning algorithms.

The strategy employed here uses an initial phase of manual feature engineering, threshold-based decision rules, and an in depth exploration of the data landscape. We prioritize human intuition and domain understanding, allowing these elements to guide our decision-making process. This initial manual phase sets the stage for interpretability, traceability, and transparency in our analytical journey.

This approach has many advantages. First, it enables us to fully leverage the domain knowledge at our disposal, making informed decisions that are rooted in an understanding of the problem's details. Second, the outcomes of manual methods tend to be more inter-

pretable. Third, it promotes thorough data exploration, helping with the identification of hidden patterns and relationships within the dataset.

Furthermore, the initial manual phase provides a baseline for performance and an opportunity to gauge the effectiveness of subsequent machine learning techniques. This "human-first" approach is often more efficient in terms of time and computational resources, especially when dealing with limited data.

However, this method is not without its limitations. Manual methods can introduce biases stemming from human decisions and may struggle to capture complex, high-dimensional patterns in the data. Recognizing these limitations, our approach involves a gradual transition to machine learning methods. This transition seeks to strike a balance between the strengths of manual methods and the power of automated algorithms. We will not train it 'purely' end-to-end (.wav file to flow rate), but still decide for a representation the network will be provided. The data will remain high dimensional though.

The hope is that during the transition from the first to the second phase (only preprocessing the data without proposing a model for the predictions) we learn something and can already see what features/investigations are promising. A good rule of thumb is "if one can already imagine the data to be visually separated, that is a good sign that a suitable ML method will exist - even if it might be difficult to estimate."

Another reason why to start with classical methods is that from the beginning it was not clear if enough data was available to train end-to-end methods. So starting incorporating as much domain knowledge as possible reduces the degrees of freedom the network has to deal with, which allows for having only few data samples available.

5.2 Software and Environment

The experiments were done in python jupyter notebooks[23] in a virtual coding environment. Mainly the packages numpy[15], pandas[47], Scipy[32] were used for processing the data, such as computing Fourier transforms and spectra. Machine learning was done with sklearn/scikit-learn[38], deep learning with tensorflow[2] and visualizations were made with matplotlib[18] and seaborn[50].

5.3 Omission of Anomaly Detection

In the context of the experiments conducted in this study, it was decided not to employ anomaly detection as a preprocessing step, despite the availability of advanced techniques that have been developed (as referenced in the audio lab done by us [19]). This decision was made comfortably due to the following two facts:

To our understanding the short-term events resulting from other sources, such as vehicular activities, were not expected to significantly compromise the overall performance.

Additionally, it is noteworthy that one facet of the initial anomaly detection framework involved the filtering of signals showing strong similarity to white noise. However, subsequent analysis done in this thesis revealed that signals exhibiting high spectral flatness, which were initially treated as anomalies, held intrinsic value as features for the extraction of extreme flow patterns within the dataset. This emergent insight was not foreseen or considered during the original investigations into anomaly detection methods.

5.4 Classical Methods

Here the focus will be on investigating the data manually, by computing well known features from the signal processing library *librosa*[31].

The question of interest is:

Are there already some visual indications for robust features that cluster the data well in terms of correlation with flow rate?

Robust in this case means that slight variations of the decision boundaries does not degrade the performance.

As already mentioned, the raw .wav file as well as the STFT of that recording constitude high dimensional data. For a raw 10 second recording with our sampling rate $48\,kHz$ this amounts to $480,000$ floating points numbers (floats). For the spectrogram (implemented by the librosa library[9] [28]) it is roughly $120,000$ floats, so still too much to be human manageable.

Due to this high dimensionality, we will utilize the features presented in the theory section to reduce the dimension.

5.4.1 Power Spectral Density (PSD)

The Power Spectral Density (PSD) is a fundamental feature that serves as the basis of our analysis. In this section, we perform an in-depth exploration of the PSD.

We initiate our examination by roughly scanning the PSD for all recorded events, see Figure 5.2 for event 1, 2 and 3 and Figure 5.3 for event 4 and 5. This is the first deeper look at the data. Notably, the observed PSD exhibits a polygonal structure, which may initially suggest the use of an extensive window size in its creation. However, this piecemeal linear

[9] If not specified the default values are used: window length = nfft = 4 * hop size = 2048, hann window, centered frames and zero padding

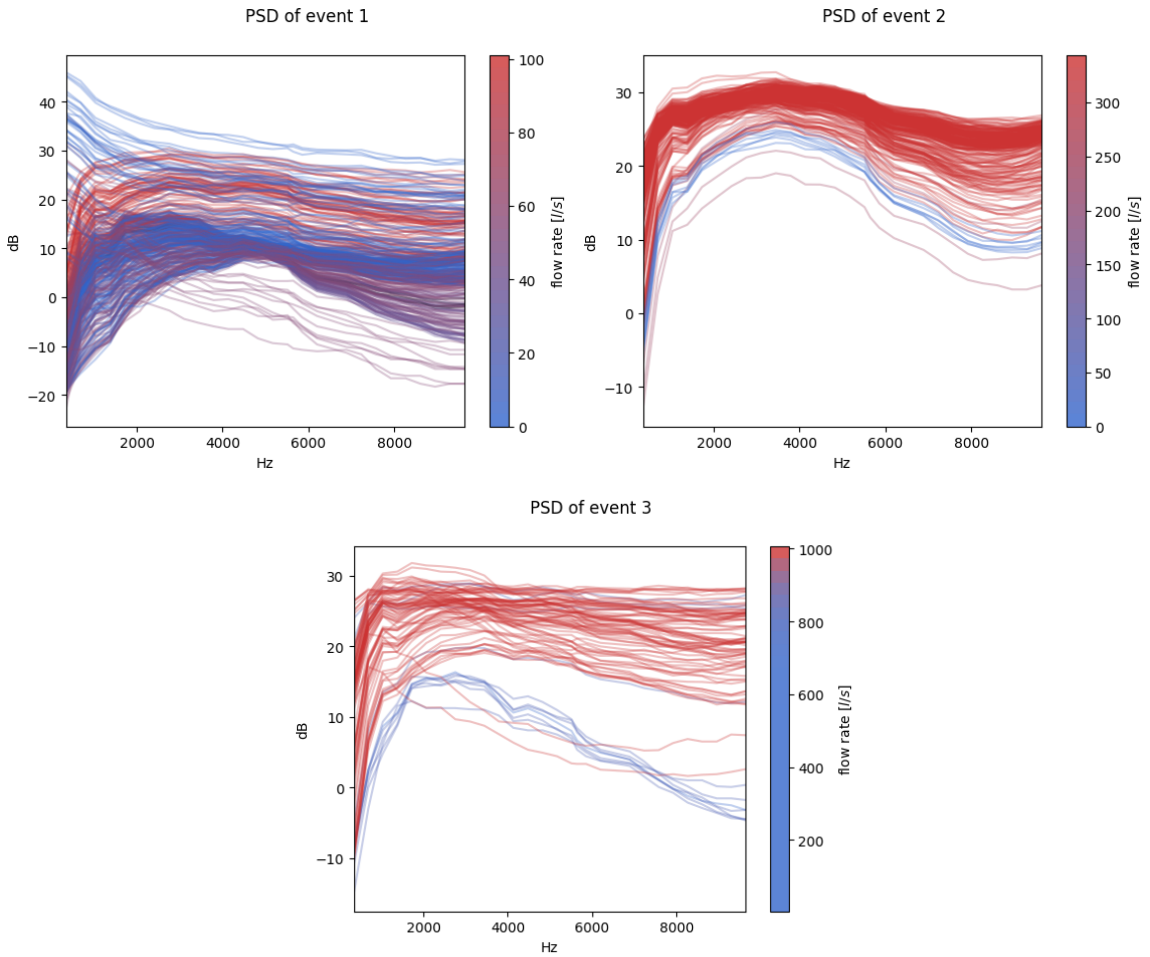

Figure 5.2: PSD of events 1, 2 and 3 with (relative) hue encoding the flow rate according to the colormap shown on the right.

structure - i.e. one can see it is discrete - is the result of pre-processing. Initially, the spectrum is computed in accordance with the sampling rate, and subsequently, down-sampling of the PSD is carried out with a precision of 344 Hz. The fact that the recordings mostly sound like noise, allow for such a rough down-sampling. In applications where greater details are required (for e.g. speech detection or speaker separation) this would, however, discard a lot of useful information.

Here however the focus lies on manual experiments of the data, for that a low dimensional representation is desired.

Upon closer examination of the PSD, several observations can be made. The following list describes/summarizes the most interesting:

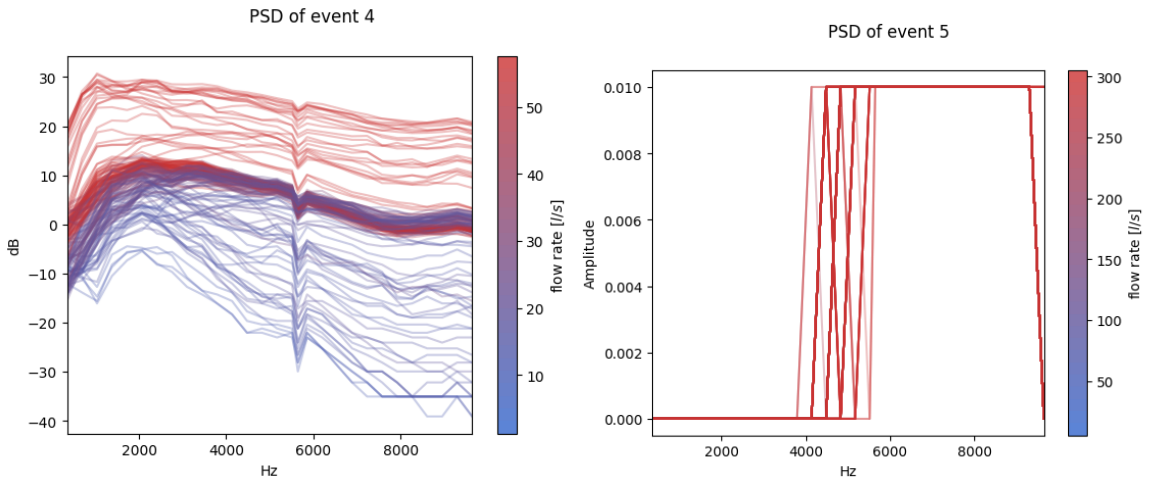

Figure 5.3: PSD of events 4 and 5 with (relative) hue encoding the flow rate according to the colormap shown on the right.

- Event 1 displays curves assigned to low flow rates with an unconventional trajectory, raising questions about its distinctive characteristics.
- Event 1 (relative) high and low flow is not as nicely separated as the other ones (except for event 5). This might be due to the low flow present in all the samples in the first event. In that regard it makes sense that they all clutter in the same region, because one can consider all of them to have low flow rates (compared to the other events and considering the domain experts assessment that everything below $200\,l/s$ can be consideres as "nothing too special")
- In events 2, 3, and 4, recordings with lower flow predominantly reach smaller values across all frequency components.
- Events 1, 2, 3, and 4, characterized by the highest flow rates, consistently peak at approximately $30\,dB$ in the PSD.
- The last event, in contrast to the others, shows anomalous characteristics. It was checked that the problem lies indeed in the data and not the computation of the PSD, see below for the explanation why that is.
- Event 3, distinguished by the highest flow rates, notably presents the flattest spectrum among all events. This observation suggests that spectral flatness may be a promising candidate as a valuable feature for discriminating the highest flow events.

Next a deeper look is taken at event 1 and event 2.

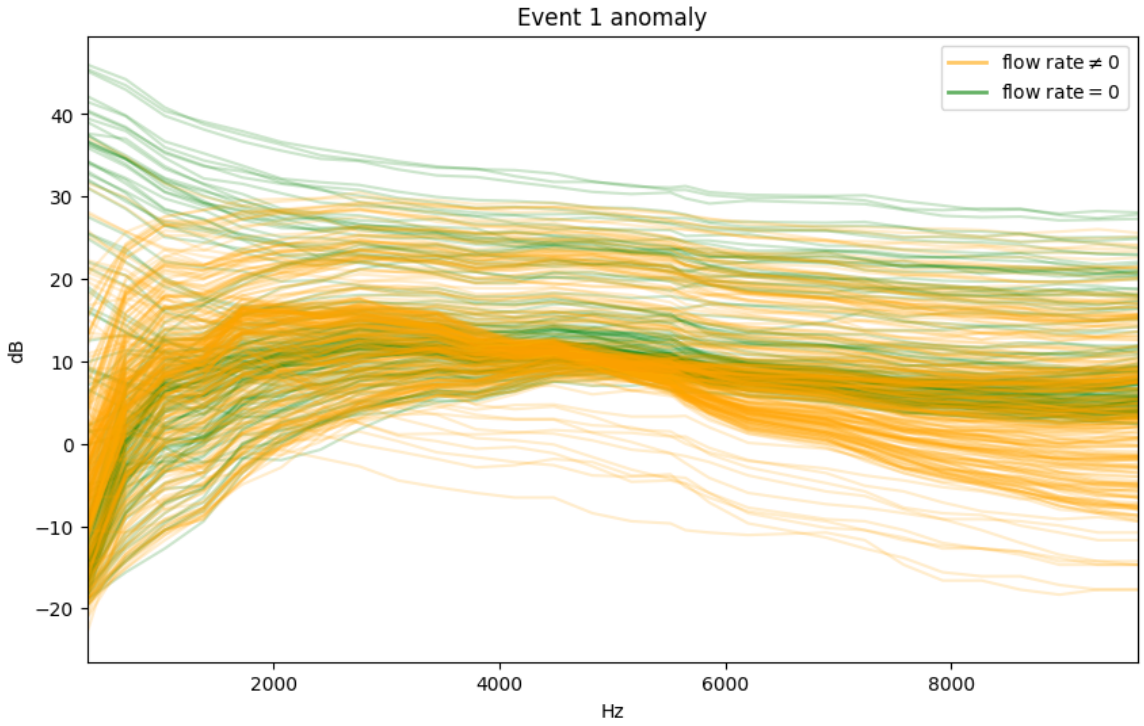

Figure 5.4: The PSD of event 1. It shows the anomaly that most of the 'strange' samples are assigned to a flow rate equal to zero.

Deeper investigation of the first event Let us delve deeper into the characteristics of the curves observed in the first event.

Upon closer examination, it becomes apparent that these curves exhibit a distinct pattern similar to a scaled version of the mathematical function $\exp(-x)$. This observation suggests the presence of a specific type of noise affecting the data. Further investigation into this matter reveals a notable observation:

When checking the labels of those events, one notices that the majority of these events are labeled with a flow rate of 0, a condition that, while possible, seems unlikely. One can see this in Figure 5.4.

This finding is particularly perplexing given that the Nivus flow meter, our gold standard tool for flow rate measurement, recorded these readings. Therefore, it raises questions about the reliability of our prototype's ability to accurately validate the ground truth measurements provided by the Nivus flow meter.

This situation presents us with two plausible scenarios: Either the sewer pipe was indeed empty, and the Nivus flow meter's measurements are accurate, or an external source of

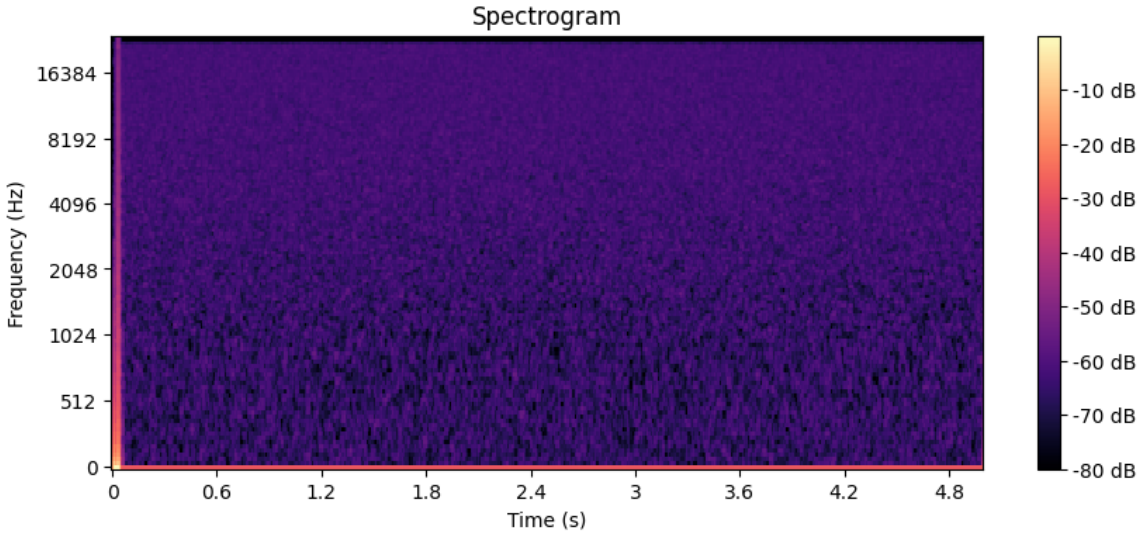

Figure 5.5: The Spectrum of event 1. Here one can clearly see a loud (relative to the rest of the recording) click at the beginning of the recording

noise affected both sensors, leading to the malfunction of both instruments.

A possible reason for this behaviour would be the automatic gain control. Lets assume that the Nivus flow meter is correctly measuring zero flow, so there actually was no flow. That would mean that the sewer pipe is silent, which would cause the auto-gain to increase the gain a lot. This might normalize, and in this case hence increase, the amplitude to a level that reaches the magnitude of the others. The difference now however is that frequencies are emphasised which do not contain any useful information, particularly lower frequencies.

One way of coping with this would be to not only record the audio, but also to capture the auto-gain currently applied during measurement. This was not possible to do a posteriori because the gain data was not available, however this might be a possible improvement step for future data acquisition.

Deeper investigation of the last event Concerning the last event, the reason for the problem was found. When listening to the recording and/or viewing the spectrum one notices that every single one starts with a loud click (difference between peak volume and rest is roughly 40 dB) at the beginning of the recording. This can be easily verified from Figure 5.5.

Note also that for this event the recordings are shorter (5 seconds instead of 10). This does not change anything in the averaging process discussed previously.

One could account for such cases in the data, by comming up with some measure that detects the shape of the spectrum and if some peak is present, just cut that region out. Another way of dealing with this would be to figure out what caused this shape in the first place. However, at that point we renounce to dig deeper into the reason of how that happened, since we do not have access to the experimental setup.

Notably, we did not perform any cleaning for this data, because we also noticed the absence of any labels.

Further investigations resulted in discovering a failure in the recording hardware of unknown reason [11]. This results in the measurement data of event 5 being unusable. Event one to three was not mentioned, because there were available way earlier as it was discussed in Section 2.1.

Choosing decision boundaries for classification Coming back to the leading question of predicting the flow rate: All of the so far made investigations (including the literature review) suggest that the accuracy will be much less detailed than the Nivus Flowmeter delivers. This motivates to use a classifier as the prediction algorithm, because that way we already impose some structure that can make the training easier, since the classes are formed by categorizing the flow rates into broad categories.

