NTNU
Norwegian University of
Science and Technology

Sensing Application System with Microphones

Edvard Rønningen Graarud

Master of Science in Communication Technology
Submission date: June 2018
Supervisor: Frank Alexander Krämer, IIK

Norwegian University of Science and Technology
Department of Information Security and Communication Technology

Problem description

With the Internet of Things being one of the most anticipated technological advancements, the amount of data being collected by sensors is increasing, and making useful applications based on the information we can obtain is an interesting prospect. Microphones are one type of hardware that is used by sensor chips to collect information, in the form of audio. However, compared to many other types of information, sound is harder to understand without human interpreters. Machine learning presents a way to deal with this, where we can train a machine to interpret the sound input, based on the learning process it has gone through.

The combination of these technologies could allow for sensing systems based on microphones, given that the weak sensors can provide data that is good enough to use for both training a good classifier and classifying new input. As these sensors do not currently contain enough resources in terms of CPU, memory and power to do classification of audio on the sensor itself, one option is to send the raw audio files to a more powerful central unit.

Raw audio files are traditionally sampled with a high sampling rate and lasts for several seconds at the least. This leads to large audio files, which is very undesirable for weak sensors. For that reason, this project will investigate using sound clips that are far below the standard used for human listeners, as the input for training and testing classifiers.

The sound clips with reduced dimensions will be considered together with a suitable data transfer protocol designed for IoT, in order to evaluate the usability of the raw audio files in a realistic IoT-setting.

Abstract

The Internet of Things is one of the most promising fields of technological advancements. Through networks of sensors, we can obtain information about the environment around the sensor and use it for various purposes. In this project, the usage of microphones as a sensor is explored, by utilizing machine learning to classify sound clips collected in a lecture hall. To make raw sound clips usable by weak sensors, a downscaling of duration and sample rate of the audio clips are done. The downscaling is measured against the classification accuracy and Matthews Correlation Coefficient and show that the audio clips used in this project could by downscaled by a factor of 197, without losing classification accuracy/MCC. Based on the results of the downscaling tests, an estimation of energy consumption using Bluetooth Low Energy to transfer the audio files is presented, and the estimation result give a sensor lifetime that the author considers viable for a sensor network. Lastly, some approaches to adaptive sensing is briefly discussed, as a function to increase the reactiveness and data value.

In the end, the results are considered mostly as a proof-of-concept for using downscaled audio files in an IoT-setting with weak sensor nodes. Even though these tests are done with a specific binary classification-based use case, and performed in a specific lecture hall, the results seem promising for other applications.

Sammendrag

Tingenes Internett er en av de mest lovende teknologiene i dagens marked. Gjennom sensornettverk kan vi samle informasjon fra miljøet rundt sensorene og bruke det til forskjellige formål. I dette prosjektet blir mikrofoner som sensorer utforsket, ved å bruke maskinlæring til å klassifisere lydklipp fra en forelesningssal. For å gjøre rå lydklipp håndterbare for svake sensorer, er det blitt gjort en nedskalering av varighet og punktprøvefrekvens. Nedskaleringen er målt mot klassifiseringstreffsikkerhet og Matthews Correlation Coefficient og viser at lydklippene i dette prosjektet kunne bli nedskalert med en faktor på 197, uten å miste treffsikkerhet/MCC. Basert på resultatene fra nedskaleringen ble det gjort en estimering av energiforbruk ved bruk av Bluetooth Low Energy for filoverføring, og estimatet ga en sensorlivstid som forfatteren vurderte til å være brukbar for et reelt sensornettverk. Etter dette blir noen metoder for adaptiv føling kort presentert og diskutert, som en funksjon for å øke reaktivitet og dataverdi.

Resultatene blir til slutt vurdert mest som et bevis på konsept for bruk av nedskalerte lydklipp i en IoT-situasjon med svake sensornoder. Selv om dette prosjektet har jobbet ut i fra et binært klassifiseringsproblem i en bestemt forelesningssal, viser resultatene at metoden virker lovende for andre applikasjoner.

Table of contents

Table of abbreviations

BLE	Bluetooth Low Energy
CPU	Central processing unit
DCASE	Detection and Classification of Acoustic Scenes and Events
FFT	Fast Fourier Transform
GRU	Gated Recurrent Unit
IMFCC	Inverted mel-frequency cepstral coefficient
IoT	Internet of Things
MCC	Matthews Correlation Coefficient
MFCC	Mel-frequency cepstral coefficient
MLP	Multilayer perceptron

List of figures

List of tables

List of equations

1. Introduction and Motivation

This project combines two of the most promising technological fields right now; Internet of Things and machine learning. Collection of data from simple sensors is a core functionality in IoT, and machine learning can provide ways of analysing this data and use it for desired purposes. There is a wide array of sensors to choose from, but we will focus on audio data gathered from microphones.

Machine learning has proven to be a very powerful problem-solving tool and has shown that it can deliver results on a level humans just can't. A good example of this can be found in chess. The chess game has existed for hundreds of years, and has been studied to great depths, and powerful computer programs has been developed to master the game. The most powerful chess engine today is called Stockfish (CCRL 40/4 Rating list, 2018) and is built as a traditional chess engine with a brute-force approach to finding the best moves (Stockfish community, 2018). The engine is very powerful, and much higher rated than any human player (FIDE, 2018). Despite this, a machine learning program called AlphaZero outperformed Stockfish after just 4 hours of training, with no other input than the rules of chess (Silver, et al., 2017).

One interesting aspect of audio data in IoT is that sound is cheap and easy to obtain, as microphones can be bought very cheaply, and in small sizes. Microphones can be especially interesting in situations or locations where other sensors are unfitting or unwanted, such as cameras in a bathroom. In many scenarios it will be beneficial and natural to use more than one type of sensor, such as a microphone and a movement sensor together to detect human presence, but we will mainly focus on audio data alone.

Another interesting aspect is the complexity of audio data compared to other types of sensor gathered data. Several other types of data, such as light level, CO2 level or temperature can be easily represented by a number, which makes it a lot easier to handle and very cheap to deal with. This is the ideal situation for sensors as they don't want to store or transmit large files, because it is taxing on their limited power and memory.

Using machines to classify audio data is not a new concept at all and is being used in various applications. One typical use case is speech recognition, where you can give commands to a machine by vocal input, for example your phone (Siri, Google Assistant, etc). Although speech recognition is not this project's area of focus, it is interesting to see that the leading

global tech companies are invested in using information in sound for some of their core applications.

1.1 The problem to be solved

Raw audio files are typically sampled with a high sampling rate and lasts for several seconds at the least. This leads to large audio files, which is very undesirable for weak sensors. For that reason, this project will investigate using sound clips that are far below the standard used for human listeners, as the input for training and testing classifiers. The idea is that if we can reduce the sound clips down to a level where the sensors are able to handle them, without ruining the classification process, there is feasibility in using audio classification in an IoT-setting. This means we have to investigate a system that is limited by typical IoT-restrictions, and see if this system would be able to operate at a satisfying level with low-level audio clips.

1.2 Use Case: A System to Monitor Lecture Hall Usage

In order to research the potential of audio data used in a cheaply operating sensor system utilizing machine learning, we want to see if we can get such a system to operate cheaply while maintaining satisfactory classification results. To do so, we will take base in a single use case, and work with that use case to obtain concrete results that can be used to analyse the performance on different system configurations.

The use case we have chosen to test the feasibility of a cheaply operating sensing system using microphones, is to try to identify if there is activity in a lecture hall, only by analysing the sound from it. The reason for choosing this is simply because of convenient collection of audio data, and there is only a selected number of activities that occur regularly in a lecture hall, judging from personal experience. We also believe this use case is realistic enough to be used as a test scenario.

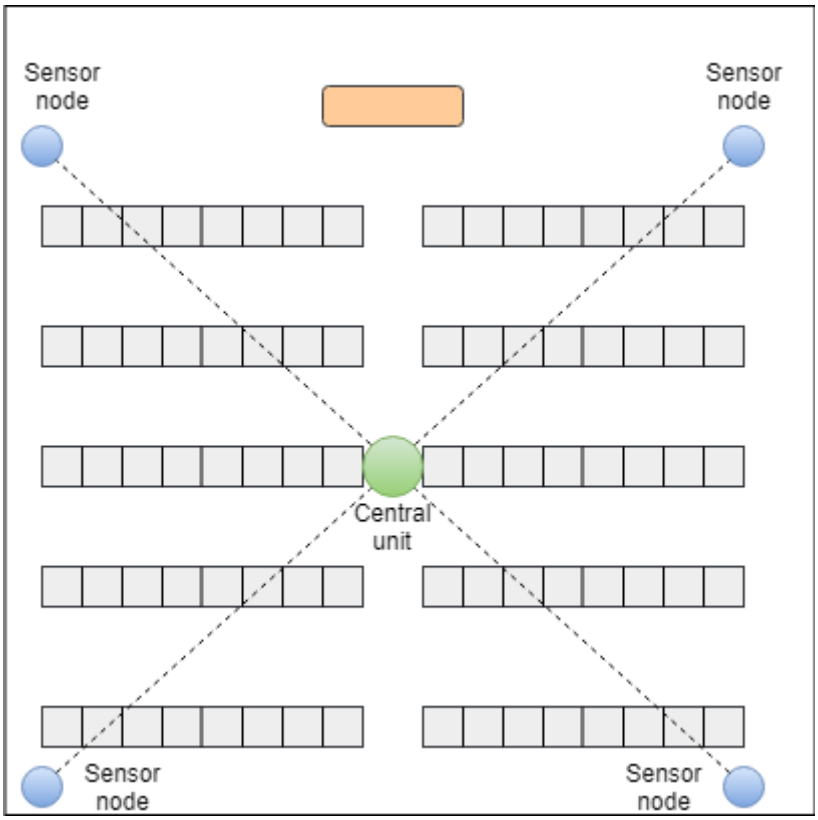

Figure 1: Abstraction of a sensor system in a lecture hall

Figure 1 shows an abstraction of how we picture the system. Simply put, we have some sensors that record sound clips and send them or some information about them, to the central unit. This central unit can be a more powerful computer, and can either run classification on itself, or be responsible of sending the data over the internet to some cloud server that does the classification process. Either way, the central unit should be the one controlling the system, so that is where the logic of the system is located.

1.3 Report Summary

The initial parts of this report present some background and related work (section 2), methodology used in the project (section 3), basic sensor functionality and at what points energy can be saved (section 4), and the machine learning solution that are used in this project (section 5).

The biggest and most time-consuming part of this project is presented in section 6, where we have performed a series of classification processes on increasingly lower quality sound clips

(reduction in duration and sample rate) and have investigated how the classifier reacted to this, presented by accuracy and MCC, compared to file size reduction. The results seemed very promising for usability in a sensor network, where the file size was reduced by a factor of 197 without a loss in accuracy/MCC, and only 4% accuracy loss/0.08 MCC loss when reduced by a factor of 391.

In section 7, several IoT communication protocols are investigated, and BLE is selected as the optimal one for our system. Some calculations of estimated energy consumption are made, and it shows that a modern sensor chip utilizing BLE and the reduced files could theoretically operate for several years.

Section 8 looks at adaptive sensing, and what other researches has achieved within that area. An idea for an algorithm that adepts the duty cycle based on observations is presented and adapting on battery level is briefly presented.

Finally, in section 9 some concluding remarks are given, and suggestions for future work is presented in section 10.

2. Background and Related Work

A key question is how to analyse and classify audio data. A modern audio file typically has a high sampling rate that contains thousands of data points per second and takes up a few megabytes of memory. This led us to research what previous work has been done on classifying audio data, and we found that feature extraction was a commonly used concept for this.

Features as classifying values on audio data is not a new concept. Over 20 years ago features were used in a distance-based system to classify a new sound after having trained their model (Wold, et al., 1996). This system was mainly used to search for and retrieve similar sounds to input sounds in a database. They successfully described different types of sound using a set of features. Later work also proved the feasibility of such an approach where features are used for classification in a large collection of audio data (Mierswa and Morik, 2005).

With the knowledge that features should be used to represent audio data, the next challenge is to decide what features to use. It seems natural that the selection of features is detrimental to get a good representation of the audio and get good classification results. The problem is selecting the set of features that suits your specific problem.

2.1 Acoustic Scene Classification

The goal of the sensing system in this project is to classify activity around the sensor, which is a lecture hall in our use case. That means the features needs to be good at representing an acoustic audio clip, which is not necessarily the same as features for other types of audio (e.g. speech recognition). Classifying what a person is saying is totally irrelevant in this setting, but sensing that a person is speaking, regardless of what words are spoken, is very relevant. This type of audio classification has its own “category”, called acoustic scene classification.

The field of acoustic scene classification has seen a lot of research in the last years, especially through the Detection and Classification of Acoustic Scenes and Event challenges. The 2017 edition of DCASE had a challenge based on acoustic scene classification, with training data already provided (Mesaros and Heittola, 2017). Participants were set to develop a system that outperforms a baseline system. Most of the background on acoustic scene classification in this project is taken from the research that was done in for this challenge, as it is some of the most modern research in the field.

Choosing exactly what features to use for a machine learning problem is far from a trivial task. Selecting the best features for a problem often requires a lot of work and is very impactful on the result of the classifier. Different type of features has been explored in the DCASE 2017 challenge in several of the papers, such as combining MFCCs (Mel Frequency Cepstral Coefficients) with IMFCCs (Inverted Mel Frequency Cepstral Coefficients) (Chandrasekhar and Gangashetty, 2017), different types of mel-spectrograms in combination (Park, et al., 2017), or extracting features from a fusion between standard spectrograms (spectrograms directly from the Short Time Fourier Transform) and Constant-Q-Transform spectrograms (Weiping, et al., 2017).

One of the most commonly used features for this type of task are mel-band energies and features derived by them, according to the developers of AuDeep (Amiriparian, et al., 2017). MFCCs are one type of features derived from these mel-band energies. They further state that manual feature selection can be a hard and demanding task, and that unsupervised representation learning has gained popularity as a substitute to conventional feature sets. They especially mention representation learning with deep neural networks. To aid their task of acoustic scene recognition they developed a recurrent sequence-to-sequence autoencoder for unsupervised representation learning.

2.2 Sensor Networks Used for Audio Classification

Salomons and Havinga has done a survey on the feasibility of sound classification on wireless sensor nodes and does mention that some classes of sound classification could be done with a lower sampling frequency than what is normally used for sound recordings. They also conclude that when using MFCC features, the calculation costs are too heavy for small sensor nodes, but that Haar-like features could be feasible to calculate on a weak sensor. However, they don't consider the payload of learning algorithms (Salomons and Havinga, 2015).