This directly brings up the question of *where the decision boundaries should be*. There are some ways to tackle this question, depending on the application in mind. For now (it will change later) this boils down to the question:

What should be considered as high/low flow rates?

So far, all the flow rates of the recordings were depicted relative to the event itself, i.e. for the first one 100 l/s was the highest flow. This however is for example an order of magnitude lower than the highest flow of the third event.

One way of determining what should be considered as high or low flow rate is to normalize it based on the available data, resulting in:

- Minimum flow = $0\,l/s$
- Maximum flow = $1000\,l/s$

When we evenly divide this range, we obtain:

- $0 - 333\,l/s$: Low flow
- $333 - 666\,l/s$: Mid flow
- $666 - 1000\,l/s$: High flow

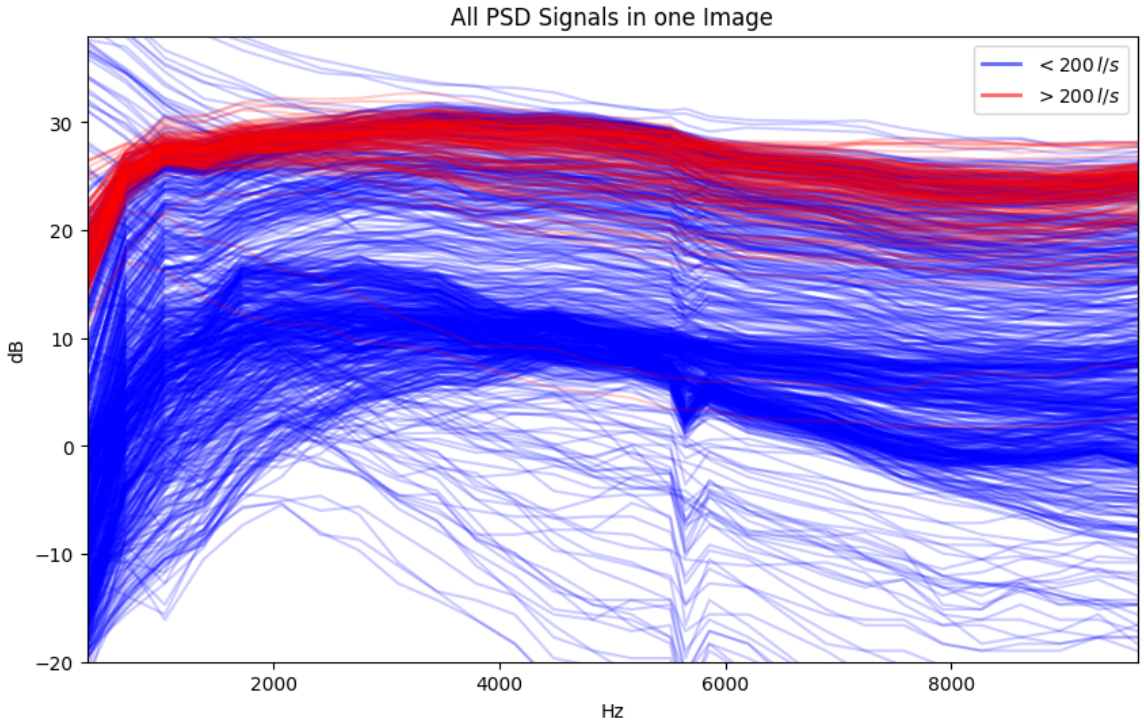

Figure 5.6: All PSD Signals in one image. This should give first insights about the chance of differentiating between above/below $200\,l/s$ using the PSD alone.

On the first glance, this approach is not significantly different from what the domain experts proposes as decision boundaries:

- $0 - 200\,l/s$: Low flow / "nothing too special"
- $200 - 800\,l/s$: Increased flow / likely to be a rain event
- $800 - 1000\,l/s$: High(est)/critical flow / heavy rain

For the further investigations we will use $\{0, 200, 800, 1000\}\,l/s$ as decision boundaries.

Using $200\,l/s$ as decision boundary Until now we only looked at features without any effort to predict something. This will change now.

The next experiments look particularly into differentiating between flow rates above and below $200\,l/s$. When looking at the PSD of all events combined while highlighting flows above and below $200\,l/s$ (see Figure 5.6) one makes the following observations.

- There is still not a clear separation between flow above or below $200\,l/s$, but almost all of the high flow rates lie in the upper range of all measured values (between 20 and $30\,dB$).

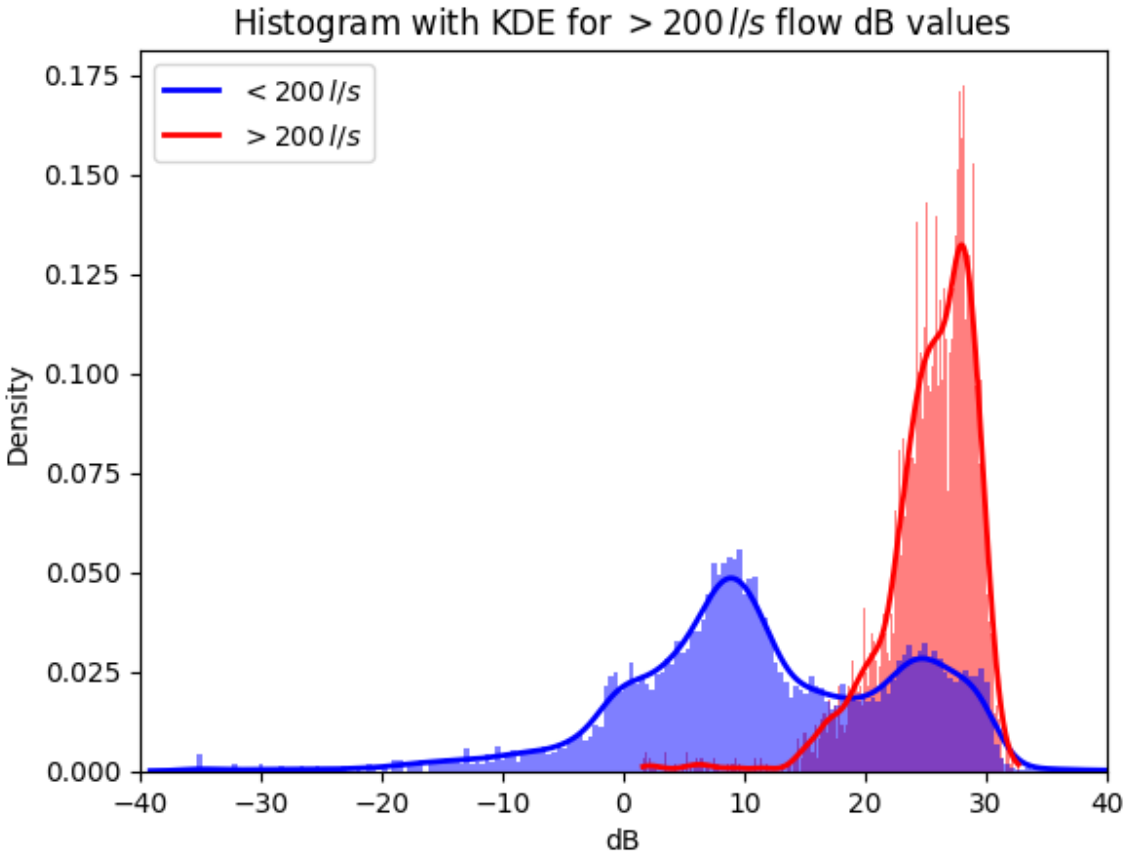

Figure 5.7: Here all the frequencies are flattened into one single array, this means that for $x = 20$ dB there are all the frequencies shown that occupy this value. The y-axis (denoted as 'Density') is a percentage share.

- None of the red curves have the $\exp(-x)$ shape of event one. This makes sense, because there are no flows higher than $100\,l/s$ present in event 1 and this were the only ones showing this anomaly.
- Even though the highest flows from event 3 were the ones showing the highest flatness, they don't seem to appear on top of the PSDs assigned to flows above $200\,l/s$.
- The representation chosen in Figure 5.6 obfuscates the number of blue curves lying 'directly behind' the red ones. Therefore another representation was chosen (see Figure 5.7.)

Figure 5.7 shows all the frequencies flattened into one single array. This confirms that most of the higher flows are above ~ 20 dB, so we identify the PSD as a promising feature for differentiating above/below 200 l/s. However, from this KDE it is not entirely clear how many false positives a simple separation by amplitude (in dB) will cause and if the resulting

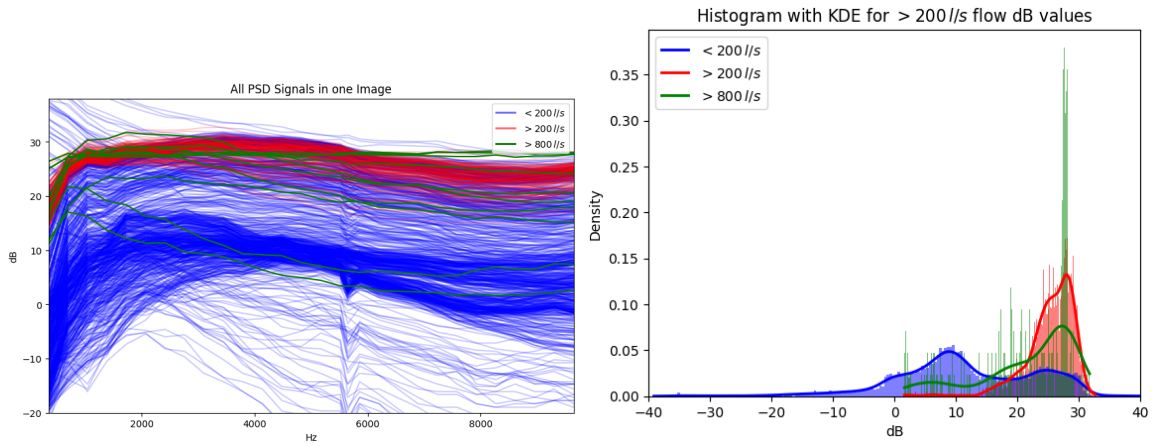

Figure 5.8: Left: All PSD Signals in one image. Right: All frequencies are summed into one single array and depicted with KDE.

model will be useful in practice. This will be checked quantitatively in the next subsection

Before discussing this, we will quickly talk about flows above 800 l/s.

Using 800 l/s as decision boundary The same plots visualising the third decision boundary proposed by the domain experts can be seen in Figure 5.8. Note that we have a total count of 10 recordings (each consisting of 10 seconds) for flows above 800 l/s.

Even though it looks more clear for the separation using 200 l/s as the decision threshold, when it comes to the highest flows measured, there is no separation possible with the use of PSDs. Therefore for the rest of the PSD investigations we will not try to predict flows above 800 l/s into a separate class.

5.4.2 First classifier using PSD

In the last section we identified that when summing the power of all frequencies into one container one sees a tendency regarding where the higher flows are. Next we try to separate the amplitude/energy contained in a frequency range, because when observing the PSD that divides between above/below 200 l/s (red/blue Figure 5.6) it seems reasonable to pick a frequency that separates the data well. Detailed histograms of all analyzed frequencies are given in the Appendix, Fig. 9.1.

From that, amongst others, *psd1378* (i.e. 1378 Hz) is a promising feature for separating the two classes, when comparing the KDEs of all frequencies with each other. When comparing aimed for a cluster of flows above 200 l/s (red) that has as few lower flows (blue)

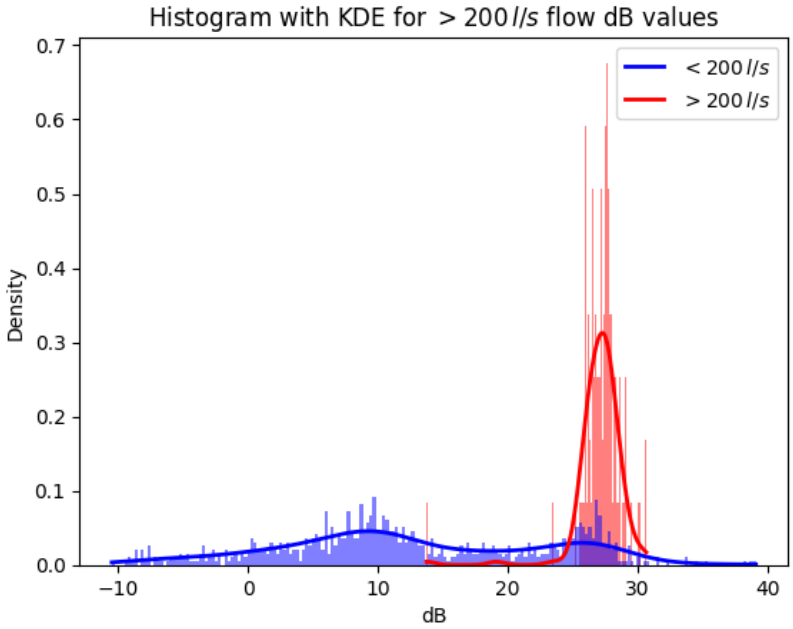

Figure 5.9: KDE for *psd1378*.

containted in it. For a plot of only *psd1378* see Figure 5.9.

Note though, this circumstance might change completely, when more data is considered, because we only have limited data available, we suppose that the resulting decision criterion might change when more data is available at a later point. We are dealing with a small sample size here and possible later changes in the hardware setup will have a huge impact on such features.

Testing the first classifier It is good practice to check the distribution of labels before training a classifier. This will prevent just learning the trivial classifier (always predicting the output to be the same no matter what the input is.) Furthermore, it gives an idea of how well the trivial classifier performs. This will set the baseline for an accuracy that should be improved by more sophisticated classifiers.

A histogram of the labels is shown in Figure 5.10. When counting the incidences one obtains that 85 % of the recordings are bellow $200\,l/s$, this means that our model has to perform better than that to be useful.

In the following we will test how well the data is actually separated by a decision rule based on the amplidute/energy in dB in the frequency bin corresponding to 1378 Hz.

When setting the separation boundary to 25 dB an accuracy of 0.84 is reached (F1 score

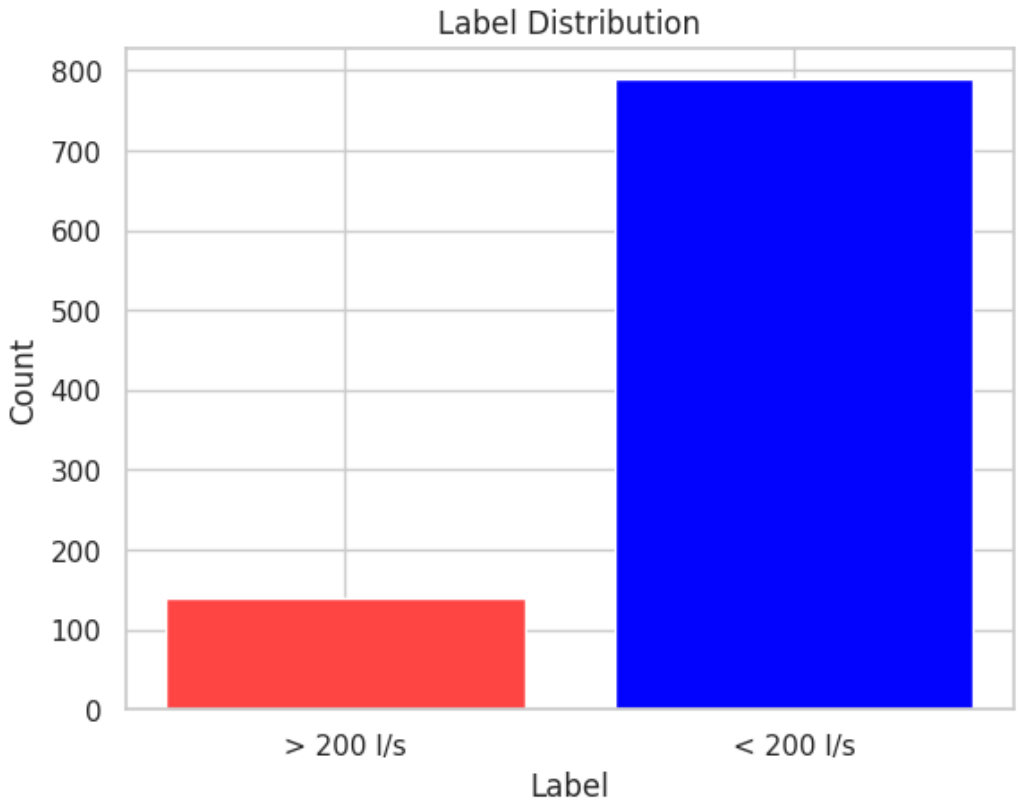

Figure 5.10: This histogram shows the occurrence of the according labels above/below 200 l/s.

and confusion matrix can be found in table 5.1). This is actually a bit worse than the trivial classifier. The amount of wrongly and correctly predicted classes for the case above 200 l/s is roughly the same (see the confusion matrix). This model has a relatively high recall, indicating that it is good at catching positive instances, but the precision is lower, meaning that there are some false positives. Depending on the specific context and goals of your model, one might need to adjust the trade-off between precision and recall to meet the objective.

Metric	**Value**
Accuracy	0.84
F1 Score	0.55

Confusion Matrix	Predicted <	Predicted >
Actually <	644	145
Actually >	4	136

Table 5.1: Classification Metrics for the 25 dB threshold.

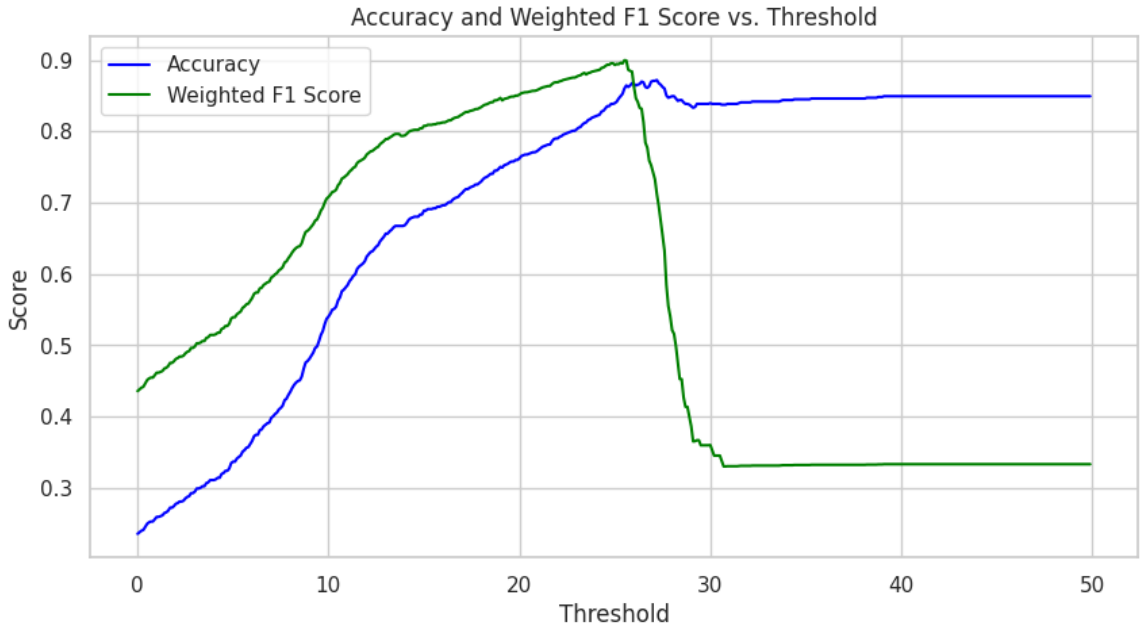

Figure 5.11: This shows the result of the systematic search for the optimal threshold that separates the two classes best.

To make sure that this failure in classification is not due to our ad hoc choice of the boundary an automatic search was performed by scanning through all the dB values and computing its accuracy and f1 score. The results of this experiment are shown in Figure 5.11.

Note that the weighted f1 score was used here for training, although it did not make a huge difference here. From the systematic search of the optimal threshold the result is 27.2 dB. The confusion matrix can be seen in table 5.2. There the confusion matrix is much more balanced.

Already here there is an application dependent choice to make: All classifiers reach approximately the same accuracy. Some however, produce much more false negatives than others. For the application of flood prevention for example, the false positives might be more tolerable than false negatives. Using one method, the alarm would be produced more frequently without there being an actual problem. However, based on another classifier would miss to warn much more often. In such an application the first behaviour is probably more desirable. For a different application this might however look different. Notably, such unreliable systems should not be used for critical situations such as flood prevention at this stage of development as a too high amount of false positives will degrade the trustworthiness of the model.

In our quest for parameter optimization, we devoted substantial effort to meticulously

Prediction for Threshold	27.2
Custom Model	
Accuracy	0.87
F1 Score	0.65

Confusion Matrix	Predicted True	Predicted False
Actually True	738	51
Actually False	68	72

Table 5.2: Classification Metrics for the 27.2 dB threshold.

fine-tuning this particular parameter. We explored its full potential and subjected its performance as much as possible.

We summarize several noteworthy insights:

Limitations of PSD as a Predictive Feature: It becomes evident that the Power Spectral Density (PSD) may not be the most effective feature for predicting flow rates exceeding $800\,l/s$ within the context of this dataset. While PSD remains a valuable tool, its suitability for accurately predicting high-flow scenarios in this specific dataset is called into question.

Trade-offs in False Positives: The choice of threshold and its implications may vary depending on the specific application. Practitioners may need to decide whether they are willing to tolerate more or fewer false positives. In practice, distinguishing between flow rates above and below 200 l/s can be achieved with a level of accuracy of up to 87 %. This Figure represents a relative improvement of 13 % when compared to previous results, equivalent to approximately 2 out of 15 instances.

Interpreting Accuracy: It is important to note that the accuracy achieved must be understood in the context of training data. Since all available data was employed for model training, the reported accuracy represents how well the model performs on the data from which it was trained. For the classical methods, there was no perceived necessity to reserve a separate set of random data for evaluation.

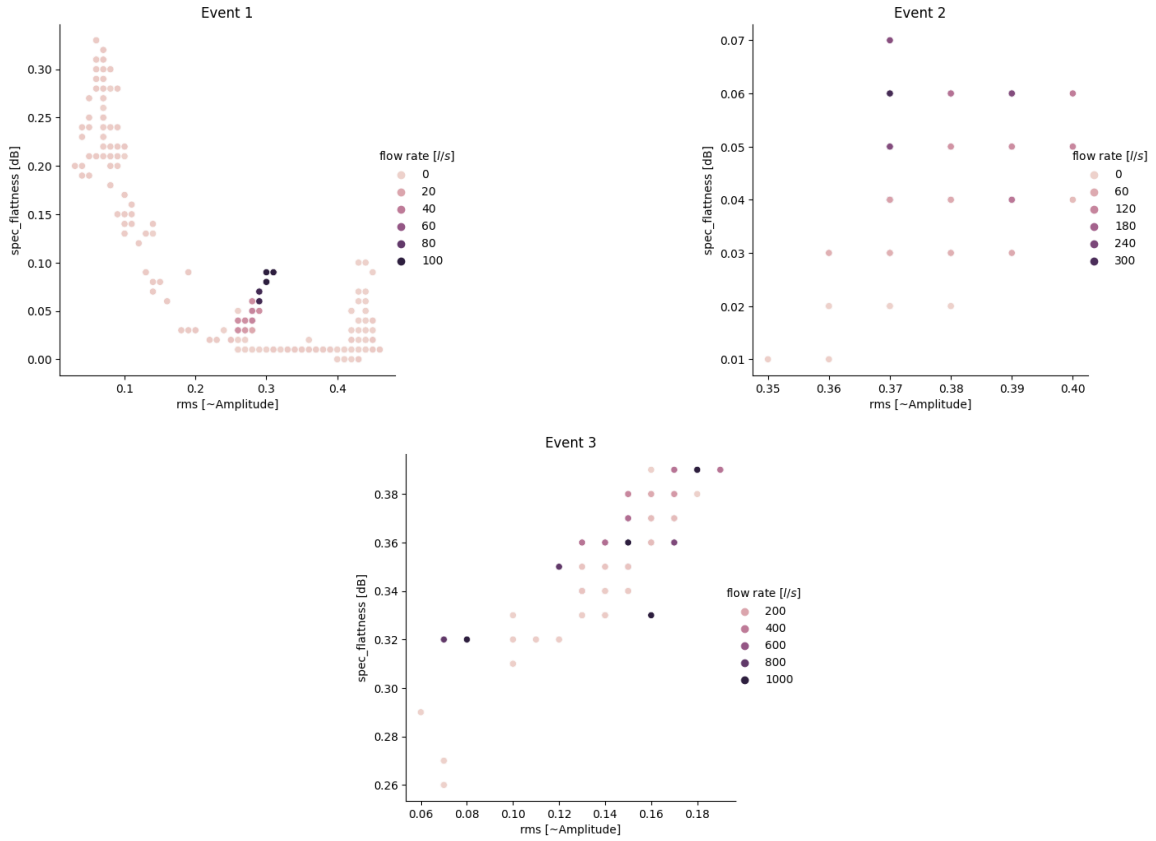

Figure 5.12: rms against spectral flatness for event 1, 2 and 3 with (relative) hue for flow rate in l/s.

5.4.3 Spectral Features

With the insights of PSD in mind, this section will delve into the exploration of alternative spectral features and their potential implications for our analysis. For a first intuition we plot all features against each other in the above introduced scatter plot matrix. The letter q in the following section (especially in the images) represents the flow rate in units of l/s.

The SPM of Event 1 can be seen in Figure 9.2. That way one can manually scan for suitable candidates for the flow rate prediction. In the SPM each point represents a 10 seconds audio recording. On the first view one can already notice that some features seem to separate the data better than others. Also none of the KDEs separate the data well, which means that none of the investigated features solves the problem right away, but the combination of two reaches good results.

The most promising feature here seems: rms against spectral flatness. The plots for event 1 to 3 are shown in Figure 5.12. For event 1 it is easy to see where the data clutters, however for 2 and 3 that is not necessarily the case. Also it is interesting, that for 2 and 3 the

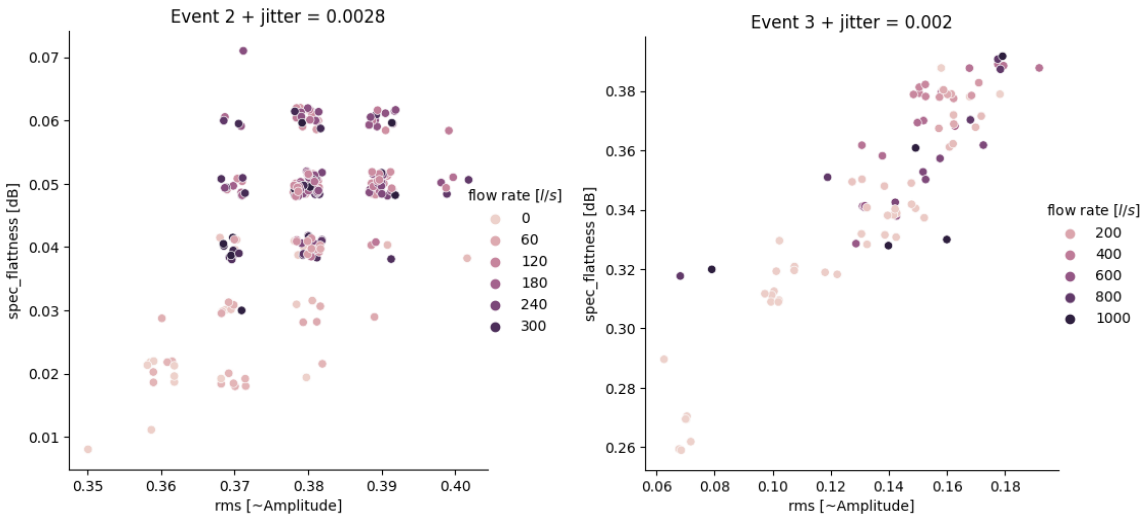

Figure 5.13: Same features as Figure 5.12 for event 2 and 3, but with an additionally added jitter vector to omit occlusion.

data representation seems discrete. Note that this is already the case for event 1, but there the data is more separated. This hits at higher sound differences for event 1 than for 2 and 3.

It is also notable that despite the values being in a different absolute range, there is no clear cluster observable, for them. Another things that comes to mind is that it seems to be less points for event 2 and 3 than for 1 even though they have a comparable amount of recordings. This combined with the discretization suggests that the chosen image probably puts many points into the same category, therefore one cannot see how many of them got the same value. To balance out this effect, we added random jitter to the points, to make them visible. This can be seen in Figure 5.13

Now this shows that for event 2 the higher flow rates tend to be in the upper left corner and for event 3 there is no useful correlation. So one has to check the features of one event relative to the other events. This can be done in two ways: Either adding all data points into one collective dataset or checking them next to each other. First we investigate the second way. The results can be found in Figures 9.3 and 9.4. The most promising per event separation for the events 2 and 3 was spectral bandwidth against center frequency, see Figure 5.14.

For a final comparison of the features, we take a look at all features in one plot. Note that here again for the spectral flatness against RMS a random jutter vector was added. See the results in Figure 5.15. From that images one can observe, that using these features, there is no clear cluster suitable for identifying where certain flows are located. Especially the highest flows are spread widely.

The consequence of this (concerning only the use of classical methods) is:

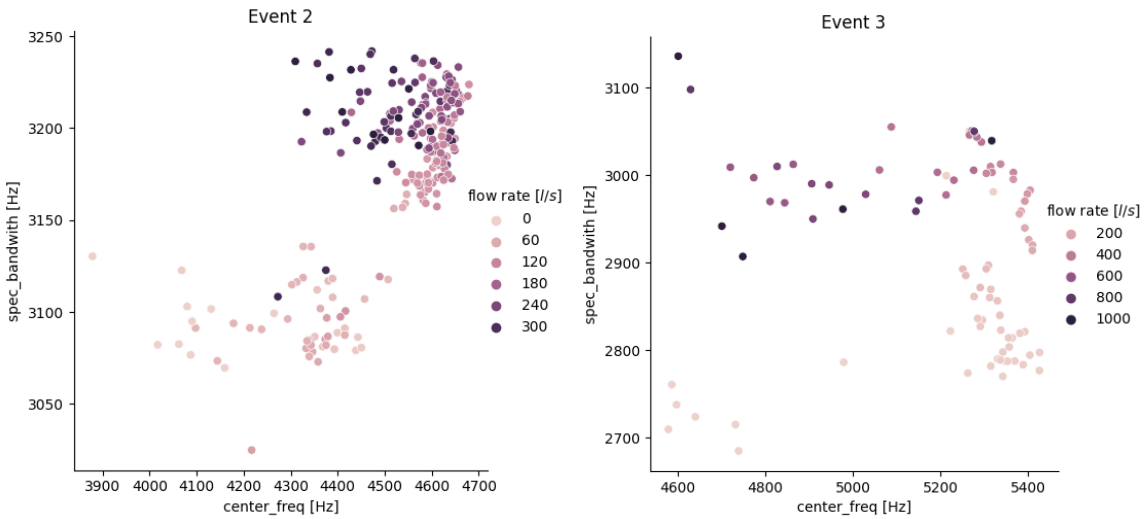

Figure 5.14: Spectral bandwidth against center frequency for event 2 and 3. This feature seems more suitable for comparable/robust features.

There is not much one can learn about a features of an event by studying the other events. They seem to cluster in different locations. This is especially true for the higher flows.

Concerning the research question (predicting flow rates from audio data) this suggests, that for the application in sewer pipes one has to identify a certain type of flow rate (low/mid/high) at least once to get an idea of the feature landscape, before predictions can be made. This circumstance will be reviewed again for the end-to-end methods.

5.4.4 Promising Features for High Flows

When observing the SPM for the dataset that concatenates all three events into one, there is one very promising feature, concerning the higher flows. It is the case for spectral flatness against center frequency, see Figure 5.16.

This nice separation already allows for a simple prediction model. We will manually draw a line and decide for high flows above that line and low flows below it, see Figure 5.17.

For that, we roughly pick two points (green cross) on the grid that make up a nice separating line (red line):

$$P1(3500|0.2) \quad \text{and} \quad P2(5500|0.4) \tag{5.1}$$

When solving the equation $y = mx + n$, one gets the following parameters:

$$m = 0.0001 \quad \text{and} \quad n = -0.15 \tag{5.2}$$

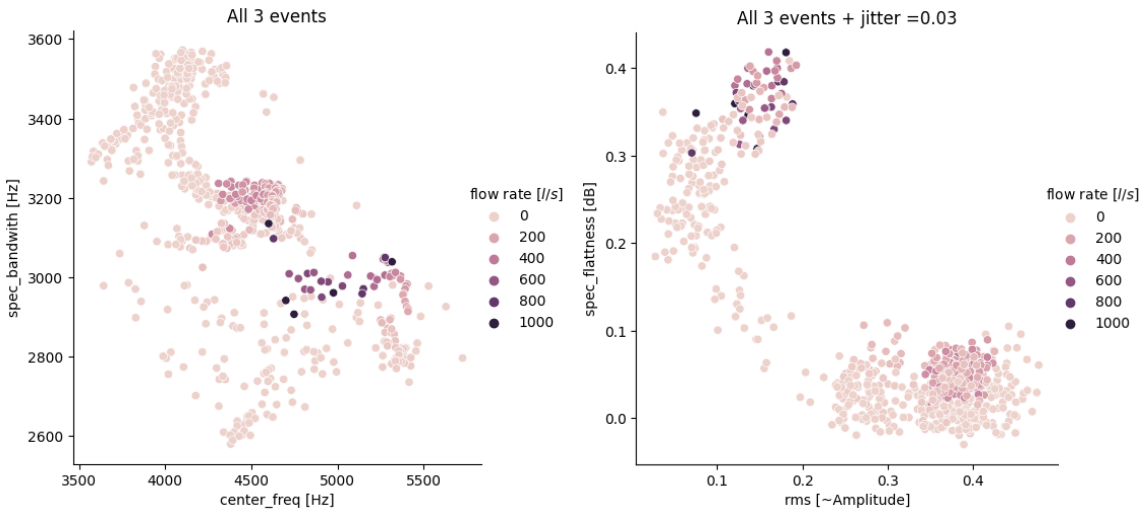

Figure 5.15: All events combined into one image for the check how well cluster translate from event to event.

Accuracy	0.99
F1 Score	0.92
Confusion Matrix	$\begin{bmatrix} 717 & 2 \\ 3 & 27 \end{bmatrix}$

Table 5.3: Performance of the line separation model for threshold = 400 l/s.

When computing the accuracy this classifier one gets the following tables (5.3 and 5.4): So we conclude with two main findings concerning that separation:

- Spectral flatness and center frequency combined allow for nice separation of values above 800 l/s. Thought, it produces many false positives
- When looking at the hue of the above plot, one notices that another flow rate threshold is naturally separated by the line, namely 400 l/s. This actually achieves an **almost perfect accuracy of 0.99** as well as a nice f1 score, so it is well balanced/robust too.

Note however that this result over fits the current circumstance, that we only have one event available with flows above 300 (Event 3). Although spectral flatness should remain a

Accuracy	0.97
F1 Score	0.46
Confusion Matrix	$\begin{bmatrix} 719 & 20 \\ 1 & 9 \end{bmatrix}$

Table 5.4: Performance of the line separation model for threshold = 800 l/s.

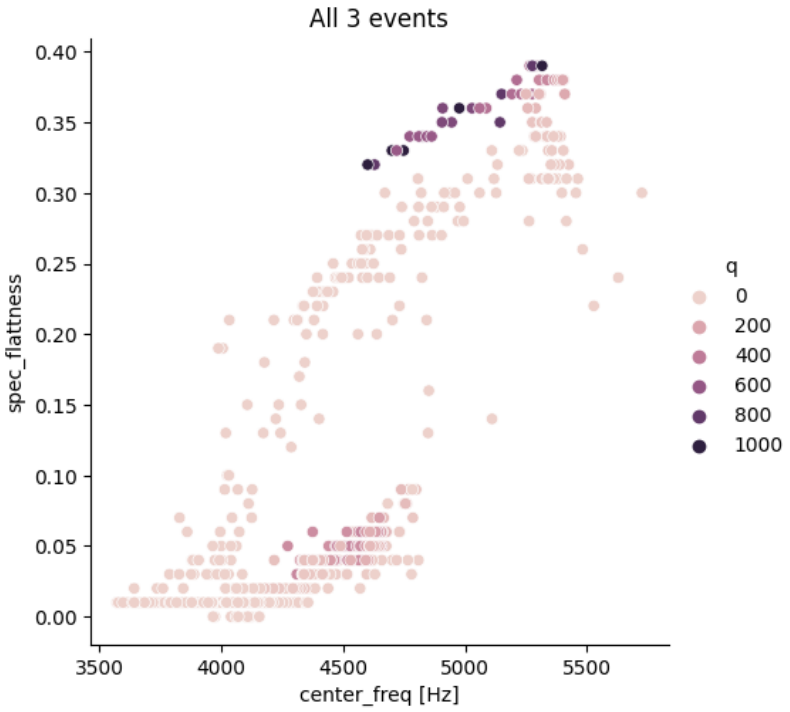

Figure 5.16: All events combined into one image and spectral flatness against center frequency. The most promising feature combination for highest flows so far.

feature of valuable insight, the center frequency is likely to vary with changing geometric conditions inside the sewer pipe and different locations.

5.4.5 Quick recap of the results so far

Concerning the big picture we can confortably state:

There are already some visual indications for robust features that clusters the data well in terms of correlation with flow rate.

Relative to the flow of the event at hand there were some trends observable concerning spectral features and the flow rate.

We looked at PSD, center_freq, spec_bandwith, spec_contrast, spec_flattness and rms. Zero_cross and pitch are discussed in the next subsection.