Within the field of assisted ambient living systems there has been proposed a solution for a low cost wireless acoustic sensor network, where acoustic audio data is classified based on their audio fingerprint and Hamming distance to the closest fingerprint neighbour. Their end results did prove that it is feasible to realize an acoustic sound sensing system, where an audio fingerprint can be calculated before the data is sent to the classifying system. Compared to the classification system we have used in this project, their system used audio clips of higher

quality than what we found necessary for strong classification results and did not discuss using low power communication protocols (Quintana-Suárez, et al., 2017).

While searching for related work and previous results reducing duration or sample rate of audio clips, there was very little work to be found that could guide our parameters. This means we had no knowledge about how classifiers would react to low duration/low sample rate audio clips. The choice of methodology had to have this in mind, so that the process is suitable for solving a problem we had little knowledge about.

3. Methodology

Approaching the problem of optimizing the parameters of the audio clips needed to be done systematically. The book of Design Science Methodology for Information Systems and Software Engineering (Wieringa, 2014a) presents a way of designing a treatment for a problem that we found fitting for this project. A treatment is way to solve a problem, so designing a treatment essentially means designing a way to solve a problem. Naturally, finding a good treatment is not a necessarily a process where you find the optimal solution immediately, but rather through a looped process.

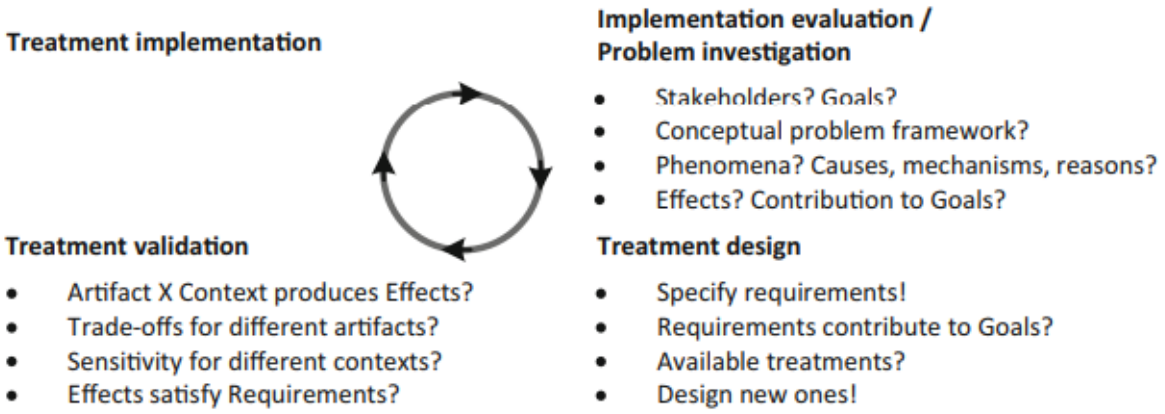

Figure 2: The Engineering Cycle, taken from (Wieringa, 2014a)

Wieringa introduces the loop in Figure 2, called the Engineering Cycle, and describes a looped process consisting of four phases. It starts of at problem investigation, where a phenomena that must be improved is defined. Moving into treatment design, we design one or more potential treatments for the problem. This treatment is then validated in the treatment validation phase, before treatment implementation happens. Lastly, we move to the implementation evaluation, where we evaluate the success of the treatment, and this is where a new iteration of the cycle could follow.

3.1 Relating the Engineering Cycle to This Project

Entering the loop, we first did some problem investigation, and found that modern audio files are difficult and expensive to handle for weak sensors. Looking at a potential chip for a system like this, the nRF52840 from Nordic Semiconductor (Nordic Semiconductor, no date),

it has 1MB flash memory onboard. Comparing this to audio files sampled at 44.1 kHz, a normal modern sampling rate, the sensor would run out of memory after recording between 5 and 6 seconds of audio, assuming the entire memory is available for audio data.

Power consumption is also a major issue with weak sensors, and data transfers are a power-demanding operation. Because of this, sensors typically don't want to transfer large data files, because it involves sending a large amount of packets, and that represent a large power drain.

Next step was to design a treatment to this problem. The idea is that if we need to reduce the size of the audio files. This would be a treatment to the memory problem and could make sending the raw audio file a viable solution. From how sound is recorded and stored on digital devices, it can be represented as a series of data points on a time axis. That means we can either reduce the number of points, or the duration in order to reduce the size.

The important relation to look at is how the classification process reacts to the change in audio file size. This relation is what defines the treatment as good or bad, and if we need to continue working on a treatment. This whole process can be seen in chapter 5, where the process is performed in this looped sequence.

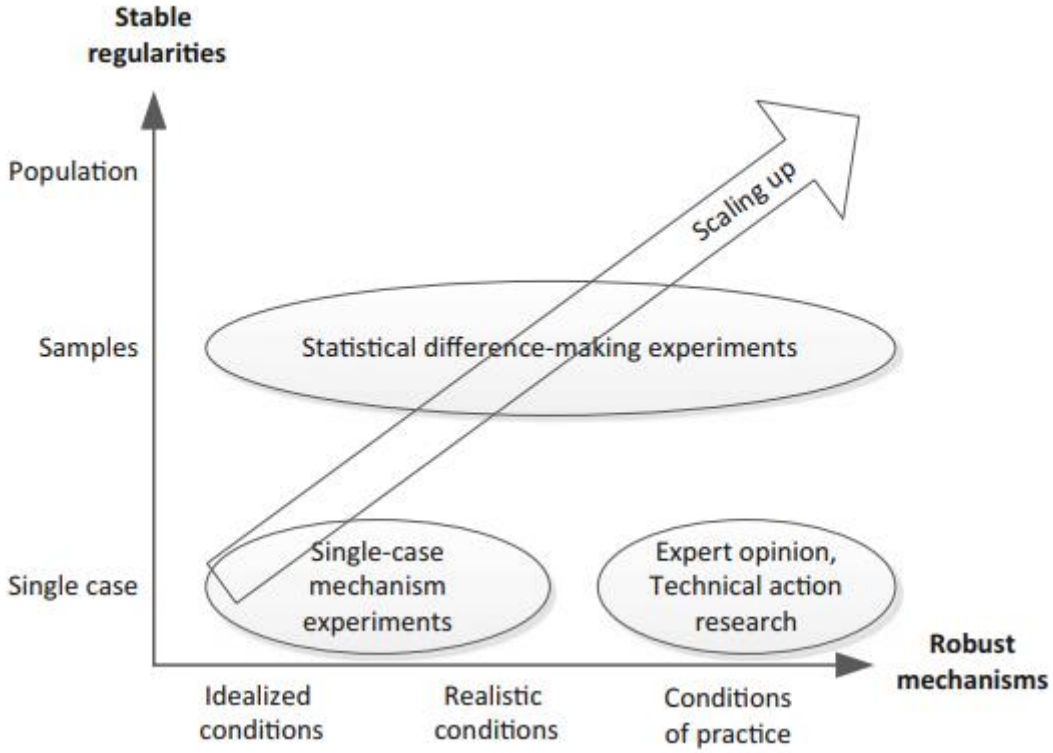

Figure 3: Scaling up, taken from (Wieringa, 2014b)

When developing and testing a system, it is important to keep the scale of it in mind (Wieringa, 2014b). What scale we are operating at indicates what information we can obtain from the results. During the development of a system, the idea is to work in increasingly realistic conditions. In this project we have mainly been operating in the bottom left corner and moved towards the right on the "Robust mechanism"-axis. The tests we have performed and their relation to this model are discussed later on.

4. The Sensor

4.1 The Duty Cycle

The sensor has three main tasks in this system: sense (record sound), compute (do some operation on the sound) and send (transfer sound file to central unit). This entire process is called a duty cycle.

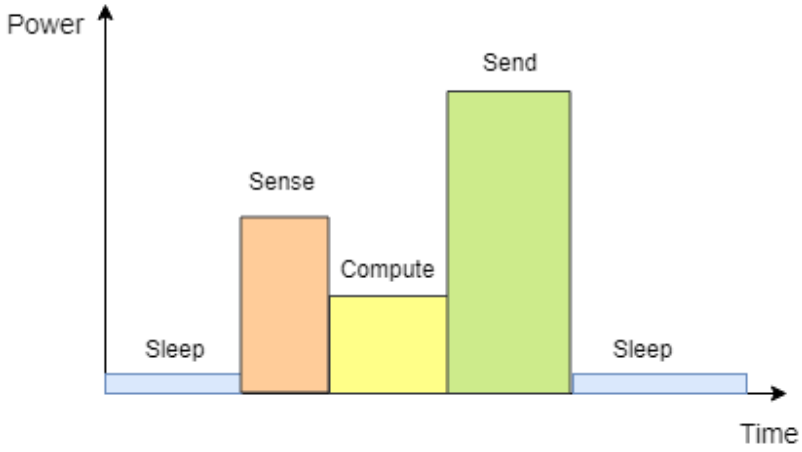

Figure 4: An abstract duty cycle

Figure 4 illustrates a duty cycle, where the sensor sleeps for some time, wakes up to do some measurements (record audio in this case), compute (sensor chip performs some computations, Fast Fourier Transform is one example), send the data, and then go back to sleep. This figure is a high-level illustration of the operations that the sensor is supposed to do. It is not correctly scaled in relation to the other activities or the power and time-axis. The purpose of the figure is to give a general insight into the sensor's operations, and to clarify this project's thought process for problem treatment.

This means there are three stages that can be influenced by configuration: Sensing, computing and sending. In addition to this, how often the cycle is run affects power consumption. Naturally, less frequent cycles leads to more time in sleep mode for the sensor, which saves power.

4.1.1 Sensing

Reducing the power spent on sensing would make a positive impact on power consumption. To reduce power consumption during sensing, the column for sensing either has to become shorter on the time axis, or shorter on the power axis. As there exists microphones can operate

with very little power drawn, such as electret condenser microphones (CUI INC, 2013), the importance of power saving in this phase is not as critical as in the sending phase.

4.1.2 Computing

What kind of computing needs to be done is system-dependent. For the system in this project, computations like Fast Fourier Transform could be relevant, but best case is that no real computations are done, so almost no power are used. Naturally, there could be some gain in power savings if the power spent computing reduces the power spent on sending by more than what the computations costs. This is discussed later, after presenting the results of audio reductions.

4.1.3 Sending

The power used for sending will depend on the amount of data that has to be sent, and how that data is sent (what protocols are used). The amount of data that needs to be sent is influenced by how much data is recorded in the sensing phase and how this data is handled before sending. Data transfer is typically the most expensive phase of the duty cycle, and for that reason it is an important factor when managing power usage.

4.2 Duty Cycle Frequency

How often the sensor executes a duty cycle is also a big factor in terms of energy usage. Say you reduced the amount of cycles performed by 50%, then power usage should also be reduced by 50%, assuming every cycle use the same amount of energy and that power spent in sleep mode is negligible. However, reducing the sensing frequency could reduce the usability and reactiveness of the system.

4.3 Why is Resource Usage Important?

On a relatively weak sensor that is running on battery or some alternative constrained power source, the sustainability of the sensor is a big concern. You typically want the sensor to be able to operate for very long periods of time, without having do any manual operations on it.

In contrast to personal computers today, all resources on a small battery-driven sensor chip is scarce. The CPU is much slower, the memory is very small, and the power is limited. Changing battery or manually recharging it is typically not a feasible solution, so the power usage must be carefully planned. Possible operations or computations that can be performed on the sensor is also limited with limited CPU and memory.

5. Finding and Testing a Machine Learning Solution

In section 2.1, AuDeep was mentioned as an acoustic scene classification tool. Using AuDeep makes it possible to do unsupervised feature learning for different sound data sets, and then use these features to train a classifier. As the tools seemed easy to use, delivered strong results in the DCASE 2017 challenge and deals with the issues of features selection, AuDeep was selected for initial testing, to see if it is compatible and suitable for this project.

As an initial test, the settings were fitted to the bottom left corner of Figure 3, meaning the test were a single case, with idealized condition. Before running the test, the hope was that the AuDeep-tools would accept the recorded audio files, and that the classifiers would give acceptable results after training. This would confirm that AuDeep could be used for this system and this use case.

The AuDeep-project comes with built-in functionality to train an SVM and an MLP-classifier. Throughout the tests, we run the classification process with both classifiers to give more trustworthy results, and to reveal how the classifiers differs as the input is changed. Testing this with self-supplied test data separated in cross-validation folds showed promising results, as demonstrated by these confusion matrices:

MLP:

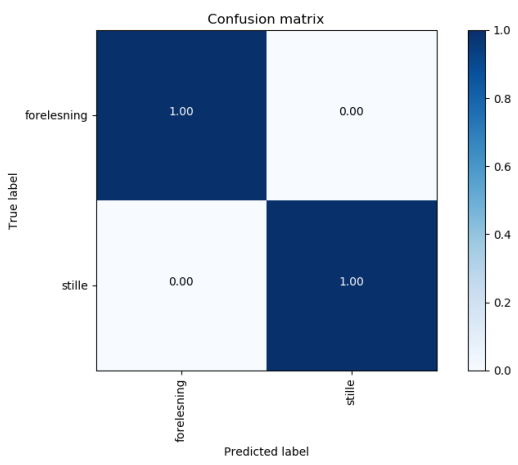

Figure 5: MLP classifier test

SVM:

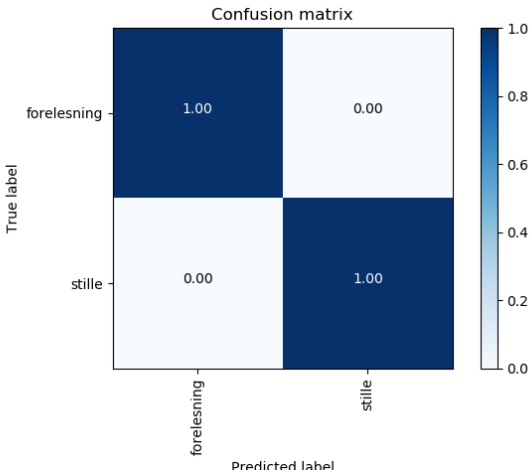

Figure 6: SVM classifier test

This was the initial test with only two labels, which are very different from each other. As the resulting confusion matrices indicates, the correct classification rate is 100%, and confirms the usability of the tools and the case itself. This test fits the description of a single case with idealized conditions, which is the bottom left corner of Figure 3. The next step is to scale up to a more realistic scenario.

Following this run, two more labels were added, so the system is now trained to label a sound into one of these four classes: “forelesning”, “stille”, “rats” or “team_act”. “Forelesning” is sound of a lecture being held in the lecture hall, “stille” is how the hall sounds when there are no people in the hall itself (but background noise exist), “rats” is sound recorded from when a large group of students is taking a test in the hall, but being quiet whilst doing so, and “team_act” is a team activity taking place in the lecture hall, meaning that a lot of chatter is going on, but the lecturer is not speaking, and no sound is coming from the lecture hall speakers.