Here some quick summary of the results:

- When using only classical methods the differentiation of above/bellow 200 l/s is possible to some extend when using PSD. It is not perfekt, but definitely better than guessing (better by 13 %). For the other features it not possible.

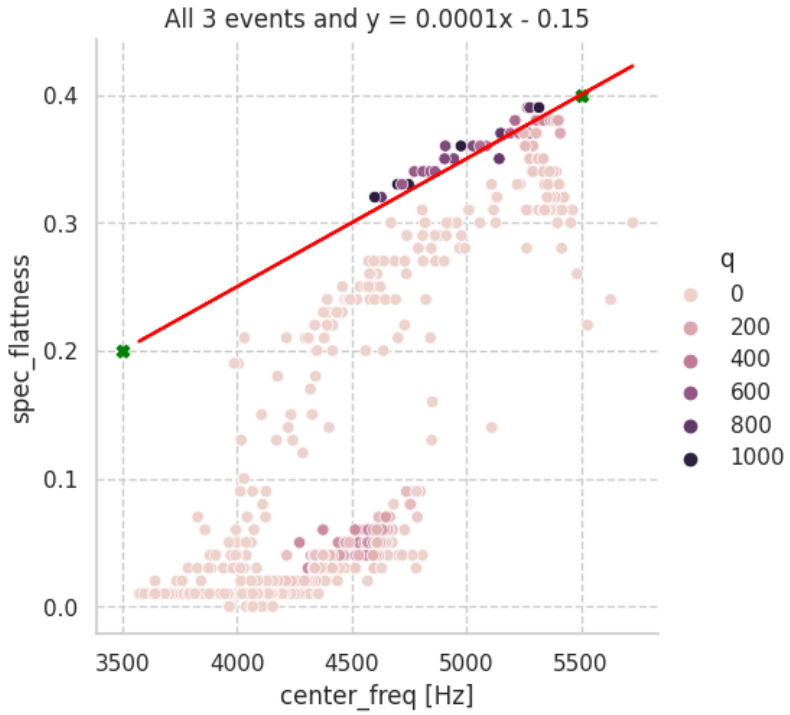

Figure 5.17: Representation of the separation model with a manually drawn line (red) based on points $P1(3500|0.2)$ and $P2(5500|0.4)$. The equation $y = 0.0001x - 0.15$ defines the separation line between high flows (above the line) and low flows (below the line).

- Using a linear separator with spectral predicting flows above $800\,l/s$ reaches an accuracy of 0.97. Almost all of the errors occur from false positives.
- When one shifts the objective from predicting above $800\,l/s$ to above $400\,l/s$ than one can accomplish an almost perfect classifier (0.99 accuracy).

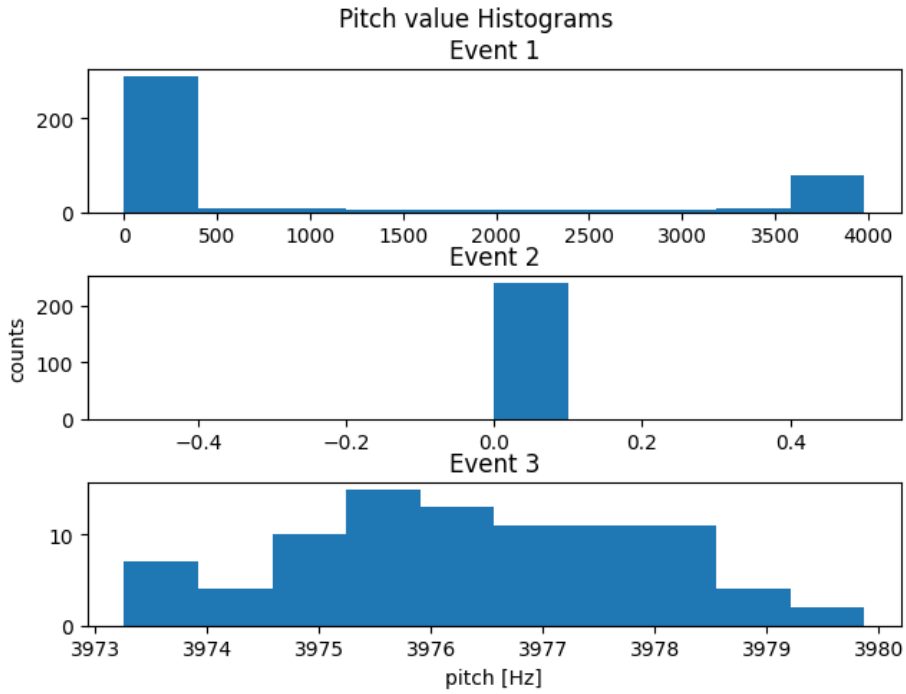

Figure 5.18: Result of the pitch estimation feature.

5.4.6 Further Results

This section will show some results that were found during experimentation that did not fit into the narrative chosen for this thesis. However, they might be interesting for followup investigations.

Pitch Estimation In our exploration of the recorded data, we encountered an expected observation. When viewing the pitch estimation feature in the SPM, they seem to be the least insightful into the data. In Figure 5.18 one can see a histogram of this feature for the first three events. Here a possible interpretation of that result:

White noise, being a random signal with equal intensity across all frequencies, lacks a discernible fundamental frequency. Pitch estimation methods are primarily designed for harmonic signals like speech or musical notes, where a clear fundamental frequency exists. In the case of pure white noise, attempting to estimate pitch can lead to inconsistent or unreliable results, often resulting in zeros.

Zero Crossing Rate One of the experiments made by GW which we were not able to reproduce, due to a lack of data is (for the sake of completeness) presented in the following:

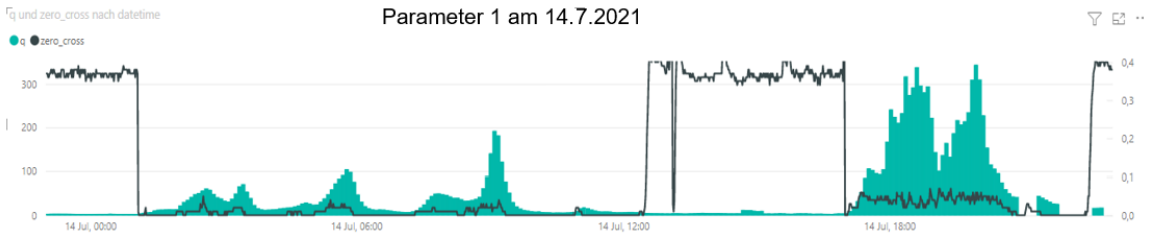

Figure 5.19: Relationship between flow rate and computed zero crossing rate. There appears to be a negative correlation between the two variables: as flow activities increase, the zero crossing rate decreases to values near 0. Conversely, when minimal flow is present, the zero crossing rate stabilizes at approximately 0.4.

Figure 5.19 shows the relation between the flow rate and the computed zero crossing rate. It seems like there is a negative correlation: When increased flow activities are present, the zero crossing rate drops to values close to 0. Likewise, when the there is almost no flow, the zero crossing rate takes the value 0.4.

When one compares the slope of that curve near the ’14 Jul, 18:00’ x-axis location with our event number 3 (at 13.07.2021, 24:00), as well as notices the max value having roughly $300\,l/s$. Is seems reasonable to conclude that they are the same. From that we can conclude that we only got a snippet of that entire event, namely the one where the zero crossing rate is close to zero. Therefore, no verification can be done by us.

When this effect appears to occur in every situation, this could be used to identify if flow is present or not. Possible implementations and applications of this are discussed in Section 8.

5.5 Hybrid Methods

In this section we will make use of machine learning techniques, more precisely, we will identify the limitations of the classical methods and try to improve the results, using more sophisticated algorithms.

For the following experiments (if not stated differently) train-test-split were performed through an initial uniformly random shuffle of the entire data and than splitting the data into two sets. We used a test size of 0.2, which means that 20 % of the data is used for testing and the rest for training.

5.5.1 Improving on the $200\,l/s$ decision boundary

The best performance for separating between above and below a flow rate of $200\,l/s$ obtained by using the energy of the PSD is shown in Table 5.2. When we train a random forest (Number estimators = 100, everything else keeps the default values of sklearn[10]) on the same data to perform the same task, we get the performance shown in Table 5.5.

Accuracy	0.82
F1 Score	0.51
Confusion Matrix	$\begin{bmatrix} 136 & 15 \\ 18 & 17 \end{bmatrix}$

Table 5.5: Performance of the random forest on separating $200\,l/s$ with the *psd1378* feature.

One can see that the performance even drops. When one tries other methods (the ones presented in the theory section) than they all perform roughly the same. This suggests that we reached the limits of the *psd1378* feature and should try something else.

So the next experiments relaxes the effort to do manual feature design and combines features. Therefore we took the entire PSD as the input for the random forest classifier. Results are depicted in Table 5.6.

This already performs better than using classical method alone[11]. This is probably, because the random forest considers much more numerical values at the same time, makes (locally) optimal decision concerning the separation of the data and does that multiple times with a final majority vote. So it is not a surprise, that this performs better than the pure human decision.

[10] Default values of RandomForestClassifier in scikit-learn are in the Appendix.

[11] Note, that this time using the weighted f1 score will not result in any performance improvements compared to normal f1 score.

Accuracy	0.94
F1 Score	0.80
Confusion Matrix	$\begin{bmatrix} 226 & 8 \\ 10 & 35 \end{bmatrix}$

Table 5.6: Performance of the random forest on separating $200\,l/s$ with the entire PSD as feature.

Next we decided to check if any other algorithm might perform better. The results are shown in Table 5.7.

Classifier	**Accuracy**	**F1 Score**
SVM	0.94	0.79
Gradient Boosting	0.94	0.73
K-Nearest Neighbors	0.93	0.73
Logistic Regression	0.97	0.86
Naive Bayes	0.84	0.55
Neural Network (MLP)	0.96	0.83

Table 5.7: Performance of Different Classifiers on separating $200\,l/s$ with the entire PSD as feature.

The winner clearly is the Logistic Regression algorithm[12], also better than the initial random forest. For its entire performance report (with confusion matrix) see Table 5.8.

Accuracy	0.97
F1 Score	0.86
Confusion Matrix	$\begin{bmatrix} 243 & 5 \\ 4 & 27 \end{bmatrix}$

Table 5.8: Performance of Logistic Regression Classifier on separating $200\,l/s$ with the entire PSD as feature.

This suggests that differentiating between above and below $200\,l/s$ is likely to be good enough in practice, for the application intended by GW. Also note that no anomaly detection (or any other data cleaning method) was performed before training, so it might even be possible, that after some appropriate cleaning the performance becomes even better.

[12]Default values of LogisticRegression in scikit-learn shown in appendix

5.5.2 Improving on the $800\,l/s$ (and $400\,l/s$) decision boundary

Next a review of the results concerning the performance of the higher flows is presented.

The best performance achieved by the classical methods is shown in Table 5.3 and Table 5.4. Now we use the same features but instead of a simple linear model, we use a random forest. Note that the other models where tried as well, but this time the random forest performed the best. See its performance in Table 5.9 and 5.10.

Accuracy	0.99
F1 Score	0.85
Confusion Matrix	$\begin{bmatrix} 360 & 2 \\ 2 & 11 \end{bmatrix}$

Table 5.9: Performance of the random forest classifier for threshold = $400\,l/s$.

Accuracy	0.99
F1 Score	0.29
Confusion Matrix	$\begin{bmatrix} 369 & 2 \\ 3 & 1 \end{bmatrix}$

Table 5.10: Performance of the random forest classifier model for threshold = $800\,l/s$.

For $400\,l/s$ there is no difference concerning the accuracy, however a worse f1 score. This is however probably not because it learned an underlying structure, but because (due to random sampling of the train and test data) there are less points inside the second class. It is still 2 false positives (FP) and 2 false negatives (FN) (probably the same points as it was for the manual line separator).

For $800\,l/s$ the performance is again a bit more dificult to interpret. The accuracy went from 0.97 to 0.99. and the f1 score from 0.46 to 0.29. When investigating the relative distribution it becomes clear that the performance is worse. The line separation had twice as many FN (upper right in confusion matrix) as it had TN (true negative, lower right in the confusion matrix) and only one (roughly 1/10 of the TN) FP.

For the random forest model it is again twice as many FN as TN, but three times as much FP as TN. So it indeed got worse (relatively speeking).

Nonetheless, since there are only 6 data points and we know how the data landscape looks like (see Figure 5.16) these is no point in further interpreting the results of that experiment. It confirms the limits of the dataset and the features spectral flatness and center

frequency.

Notably, when we tried the enrich the input data with the other features (spec_bandwidth, zero_cross, pitch, rms) the performance dropped. This is the opposite behavior that it was for the PSD and $200\,l/s$. However, this was expected since it was already visually clear that all the other features don't capture any useful information concerning high flows. Therefore that model first has to learn to ignore them, which also takes time and effort.

At that point there is no further need to train more sophisticated machine learning algorithm for this particular task, since their effectiveness seems to saturate.

However, this brought up a new research direction that slightly deviates from the application of GW. Namely: **There may be granularity and location of the decision boundaries that intrinsically separate the data well**. Intrinsically, here can be understood as naturally by the data and underlying phenomenon. Maybe she sound of changes drastically due to some physical turbulence effect that is especially dominant in this certain sewer pipe architecture, so it is especially easy to hear and therefore classify.

To demonstrate this with the current experiment see Figure 5.20. It demonstrates how a different decision boundary reaches different accuracies.

The reason for those performance performance differences can be clearly seen in Figure 5.21. For $800\,l/s$ the points cluster in one region but spread across the above lengthy cluster. For $400\,l/s$ it appears to be the entire above lengthy cluster, so it is clear that this performs well. For $200\,l/s$ one can see that other points start to appear in another place, making spectral_flatness and center_frequencies not suitable anymore. For that another feature should be better, for example the random forest on PSD presented earlier in this section. This alone does not necessarily mean that it is not separable, but the new points appear relatively wide spread inside another cluster.

However, notice how the performance goes up again for roughly $50\,l/s$ when analysing Figure 5.20. This once again confirms that some frequencies (particularly the ones below $400\,l/s$) seem to be better separable than others.

5.5.3 Summary of the results so far

The integration of hybrid methods demonstrates a notable improvement over the exclusive utilization of classical methods for the PSD, however not necessarily for the highest flows. This was somehow expected, since we have only a hand full of data point up there. Therefore, combining these methods, The best performance is to be expected.

Another interesting finding is, that some decision boundaries seem to emerge naturally from the data. This especially seems to be the case or lower flows. However, this can also

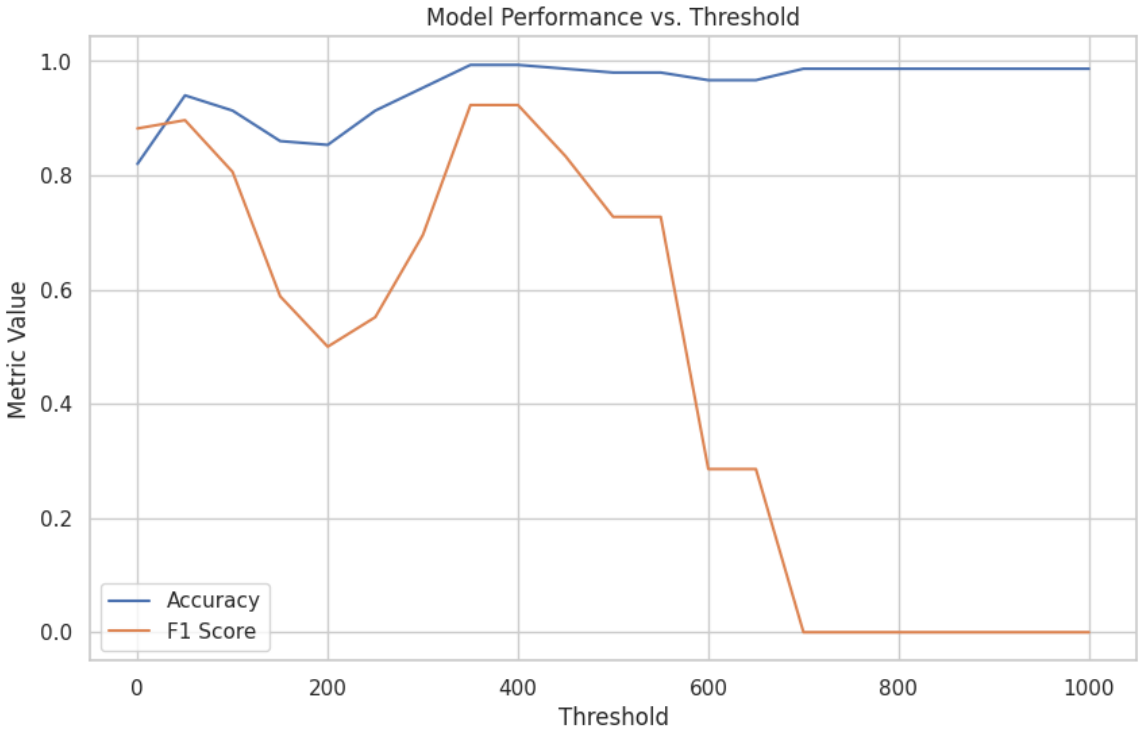

Figure 5.20: Checking what threshold can be separated best for the features spectral flatness and center frequency depicted in Figure 5.16. It demonstrates how a different decision boundary reaches different accuracies.

be the effect of more data being available there, therefore the machine learning models can leverage their capabilities better.

However, the ultimate evaluation and comparison of these proposed methods will be presented in Section 5.7. There we will combine the methods into one custom model.