From personally observing the lecture hall, this situation seems much more realistic, and covers most of the activity that takes place in the lecture hall. Note that there will always be exceptions to this, like construction work or washing being done. These cases could trick the system, as it has to label a sound with one of the given labels, but as we are not developing a commercial product, we will not implement any handling of these because it does not change how the system behaves.

The reason these classes were added was to get a better understanding of what type of sound comes closer to each other in classification, as parameters are changed. In the end, the system should do a binary classification where “forelesning”, “rats” and “team_act” all are classified as the lecture hall being busy, while “stille” means the lecture hall is classified as empty.

An initial concern here was that the system would not be able to separate between rats and silence, as they sound very similar to a human. Sitting in the hall with your eyes closed could fool you at the right time, unless you pick up the sounds of rattling paper, someone coughing, etc. However, this proved to be much less of a problem than expected, as shown by these confusion matrices:

MLP:

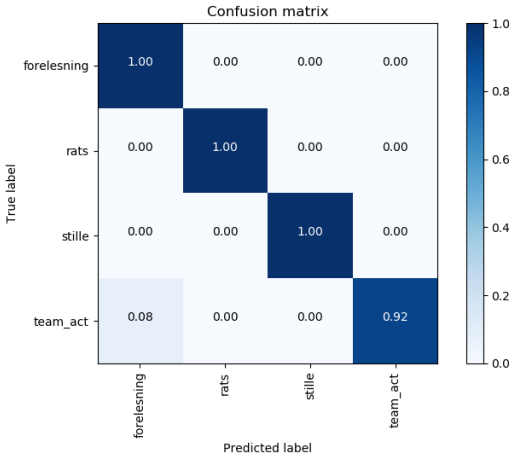

Figure 7: MLP with no reduction on audio clips

SVM:

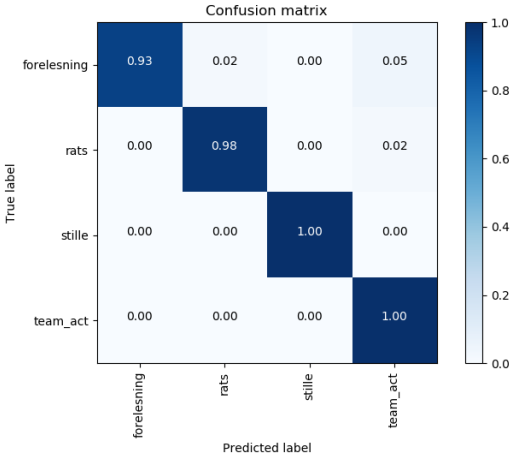

Figure 8: SVM with no reduction on audio clips

As shown, the results seem very strong, and if we look at how the binary classifier would evaluate these results, it would result in 100% accuracy, as we don't really care about making some wrong classifications between classes that represent the same end result.

The initial test data set used to set this standard consists of 158 audio clips, divided like this: 44 "forelesning", 50 "rats", 39 "stille" and 24 "team_act". The imbalance in number of audio clips per class is due to hardware problems during recording resulting in corrupted clips, and therefore some data had to be cut from the final data set after the data collection phase. This imbalance is taken into account when evaluating the classifier later on. The audio clips are put in two folds of almost equal size, in order to use cross-validation for evaluation, which is handled by AuDeep.

A Raspberry Pi 3 Model B together with a Cirrus Logic Audio Card (Element14, 2014) were used to record audio data and transfer it to a more powerful computer for analysis. The built-in microphones of the audio card seem well suited for a test environment that resembles an IoT-sensor setting. The data collection has only been done with a single sensor.

These original tests are done with 9 second audio clips, sampled at 44100 Hz, which is what we use as our upper limit for duration and sample rate. This combination seems representable for a typical classification problem that does not restrict itself because of constrained resources.

The spectrogram extraction is done with 0.08 FFT window width, 0.04 window overlap, and 128 mel frequency bands extracted.

The autoencoder has recurrent layers, with 256 GRU cells as is default in AuDeep. The decoder is set to be bidirectional. The training of the autoencoder is done with 64 epochs, learning rate 0.001 and 20% dropout. Because of memory issues, the batch size on the home system that were used was set to 16. In the later test, this is turned up to 32, just to speed up the training process slightly.

With this autoencoder we could generate features from the spectrograms and use the cross-validation setup to evaluate the classifiers based on these features. The SVM evaluation was done with a complexity parameter of 1, while the MLP evaluation was done with two hidden layers, 150 hidden units per layer, for 400 epochs, a learning rate of 0.001 and 40% dropout.

All classification tests are done five times and the results shown are the average of those five runs. This is done because the classifiers show some randomness, and therefore the average of five runs gives a better picture of the real performance.

The choice of these parameters is simply based on the original AuDeep experiment. The AuDeep developers showed how their classification accuracy was affected by changing these parameters, and what were the optimal settings for their data sets. We used rather low settings in this system, but the classification accuracy is still very high in the best cases. The most important factor in these tests is keeping these AuDeep-parameters static, so comparison of the classifier can be made when the only change in the system are the sound files themselves.

There could be a lot more experimentation with parameters and machine learning techniques to perfectionate the correct prediction rates, but this project focuses more on the differences between different sound clip parameters than perfectionating the machine learning part of the system, and it is much more interesting for us to investigate efficiency-aspects of the full system. For this reason, we settle for the learning parameters we have found satisfactory so far.

6. Classification on Downscaled Audio Clips

One way of making the system more efficient is reducing the amount of data needed. There is strong motivation to reduce the size of each sound file. If we could reduce the file size, there would be less memory usage on the sensor, there would be less power spent on transferring data and smaller sound files are potentially cheaper to record and do operations on, saving both power and CPU. There are two obvious ways this could be done: reduce the number of seconds of each sound clip; or reduce the sample rate of each sound clip.

For all following tests, a four-label confusion matrix is presented for both classifiers to give a better idea of how the classifiers reacts for each data set, and what classes gets harder for the classifiers to distinguish as the sound clips are cut down. However, the use case that is considered in this project is a binary classification problem, with two labels that describes the state of the lecture hall: busy or empty. In the end of each subset of tests, a table with the binary classification accuracy is presented, with a corresponding graph. All classification accuracy percentages presented are for the binary classification case. Percentages are also rounded to the closest integer.

The tests performed in this section is done with more realistic conditions than the initial test with only two classes, moving the field of operation further to the right of Figure 4. The samples are still only collected in a single lecture hall and therefore more tests from several rooms are needed to move into the centre of the graph where we are looking at multiple samples of realistic condition tests.

One assumption done here is that the reduction of sampling rate and sound clip duration are independent, so that one dimension can be altered without the other dimension being affected. As illustrated by the graph in Figure 9, this assumption gives us the possibility to explore values along one axis at a time. If the assumption is true, this should mean that when both dimension are explored, doing a combined reduction on both dimension should yield a classification accuracy very close to the weakest accuracy shown when only reducing one dimension. An example would be if cutting the duration in half gave 90% accuracy and reducing the sample rate by half gave 87% accuracy, then the combination of the two should still give around 87% accuracy.

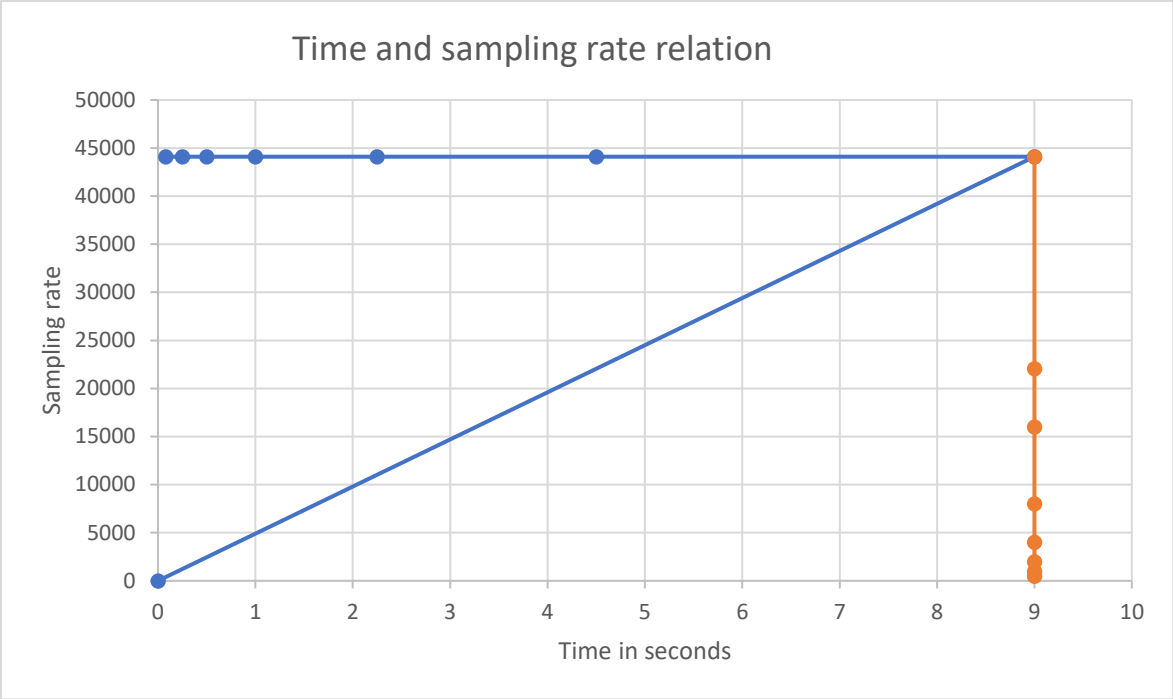

Figure 9: Time and sampling rate relation

6.1 Reducing Duration by Cutting Clips in Smaller Parts

For this part of the project, the work is done in cycles like indicated by Figure 2. The machine learning process is the same, but with constantly changed sound clip durations. For each step we look at gain versus loss and change the next run slightly to find the optimal settings for the system.

One concern here is that the shorter sound clips doesn't capture for a duration that is long enough to obtain sound data that is pivotal for the class, such as recording in between two words or two sentences of the lecturer, making the room appear silent to the classifier.

First test done is cutting all clips used in the initial test in half, meaning that the number of sound clips used are now doubled, while the total duration stays the same. The confusion matrices below show the results for 4.5 seconds sound clips:

4.5 second clips:

MLP:

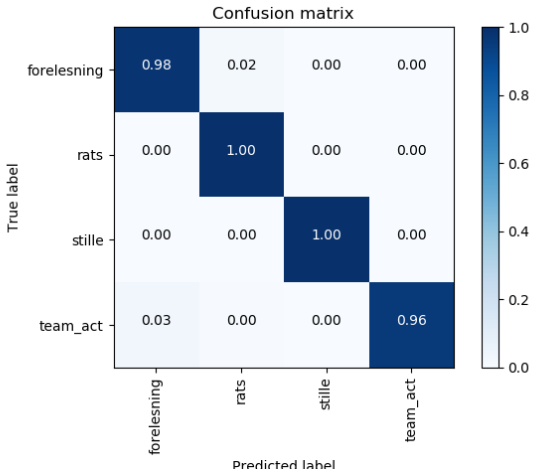

Figure 10: MLP 4.5 seconds

SVM:

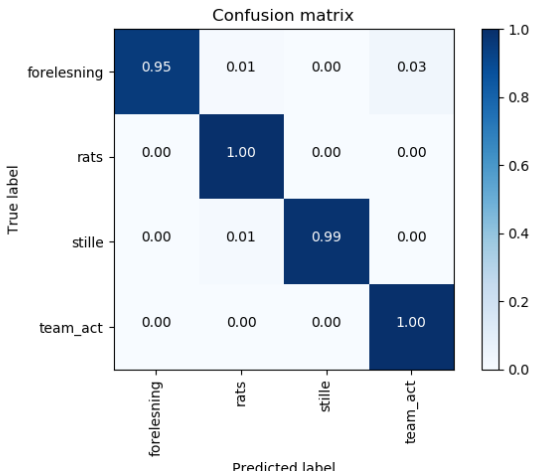

Figure 11: SVM 4.5 seconds

Both classifiers still show a ~100% accuracy, showing no real loss compared to the original run.

2.25 second clips:

MLP:

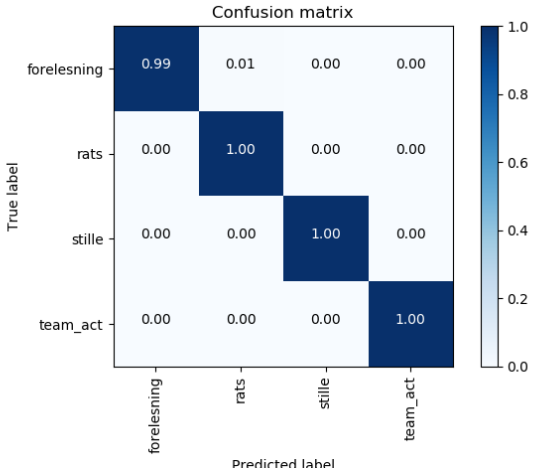

Figure 12: MLP 2.25 seconds

SVM:

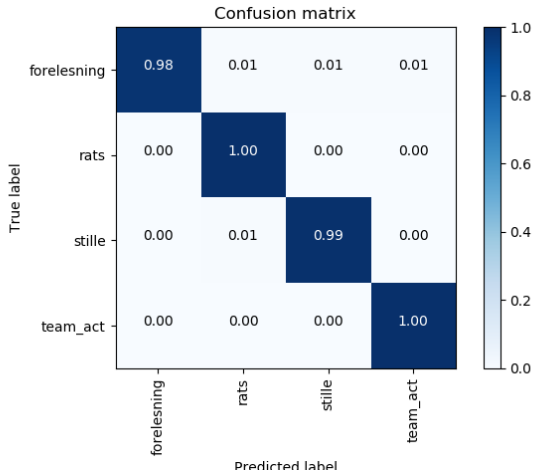

Figure 13: SVM 2.25 seconds

The binary classification accuracy is still maintaining 100% accuracy for MLP, and dropping to 99% for the SVM-classifier.

1 second clips:

MLP:

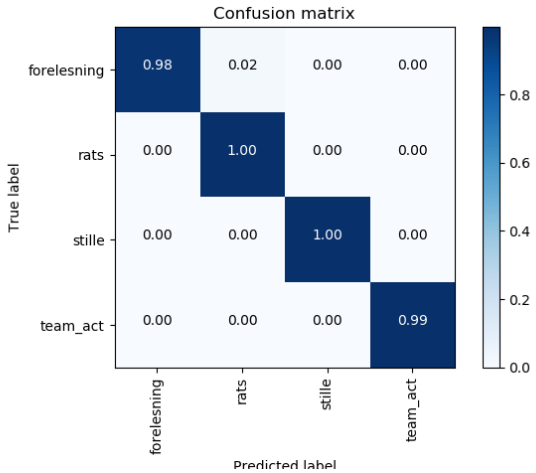

Figure 14: MLP 1 second

SVM:

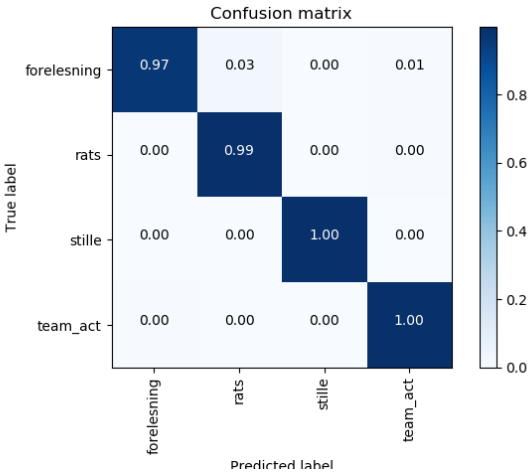

Figure 15: SVM 1 second

The original clips are now cut in 9 equal parts, but the system still performs classification at a near perfect rate, rounded to 100% for both classifiers.

0.5 second clips:

MLP:

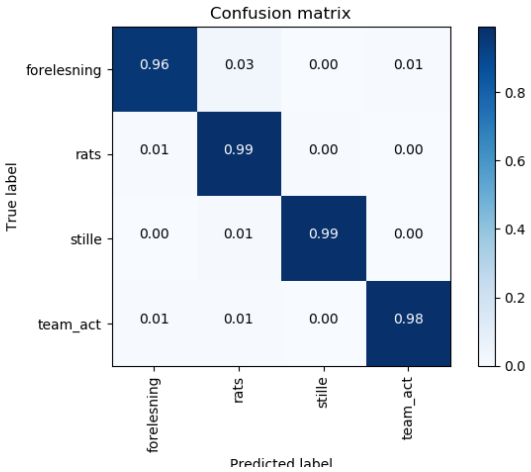

Figure 16: MLP 0.5 seconds

SVM:

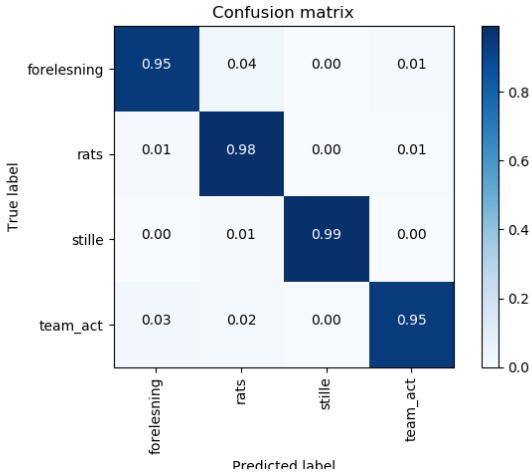

Figure 17: SVM 0.5 seconds

The confusion matrices show a very slight drop-off for 0.5 second clips, but the accuracy is still high enough to be rounded to 100%.

0.25 second clips:

MLP:

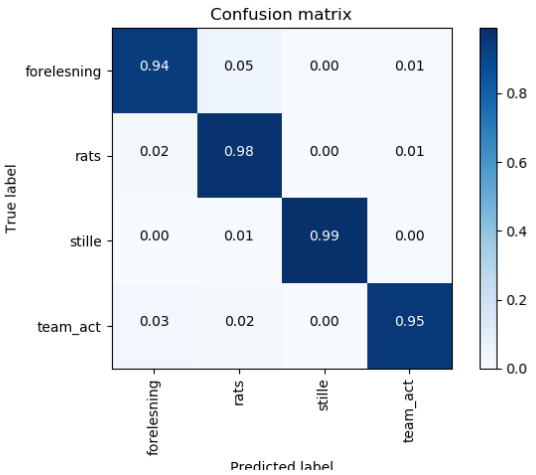

Figure 18: MLP 0.25 seconds

SVM:

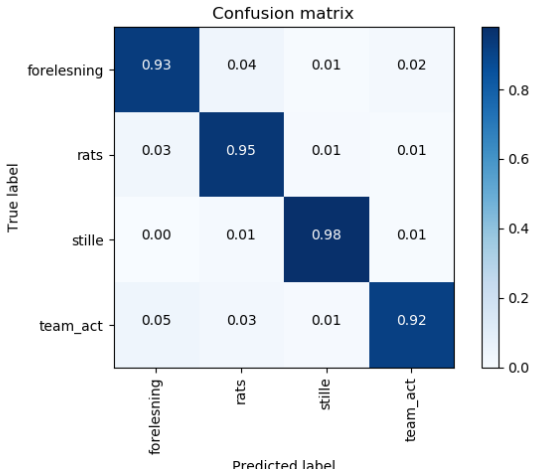

Figure 19: SVM 0.25 seconds

Once again cutting all previous clips in half, but the classification stays at 100% for MLP and drops to 99% for SVM.

0.08 second clips:

MLP:

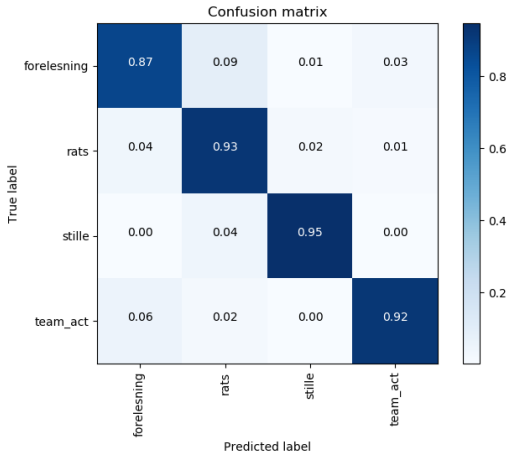

Figure 20: MLP 0.08 seconds

SVM:

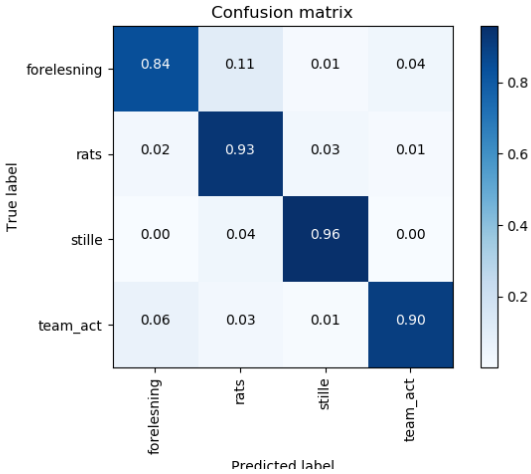

Figure 21: SVM 0.08 seconds

Going down to the lowest clip duration our settings allows, and the system still performs very strong classification, with both classifiers correctly classifying 98% of the total clips.

Results visualizations:

Time	MLP	SVM
9	100	100
4,50	100	100
2,25	100	99
1	100	100
0,5	100	100
0,25	100	99
0,08	98	98

Table 1: Classification accuracy vs time reduction

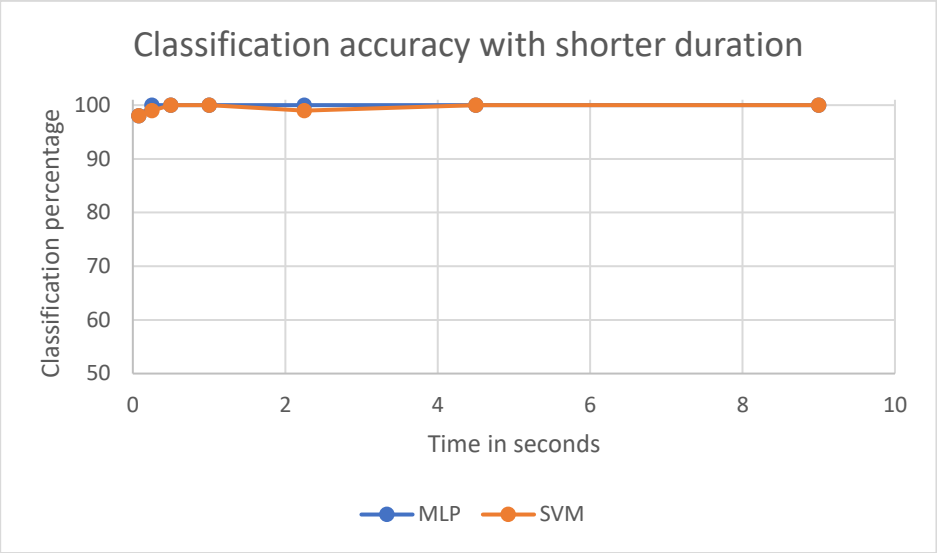

Figure 22: Graph of classification accuray vs duration

As seen in the table with the corresponding graph, the overall accuracy is very high, only showing a slight drop-off for the very lowest category. These results were surprisingly high, especially for the 0.08 second clips. Despite one of the initial concerns being that when given very short sound clips, there is a chance of not recording sound that is typical for the class, it seems this was not a very big problem in this case.

6.2 Reducing Sample Rate of Each Audio Clip

Lowering the sample rate of the clips, rather than the shortening of the recordings, gives a similar reduction in memory used. On the sensor node this would mean the sensor would have to sense for a longer period of time for each duty cycle, but the sensing itself is possibly cheaper due to the lower sampling rate of the audio clip itself. For the sake of this system, we assume the sensor power consumption is directly related to the file size, which makes changing the duration and sample size equally important.

Sample rate: 22050 Hz

MLP:

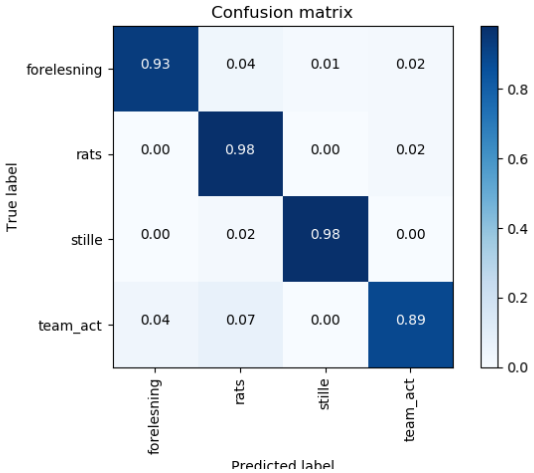

Figure 23: MLP 22.05 kHz

SVM:

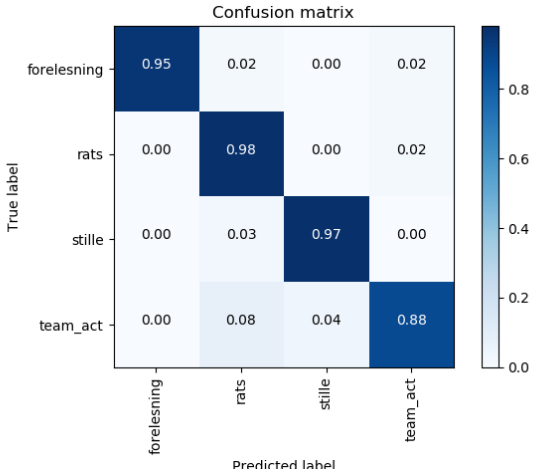

Figure 24: SVM 22.05 kHz

Reducing the sample rate by 50% does have a larger impact than what cutting the duration in half had. Both classifiers still perform very well but does go down from 100% to 99%.

Notice: a severe dip in the accuracy for “team_act”, somehow caused by this sampling rate in particular. Why this happened is unclear to me, despite trying to compare the frequency spectrum of these clips to the originals. This will not be a problem in when working with this use case but shows that reduction of sample rate could have some unexpected effects.

Sample rate: 16000 Hz

MLP:

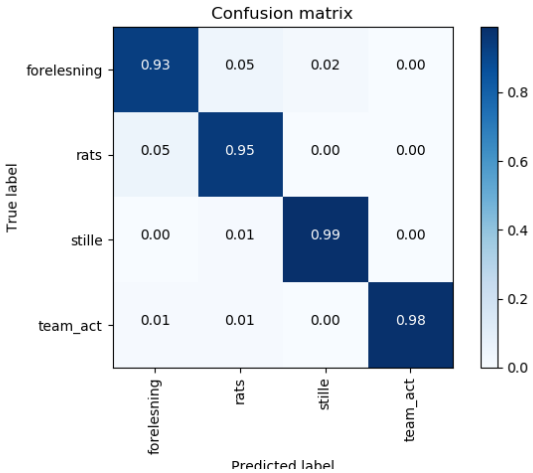

Figure 26: MLP 16 kHz

SVM:

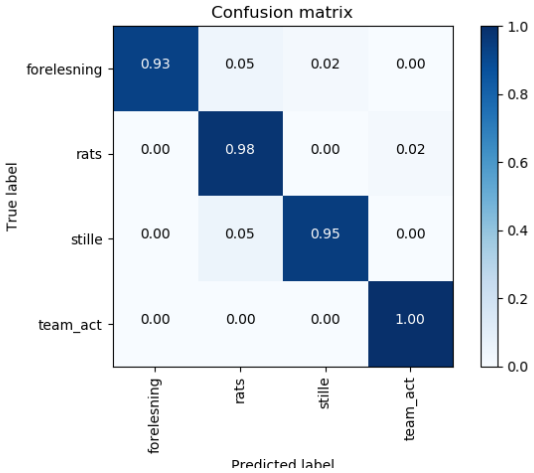

Figure 25: SVM 16 kHz

There is a very slight loss going from 22050 Hz to 16000 Hz, tipping the SVM-classifier over to 98% accuracy.