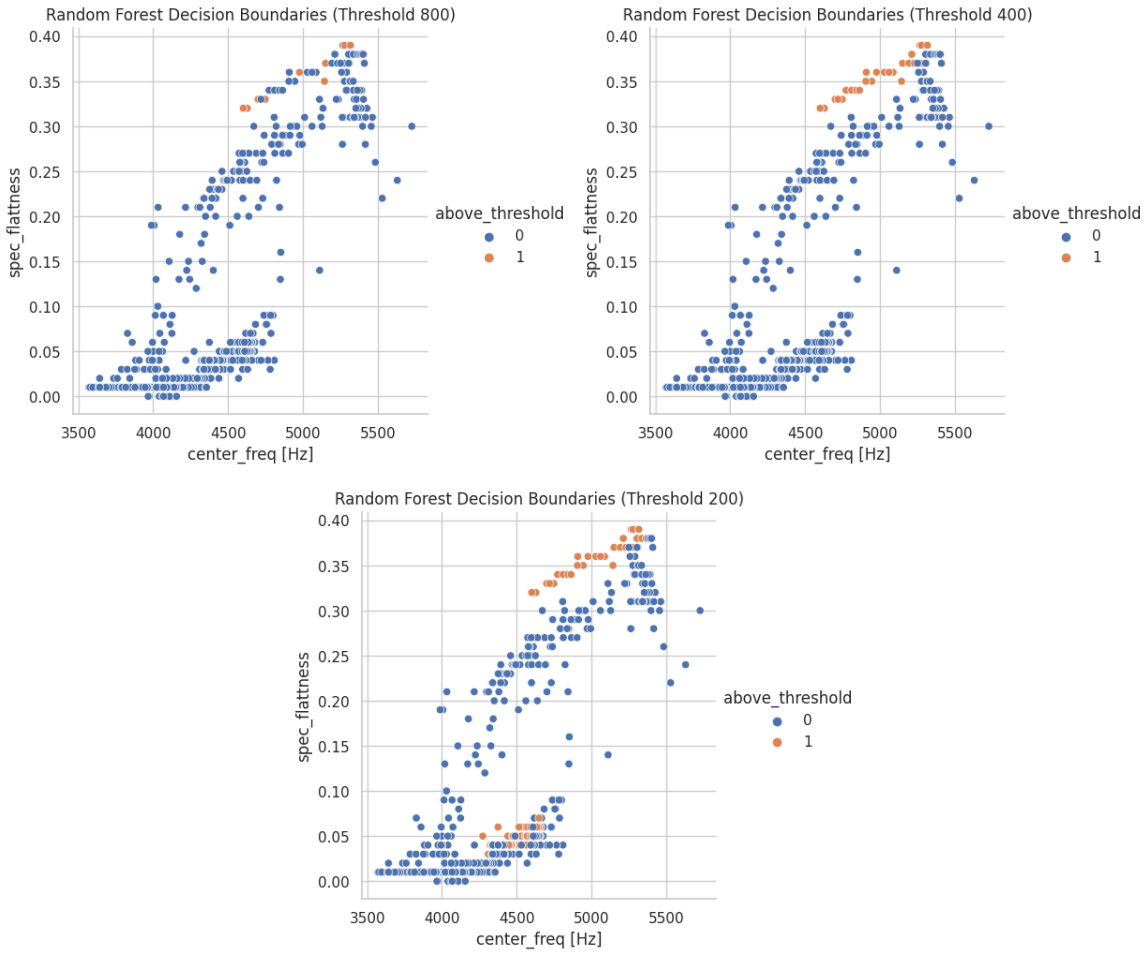

Figure 5.21: Color coding the recordings due to their flow rate. This explains why the random forest performs best for 400 l/s and worse (in terms of accuracy, not f1 score) for 200 l/s.

5.6 End-to-End Methods

Although we call these methods end-to-end, this is not the most pure form of it. In our case that would be a neural network that is trained on the .wav files directly to predict flow rates. In principal this is possible, but for that much more data is needed.

As already mentioned for a raw 10 second recording saved as as waveform with a sampling rate 48 kHz that means 480,000 floats per sample. For the spectrogram (implemented by the librosa library [30]) it is roughly 120,000 floats per sample. So 1/4 of it, which is still far from human readable/manually processable. This is why we still call this end-to-end, even though some pre-processing (computation of the spectrum) was performed.

The baseline of this section will be training a fully connected neural network on log-mel-

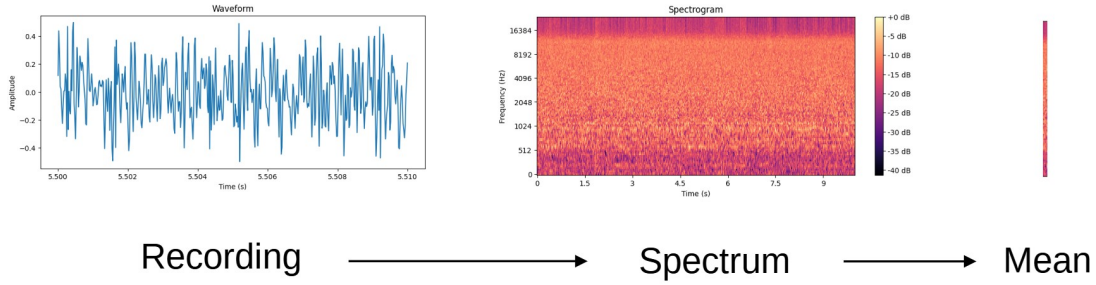

Figure 5.22: Schematic for visualizing the pre-processing procedure. First the recording (as a .wav file) is converted into a spectrum and than the mean is computed over all frequencies.

spectra.

5.6.1 Training the model

For training we tried to 'be as end-to-end es possible', however just presenting the spectrum is still too much information to handle for the network. Within training 500 epochs it was still not able to converge to a useful result, so we decided to compress it even further. So after the compute the spectrum we also calculate the mean for it. For a schematic see Figure 5.22.

For the mean signal a length of 0.2 seconds was chosen, which is still above the lower bound that would allow to still resolve 20 Hz, due to the Nyquist theorem.

Notably, such a aggregation is only feasible, if the signal remains constant during the entire length of the recording, which is 10 seconds in our case. This can be confirmed since the (properly scaled) FT of those audio files did not deviate from the (full length) original signal, when there are no sudden events.

Furthermore down sampling was performed again to support the training. Again, this should only be done with care, because it reduces the amount of information available. Here we compressed it down to roughly 30 floats. This is quite a lot, but reasonable for this application, since the overall shape of the PSD seems enough.

Also note that decision boundaries were apposed right from the beginning, except for one experiment, which is discussed in Section 6.2. This is so we can easily compare it to the other models and because it simplifies the training since more 'knowledge' went into the model already.

The architecture of the model used for the following experiments (including the chosen

hyperparameters of the model and the training) are presented in Table 5.11.

Parameter	**Value/Description**
Model Architecture	Sequential fully connected model
Layers	Input layer: Shape - (Number of features) Dense layer 1: 128 neurons, ReLU activation Batch Normalization Dense layer 2: 64 neurons, ReLU activation Batch Normalization Output layer: Softmax activation for all target columns
Optimizer	Adam
Loss Function	Categorical Crossentropy (softmax entropy)
Metrics	Accuracy
Training Parameters	
Epochs	500
Batch Size	516
Validation Split	0.2

Table 5.11: Description of the Trained Neural Network Model and Training Parameters.

The model reached an accuracy of 0.89 so already similar useful as the other ones derived so far (after aggregation and down sampling). This demonstrates the power of machine learning since much less understanding of signal processing and the data was needed to achieve a comparable accuracy right away. The confusion matrix is depicted in Figure 5.23.

5.6.2 Adding data augmentation: Mixup

To effectively increase the number of available training samples, existing recordings can be manipulated randomly during training. Here we perform mixup, an introduction on this was given in the theory section.

There is one value β that has to be chosen for Mixup. It is the amount of blending between the actual and different sample. We tried the values$(1, 0.8, 0.5, 0.2)$. A report of the achieved performance (accuracy and confusion matrix) is shown in Figure 5.24.

$\beta = 1$ (Accuracy = 0.89)	$\beta = 0.8$ (Accuracy = 0.85)
$\beta = 0.5$ (Accuracy = 0.91)	$\beta = 0.2$ (Accuracy = 0.86)

Table 5.12: Table of β values and accuracies for corresponding plots in Figure: 5.24

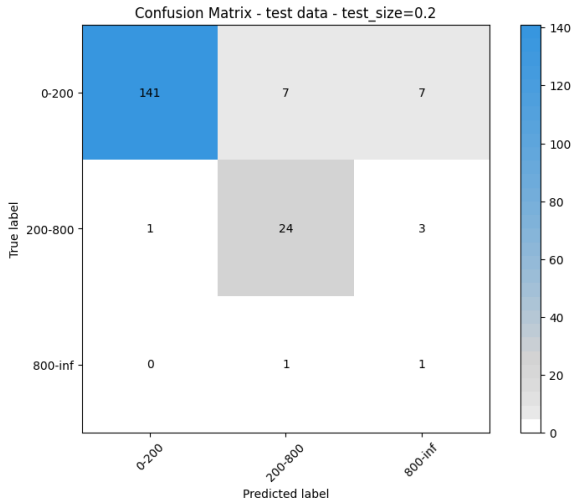

Figure 5.23: Confusion Matrix that shows the performance achieved by the model presented in Table 5.11.

There we make two main observations, namely for $\beta = (0.2, 0.8)$ the performance goes down and for $\beta = 0.5$ the performance is slightly better. It increased a bit (0.91 accuracy) and the false positives decreased compared to without Mixup.

Since the addition of Mixup performed a bit better, all the following experiments were performed with it.

5.6.3 Data Augmentation Decision

In the context of this thesis, the application of data augmentation techniques was carefully considered, including the use of Mixup. However, it is important to clarify the reasons behind the decision not to pursue data augmentation further, as the results were not as promising as initially hoped.

The decision to employ data augmentation was primarily driven by the nature of the available dataset, which consists of a spectrum of flow rates, including both lower and higher flow rates, recorded under various conditions.

For the lower flow rates, there existed a sufficient volume of data suitable for experimentation. Augmenting this segment of the data would essentially generate additional instances of an already well-represented category within the dataset, offering limited potential for enhancing model training.

Conversely, the dataset contained only a hand full of instances for higher flow rates. Augmenting this small data sample would lead to artificially created samples that closely resemble the original data. Such artificially generated data may not fully capture the unique

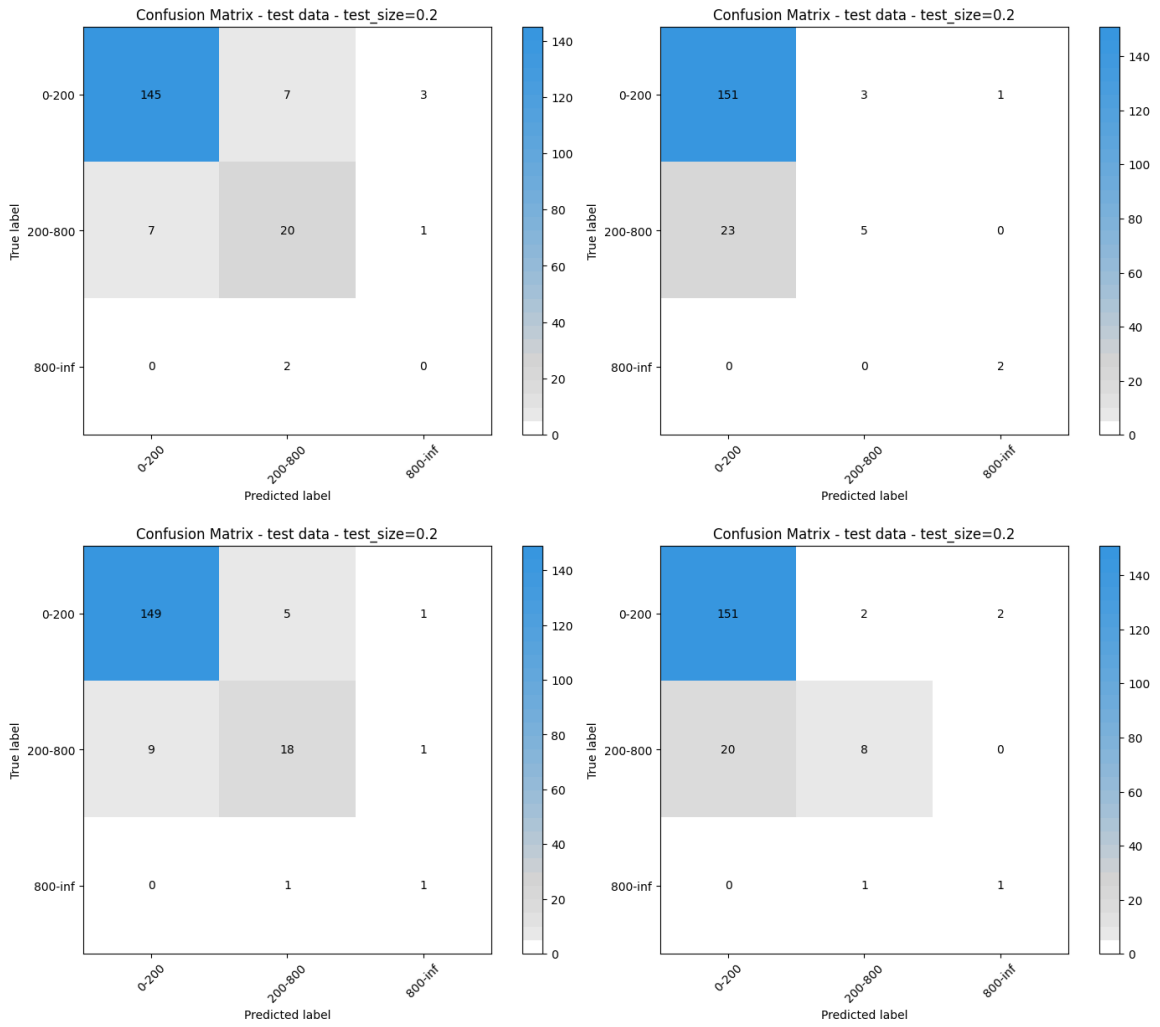

Figure 5.24: Confusion matrices of the models that were trained with augmented data. The corresponding β values can be viewed in Table 5.12.

characteristics, potential outliers, or nuances associated with high flow rate events.

Furthermore, the effectiveness of data augmentation techniques, such as Mixup and SpecAugment[35], often relies on the availability of a substantial and diverse dataset. Due to the dataset being small and not very varied, traditional data augmentation methods were thought to be less fitting for our research goals.

It is crucial to emphasize that while data augmentation was applied in this study and even caused slightly improved performance, other approaches and strategies were explored to enhance the performance of the flow rate estimation model. The later strategies resulted in greater improvements.

5.6.4 Concluding words

One can perform various kinds of hyper-parameter optimization to try to increase the performance further. Here a non-exhaustive list of tweaks we tried:

- Trying different learning rates.
- Adjusting batch size.
- Experimenting with the number of neurons in each layer.
- Trying adding or removing hidden layers.
- Introducing dropout layers to prevent over-fitting.
- Applying L1 or L2 regularization to the weights.
- Exploring different regularization techniques.
- Trying different activation functions.
- Adjusting the number of training epochs.

All of these techniques did not help to perform much better than the results already presented. Since investigations like that are very common and already intensively discussed in the literature, we will not go into any detail here. The interested reader can find more about it in [51].

One could obviously improve performance even further, by choosing more sophisticated networks, for example the attention mechanism[48]. However, figuring out a way of applying this technique (mostly used in transformers or recurrent neural networks for Natural Language Processing[43]) for the situation at hand might be a topic of a thesis on its own.

5.7 Comparing Methods

We will now compare classical, hybrid, with the end-to-end methods with each other, even though the previous sections already suggest, that the pure classical methods perform worse, than the others.

5.7.1 Combining models into a single classifier

So far all the methods presented in the section about classical and hybrid methods focused on binary classification, namely predicting of flow rates are above or below a certain region. Now we want to combine those models into a single classifier that predicts more than just two classes.

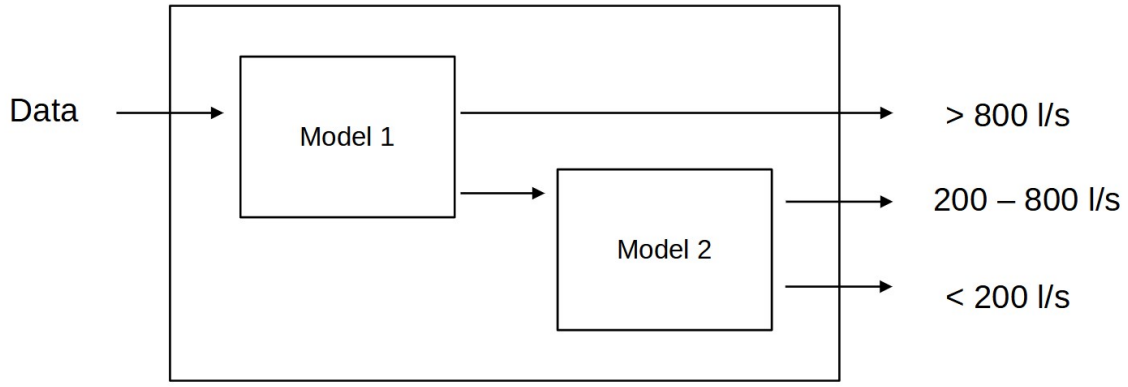

Figure 5.25: Illustration of the combined model that classifier three classes, however being composed of two models that perform binary classification.

There are many ways to do it, we will now present the way we diced for: First we use the trained model that differentiates flow rates above and below $800\,l/s$ since it is the more specific one in terms of how much data is available. If the prediction says it is bellow $800\,l/s$ the same data is shown to the next model trained to differentiate flow rates above and below $200\,l/s$. This architecture is illustrated in Figure 5.25

5.7.2 Final comparison

Now we will compare all the best performing (combined) methods with each other. The final comparison is depicted in Table 5.13.

Classical Methods	**Hybrid Methods**	**End-to-End Method**
Accuracy 0.84	**Accuracy** 0.92	**Accuracy** 0.92
F1 (weighted) Score 0.83	**F1 (weighted) Score** 0.92	**F1 (weighted) Score** 0.91
Confusion Matrix $\begin{bmatrix} 561 & 42 & 2 \\ 59 & 53 & 18 \\ 1 & 0 & 9 \end{bmatrix}$	**Confusion Matrix** $\begin{bmatrix} 229 & 13 & 0 \\ 6 & 43 & 3 \\ 0 & 2 & 2 \end{bmatrix}$	**Confusion Matrix** $\begin{bmatrix} 235 & 10 & 0 \\ 9 & 39 & 0 \\ 2 & 3 & 0 \end{bmatrix}$

Table 5.13: Comparison of accuracy measures and confusion matrices for all the so far proposed (best performing) methods.

Even though the metrics (accuracy, F1) are basically the same for hybrid and end-to-end, One can clearly see that hybrid performs better for the latter class. It seems 'convinient' for

the network to ignore the 'above $800\,l/s$ class. However, if one needs to tweek a bit more for certain classes the hybrid approach seems most promising.

This result was somewhat expected, because when a network optimizes for statistical measures it is to be expected, that underrepresented classes ar investigated and weighted less. In our application/research however this is was the more important differentiation, so more (classical) investigation had to be performed, to understand the circumstances better.

So this suggests that if the given dataset is a representative depiction of the distribution of interesst (not just of availability!) than utilizing end-to-end methods would have performed comparable to an in depth investigation of a research scientist with domain knowledge. However, if the given data underrepresents the properties of interesst than deeper investigations have to be performed.

Note that the performance of the classical methods is rather a training performance than a test performance, because we used the entire dataset for the manual investigations and also for the evaluation of the presented performance measures. Still it gives a rough idea of its capabilities. Again, both hybrid and end-to-end methods were trained with a test size of 0.2.