Sample rate: 8000 Hz

MLP:

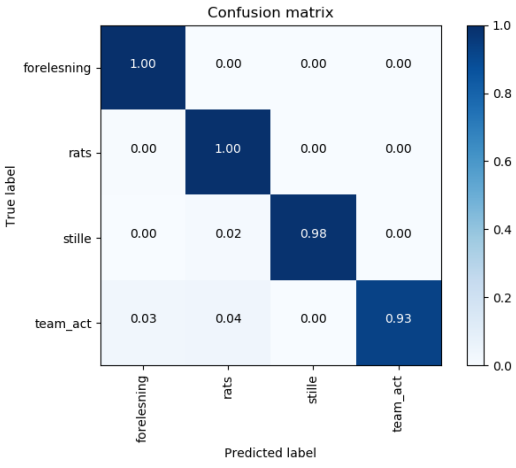

Figure 28: MLP 8 kHz

SVM:

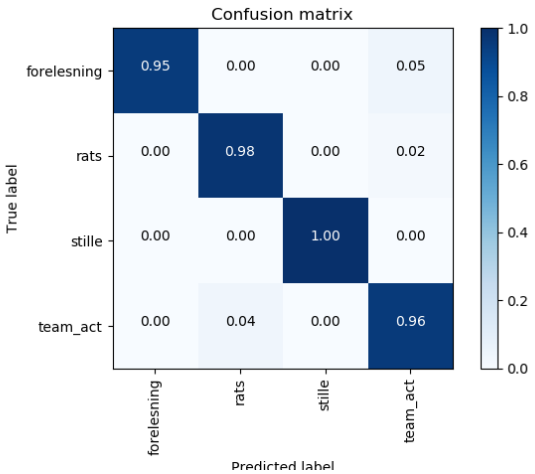

Figure 27: SVM 8 kHz

Just as with 16000 Hz, the accuracy increases slightly when reducing the sample rate. 8000 Hz is pretty much the lowest frequency used to sample sound these days and is already

considered old-fashioned because the sound quality is too low for a good listener experience. The classifier still performs well, and SVM goes all the way back up to 100%.

Sample rate: 4000 Hz

MLP:

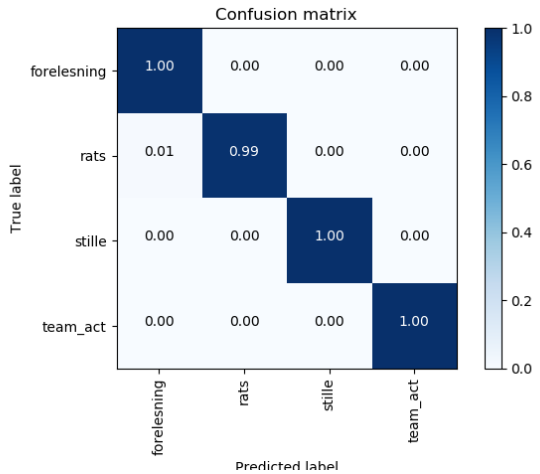

Figure 29: MLP 4 khz

SVM:

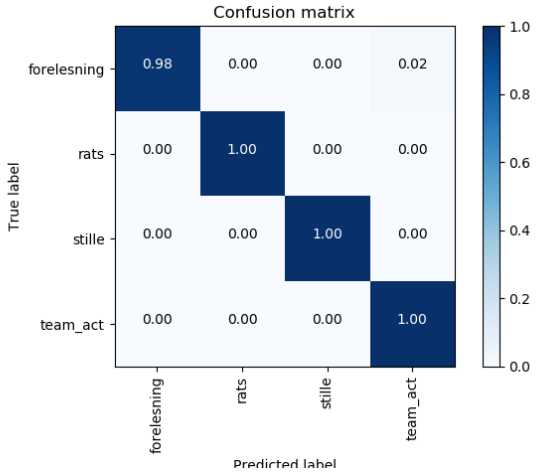

Figure 30: SVM 4 kHz

Going down to 4000 Hz, the audio quality is terrible to listen to, but the system performs extremely well under these conditions, showing 100% accuracy on both classifiers. This seems very promising when the goal is to make a system that operates as cheap as possible.

Sample rate: 2000 Hz

MLP:

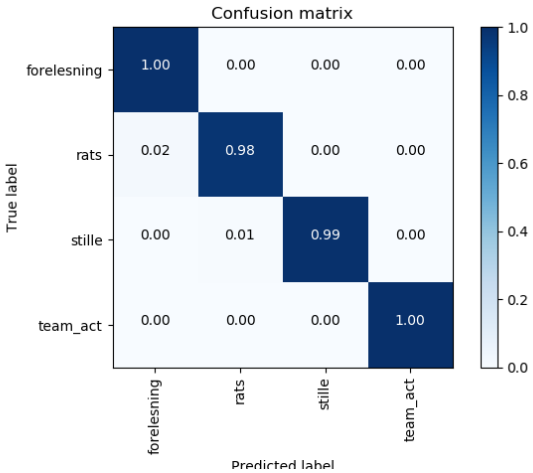

Figure 32: MLP 2 kHz

SVM:

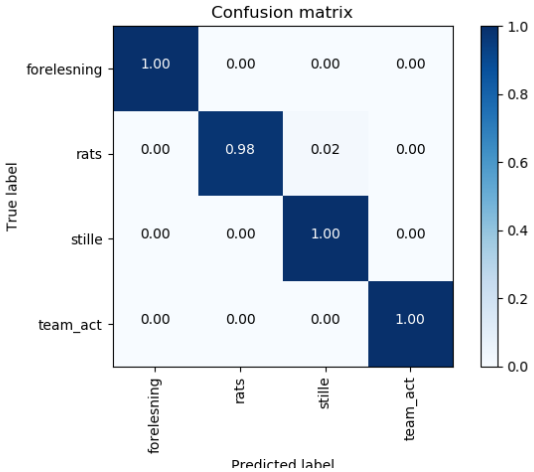

Figure 31: SVM 2 kHz

At 2000 Hz it gets very hard to tell what the lecturer is saying, but any human could still tell it's a lecture, and so can the system. Still near perfect accuracy despite having 22.05 times lower sampling rate.

Sample rate: 1000 Hz

MLP:

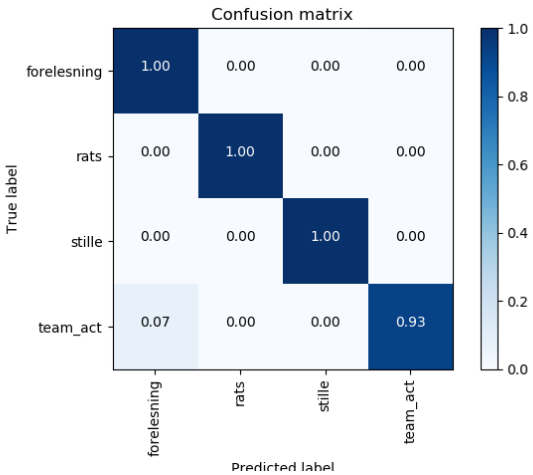

Figure 34: MLP 1 kHz

SVM:

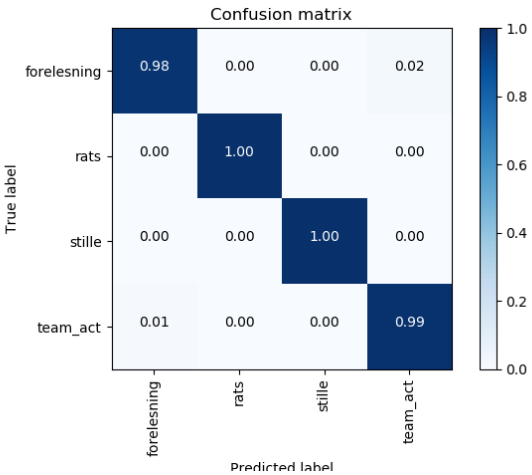

Figure 33: SVM 1 kHz

At 1000 Hz it is almost impossible to make out any words the lecturer is saying, but it is still possible to understand that the sound clip is a human speaking. The system shows extremely good performance, despite having reduced the sampling rate by a factor of 44.1. In fact, the system does not wrongly predict nearly any labels, so seemingly there is no critical loss of information that is needed to separate the audio clips.

Sample rate: 500 Hz

MLP:

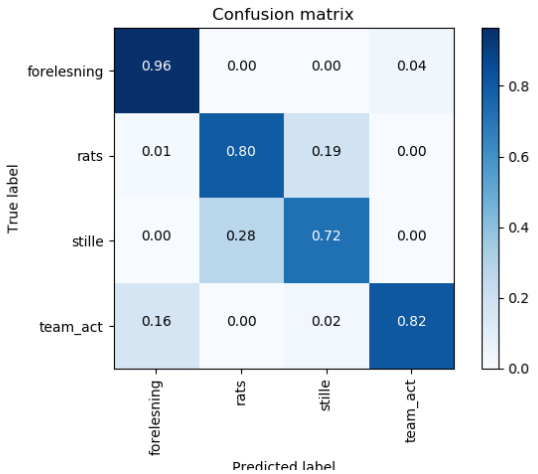

Figure 36: MLP 0.5 kHz

SVM:

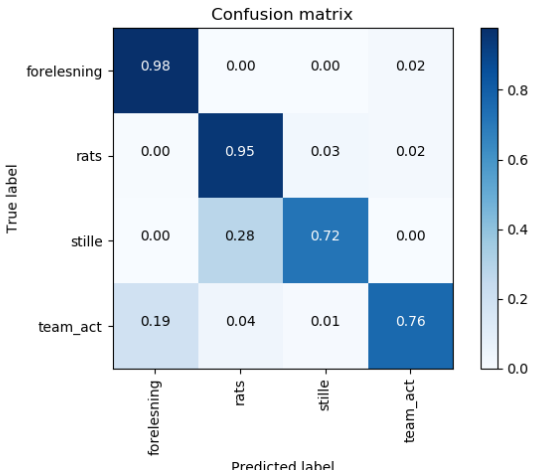

Figure 35: SVM 0.5 kHz

At this sample rate, some voice gets cut out, and the system starts struggling with classification finally, with a very large drop-off in accuracy from 1000 Hz.

Result visualization:

Sample rate	MLP	SVM
44100	100	100
22050	99	99
16000	99	98
8000	99	100
4000	100	100
2000	100	99
1000	100	100
500	87	92

Table 2: Classification accuracy vs sample rate reduction

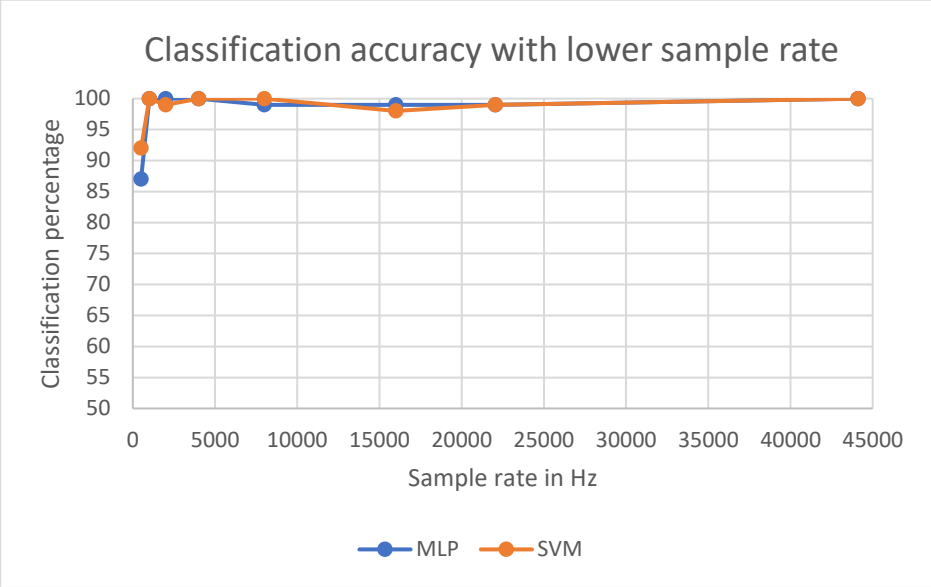

Figure 37: Graph of classification accuracy vs sample rate

After doing a frequency analysis on a selection of sound files from each class, it becomes clear that the majority of all sound in the clips is found from ~2000 Hz and below, while there is some noise from ~3000 Hz to ~4000 Hz and also a little bit of noise from ~12000 Hz and above, while there is nothing to report between ~4000 Hz and ~12000 Hz. This lead us to believe that 4000 Hz would be the ideal sampling rate for the system, but as it turns out, the sound that is important for classification seems to be present even when sampling with only 1000 Hz.

Overall, reducing the sample rate also showed very promising results, where we were able to reduce the sample rate all the way down to 1000 Hz without losing accuracy. In fact, the low sample rates of 4000 and 1000 Hz both managed an 100% accuracy, which was slightly better than some of the higher sample rates.

Based on the tests performed so far, both for duration and sample rate, both options seem very strong, and both options bring great benefits to the system without losing classification accuracy.

6.3 Combining the Two Downscaled Factors

Our initial assumption about duration and sampling rate is that they are independent of each other, and that leads us to try both parameters at the same time in this next step. The idea behind this is illustrated in Figure 9, where we assumed we could find the best solution for the system along the diagonal line somewhere. The straight lines along the X and Y-axis is where we have tested the classification accuracy at each point.

First test done was with very low values, the second lowest tested in both categories, being 0.25 seconds duration and sampled at 1000 Hz. Best case that could happen here is that the combination shows no loss compared to the 0.25 seconds, high sampling rate run, and we obtain a classification accuracy that is close to 100%.

Sound files of 0.25 seconds, sampled at 1000 Hz:

MLP:

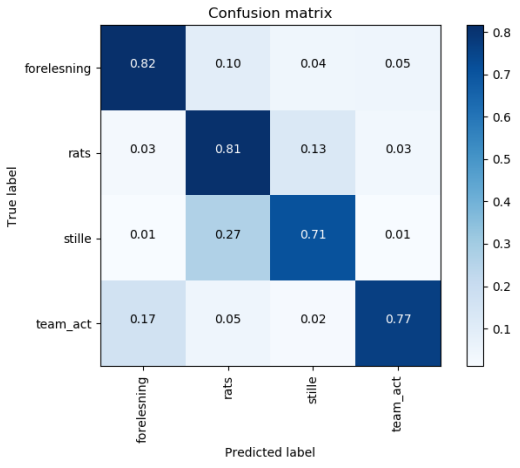

Figure 38: MLP 0.25 seconds, 1 kHz

SVM:

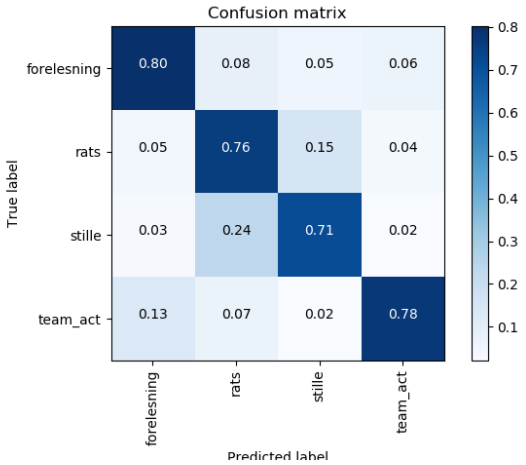

Figure 39: SVM 0.25 seconds, 1 kHz

So the assumption of complete independency did not hold, and the combined reductions on the audio clips lead to a major loss in classification accuracy. With this combination, the accuracy drops off significantly, to a point where it is unacceptable in pretty much any classification system. This means more tests need to be done to get a better understanding of how changing both parameters at the same time affects the classification accuracy.

Sound files of 0.5 seconds, sampled at 1000 Hz:

MLP:

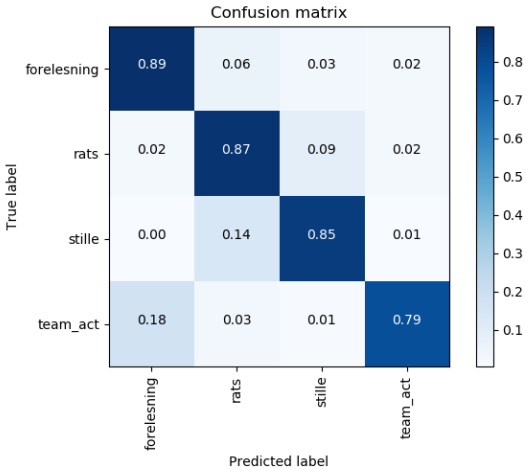

Figure 40: MLP 0.5 seconds, 1 kHz

SVM:

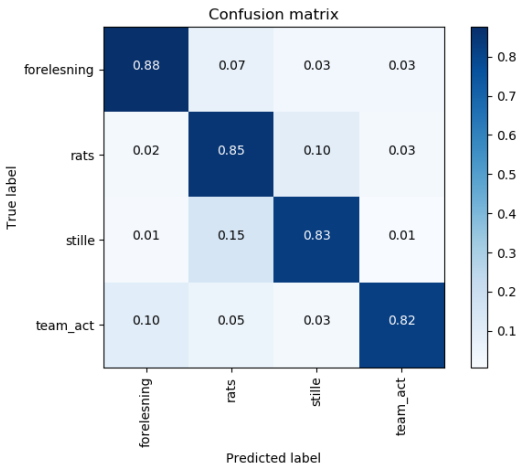

Figure 41: SVM 0.5 seconds, 1kHz

Sound files of 1 second, sampled at 1000 Hz:

MLP:

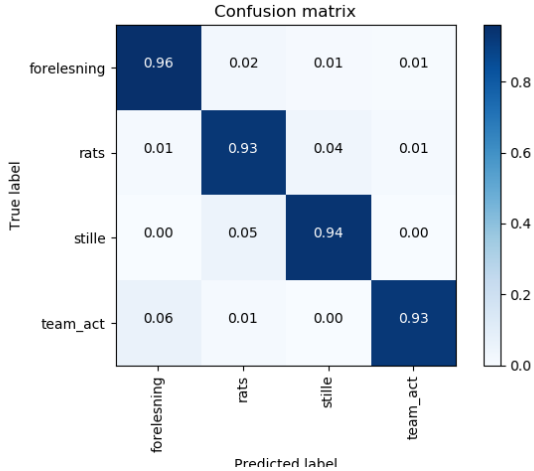

Figure 42: MLP 1 second, 1 kHz

SVM:

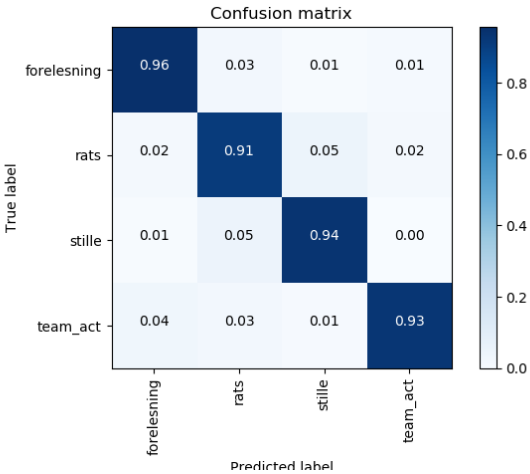

Figure 43: SVM 1 second, 1 kHz

Sound files of 1 second, sampled at 2000 Hz:

MLP:

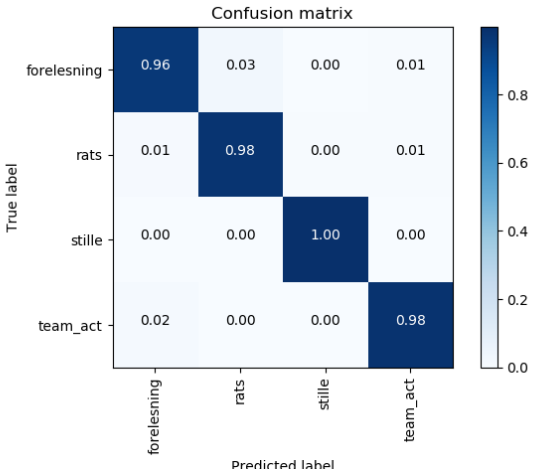

Figure 44: MLP 1 second, 2 kHz

SVM:

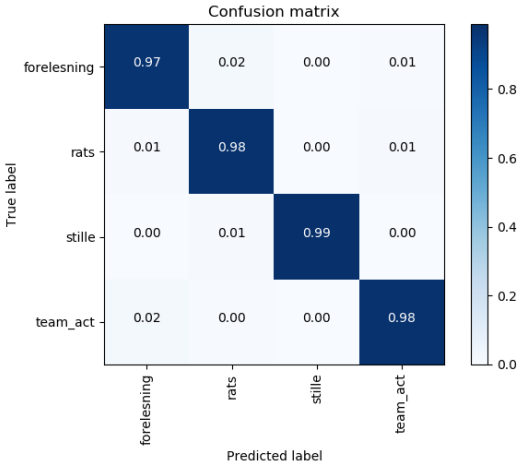

Figure 45: SVM 1 second, 2 kHz

Sound clips of 0.08 seconds, sampled at 500Hz:

MLP:

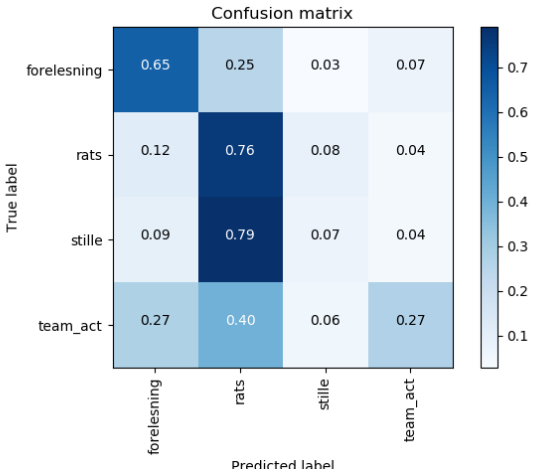

Figure 46: MLP 0.08 seconds, 0.5 kHz

SVM:

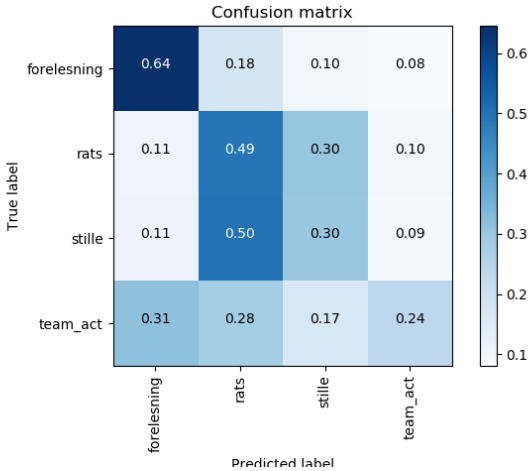

Figure 47: SVM 0.08 seconds, 0.5 kHz

To give an idea of how the system would perform with the worst settings tested in combination, one test was run with 0.08 second clips, sampled at 500 Hz. As seen in the confusion matrices, the result is extremely unreliable, and the system wrongly classifies two of the classes more often than it correctly classifies them.

Sound files at 1 second, sampled at 4000 Hz:

MLP:

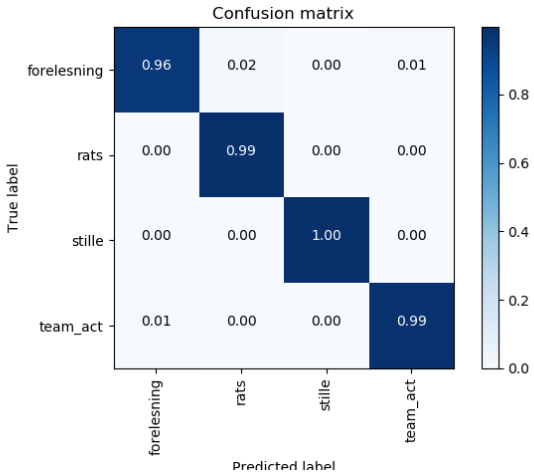

Figure 48: MLP 1 second, 4 kHz

SVM:

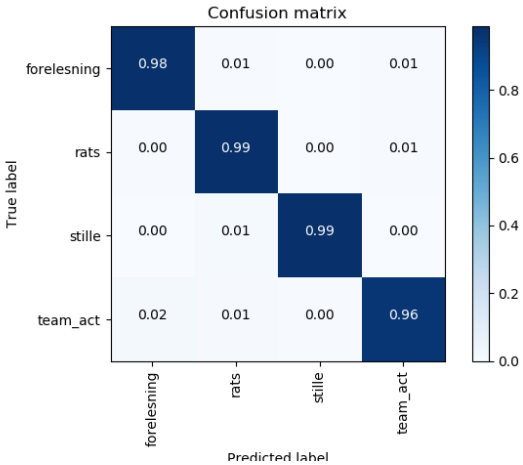

Figure 49: SVM 1 second, 4 kHz

Both 1 second and 4000 Hz showed 100% accuracy on their own, and the combination is almost equally good, yielding 100% accuracy with the MLP classifier.

6.4 Accuracy to File Size Reduction Ratio

An interesting relation to look at is how much classification accuracy is sacrificed in order to reduce the file size. This is the essential factor that has to be considered when designing the system. A product owner could use this to make a decision, based on his system specification, of what file size to use in order to keep a satisfactory classification accuracy.

The tests with reducing both dimensions of the audio clips in this data set is presented in this table with the corresponding graph (MLP accuracy is used as it performed slightly better overall):

Classification accuracy	File size reduction
100	0
100	99
100	197
96	391
92	771
87	1500
73	7283

Table 3: Classification accuracy vs file size reduction

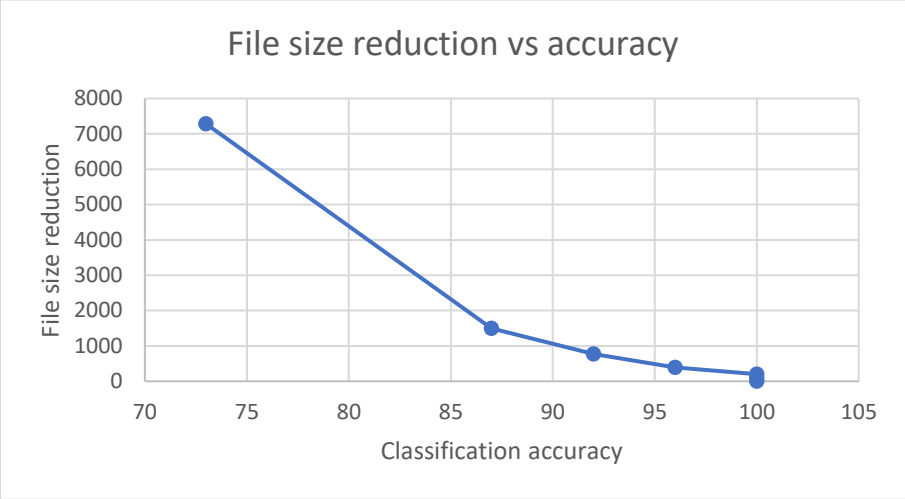

Figure 50: Graph of size reduction vs classification accuracy

6.5 Suspicion of Overfitting

As the classification accuracy is very high even when reducing the duration and sample rate, there could be a danger of overfitting. Overfitting would mean that the correct classification rate is artificially high due to a too one-sided data set or the test data being artificially close to the training data, and that the test results are not realistic. This would also mean that the system may not be fit to predict labels on new data samples. To make sure that the cross-validations were not overfitted, tests of data that was not available during training of the models were used. These tests show similar results to the original cross-validation tests (within reasonable boundaries of a few percent), and at least proves that the previous tests are not majorly overfitted. It should be noted however, that these tests were only done for a few combinations in the category of both dimensions being reduced, and that more testing is necessary to get a strong understanding of the classification accuracy over time.

6.6 Using Matthews Correlation Coefficient to Evaluate the Classification

Looking at the classification accuracy in the table above, it could seem that accuracy can be a misleading metric for classification performance. Given that the classes are imbalanced in the

binary case, the accuracy is as high as 73% despite the confusion matrix showing horrible results, as the system would almost always wrongly classify silence with these parameters. Matthews correlation coefficient is another metric and has shown to be suitable for imbalanced data (Boughorbel, et al., 2017). A great benefit of using MCCs in this project is that it can be directly calculated from the confusion matrices and takes all squares of the matrix into account. The formula for MCC can be written like in Equation 1

$$\text{MCC} = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Equation 1: Matthews correlation coefficient

where TP is True Positives, TN is True Negatives, FP is False Positives and FN is False Negatives.

The MCCs are only calculated for the combined reduction section, because it would give very little valuable information elsewhere. The ideal value for the MCC is 1, and if the value goes below 0, the classifier performs worse than what a random classification would do. The results from the MCC calculations are displayed with a table and graph below:

MCC	File size reduction	Size in kB
1	0	1587
1	99	16
1	197	8
0,92	391	4
0,8	771	2
0,65	1500	1
0,03	7283	0,2

Table 4: MCC vs file size reduction, with file size in kB

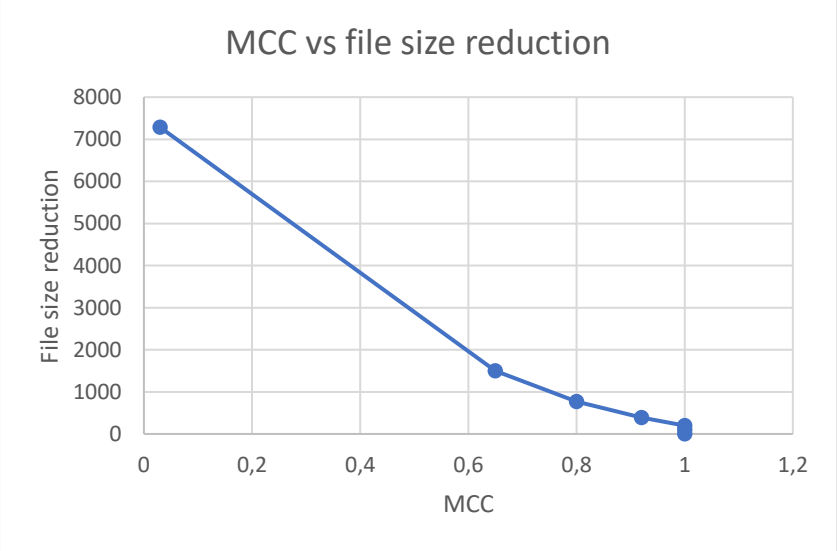

Figure 51: Graph of MCC vs file size reduction

These results paint a more telling picture of the system's performance, and actually shows that the combination of 0.08 seconds and 500 Hz is almost equal to randomly labelling sound clips. It still shows that the 197 times reduction is equal to that of the full-size clips, and the loss of going down to 391 times file size reduction is relatively small.