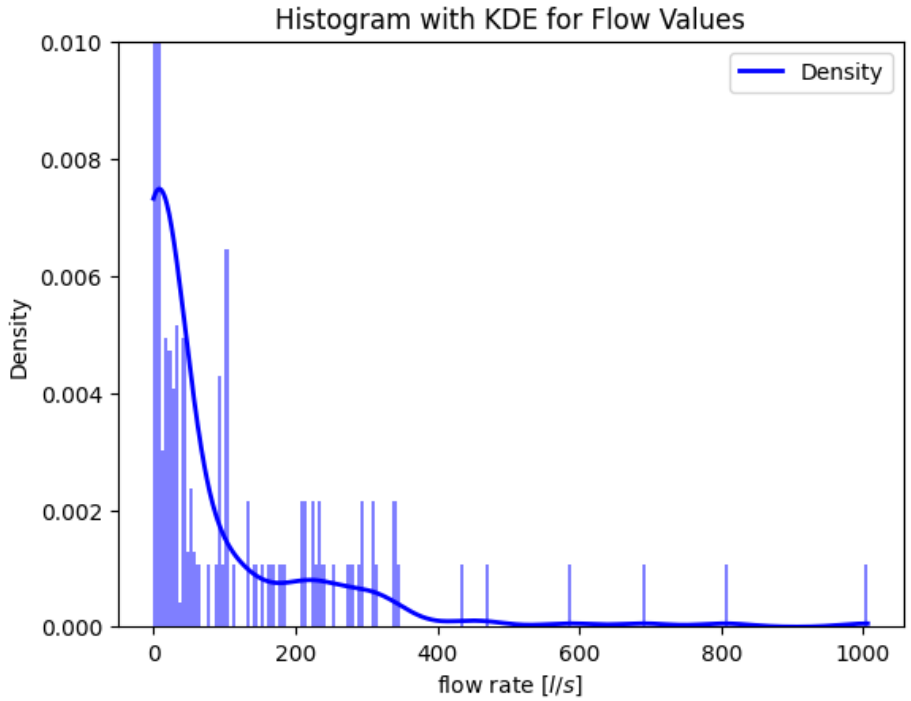

Figure 6.1: KDE for all measured flow rates.

6 Complementary Experiments

The last section aimed at performing as good as possible for the application of GW. This section will dive a bit deeper into other interesting findings that were discovered during experimentation, but were heading into a slightly different direction.

6.1 Focusing on Lower Flow Rates

This section was motivated by the observation that the proposed methods perform roughly the same on the task encouraged by GW, especially for the highest flows. Despite GWs application maybe relaxing the scope of flow rates will result in better performance for a task can still be considered as part of the research question asked here. The question now is:

What if we throw away the high flows and try to make more precised predictions with the lower ones? Is the error rate and accuracy better for lower flows?

The experiments performed in the previous section already suggested, that some decision boundaries might separate the data better due to some intrinsic nature of the data. The first decision that hast do be done here is the choice of where the new decision boundaries should be.

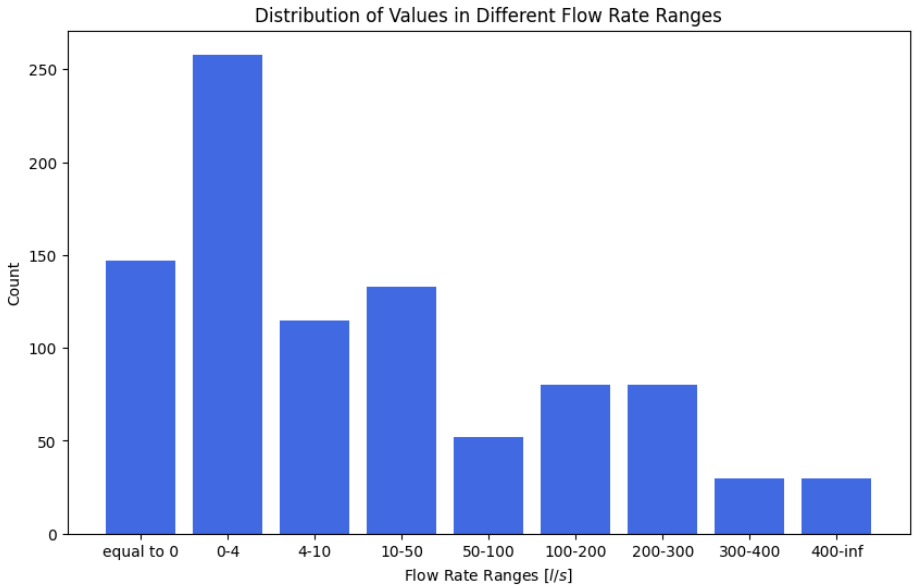

Figure 6.2: Histogram of flow rates for finding decision boundaries, so the amount of data points are distributed roughly equal to the bins.

One possibility would be to use 10, 50, 100, 200, 300 as classes because by doing so higher flow rates are ignored while ensuring a finer granularity for low flow rates.

When the KDE of all measured flow rates one gets an idea of how many counts of certain flows are present in the data. Such a KDE is shown in Figure 6.1
From this KDE we see that most values are between 0 and 100 than there some more values between 200 and 400 and from there on it is only a few different values. Since the KDE only gives us relative values, we will now look into a histogram to find boundaries, so the amount of data points are distributed roughly equal to the bins. This was done manually.

The mentioned histogram can be seen in Figure 6.2. For the rest of this section these values will be the decision boundaries our classifier tries to differentiate.

For the training we decided for the aggregated and down sampled spectra again. For the model we used a random forest and a Neural Network (the same as in Table 5.11). The result of the training can be seen in Figure 6.3 for the confusion matrices and Table 6.1 for the corresponding settings and reached accuracies. We gradually increased the size of the training data to show its effect on the performance.

Here a list of interessting observations:

- Predicting 300-400 fails most of the time for all of the experiments
- It seems to be easy to differentiate between flows above and below $10\,l/s$. There are

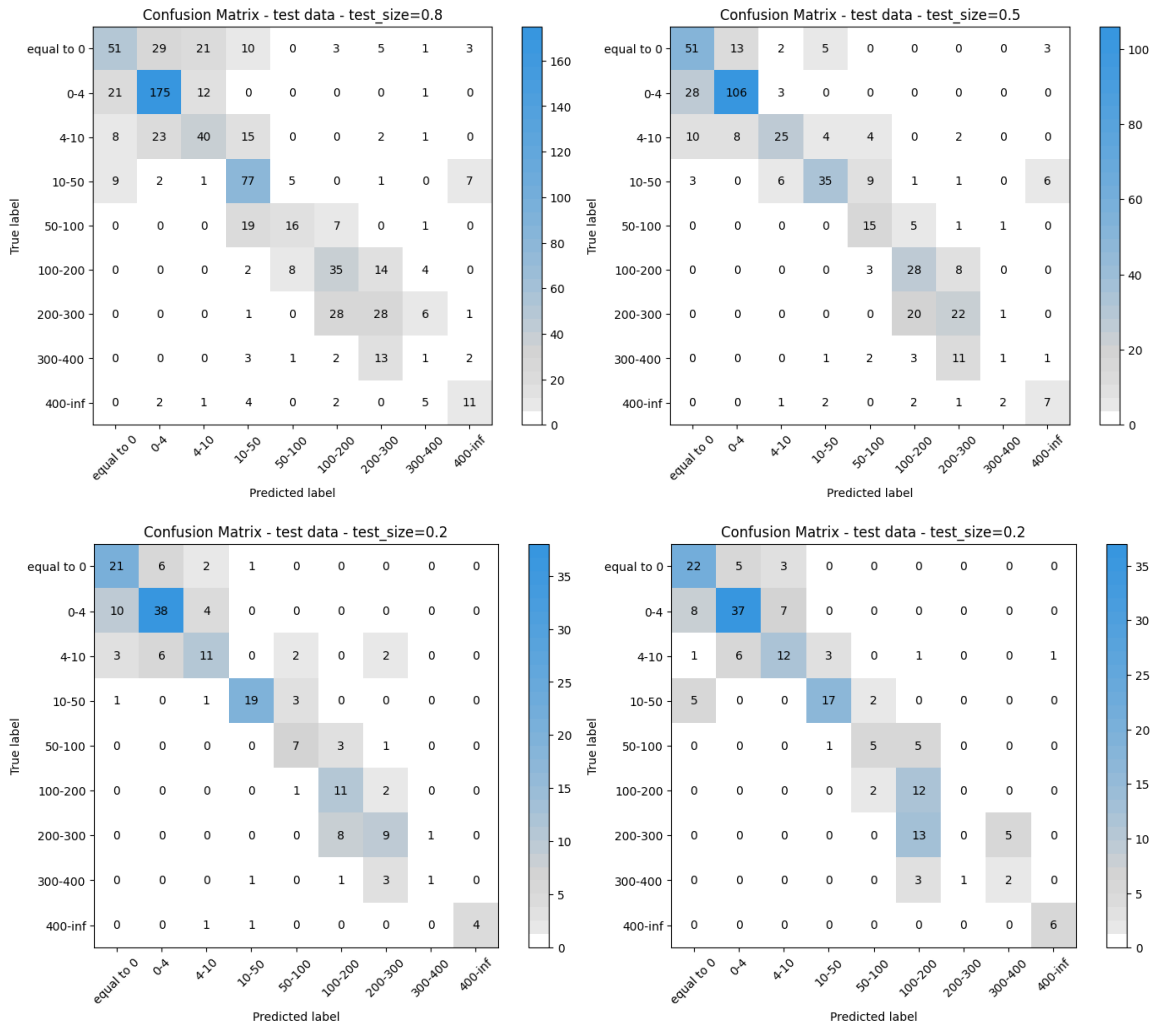

Figure 6.3: Confusion Matrices for the classifiers of lower flow. The corresponding settings and achieved performances are shown in Table 6.1.

only few FP and FN.

- From 50 l/s upwards it seems more difficult to clearly assign the proper flow rates. The error increases, but still allows for estimating roughly. Most of the FP/FN predict the direct neighbor of the correct class. This suggests that the relative error is smaller for lower flow rates than it is for higher flow rates. This can mean that for the lower flows minor changes in the flow speed result in a more distinct sounding recording than for higher flows.

- The size of training set matters. The more training data, the better the accuracy. However, the effect seems to saturate quickly. When comparing test_size = 0.2 (which means 20 % of the data is used for testing. In our case that's 185 our of 929 recording.) with test_size = 0.5 (464 out of 929, so less than half the training data) the only

Random Forest (test_size=0.8) Test accuracy = 0.59, Weighted F1 Score = 0.58	Random Forest (test_size=0.5) Test accuracy = 0.63, Weighted F1 Score = 0.62
Random Forest (test_size=0.2) Test accuracy = 0.65, Weighted F1 Score = 0.65	Neural Network (test_size=0.2) Test accuracy = 0.61, Weighted F1 Score = 0.62

Table 6.1: Table of Model Performance for the plots shown in 6.3. The Neural Network here is the same model as in Table 5.11.

noticeable difference is the separation between flows above and below $10\,l/s$. For the higher flows the difference is not noticeable. For the separation for flows above/below $10\,l/s$ the increased dataset made a clear noticeable difference. With test_size = 0.5 the separation was still a bit blurry, with 0.2 though, it was not.

- Regarding the improvements, another interesting observation is that while for the bigger test size and flow rates above $10\,l/s$ most FP/FN chose a direct neighbor relatively symmetrically, it becomes much more asymmetric for smaller test sizes. Visually speaking: the model overestimates the flow much less often, but stays similar with underestimating.
- This last experiment used a more sophisticated fully connected neural network. However, there are no noticeable differences, except for a few more FP/FN and a bit worse accuracy.

Concluding words

The exploration of refining groundwater prediction models revealed intriguing nuances, especially in predicting lower flow rates. Focusing solely on lower flows demonstrated reduced errors and clearer distinctions in predictions.

Decisive boundaries, identified through analysis of flow rate distributions, notably impacted model performance. Larger training sets significantly enhanced accuracy for lower flow separations. However, employing a more sophisticated neural network did not yield substantial improvements over other models. The findings suggest potential for enhanced predictive accuracy in specialized subsets, emphasizing the distinctiveness of lower flow predictions.

One could interpret this observation regarding the high flow separation task the following way: Either we do not have enough data or this task is not solvable in the sense that it is not distinguishable with the audio data provided, maybe not even with audio data in general. Without more of the high flow data it is hard to tell the reason.

6.2 Domain Shift

The concept of domain shift and domain generalization encapsulates the challenge of making AI methods robust. This small section aims at testing this robustness. In our context we investigate into two different types of domains, different *location* and different *event* in terms of contained flow rate. The question is then:

How well can the trained models generalize different domains (locations and situations)?

In the the previous section about classical methods (Section 5.4) we already saw that the data clusters in unpredictable locations, making generalisation impossible. Now we focus on it in slightly more detail.

6.2.1 Training and testing on similar flow rates but different locations

There are no investigations concerning the improvement of those methods, since we have only two locations for labeled audio samples available, and the second one does not cover the range of values that is of interest (58 l/s is the max flow).

Overall Test accuracy	0.88
Weighted F1 Score	0.86
Confusion Matrix	$\begin{bmatrix} 141 & 4 & 0 & 0 & 0 \\ 10 & 2 & 2 & 0 & 0 \\ 1 & 0 & 12 & 2 & 1 \\ 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
Decision Boundaries	['0-10', '10-25', '25-50', '50-100', '100-inf']

Table 6.2: Performance metrics and confusion matrix of a random forest trained on the dataset that combined event 1 and event 4.

Still we wanted to at least check it. So we trained a random forest again on the PSD of only the first event (Ueckendorfer_Str) - with a max flow of roughly 100 l/s - and test on the forth event (Holtkamp). Note that the first event has roughly 2.3 times as many data points, which can be understand as a test size of 0.43.

For that the following bins have been chosen ['0-10', '10-25', '25-50' , '50-100', '100-inf']. Note that for the test confusion matrix it is to be expected that there is no data in the last bin and only very few in the one before that.

In order to be able to interpret the results, we trained two models and compare them. The first one shuffles the data of event 1 and event 4 into one dataset and then separates it randomly again with a test size of 0.3. The results of that can be seen in Table 6.2

The results of experiment where the random forest was trained on event 1 and tested on event 4 is shown in Table 6.3.

Overall Test accuracy	0.78
Weighted F1 Score	0.70
Confusion Matrix	$\begin{bmatrix} 137 & 0 & 0 & 0 & 0 \\ 25 & 1 & 1 & 0 & 0 \\ 8 & 1 & 1 & 2 & 2 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
Decision Boundaries	['0-10', '10-25', '25-50', '50-100', '100-inf']

Table 6.3: Performance metrics and confusion matrix of a random forest trained on event 1 and tested on event 4.

As it was expected, the performances decreases. Especially the beforehand relatively well performing range 25-50 degraded significantly. And since this was actually a range well represented by the data, this hints at the training data not being general enough for capturing relevant features.

6.2.2 Training and testing on different flow rates, but the same location

For this experiment we used the neural network depicted in Table 5.11 again, however with one crucial modification. The output now does not predict a particular class, but performs regression, so it returns a floating point number representing a flow ware in l/s [13].

Here we did two experiments. For the first we trained the network on event 1 and tested them on event 2 and 3. For the second experiment we trained on event 3 and tested on event 1 and 2. The results of both experiments can be seen in Figure 6.4.

For the first experiment one can see that the model is able to perform appropriately on the train subset, but absolutely fails on the test subset. It seems like the model is not able to generalize flow rates that exceed the limits of the ones it had been provided. Therefore no matter how high the labels become, the network always stays under $100\,l/s$ which is the

[13] This strategy was not presented earlier, because it does not change anything in the performance, but makes it unnecessary complicated to compare it to the other models (classical, hybrid).

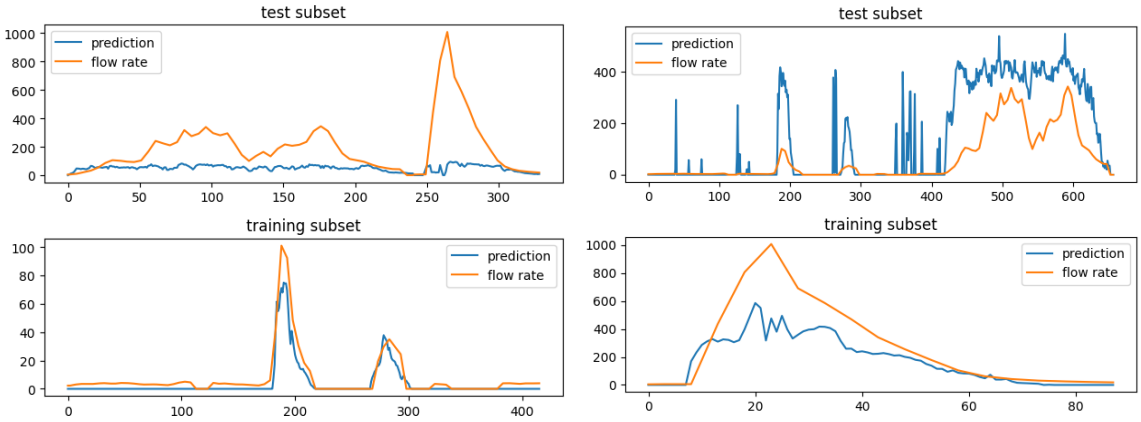

Figure 6.4: Training a neural network for predicting flow rates in a regressive manner. Left: Training on event 1 and testing on event 2 and 3. Right: Training on event 3 and testing on event 1 and 2.

maximum value of the train subset.

In the second experiment the performance is even worse. There the model already fails at producing good results on the training subset. It seems to stagnate at roughly $400\,l/s$, and does not go higher than this, even though it is part of the training subset. This might be due to the fact that the subset has less than 80 points.

For the prediction one can also see that it fails to predict the lower flows, even though such flows are part of the training subset. One can see that most most of the time when some decreased flow is present the network jumps up to its max flow $400\,l/s$ (for the second smaller peak of event 1 to $200\,l/s$). Sometimes it jumps up without any noticeable reason, which might be caused by noise.

All the experiments presented in this section support the hypothesis that extrapolating different domains is not possible without special care. If this is something of interest, than suitable datasets have to be collected to learn more about the underlying behavior causing the differences for shifting domains.

7 Conclusions

The design of algorithms for acoustic flow rate estimation within sewer pipes was presented here. For other applications, there are works focusing on acoustic flow rate estimation. However, usually these attempts are performed in lab-like environmnents and with the possibility to generate more data when needed, which is not the case for sewer pipes, since simulating high flow rates in sewage systems is very expensive and thus the data collection depends on the weather. The techniques found in the literature consist of manually investigating the amplitude of certain frequency bands and applying machine learning to obtain data-driven models. Both have their advantages and disadvantages, which were reviewed in this thesis.

One important finding is that when we combine manual and machine learning methods, we combine their strengths. However, while conceptually clear, these mixed methods did not perform much better than just using machine learning. This shows that machine learning works well when the data is prepared carefully.

Also we found that, depending on the situation/phenomenon and data available, certain flow rate ranges can be predicted with better resolution than for other ranges. This turned out to be related to the amount of training data available, but also some resolutions stayed the same for a wide range of available training data, hinting at some intrinsically non-differentiable regions of flow rate values.

The third major finding was that shifting the location decreases the precision substantially. This behavior upon domain shifts suggests that the acoustic flow rate estimation performs best as a permanently installed long term monitoring system calibrated for the location where it is used. It seems to be less suited for flow rate measurements on the fly, due to the performance decrease for varying locations.