6.7 Discussion of Results

The numbers presented from the tests speaks for themselves to some degree, where the drop-off in classification quality clearly shows when the clips are reduced more and more. Without a full implementation of a system and a product owner with some demands for the system, it is difficult to pick the "correct" file size, but with a presentation like this it is possible to set a requirement for either file size or MCC/accuracy and tell the other factor from that. Example: The university demands an MCC of above 90 in the system, and that means we are able to reduce the file size about 400 times while still satisfying the requirements.

The two classifiers gave very similar results throughout most of the tests, except for the 0.08 seconds, 500 Hertz audio clips. From this it can be seen that the representation learning process gives a very good basis for the classifiers, and that the choice is final classifier is not necessarily critical. Other classifiers have not been tested, but it is plausible that they would show similar results, and that the final application can be flexible in the choice of classifier.

The degree of reduction needed could also be hardware specific. Some sensors have very little memory and might not be able to store files over a few tens of kilobytes, while stronger sensors could be able to store a few megabytes. In this case it comes down to selecting the proper hardware for the desired system.

A side effect of using audio clips of short duration and very low sampling rate, is that they become uninterpretable to a human listener. This does cause the system to become naturally non-intrusive, as it does not capture data that has any real value outside of the classification system. If security is down-prioritized because of resource constraints, this could be a very positive effect to inherit.

These results are obtained from sound files recorded in a single lecture hall and is therefore mostly a proof of concept in this context. However, the results seem very strong and is promising for a more general setting. The possibility of training a system that can classify

sound from any given room is unclear, but an interesting prospect for future systems, and with a powerful common cloud server there is potential for a very scalable solution, suitable for something like an entire university.

7. Data Transfer Protocol

The reason that choice of data transfer protocol is an important aspect of system design, is that data transfer is the most expensive operation for a lot of sensors systems, including the one in this project. A lot of the protocols designed for IoT is not designed to handle larger file transfers and works best when the information that needs to be sent can be stored in a single packet.

Finding the most suitable protocol to use for an IoT-system is a challenging task, because it depends on a lot of variables. There is no single protocol that simply beats out all the others, they all have different perks and drawbacks, and the choice comes down to what environment the system operates in and what the system needs to accomplish. In our case, the system is set to work within one room in an indoor environment, and it needs to transfer files of some kilobytes (around four to eight, as shown in the previous section).

The protocol used in a sensor system is a contributing factor in defining the system's capabilities. It is the protocol that determines how far the sensor can send data, how fast it can send the data, and how much power it uses doing so.

7.1 Finding a Set of Potential Protocols

The list of protocols to choose from is long (Schatz, 2016), but luckily there has been several comparative studies between some of the most popular ones. For the system we are aiming for, the most important value is low power consumption, as long as some transmission rate and range are preserved. The transmission range for this indoor sensing system will be some tens of meters, so we can exclude the very low range protocols such as NFC, and high power/high range technology such as cellular is not relevant for the system either. That leaves the protocols that operates within a range of some tens of meters, and keeps an acceptable data rate, while still consuming very little power. Some of the protocols that fits this description are 6LoWPAN, Zigbee, Bluetooth Low Energy, Z-wave and ANT, so these are the ones we will compare and decide between in the next section.

7.1.1 6LoWPAN

6LoWPAN – "IPv6 over Low-Power Wireless Personal Area Networks" is as the name suggest, a low power protocol that uses IPv6-packets to efficiently transfer data. This protocol

does what it promises and is capable of operating cheaply in a duty cycle with sleeping periods (Olsson, 2014). The big advantage of 6LoWPAN is usage of IP communication which is used by an extreme number of devices already, and very well known by most developers, and it also has good support for mesh networks. In this system however, it falls behind some of the other protocols like Bluetooth Low Energy, because of higher energy consumption and no need for mesh network capabilities (Tabish, et al., 2013;López, et al., 2013)

7.1.2 Zigbee

Zigbee is another low power, low data protocol that is designed for sensor networks. It is created by the Zigbee Alliance which consists of several of the world's largest communication and IT-based companies like Huawei and Comcast (Zigbee Alliance, no date). This means there are strong support for the protocol, and it is well established in the market. Zigbee has been part of several comparison articles, and the capabilities of the protocol is well tested and documented through all of this research (Siekkinen, et al., 2012; Dementyev, et al., 2013).

7.1.3 Bluetooth Low Energy

Another promising low power protocol is Bluetooth Low Energy, which is the newer versions of Bluetooth. The Bluetooth protocol exist in a very large portion of commercial products already, such as most smartphones. Originally the protocol had shorter range and higher data rate than Zigbee (Ray, 2015), but with the introduction of Bluetooth 5, the range has been quadrupled and the maximum rate doubled, without an increase in power consumption, making in a very strong contender in the IoT-sensor protocol market. For the system in this project it seems like a good fit, with a star network topology and long sleeping durations between data bursts (Gomez, et al., 2012; Collotta, et al., 2017).

7.1.4 Z-wave

Z-wave is a more specialized low power protocol than Zigbee and BLE, but with many similarities. The main differences are that it operates in a lower frequency band, and that it offers low data rate, especially compared to BLE (40 kbps vs 2Mbps). Like Zigbee, it operates as a mesh network, but it has no significant advantages in this project's system (Ray, 2017).

7.1.5 ANT

ANT is another low power protocol that is capable of operating in a cyclic sleep scenario. With ANT+, and extension of the ANT protocol it is possible to build a topology-flexible sensor network, and ANT+ offers interoperability between all ANT+-devices (ANT / ANT+ DEFINED, no date). However, ANT doesn't seem to provide any significant advantage over the other protocols for this project, as it loses out in power consumption to both Zigbee and BLE (Dementyev, et al., 2013).

7.2 Choosing a Suitable Protocol

Judging from these descriptions and comparisons, Bluetooth 5's Low Energy seems to be the most promising alternative, as it has sufficient range, high data rate and very low power consumption. Bluetooth is also a widely used protocol and there are several chips on the commercial market with Bluetooth 5 capabilities. With the improved performance in Bluetooth 5, the sensor should be able to transmit at least 40 meters indoors, and the data rate can be configured to 125kbps/500kpbs/1Mbps/2Mbps. Noting that an increase in data rate means a decrease in range, makes it so that choosing data rate becomes sensor specific. Despite all of this, the sensor could transmit data at 2Mbps using only half the power its previous version would have used, because the radio frequency decides the power consumption, not the data rate itself, and the radio frequency has remained 2.4GHz (Collotta, et al., 2017).

What is especially interesting in this case, is how much energy is spent on the file transfer. This could be a very costly operation in terms of power usage and can have a heavy influence of how the sensor operates in general, if the number of data transfers has to be restricted to save battery power.

7.3 Making an Estimation of Power Consumption

To try to understand more about how this system could operate, making an estimation of power consumption could be useful. Although an estimation won't match a real-life situation, it should give some idea about how the system will perform, and if it is within reasonable boundaries for the system requirements.

There is a lot of parameters to consider when trying to estimate energy usage by the sensor node. The number of packets that needs to be retransmitted is one of these parameters, which can be different for every transmission. To make an estimation, the number of retransmissions is assumed to be 0.

To give a general idea of the power consumption of the data transfer, we set a series of parameters that could be used in the system. In order to keep a connection between the master and slave alive, there has to be at least one notification sent every 4 seconds. Assuming latency is a non-issue in the system, the sensor could send the file in packets, with one packet every 4 seconds. Choosing a sensible file size, such as 8058 bytes which showed no real loss in accuracy, with DLE enabled, meaning we can fit 244 bytes of payload in each packet, we need a total of 35 packets to transfer the entire file. Using the 1MB PHY data rate, sending 1 packet per 4 seconds, means the sensor transfers one full audio file in 140 seconds. This way we essentially spend no excess power transferring the data, except the power difference between sending an empty packet to keep the connection alive and sending a full 244-byte payload.

To do the calculations, we used STMicroelectronics' BlueNRG current consumption estimation tool (STMicroelectronics, 2017). Their BlueNRG-2 chip does support Bluetooth 5, and is therefore suitable to use in these calculations (STMicroelectronics, no date). Assuming a battery capacity of 500 mAh and using the power consumption values for the BlueNRG-2 for the different operations, the estimated lifetime of the battery is 7 years, 4 months, and 27 days. Keep in mind these calculations only take the Bluetooth data transfer into account, so that all sensing, computing and other operations done on the sensor chip are not included. The plot given of the power consumption in one connection interval is shown in the plot in Figure 52.

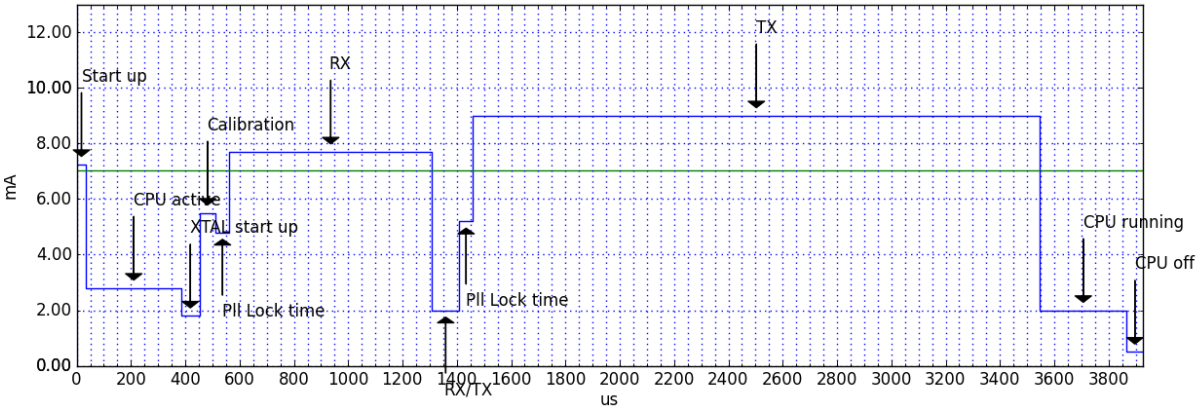

Figure 52: BlueNRG-2 energy consumption for one packet sent

These settings would allow the sensor to send one audio file to the central unit every ~2.33 minutes, meaning the situation of a lecture hall is mapped out by around 25 measurements per hour, without having to worry too much about power consumption. Even if the real energy consumption of the sensor proved to be double of what this estimation shows, it would still last for over 3.5 years. In fact, if we chose to send 35 packets per 4 second interval, the battery would still last for 3 months and 25 days if only spending energy on data transmission.

7.4 If We Could Do Classification on the Sensor

Currently, the sensors do not have the capability to do classification on the sensor themselves, but if they did, there could be a huge power saving potential, by only sending very small packets indicating the class (1 byte), and even choose to only send data when a change in class is observed. Using the same settings as the calculation above, but with 1-byte payloads instead of 244, we are looking at over 17 years of battery life, although this is slightly misleading as the operations on the sensors would be much heavier. Without a significant increase in memory and CPU power on the sensor, classification on the sensor is not feasible, but it is an interesting aspect of the future.

Doing the pre-processing of the audio clips on the sensor also proved inefficient when the audio clips were small, as the files made by the pre-processing were significantly bigger than the raw audio clips.

8. Adaptive Sensing

A lot of work has been done on sensing smarter than just getting samples from all the sensors in a network at a static interval. The motivation behind this can be several factors, but one of the most popular is saving energy without losing data value. This is usually achieved by adapting the duty cycle. Studies has been done where the duty cycle is adapted based on traffic load (Oliveira, et al., 2013), distance to the central unit (Zhang, et al., 2010), and also sensor redundancy where some sensors can be left sleeping to conserve power (T, 2015). However, these are not the only factors that can be considered to trigger a change in duty cycle frequency.

How often a duty cycle is performed can also be related to the reactive needs of the system. One example of a system that is dependent on fast reactions is an alarm system, where the earlier you can raise the alarm, the better. In a system like this, sensing frequently is crucial, but would be very taxing on the battery, so finding the correct trade-off is the key to obtain a good system overall.

Another factor to consider is the current battery level of the sensor. In many cases, loss of functionality in the system is worse than having a weaker data foundation. In case of low battery level, a decision to reduce the duty cycle frequency could be made in order to maintain the sensing functionality.

The adaption based on observations is the factor this project will take a closer look at, as it seems the most interesting in a single-hop network with relatively low amount of sensors per central unit. Adaption based on battery level is also briefly discussed in combination with energy harvesting techniques.

8.1 Adaption Based on Observations

When dealing with sound as an information source, noise is a challenge that must be dealt with in some way. Noisy sound can even be assumed to always be present in some way, and noise can affect the classification. If the environment around the sensor proved to be noisy on a regular basis, it can be hard to trust a single sound clip coming from it.

In the calculations for BLE power consumption, a constant duty cycle interval was used, making the system very predictable, but in a situation where we choose to not trust the observation before we see several equal classifications in a row, the reactiveness of the system

is weak. There is considerable motivation to increase the duty cycle frequency in some periods of time to increase the data value and reactiveness of the system. However, this would mean an increase in power consumption, which is very undesirable. The ideal system would be able to dynamically change the duty cycle frequency to achieve higher data value and reactiveness, without significantly increasing the power consumption.

As previously mentioned, doing classification on the sensor is not viable yet, which means that if the system were to adapt its duty cycle frequency based on observations, it needs to be organized by the central unit.

The idea behind this is that a lecture hall typically goes through longer periods of the same activity. One single lecturing session often indicates activity in the room for around 2 hours, and a full schedule could mean the lecture hall is almost always in a busy state from 08:15 in the morning to 16:00 in the evening. This description of a lecture hall calls for a low duty cycle frequency.

Although the low duty cycle frequency would be a good solution most of the time, it would come at a cost, which is reactiveness. As changes in the environment around the sensor is assumed to be infrequent, it is fair to assume that the central unit would be in a situation where it classifies the new incoming sounds as the same class many times in a row. When a change in class finally appears, it can be hard to trust that a change has really taken place. In this situation, it could be beneficial to be able to take more samples in a shorter amount of time to confirm that the classification was not a false positive.

We picture an algorithm that change the duty cycle frequency whenever a streak of several similar classifications is broken, illustrated by Figure 53 where each rectangle indicates a duty cycle. This could be achieved by the allowing the central unit to ask the sensor for a measurement and set the connection interval to a shorter period/send more packets per interval.

Theoretically, the new BLE connection interval could be set too 100 milliseconds, which is enough to send 35 packets, and do one sensing per second, meaning a change in the activity could be confirmed by 4.4 seconds (1 second spent sensing, 0.1 seconds spent sending), assuming we set the streak threshold before reducing frequency to four cycles. This would be draining on the battery, but compared to the static interval of one sensing per 2.33 minutes, assuming the same four streak requirements, confirming a change in the activity would take about 9 minutes and 20 seconds.

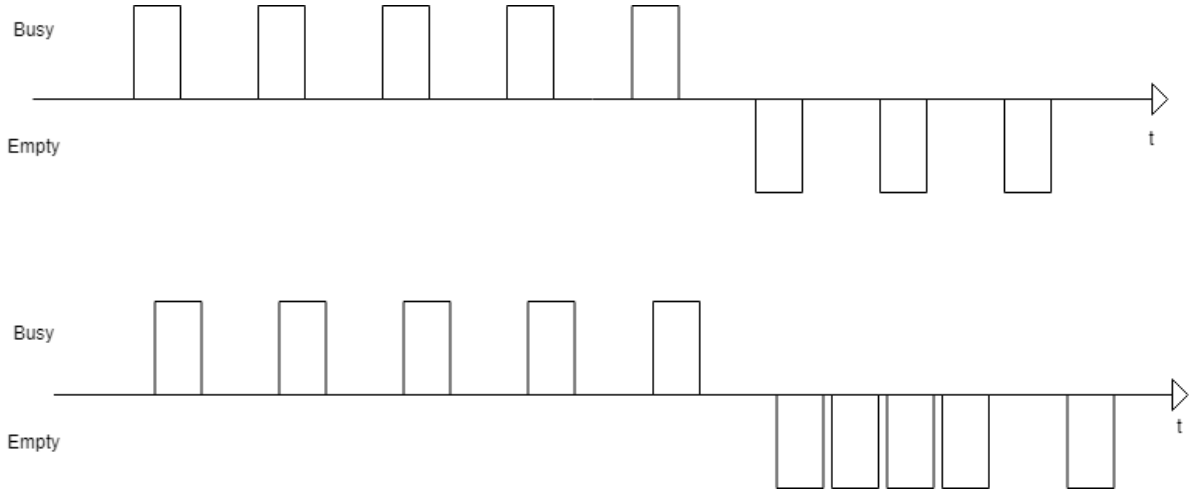

Figure 53: Adaptive sensing on observations idea visualization

The time limit of this project made implementing and testing an algorithm like this undoable. Without a proper implementation and simulation, there is sadly little value to draw from this result-wise. We still choose to present the idea, to show future work that could aid the reactiveness of a sound sensing system that cannot do classification on a sensor level.

There are some potential dangers to consider when designing an algorithm like this, such as ending up with a much higher than normal power consumption due to noise making a change in classification very often. It is hard to say what should be done in these cases. Some systems might be satisfied with just putting the sensor to sleep for a while and hope that the source of the noise disappears, while some systems might not tolerate down-time at all and wants to keep the sensor on a high frequency duty cycle plan. If noise is too much of a problem, then a microphone-based sensing system might not even be feasible.

8.2 Adaption Based on Battery Level

Energy harvesting sensors are an interesting option in any power constrained sensor network, and several ways to harvest energy on a sensor exists (Shaikh and Zeadally, 2016). However, state-of-the-art energy prediction models was described as immature, and showed high error rates. A better solution could be to not try to predict future energy gain, but rather look at the available power at the current time.

Estimating energy level on the sensor and sending it to a central unit has proven to be a possibility without any additional hardware, and the estimations are very close to the actual level, which enables adaption on the current battery level, controlled by the central unit (Tamkittikhun, et al., 2017).

The possibility of both down and upscaling the duty cycling frequency based on how much energy a sensor has could be an interesting option, especially when combined with energy harvesting techniques for sensors. This has proven to be a feasible option which manages to avoid any sensor dead time due power shortage (Vigorito, et al., 2007).

Although this project did not find the time to implement any functionality for adaption on battery level, it is presented as an interesting extension of the system for future work, and also to give a better picture of the options that are available to a sensor system. These options should be taken into account when discussing the feasibility of audio data, as it could enable high data value without causing too high power consumption.

9. Summary and Concluding Remarks

In this project we have investigated the potential of using microphones as a sensor in an IoT setting, in combination with machine learning. To accomplish this, we have investigated popular methods to do acoustic scene classification and ended up using a tool that utilize unsupervised feature representation instead of hand-crafted feature sets.

The tests performed shows that classification of sound files with low duration and sampling rate is possible, even with large reductions, but further testing with more variance is needed to get a better idea of how reductions impacts classification on a general basis.

An investigation of communication protocols designed for low power IoT sensor networks has been performed, and BLE was chosen as it seemed the best candidate for the system in this project. Estimated power usage showed that a weak sensor should be able to operate with high classification accuracy for a very long time if downscaled audio clips are used in combination with BLE.

Lastly, adaptive sensing was discussed as a way to make the system more reactive. Other projects have shown that adapting the duty cycle is a feasible option to achieve goals like saving power without losing system capabilities. An idea of adapting based on observations has been presented briefly, but a proper implementation of an observation-based adaptive sensing algorithm needs to be tested and compared to a static interval sensing application to investigate the gained value vs power consumption.

Utilizing machine learning, low power communication protocols, and possibly smart sensing techniques, we believe the results show that audio is a feasible information source with a lot of potential in an IoT network. This project investigated audio data with a binary classification problem as its basis, but the tests done involving machine learning shows that it could be viable to more complex problems with several different classes.

10. Future Work

A full-scale prototype of a system should be implemented to get a better understanding of how the system would work in a real-world scenario. There are also a lot of more work that could be done optimizing the machine learning process, potentially allowing for even bigger reductions in file size without losing accuracy/MCC.

Adaptive sensing seems to have a lot of potential and pursuing a good solution for adapting based on observations further could be a strong addition to systems like these.

We believe that both wireless sensor networks and machine learning will see a lot of research and improvements in the future, and that possible solutions to classification on the sensor side will be discovered. This could make file size reduction less needed as we don’t need to transfer a full data file anymore. It could still be energy efficient because of the energy saved on sensing and computations, and there is also a possibility that reducing the sound clips before classification could be an enabling factor in sensor-side classification, but that remains to be seen.

11. References

Amiriparian, S. *et al.* (2017) Sequence to Sequence Autoencoders for Unsupervised Representation Learning From Audio, *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017),* pp. 17-21

ANT / ANT+ DEFINED (no date)
Available at: https://www.thisisant.com/developer/ant-plus/ant-antplus-defined/
(Accessed 23 May 2018)

Boughorbel, S., Jarray, F. and El-Anbari, M. (2017) Optimal classifier for imbalanced data using. *PLoS ONE,* 12(6), pp. e0177678. doi: 10.1371/journal.pone.0177678

CCRL 40/4 Rating list (2018)
Available at: http://www.computerchess.org.uk/ccrl/404/
(Accessed 05 Juni 2018)

Chandrasekhar, P. and Gangashetty, S. V. (2017) *ACOUSTIC SCENE CLASSIFICATION USING DEEP NEURAL NETWORK,* Munich: s.n.

Collotta, M. *et al.* (2017) Bluetooth 5: a concrete step forward towards the IoT. *arXiv:1711.00257*

CUI INC (2013) *CMA-4544PF-W Datasheet.*
Available at: https://www.cui.com/product/resource/cma-4544pf-w.pdf
(Accessed 06 June 2018)

Dementyev, A. *et al.* (2013) Power Consumption Analysis of Bluetooth Low Energy, ZigBee and ANT sensor nodes in a cyclic sleep scenario, *2013 IEEE International Wireless Symposium (IWS).* Beijing, April 14-18, 2003. IEEE, pp. 1-4.

Cirrus Logic Audio Card for Raspberry Pi A plus and B plus (2014)
Available at: https://www.element14.com/community/community/raspberry-pi/raspberry-pi-accessories/cirrus_logic_audio_card
(Accessed 28 May 2018)

FIDE (2018) *FIDE rating list.*
Available at: https://github.com/official-stockfish/Stockfish
(Accessed 05 June 2018)

Gomez, C., Oller, J. and Paradells, J. (2012) Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology, *Sensors*, 12(9), pp. 11734-11753. doi: 10.3390/s120911734

López, P. *et al.* (2013) Survey of Internet of Things Technologies for Clinical Environments, *2013 27th International Conference on Advanced Information Networking and Applications Workshops Barcelona.* Barcelona, March 25-28, 2013. IEEE, pp. 1349-1354.

Mesaros, A. and Heittola, T. (2017) *DCASE 2017 Task 1.*
Available at: http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/task-acoustic-scene-classification
(Accessed 18 May 2018)

Mierswa, I. and Morik, K. (2005) Automatic Feature Extraction for Classifying Audio Data. *Machine Learning*, 58(2-3), pp. 127-149. doi: 10.1007/s10994-005-5824-7

Nordic Semiconductor (no date) *nRF52840.*
Available at: https://www.nordicsemi.com/eng/Products/nRF52840
(Accessed 28 May 2018)

Oliveira, C. H. S., Ghamri-Doudane, Y. and Lohier, S. (2013) A duty cycle self-adaptation algorithm for the 802.15.4 wireless sensor networks, *Global Information Infrastructure Symposium - GIIS 2013.* Trento, October 28-31, 2013. IEEE, pp. 1-7.

Olsson, J. (2014) *6LoWPAN Demystified.*
Available at: http://www.ti.com/lit/wp/swry013/swry013.pdf
(Accessed 26 May 2018)

Park, S. *et al.* (2017) Acoustic Scene Classification Based on Convolutional Neural Network Using Double Image Features, *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017),* pp. 98-102.

Quintana-Suárez, M. A. *et al.* (2017) A Low Cost Wireless Acoustic Sensor for Ambient Assisted Living Systems. *Applied Sciences,* Issue 7(9), p. 877. doi: 10.3390/app7090877

Ray, B. (2015) *A Bluetooth & ZigBee Comparison For IoT Applications.*
Available at: https://www.link-labs.com/blog/bluetooth-zigbee-comparison
(Accessed 20 May 2018)

Ray, B. (2017) *Z-Wave Vs. Zigbee.*
Available at: https://www.link-labs.com/blog/z-wave-vs-zigbee
(Accessed 23 May 2018)

Salomons, E. L. and Havinga, P. J. M. (2015) A Survey on the Feasibility of Sound Classification on Wireless Sensor Nodes. *Sensors (Basel, Switzerland),* 15(4), p. 7462–7498. doi: 10.3390/s150407462.

Schatz, G. (2016) *The Complete List Of Wireless IoT Network Protocols.*
Available at: https://www.link-labs.com/blog/complete-list-iot-network-protocols
(Accessed 24 May 2018)

Shaikh, F. K. and Zeadally, S. (2016) Energy harvesting in wireless sensor networks: A comprehensive review. *Renewable and Sustainable Energy Reviews,* Volume 55, pp. 1041-1054. doi: 10.1016/j.rser.2015.11.010

Siekkinen, M. *et al.* (2012) How low energy is bluetooth low energy? Comparative measurements with ZigBee/802.15.4, *2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW).* Paris, April 1-1 2012. IEEE, pp. 232-237.

Silver, D. *et al.* (2017) Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv:1712.01815*.

STMicroelectronics (2017) *STSW-BNRG001*.
Available at: http://www.st.com/en/embedded-software/stsw-bnrg001.html
(Accessed 1 June 2018)

STMicroelectronics (no date) *BlueNRG-2.*
Available at: http://www.st.com/en/wireless-connectivity/bluenrg-2.html
(Accessed 1 June 2018)

Stockfish community (2018) *Stockfish GitHub repository.*
Available at: https://github.com/official-stockfish/Stockfish
(Accessed 05 June 2018)

Tabish, R. *et al.* (2013) A Comparative Analysis of BLE and 6LoWPAN For U-HealthCare Applications, *2013 7th IEEE GCC Conference and Exhibition (GCC).* Doha, November 17-20, 2013. IEEE, pp. 286-291.

Tamkittikhun, N., Hussain, A. and Kraemer, F. A. (2017) Energy Consumption Estimation for Energy-Aware, Adaptive Sensing Applications. In: S. Bouzefrane, et al. eds. *Mobile, Secure, and Programmable Networking. MSPN 2017. Lecture Notes in Computer Science, vol 10566.* Cham: Springer, pp. 222-235.

T, J. (2015) Algorithms for Duty Cycle Control in Wireless Sensor Networks- A Survey. *International Journal of Computer Science and Information Technologies,* 6(4), pp. 3881-3884.

Vigorito, C. M., Ganesan, D. and Barto, A. G. (2007) Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks. *2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. San Diego, June 18-21, 2007. IEEE, pp. 21-30.

Weiping, Z. *et al.* (2017) Acoustic Scene Classification Using Deep Convolutional Neural Network and Multiple Spectrograms Fusion, Munich: s.n.

Wieringa, R. J. (2014a) Scaling Up to Stable Regularities and Robust Mechanisms. In: *Design Science Methodology for Information Systems and Software Engineering.* London: Springer, pp. 66-67.

Wieringa, R. J. (2014b) The Design and Engineering Cycles. In: *Design Science Methodology for Information Systems and Software Engineering.* London: Springer, pp. 27-28.

Wold, E. *et al.* (1996) Content-Based Classification, Search, and Retrieval of Audio. *IEEE MultiMedia,* 3(3), pp. 27-36. doi: 10.1109/93.556537

Zhang, Y. *et al.* (2010) Energy-Efficient Duty Cycle Assignment for Receiver-Based Convergecast in Wireless Sensor Networks, *2010 IEEE Global Telecommunications Conference GLOBECOM 2010.* Miami, December 6-10, 2010. IEEE, pp. 1-5.

Zigbee Alliance Members (no date)
Available at: http://www.zigbee.org/zigbeealliance/our-members/
(Accessed 22 May 2018)