It is still to be evaluated if there is a need in the industry for such systems, since Gelsenwasser aimed at utilizing this technique to calibrate their hydraulic simulations. It is also not clear if the acoustic flow rate estimation would indeed be cheaper, since it involves quite some work, to generate the data necessary for calibrating the network. In any case more data, with varying flow rates and varying locations would allow for more precise and reliable answers to these questions.

The investigations showed that it is definitely possible to predict the flow rate in urban sewage systems. It depends on the application if the accuracy is good enough, and on how accessible it is for the operator to generate data, since it is not easily produced. It is part of further investigations to study how much additional knowledge and data is necessary to make the domain shift possible with less degradation of the accuracy. This is also the case for experimental setup for the later applications as well es the data generation process.

This work can be seen as a first step towards developing flow rate estimation algorithms in real world data recorded in sever pipes. The path is free to improve until its predictions improve the precision of hydraulic simulations so communities and cities, that are in danger of flood, are better equipped in heavy rain events.

8 Outlook and Further Work

This section continues to discuss which investigations can be done to further improve the performance.

Increase quality and quantity of data

For any kind of empirical investigation a representative sample of the underlying phenomenon is the basis of all analysis, this is especially true for machine learning techniques. The quality of the data boils down to the experimental setup. Reducing potential sources of error will provide more reliable data. Also developing regimes for quickly setting up the recording boxes in the case of predicted rain events will reduce the probability of missing out the opportunity for collecting valuable data.

Coupling between sensors

In terms of GWs interest to develop a monitoring system that can be installed inside a sewer pipe one of the time and resource intensive tasks is the calibration. We have seen that location shifts drastically degrade the accuracy, so it is not possible right away. One idea that could help the calibration is to allow for some kind of communication between the sensors, to form an ensemble. This would allow for redundancy and alignment of the predicted values. One could set up the system so that the sensors know that an adjacent sensor is located in the same pipe. That way on could estimate a boundary in which the predicted value can be. Defining an estimation of an error rate might also help: A sensor, that was directly calibrated by a commercial sophisticated 'ground truth device' will have a smaller error than a sensor that is far away from any of such ground truth devices and other sensors so compare and double check the prediction. Note that one can also verify the result with previous predictions made by the model, so like a momentum term or a flattening between multiple time steps.

Providing domain knowledge to the network

When more data would be collected for multiple locations, one could also provide some additional information to the neural network. For example how deep the pipe is or the diameter of the pipe. Maybe even some more facts about the circumstances, which might be apparent during the experimentation. This is expected to improve the performance in a transfer learning situation, especially when a relationship can be obtained manually. Observations one might keep an eye on could be how the distance to the ground of the pipe relates to the loudness of the signal and how the diameter of the pipe relates to the location of the main resonance of the signal.

Battery live safer through software

One of the challenges of the experimental setup was the battery lifetime. For now the hardware recorded the entire time between mount to dismount (except of course, when the battery died). One could use a more lightweight and efficient chip that only starts all of its processes and electronically devices (like amplification of the microphone and wireless connection) from a predefined signal. If could be an external signal, or when a certain threshold of loudness is reached. This is however not suggested to do as a first step to increase the battery lifetime. It is likely that investing in a stronger battery will have a larger effect. As discussed in the experimental Section, the zero crossing rate could also provide information about, when to start the entire measuring machinery.

Auto-gain and Normalization

From the beginning we discovered the auto-gain performed by the sensor. In the literature one can find the reason and implementation behind such techniques [21]. It seems reasonable for our task, however it modifies the data and is something a data-driven model has to learn about as well. One advantage of auto-gain however is that one does no have to control it manually. One way of dealing with this is to read out the chosen value for the auto gain and provide it to the network.

Exploration of Ensemble Techniques for Model Combination

In the course of this study, the use of a custom `sklearn` model was employed. Sequential models were engaged based on the output of the preceding one. However, an inherent limitation was identified: In cases of false positives from the initial model, subsequent models were not utilized for further predictions. To address this limitation and enhance predictive accuracy, further exploration into ensemble techniques is recommended. These techniques aim to combine multiple models to achieve superior performance. Potential methods for consideration include:

1. **Voting Classifier**: Implementing a Voting classifier to aggregate predictions from multiple models, either through majority voting or by averaging probabilities [4].
2. **Stacking**: Exploring the use of a meta-model trained on predictions of base models as features [37].
3. **Bagging and Boosting**: Investigating Bagging (e.g., Random Forest) and Boosting (e.g. Catboost, AdaBoost, Gradient Boosting) techniques to create diverse model ensembles [39].

Experimentation with these ensemble techniques aims to mitigate false positives, increase model robustness, and potentially improve overall predictive performance. Furthermore, assessing the diversity among models is essential to prevent correlated errors and maximize the effectiveness of ensemble methods.

9 Appendix

9.1 Sklearn default values

9.1.1 Default values of RandomForestClassifier in scikit-learn

- `n_estimators`: 100 (*The number of trees in the forest*)
- `criterion`: "gini" (*Function to measure split quality*)
- `max_depth`: None (*Maximum depth of trees*)
- `min_samples_split`: 2 (*Minimum samples required to split an internal node*)
- `min_samples_leaf`: 1 (*Minimum samples required to be at a leaf node*)
- `min_weight_fraction_leaf`: 0.0 (*Minimum weighted fraction of samples required to be at a leaf node*)
- `max_features`: "sqrt" (*Number of features to consider at each split*)
- `max_leaf_nodes`: None (*Grow trees with max leaf nodes in best-first fashion*)
- `min_impurity_decrease`: 0.0 (*Minimum impurity decrease for node split*)
- `bootstrap`: True (*Whether to use bootstrap samples*)
- `oob_score`: False (*Whether to use out-of-bag samples to estimate generalization score*)
- `n_jobs`: None (*Number of parallel jobs*)
- `random_state`: None (*Controls randomness during tree building*)
- `verbose`: 0 (*Controls verbosity*)
- `warm_start`: False (*Whether to reuse solution of previous fit*)
- `class_weight`: None (*Weights associated with classes*)
- `ccp_alpha`: 0.0 (*Complexity parameter for Minimal Cost-Complexity Pruning*)
- `max_samples`: None (*Number of samples to draw for each base estimator*)

9.1.2 Default values of LogisticRegression in scikit-learn

- `penalty`: 'l2' (*Specify the norm of the penalty*)
- `dual`: False (*Dual or primal formulation*)
- `tol`: 1×10^{-4} (*Tolerance for stopping criteria*)
- `C`: 1.0 (*Inverse of regularization strength*)
- `fit_intercept`: True (*Specifies if a constant should be added to the decision function*)
- `intercept_scaling`: 1 (*Useful for 'liblinear' solver with intercept*)
- `class_weight`: None (*Weights associated with classes*)
- `random_state`: None (*Used when certain solvers are employed*)
- `solver`: 'lbfgs' (*Algorithm for the optimization problem*)
- `max_iter`: 100 (*Maximum number of iterations for solvers to converge*)
- `multi_class`: 'auto' (*Strategy for handling multiclass problems*)
- `verbose`: 0 (*Verbosity for certain solvers*)
- `warm_start`: False (*Whether to reuse the previous fit solution*)
- `n_jobs`: None (*Number of CPU cores used for parallelizing*)
- `l1_ratio`: None (*Elastic-Net mixing parameter*)

9.2 Images

The rest of the appendix is used as a collection of all the images that did not fit into the text, so they are located in the end of the thesis. One can find them right after the list of references in the very last pages of this work.

List of Figures

List of Tables

References

[1] *10th IEEE International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing, HPCC/EUC 2013, Zhangjiajie, China, November 13-15, 2013*. IEEE, 2013. ISBN: 978-0-7695-5088-6. URL: https://ieeexplore.ieee.org/xpl/conhome/6823827/proceeding.

[2] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[3] Balamurali B T et al. "Acoustic prediction of flowrate: varying liquid jet stream onto a free surface". In: (June 2020).

[4] Eric Bauer and Ron Kohavi. "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants". In: *Machine Learning* vv (1998). Ed. by Philip Chan, Salvatore Stolfo, and David Wolpert. c© 1998 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands, pp. 1–38.

[5] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

[6] Carlos Alexis Bonilla Granados, Jarol Derley Ramón Valencia, and Diego Ivan Sanchez Tapiero. "Monitorización y análisis de flujo en sistemas de alcantarillado sanitario en Cúcuta, Colombia". In: *Respuestas* 26.2 (2021), 6–13. DOI: 10.22463/0122820X.3207. URL: https://revistas.ufps.edu.co/index.php/respuestas/article/view/3207.

[7] S. Dubnov. "Generalization of spectral flatness measure for non-Gaussian linear processes". In: *IEEE Signal Processing Letters* 11.8 (2004), pp. 698–701. DOI: 10.1109/LSP.2004.831663.

[8] Khalid Elgazzar et al. "Revisiting the internet of things: New trends, opportunities and grand challenges". In: *Frontiers in the Internet of Things* 1 (2022). ISSN: 2813-3110. DOI: 10.3389/friot.2022.1073780. URL: https://www.frontiersin.org/articles/10.3389/friot.2022.1073780.

[9] Robert P. Evans, Jonathan D. Blotter, and Alan G. Stephens. "Flow Rate Measurements Using Flow-Induced Pipe Vibration ". In: *Journal of Fluids Engineering* 126.2 (May 2004), pp. 280–285. ISSN: 0098-2202. DOI: 10.1115/1.1667882. eprint: https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/126/2/280/5542071/280_1.pdf. URL: https://doi.org/10.1115/1.1667882.

[10] James Fogarty, Carolyn Au, and Scott Hudson. "Sensing from the basement: A feasibility study of unobtrusive and low-cost home activity recognition". In: Oct. 2006, pp. 91–100. DOI: 10.1145/1166253.1166269.

[11] *Gelsenwasser, Private Communication on Recording Hardware*. 2023.

[12] *General Anzeiger - Flutkatastrophe in NRW und Rheinland-Pfalz*. Accessed on December 29, 2023. 2021. URL: https://ga.de/thema/flutkatastrophe-in-nrw-und-rheinland-.

[13] Theodoros Giannakopoulos and Aggelos Pikrakis. *Introduction to Audio Analysis: A MATLAB Approach*. 1st. Various: Academic Press (Elsevier), 2014. ISBN: 978-0-08-099388-1.

[14] Fabien Gouyon, Francois Pachet, and Olivier Delerue. "On the Use of Zero-Crossing Rate for an Application of Classification of Percussive Sounds". In: (Aug. 2002).

[15] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[16] Jacobs He et al. "Correlating Sound and Flow Rate at a Tap". In: *Procedia Engineering* 119 (2015). Computing and Control for the Water Industry (CCWI2015) Sharing the best practice in water management, pp. 864–873. ISSN: 1877-7058. DOI: https://doi.org/10.1016/j.proeng.2015.08.953. URL: https://www.sciencedirect.com/science/article/pii/S1877705815026235.

[17] Tin Kam Ho. "Random Decision Forests". In: *Proceedings of the 3rd International Conference on Document Analysis and Recognition*. Archived from the original (PDF) on 17 April 2016. Retrieved 5 June 2016. Montreal, QC, 1995, pp. 278–282. URL: https://web.archive.org/web/20160417030218/http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf.

[18] John D Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in science & engineering* 9.3 (2007), pp. 90–95.

[19] Johannes Schmidt. "Audio Lab Report: Anomaly Detection for Acoustic Sensor Data". en. In: (2022). URL: https://jocho.de/media/Audio_Lab_report.pdf.

[20] James D. Johnston. "Transform coding of audio signals using perceptual noise criteria". In: *IEEE J. Sel. Areas Commun.* 6 (1988), pp. 314–323. URL: https://api.semanticscholar.org/CorpusID:5999699.

[21] George S. Kang and Mark L. Lidd. "Automatic Gain Control". In: *Naval Research Laboratory Technical Report* (Year of Publication).

[22] Anssi Klapuri and Manuel Davy. *Signal Processing Methods for Music Transcription*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387306676.

[23] Thomas Kluyver et al. "Jupyter Notebooks – a publishing format for reproducible computational workflows". In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Ed. by F. Loizides and B. Schmidt. IOS Press. 2016, pp. 87–90.

[24] Jan Krhut et al. "Comparison between uroflowmetry and sonouroflowmetry in recording of urinary flow in healthy men: Sonoflowmetry in men". In: *International Journal of Urology* 22 (May 2015). DOI: 10.1111/iju.12796.

[25] Paul Kubesh. *One Method to Identify Inflow and Infiltration (I/I): Flow Monitoring*. https://www.sehinc.com/news/one-method-identify-inflow-and-infiltration-ii-flow-monitoring. [Online; accessed 21-June-2023].

[26] Alexander Lerch. *An Introduction to Audio Content Analysis: Music Information Retrieval Tasks & Applications*. Second. Hoboken, New Jersey: John Wiley & Sons, Inc., 2023. ISBN: 9781119890942 (cloth), 9781119890966 (adobe pdf), 9781119890973 (epub).

[27] librosa development team. *librosa.piptrack — librosa 1.11.0 documentation*. URL: https://librosa.org/doc/main/generated/librosa.piptrack.html (visited on 10/21/2023).

[28] Librosa Documentation. *librosa.feature.melspectrogram*. Accessed on 3 November 2023. 2023. URL: https://librosa.org/doc/main/generated/librosa.feature.melspectrogram.html.

[29] Iamir Masoud. *Understanding Micro, Macro, and Weighted Averages for scikit-learn Metrics in Multi-Class Classification with Example*. 2022. URL: http://iamirmasoud.com/2022/06/19/understanding-micro-macro-and-weighted-averages-for-scikit-learn-metrics-in-multi-class-classification-with-example/ (visited on 10/27/2023).

[30] Brian McFee et al. *LibROSA: A Python package for music and audio analysis*. https://librosa.org/doc/main/generated/librosa.feature.melspectrogram.html. Accessed on: Date Accessed. Year Accessed.

[31] Brian McFee et al. "librosa: Audio and music signal analysis in python". In: *Proceedings of the 14th Python in Science Conference*. 2015, pp. 18–25.

[32] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56 –61. DOI: 10.25080/Majora-92bf1922-00a.

[33] Meinard Müller. *Fundamentals of Music Processing, Second Edition*. Erlangen, Germany: Springer Nature Switzerland AG, 2021. ISBN: 978-3-030-69807-2. DOI: 10.1007/978-3-030-69808-9.

[34] *NIST/SEMATECH e-Handbook of Statistical Methods*. http://www.itl.nist.gov/div898/handbook/. A significant update was made to the Handbook in April 2012. Printer friendly versions of each chapter can be found at http://www.itl.nist.gov/div898/handbook/. Feedback on the Handbook can be sent to handbook@nist.gov. DOI: 10.18434/M32189.

[35] Daniel S. Park et al. *SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition*. Sept. 2019. DOI: 10.21437/interspeech.2019-2680. URL: http://dx.doi.org/10.21437/Interspeech.2019-2680.

[36] Emanuel Parzen. “On Estimation of a Probability Density Function and Mode”. In: *The Annals of Mathematical Statistics* 33.3 (1962), pp. 1065–1076. ISSN: 0003-4851. DOI: 10.1214/aoms/1177704472. URL: https://doi.org/10.1214/aoms/1177704472.

[37] Bohdan Pavlyshenko. “Using Stacking Approaches for Machine Learning Models”. In: (2018), pp. 255–258. DOI: 10.1109/DSMP.2018.8478522.

[38] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python”. In: *Journal of machine learning research* 12.Oct (2011), pp. 2825–2830.

[39] Liudmila Prokhorenkova et al. *CatBoost: unbiased boosting with categorical features*. 2019. arXiv: 1706.09516 [cs.LG].

[40] Ruben Ruiz-Gonzalez et al. “An acoustic method for flow rate estimation in agricultural sprayer nozzles”. In: *Computers and Electronics in Agriculture* 141 (2017), pp. 255–266. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2017.08.003. URL: https://www.sciencedirect.com/science/article/pii/S0168169917306282.

[41] “Sanitary Sewer Flow Monitoring and Data Analytics”. en. In: (2019). URL: https://www.wef.org/globalassets/assets-wef/direct-download-library/public/03---resources/wsec-2019-fs-011---csc---flow-monitoring-and-data-analytics---final.pdf.

[42] *PARSHL: A Program for the Analysis/Synthesis of Inharmonic Sounds Based on a Sinusoidal Representation*. STAN-M-43. Champaign-Urbana, Illinois, 1987. URL: https://ccrma.stanford.edu/files/papers/stanm43.pdf.

[43] Derya Soydaner. “Attention mechanism in neural networks: where it comes and where it goes”. In: *Neural Computing and Applications* 34.16 (2022), pp. 13371–13385. DOI: 10.1007/s00521-022-07366-3. URL: https://doi.org/10.1007%2Fs00521-022-07366-3.

[44] A. Spanias, T. Painter, and V. Atti. *Audio Signal Processing and Coding*. Wiley, 2006. ISBN: 9780470041963. URL: https://books.google.de/books?id=Z_z-0QbadPIC.

[45] Stephen V. Stehman. “Selecting and interpreting measures of thematic classification accuracy”. In: *Remote Sensing of Environment* 62.1 (1997), pp. 77–89. DOI: 10.1016/S0034-4257(97)00083-7.

[46] Ahmed A. Taha and Allan Hanbury. “Metrics for Evaluating 3D Medical Image Segmentation: Analysis, Selection, and Tool”. In: *BMC Medical Imaging* 15 (2015), p. 29. DOI: 10.1186/s12880-015-0068-x.

[47] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[48] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL].

[49] Warrakkk. *White Noise Spectrum*. Own work. 2012. URL: https://commons.wikimedia.org/wiki/File:White_noise_spectrum.svg.

[50] Michael Waskom et al. *mwaskom/seaborn: v0.8.1 (September 2017)*. Version v0.8.1. Sept. 2017. DOI: 10.5281/zenodo.883859. URL: https://doi.org/10.5281/zenodo.883859.

[51] Tong Yu and Hong Zhu. *Hyper-Parameter Optimization: A Review of Algorithms and Applications*. 2020. arXiv: 2003.05689 [cs.LG].

[52] Hongyi Zhang et al. *mixup: Beyond Empirical Risk Minimization*. 2018. arXiv: 1710.09412 [cs.LG].

[53] Slobodan Đukanović, Jiri Matas, and Tuomas Virtanen. "Acoustic Vehicle Speed Estimation From Single Sensor Measurements". In: *IEEE Sensors Journal* PP (Sept. 2021), pp. 1–1. DOI: 10.1109/JSEN.2021.3110009.

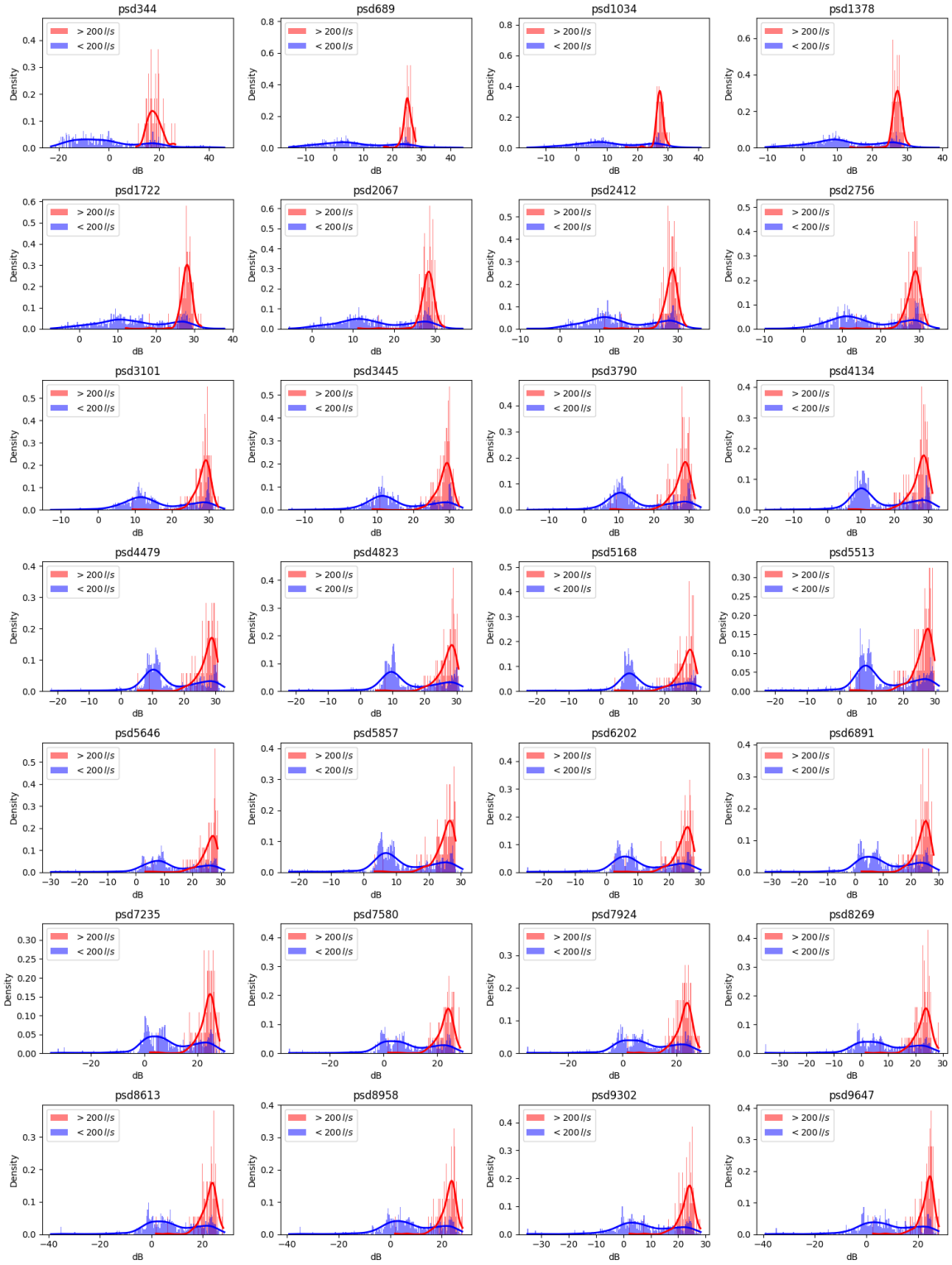

Figure 9.1: This figure is similar to figure 5.7 but this time there is no flattening. So every frequency is depicted by its own.

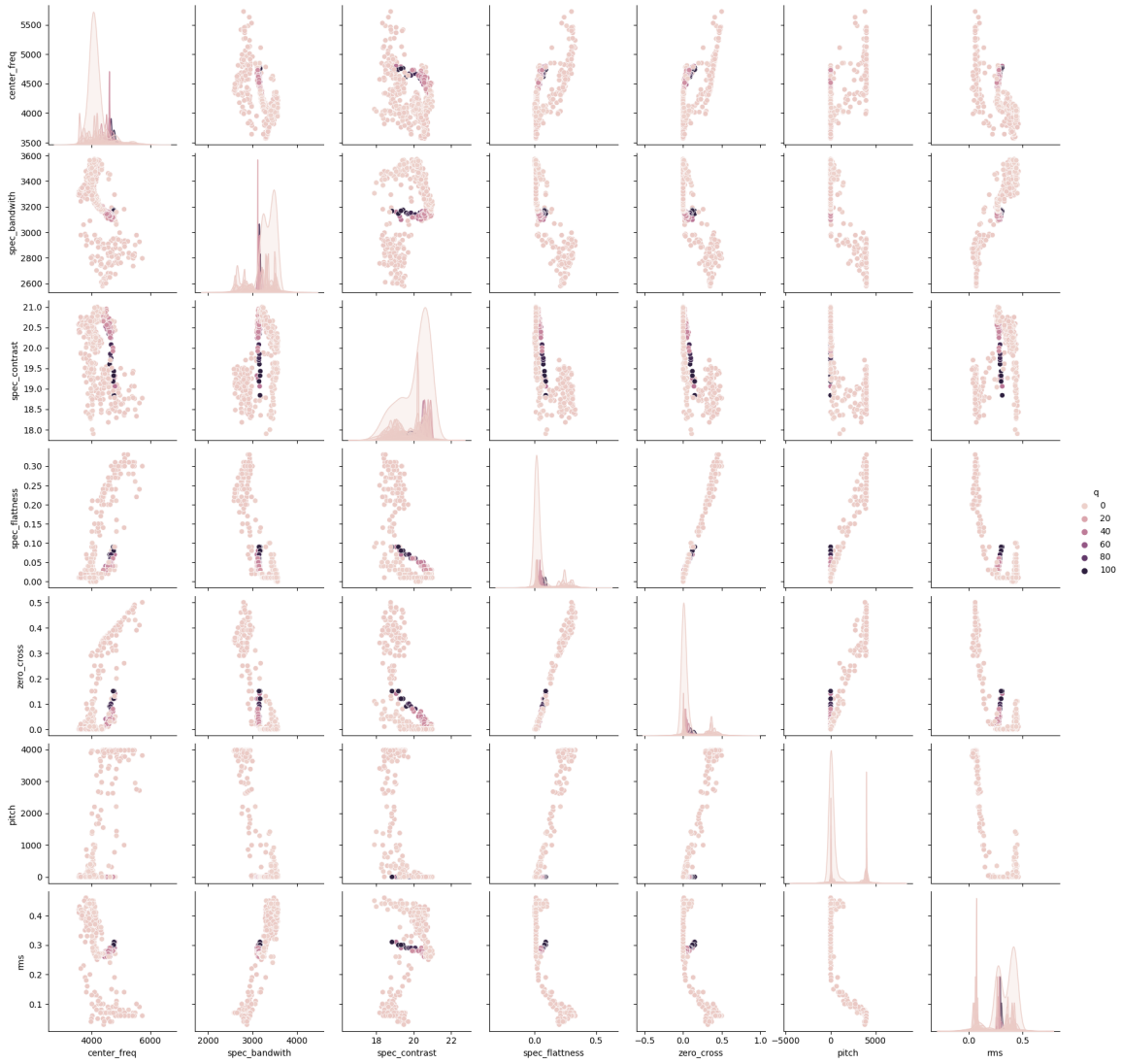

Figure 9.2: Scatter plot matrix for the first event. It depicts all the features of table 4.1.

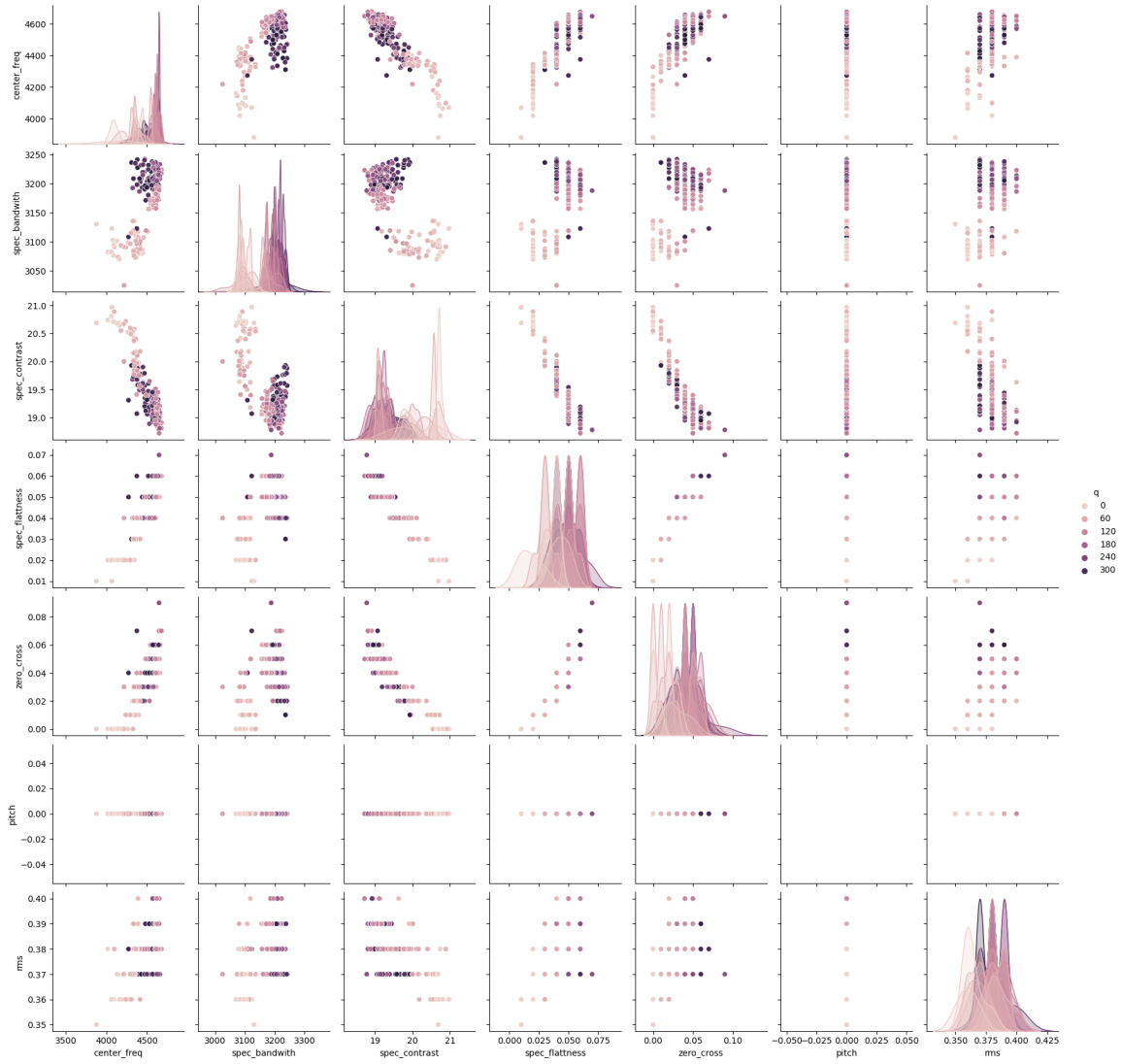

Figure 9.3: Scatter plot matrix for the second event. It depicts all the features of table 4.1.

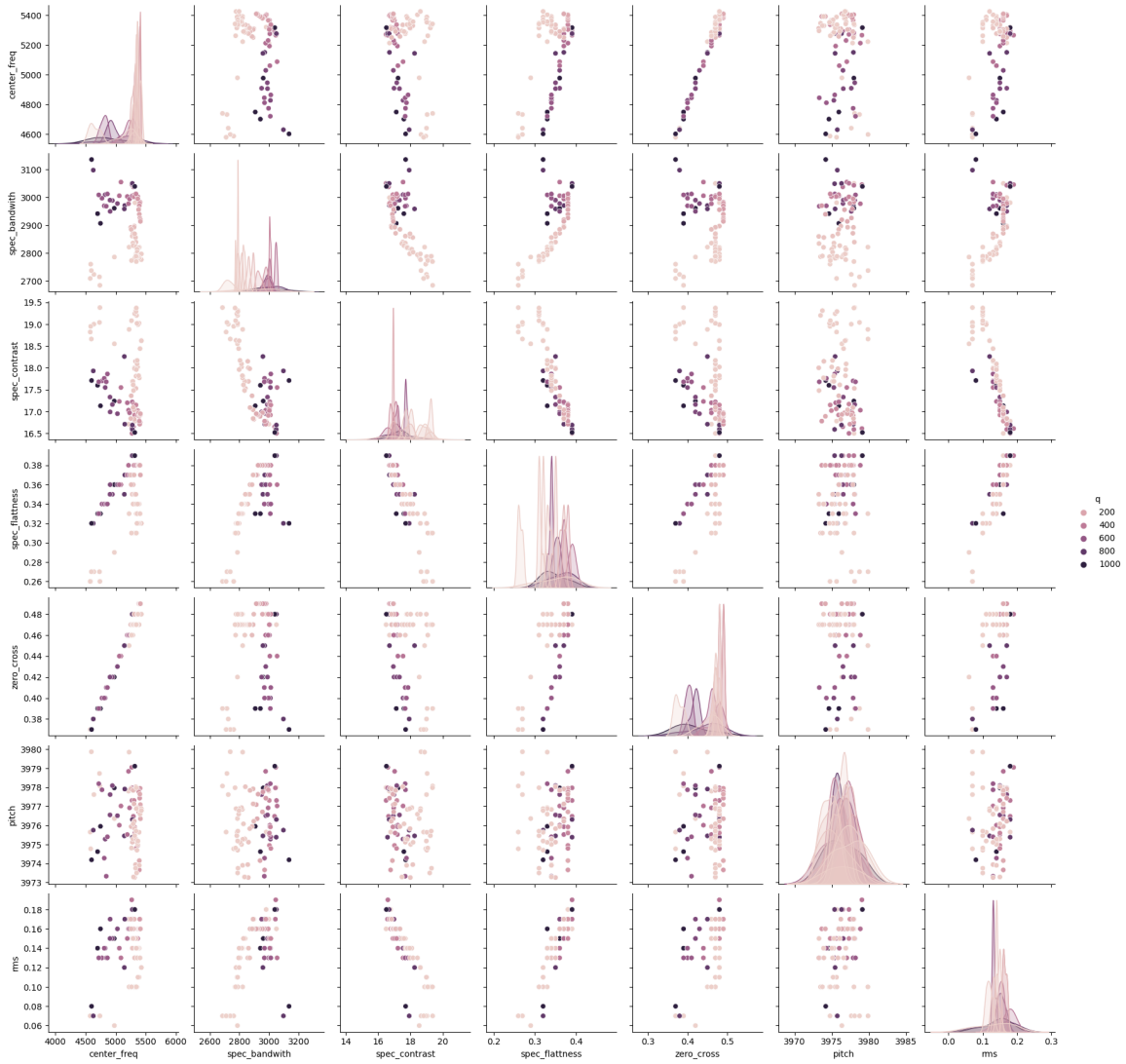

Figure 9.4: Scatter plot matrix for the third event. It depicts all the features of table 4.1.

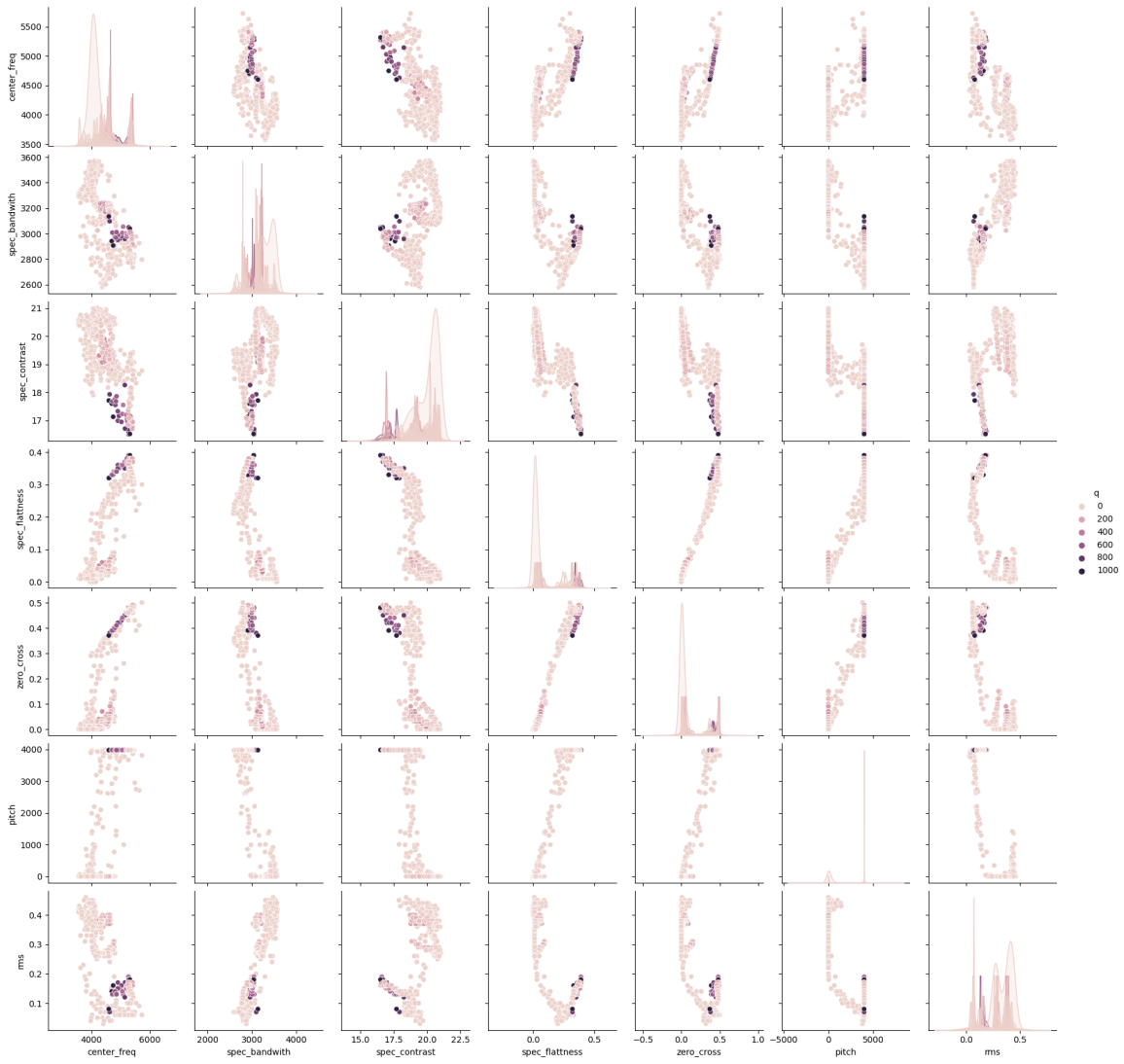

Figure 9.5: Scatter plot matrix for all three events in every plot. It depicts all the features of table 4.1.